

**UNIVERSIDAD MAYOR DE SAN ANDRES
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



PROYECTO DE GRADO

“SISTEMA DE CONTROL Y GESTION HOTELERA”

**PARA OPTAR AL TITULO DE LICENCIATURA EN INFORMATICA
MENCION: INGENIERIA DE SISTEMAS INFORMATICOS**

Postulante: Grover Milton Carrasco Quino.

Tutor: Mg. Sc. Fatima Dolz Salvador.

Revisor: Mg.Sc. Luisa Velásquez López.

LA PAZ - BOLIVIA

2009



DEDICATORIA

Dedico el presente trabajo a Dios por haberme dado el gran regalo de la vida y estar ahí en todo momento cuidándome.

A mi familia por brindarme todo su apoyo durante estos años, en especial a mi Mamá que siempre estuvo ahí dándome su apoyo, cariño y comprensión, a mi Papa por su apoyo, a mi tía Angélica por estar siempre ahí dándome consejos, confianza y apoyo, a mis hermanas Gabriela, Patricia y hermano Iván.

Y a todos los compañeros que conocí en la Carrera.

Correo electrónico: dx.hbk.wwf@gmail.com

AGRADECIMIENTOS

Es el momento de dar gracias a las personas que colaboraron para la culminación de mis estudios en esta etapa de mi vida. A mi Tutora Mg. Sc. Fatima Dolz Salvador Docente de Taller de Licenciatura II por coadyuvar a organizar y estructurar, el contenido del presente Proyecto de Grado.

A mi Revisora Mg.Sc. Luisa Velásquez López, por su comprensión y paciencia a la hora de realizar las revisiones, absorbiendo mis dudas en todo momento además de todo el tiempo que invirtió en el desarrollo del Proyecto de Grado.

Al Ing. Edgar Salinas Fuentes, Administrador del Hotel Sayari, por darme la oportunidad de desarrollar el presente Proyecto y dedicarme su valioso tiempo, brindándome toda la información necesaria para el desarrollo del “Sistema de Control y Gestión Hotelera”.

A Todos Ellas Muchas Gracias...

RESUMEN

El presente Proyecto toma como objeto de estudio al "Hotel Sayari" catalogado de tres estrellas, es una empresa que ofrece el servicio de alojamiento, lavandería, fax, Tv cable, entre otros a personas sean nacionales o extranjeros, difundiendo sus servicios a través de grandes Empresas de Turismo.

Al Hotel llega una gran afluencia de huéspedes es por eso que es necesario desarrollar e implementar un sistema de información que permita realizar el control y seguimiento de los procesos, el Proyecto de Grado titula "**Sistema de Control y Gestión Hotelera**", esta compuesto por cinco capítulos.

En el primer capítulo se realiza el análisis del sistema que concierne a la identificación y descripción de procesos dentro del área que repercuten en el planteamiento de las problemáticas y necesidades del Hotel. Asimismo contiene el análisis de la situación actual, este análisis genera el problema y objetivos a ser alcanzados con la implementación del producto software.

En el segundo capítulo se desarrolla el marco teórico que sustenta el proceso de desarrollo del software, engloba varios conceptos, entre ellos, cifrado, base de datos, y metodología de desarrollo ágil de sistemas informáticos XP acrónimo de Programación Extrema.

El tercer capítulo presenta el sistema físico y lógico, el primero muestra el análisis del antiguo sistema, el segundo da una idea del sistema a ser implantado.

El cuarto capítulo comprende el ciclo de desarrollo de software y muestra los puntos de la XP. Las fases de la metodología XP son: planeación, diseño, codificación y pruebas, Se obtiene 10 historias de usuario las cuales se van desarrollando en función de la XP.

En este capítulo además se encuentra la Demostración de Objetivos y la Calidad.

En la última parte del documento se realizan las Recomendaciones y Conclusiones del presente Proyecto.

INDICE DE PROYECTO

PRESENTACION

1.1. INTRODUCCION.....	1
1.2. ANTECEDENTES.....	3
1.2.1 Institucionales.....	3
1.2.2 Temas a fines.....	3
1.3. DESCRIPCION DEL PROBLEMA.....	5
1.4. PLANTEAMIENTO DEL PROBLEMA.....	5
1.5. OBJETIVOS.....	6
1.5.1 OBJETIVO GENERAL.....	6
1.5.2 OBJETIVOS ESPECIFICOS.....	6
1.6. OBJETO DE ESTUDIO.....	6
1.7. JUSTIFICACIÓN Y RELEVANCIA.....	7
1.8. ALCANCES Y LIMITES.....	7
1.9. METODOLOGÍA.....	8
1.10. APORTE.....	8

MARCO TEÓRICO

2.1 INTRODUCCIÓN.....	9
2.2 ANTECEDENTES DE LA ORGANIZACIÓN.....	9
2.2.1 HOTEL SAYARI.....	9
2.3 METODOLOGIA.....	10
2.3.1 Programación Extrema (XP).....	10
2.3.1.1 Valores de XP.....	11
2.3.1.2 Principios de la XP.....	11
2.3.1.3 Prácticas de la XP.....	12
2.4 PARADIGMA ORIENTADO A OBJETOS.....	13
2.4.1 Características de la Programación Orientación a Objetos.....	14
2.5 UML.....	16
2.5.1 Características de UML.....	17
2.6 LENGUAJE DE PROGRAMACION.....	19
2.6.1 PHP.....	19
2.6.2 JavaScript.....	20

2.6.2.3 jQuery.....	20
2.7 BASES DE DATOS RELACIONALES.....	20
2.9 SISTEMAS DE MANEJO DE BASE DE DATOS (DBMS).....	21
2.9.3 MySQL.....	21
2.10 ESTRUCTURA DE DATOS.....	22
2.10.1 Arrays Unidimensionales: Los Vectores.....	22
2.10.2 Array Bidimensionales (Tablas/Matrices).....	23
2.10.3 Listas.....	24
2.10.4 Listas Dblemente Enlazadas.....	25
2.11 INTERFAZ GRAFICA DE USUARIO.....	25
2.11.1 Diseño de la Interfaz de Usuario.....	25
2.12 ESTRATEGIA DE PRUEBA.....	29
2.12.1 Pruebas de Caja Negra.....	29
2.12.2 Pruebas de Caja Blanca.....	29
2.13 CRIPTOGRAFIA.....	30
2.13.1 Criptografía de Clave Secreta (simétrica).....	31
2.13.1.1 Cifrado de Vernam.....	32
2.13.1.2 Algoritmo Rijndael.....	30
2.13.2 Criptografía de Clave Pública (asimétrica).....	32
2.13.2.1 Sistema de ElGamal.....	32
2.13.2.2 MD5.....	33
2.14 ARQUITECTURA CLIENTE/ SERVIDOR.....	34
2.14.1 Arquitectura cliente-servidor de tres niveles	34
2.15 CALIDAD.....	35

MARCO PRÁCTICO

3.1 Sistema Físico Actual.....	37
3.2 Sistema Lógico.....	40
3.2.1 Diagrama de Casos de Uso.....	40
3.3 Requerimientos del Ingeniero de Software.....	40
3.3.1 En Cuanto a Hardware.....	41
3.3.2 En cuanto a Software.....	41

PROCESO DE INVESTIGACIÓN

4.1 Elaboración de las Historias de usuario.....	42
4.2 Planificación.....	45
4.3 Diseño y Desarrollo.....	45
 4.3.1 Primer incremento.....	45
 4.3.2 Segundo incremento.....	47
 4.3.3 Tercer incremento.....	56
 4.3.4 Cuarto incremento.....	58
 4.3.5 Quinto incremento.....	65
 4.3.6 Sexto incremento.....	70
 4.3.7 Séptimo incremento.....	73
 4.3.8 Octavo incremento.....	76
 4.3.9 Noveno incremento.....	78
 4.3.10 Décimo incremento.....	79
4.4 Demostración de Objetivos.....	81
4.5 CALIDAD.....	87
 4.5.1 Funcionalidad.....	87
 4.5.2 Fiabilidad.....	89
 4.5.3 Usabilidad.....	91
 4.5.4 Eficiencia.....	91
 4.5.5 Mantenibilidad.....	92
 4.5.6 Portabilidad.....	93

PARTE FINAL

5.1 CONCLUSIONES.....	94
5.2 RECOMENDACIONES.....	95

REFERENCIAS BIBLIOGRAFICAS.....	96
--	-----------

Glosario de términos

Anexos

A. Historias de Usuario

B. Tareas

INDICE DE FIGURAS

Figura 1.1. Estructura Orgánica	4
Figura 2.1 Vista del Paradigma XP	11
Figura 2.2 Ejemplo de Herencia	15
Figura 2.3 Vista general de UML.....	16
Figura 2.4 Vectores.....	23
Figura 2.5 Matriz.....	23
Figura 2.6 Matrices Ejemplo (subíndices).....	24
Figura 2.7 Lista Dblemente Enlazada.....	25
Figura 2.8 Evaluación del diseño de interfaz.....	27
Figura 2.9 Clave Privada	31
Figura 2.10 Arquitectura de tres Niveles	34
Figura 3.1. Actores del Sistema.....	37
Figura 3.2. Diagrama de Actividades Actual	39
Figura 3.3 Diagrama de casos e Uso	40
Figura 4.1 Diagrama de Clases.....	46
Figura 4.2 Modelo Entidad - Relación.....	47
Figura 4.3 Diagrama Jerárquico.....	51
Figura 4.4 Estructura Cliente Servidor Hotel Sayari.....	52
Figura 4.5 Comportamiento de la arquitectura tres capas.....	52
Figura 4.6 Registro	53
Figura 4.7 Clases del modulo Registro.....	54
Figura 4.8 Reserva.....	55
Figura 4.9 Clases del modulo Reserva.....	55
Figura 4.10 Interfaz.....	57
Figura 4.11 Validación.....	57
Figura 4.12 Ventana Menú Principal.....	58
Figura 4.13 Ventana datos del huésped.....	58
Figura 4.14 Clases del modulo, datos del huésped.....	59
Figura 4.15 Ventana Habitación.....	60
Figura 4.16 Clases del modulo, Habitación.....	61
Figura 4.17 Ventana Servicios y Cuenta.....	63
Figura 4.18 Clases del modulo, servicios y cuenta.....	64

Figura 4.19 Clases del modulo ,factura.....	65
Figura 4.20 Ventana Factura.....	66
Figura 4.21 Ventana Factura por habitación.....	67
Figura 4.22 Ventana Factura Grupal.....	69
Figura 4.23 Factura, Impresión.....	70
Figura 4.24 Busca factura.....	71
Figura 4.25 Ejemplo de busca factura.....	71
Figura 4.26 Cotización.....	72
Figura 4.27 ventana ingresos.....	73
Figura 4.28 ventana informes.....	73
Figura 4.29 listado de huéspedes.....	74
Figura 4.30 Estadísticas.....	74
Figura 4.31 Estadísticas, gráficos.....	75
Figura 4.32 Configuración, Habitación.....	76
Figura 4.33 Clases del modulo Registro	77
Figura 4.34 Encriptación.....	79
Figura 4.35 Interfaz utilizando matrices.....	81
Figura 4.36 Matriz 7x10.....	82
Figura 4.37 Base de Datos, MD5.....	85
Figura 4.38 Modelo del Sistema.....	90

INDICE DE TABLAS

Tabla 1.1: Servicios.....	3
Tabla 1.2: Relación Causa-Efecto.....	5
Tabla 4.1: Cronograma de desarrollo de incrementos.....	45
Tabla 4.35: Matriz de Punto Función.....	87
Tabla 4.36 Punto Función.....	87
Tabla 4.38 Confiabilidad por componente del Sistema.....	90



1. PRESENTACION

1.1 INTRODUCCION

Antiguamente existían varias clases de establecimientos, a lo largo de los caminos y dentro de las poblaciones había tabernas y posadas para satisfacer las necesidades de los viajeros y de la población local. Posteriormente, las tabernas y posadas que ofrecían alojamiento, comida y bebida se habían convertido en una realidad común.

En la actualidad se puede decir que un Hotel es un edificio planificado y acondicionado para albergar a las personas temporalmente y que permite a los viajeros, alojarse durante sus desplazamientos. Los Hoteles proveen a los huéspedes servicios adicionales como restaurantes, piscinas y guarderías. Algunos hoteles tienen servicios de conferencias y animan a grupos a organizar convenciones y reuniones en su establecimiento.

Existe una gran clasificación por tipos como son: Hoteles urbanos, Hoteles de aeropuerto, Hoteles de playa, Hoteles de naturaleza, Hoteles Albergues turísticos, Hoteles familiares, Hoteles posada, Hoteles balneario, Hoteles de paso o Alojamiento, Hoteles-casino, Hoteles-clubes, Hoteles deportivos, Hoteles Gastronómicos, Hoteles de montaña, Hoteles boutique, Hoteles de Negocios, Hoteles Vacacionales, Cadenas hoteleras, etc...
[Roy, 2009]

En Bolivia, los Hoteles son aquellos establecimientos que presentan en forma permanente el servicio de hospedaje en unidades habitacionales, la construcción deberá constituir un todo homogéneo con escaleras y/o ascensores de uso exclusivo.

Debe tener un mínimo de 20 habitaciones, contar con el servicio sanitario privado y/o común según su categoría.

Los Establecimientos de Hospedaje Turístico serán categorizados según su tipo y clasificación.

Las diferentes categorías genéricas de los Establecimientos de Hospedaje Turístico son: [Téllez, 2002].

Hoteles	1 a 5 estrellas;
Apart Hoteles	3 a 5 estrellas;
Hoteles Suites	4 y 5 estrellas;
Hostales o Residenciales	1 a 5 estrellas;
Casas de Huéspedes	Categoría única;
Complejos Turísticos	3 a 5 estrellas;
Flóteles	Categoría única;
Establecimiento de Hospedaje Complementario	Categoría única;

En el presente, gracias a la competitividad es vital el manejo de la información para una organización, es así que los sistemas de información basados en computadoras son fundamentales para un Hotel. Los Hoteles manejan una gran cantidad de información gracias a todos los procesos, en consecuencia toda esta información debe ser procesada con el fin de dar una base sólida cuando los gerentes tomen decisiones.

La industria Hotelera en nuestro País, es muy importante ya que además de generar ganancias para el Hotel también Genera ganancias directas o indirectas en todo su entorno.

El presente trabajo desarrolla un sistema informático para el Hotel SAYARI para automatizar los procesos de reservas, registros, servicios, reportes, caja y auditoria nocturna de forma rápida y eficiente.

1.2. ANTECEDENTES

La clasificación de los Hoteles:

Nota: SI = Requisito Obligatorio, OP = Requisito Opcional;

Solo se toma lo referente a los Servicios al Huésped, de Hoteles de 1 a 3 Estrellas, no así de 4 y 5 Estrellas. [Téllez, 2002]

Tabla 1.1 Servicios

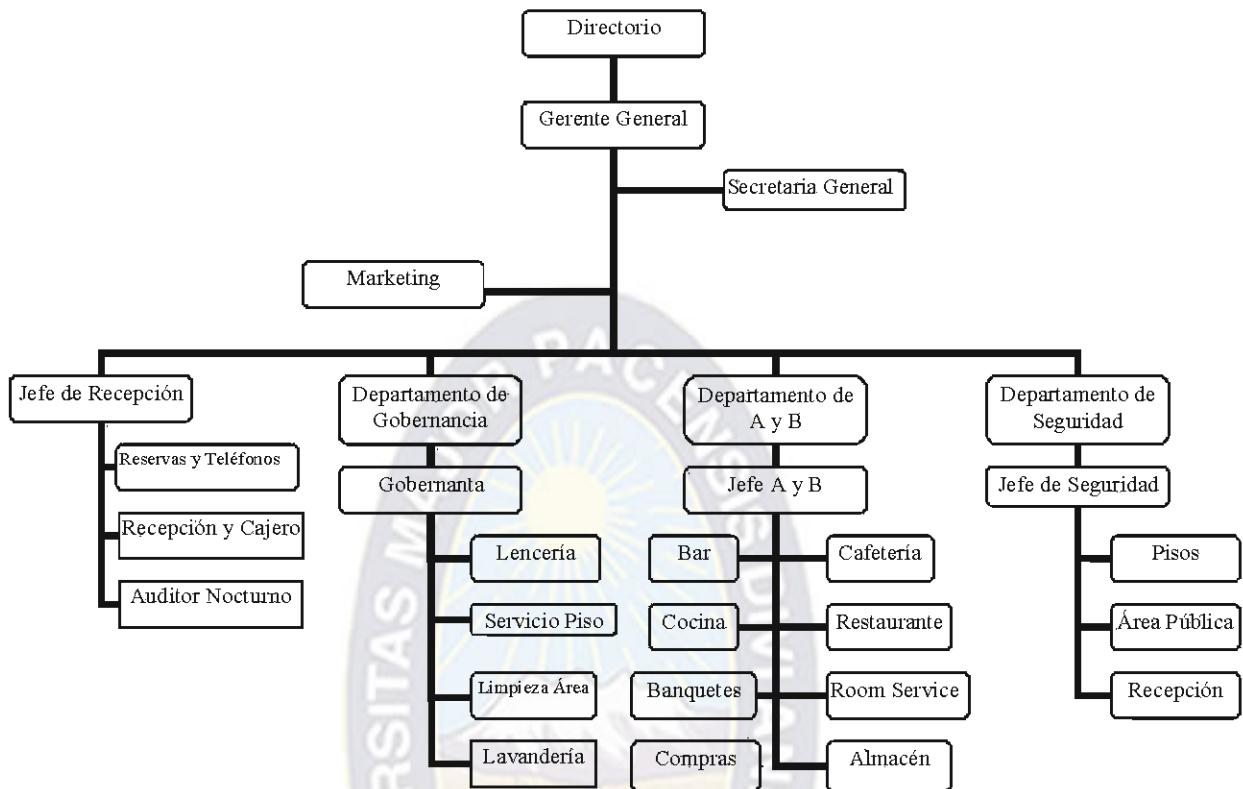
Categoría del hotel	3(***)	2 (**)	1 (*)
Servicio al Huésped			
Servicio de lavado y planchado de ropa de acuerdo a la política de la empresa	SI	SI	SI
Servicio de portería las 24 horas	SI	SI	SI
Servicio de información general con personal capacitado	SI	OP	OP
Servicio de estacionamiento dentro, contiguo o en las cercanías del establecimiento	OP	OP	OP
Servicio de Botones	SI	OP	OP
Servicio de Masajes	SI	SI	SI
Servicio en habitaciones en horas	8	OP	OP
Servicio de Telefonía local, nacional en habitaciones	SI	OP	OP
Agua fría en grifos	SI	SI	SI
Agua caliente en grifos	SI	OP	OP
Custodia de equipajes y valores	SI	OP	OP
Botiquín de primeros auxilios y emergencias	SI	SI	SI
Servicios de equipos postales y aire acondicionado, ventilación y/o calefacción para habitaciones a solicitud del huésped	SI	SI	SI

Fuente:[Téllez, 2002]

1.2.1 Institucionales.

El Hotel Sayari, catalogado de tres estrellas ubicado en la ciudad de La Paz en la Avenida Manco Cápac.

Figura 1.1 Estructura Orgánica



El jefe de recepción se encarga de lo que es las reservas, registros, reportes, caja y auditoria nocturna, de los servicios se encarga el Departamento A y B, lo referente ha limpieza de habitaciones es absorbido por el departamento de gobernanza,

1.2.2 Temas afines.

Revisando Bibliografía y trabajos relacionados con la Gestión Hotelera en la carrera de Informática se tiene.

El proyecto de Cabero, 2000,"Sistema de Información Hotelera", La Paz, cuyo propósito es mejorar los procesos operativos y administrativos para una atención más eficiente al huésped.

El proyecto de Loza, 2001,"Sistema de Seguimiento y Control para la Hotelería Boliviana", La Paz, cuyo fin es resolver las dificultades y deficiencias que se presentan en algunos procesos del Hotel.

El proyecto de grado de Cuela, 2007, "Sistema de Seguimiento y Control Caso: Hotel Sagarnaga", La Paz Bolivia, cuyo objetivo es Analizar, diseñar e implementar un sistema informático que brinde información confiable y oportuna para una buena toma de decisiones.

El presente proyecto implementará un sistema informático que realizará un adecuado control y gestión de la información para la toma de decisiones, mejorará la atención en el Hotel, y será capaz de realizar los procesos de reservas, registros, servicios, reportes, caja y auditoria nocturna de forma rápida, para tal fin se realizará un estudio a fondo de los procesos existentes en el Hotel, se utilizará listas dobles para las reservas, matrices para un fácil manejo de las habitaciones y Base de Datos, además se utilizará el lenguaje de programación PHP.

1.3. DESCRIPCION DEL PROBLEMA

Tabla 1.2 Relación Causa-Efecto

PROBLEMA	CAUSA	EFFECTO	SOLUCION
Existe un registro inadecuado de huéspedes.	Registro manual en varios libros.	Desorden y pérdida de información.	Almacenar en una BD para proporcionar esta información.
No existe una adecuada asignación de habitaciones.	Debido al registro manual, se toma los registros antiguos como validos.	El huésped es asignado a una habitación no disponible.	Almacenar en una matriz las habitaciones del hotel con el fin de realizar una asignación grafica.
No hay control de las reservas.	Reserva manual en distintos libros.	El huésped es asignado a una habitación no disponible, o en otro caso una habitación disponible figura como reservado.	Módulo que realice un seguimiento y control de reservas
Existe una deficiente información acerca de los distintos servicios que obtuvo el huésped.	Registro manual en varios libros.	Pérdida económica y molestia del huésped.	Módulo que realice un seguimiento y control de los servicios que obtuvo el huésped.
Existe una Facturación manual	Registro manual en varios libros.	Facturación lenta	Módulo que pueda facturar de manera rápida.

Fuente: [Velásquez, datos propios]

1.4. PLANTEAMIENTO DEL PROBLEMA

¿El Hotel SAYARI, funciona con un entorno de trabajo manual en consecuencia los procesos de reservas, registros, servicios, reportes, caja y auditoria nocturna son lentos, existe un sistema informático que pueda solucionar los problemas de forma rápida?

De acuerdo al análisis se plantea las siguientes interrogantes:

1. ¿Cómo ayudará el sistema informático a los Administradores del Hotel?
2. ¿Es posible implementar un sistema de informático?
3. ¿Qué medios pueden ayudarán a implementar el sistema informático?
4. ¿Cómo mostrar datos de manera comprensible para los que administrarán el sistema?
5. ¿Es posible que las metodologías que se usarán, contribuirán con el desarrollo del sistema de una forma clara y sencilla?

Tomando en cuenta todas las interrogantes, una opción de disponer de toda la información es el Sistema de Control y Gestión Hotelera.

1.5 OBJETIVOS

1.5.1 OBJETIVO GENERAL.

Realizar un sistema informático que permita un adecuado control y gestión de la información para la toma de decisiones aplicando modelos criptográficos.

1.5.2 OBJETIVOS ESPECIFICOS.

- Desarrollar un módulo que pueda facturar, integrando procesos.
- Implementar una interfaz grafica de las habitaciones del hotel, utilizando Matrices.
- Realizar la Interfaz Grafica de Usuario, aplicando métricas.
- Implementar contraseñas, cifrando datos.
- Desarrollar un sistema multiusuario, utilizando el modelo Cliente Servidor.
- Implementar el proceso de Reservas mediante Listas Dobles.

1.6 OBJETO DE ESTUDIO

La razón para desarrollar este trabajo “Sistema de Control y Gestión Hotelera”, es poder dar una herramienta que permita a los administradores del Hotel Sayari realizar un trabajo mucho más rápido, eficiente y seguro, ya que en este momento tienen un sistema de trabajo manual.

1.7 JUSTIFICACIÓN Y RELEVANCIA

El desarrollo del Sistema es muy importante por que va ha servir a los administradores del Hotel, brindándoles información precisa y rápida para la toma de decisiones oportuna.

La posibilidad de aplicar teorías y modelos del ámbito científico a la resolución de problemas en los campos reales siempre dará como resultado una solución óptima, el presente sistema integrara distintos módulos. El Sistema contará con una interfaz agradable, utilizará listas para el proceso de reservas, manejará matrices en la interfaz gráfica de las habitaciones además de integrar todos los módulos para la facturación.

El Sistema para un optimo funcionamiento requiere en cuanto a Hardware, computadoras Pentium III en adelante, en cuanto a software sistema operativo Windows Xp en adelante, además el Hotel cuenta con todos los recursos tecnológicos, materiales y equipos que puedan ser aprovechados de manera eficaz y eficiente.

El Sistema informático ayudará al personal involucrado a un mejor desenvolvimiento y una mejor atención, atrayendo más huéspedes, ya que al Hotel acuden personas de distintas clases sociales sean nacionales o extranjeros, además la presencia del Hotel provoca un impacto económico en todo su entorno generando ingresos ya sean directos o indirectos a negocios o personas particulares.

1.8 ALCANCES Y LIMITES

- El sistema Informático permitirá realizar la reserva de un huésped, el registro, los servicios que ha adquirido durante su estancia en el Hotel, la habitación o habitaciones que ha ocupado y además la facturación;
- También contará con los reportes, es decir todo lo que sucedió durante un determinado tiempo;
- Al hacer el cierre del día, el sistema automáticamente carga la noche a la cuenta del huésped y prepara todo para el nuevo día;
- Capacitar al administrador del sistema, explicar el Manual de usuario;
- El Sistema no incluye lo que es el Sistema de Restaurante;

1.9 METODOLOGÍA

Se utilizara un modelo Ágil de desarrollo de software utilizando el paradigma Orientado a Objetos, con UML.

1.10 APORTE

La implantación del presente proyecto es un aporte significativo para Hotel ya que este será de gran ayuda para la toma de decisiones.

El sistema será desarrollado bajo la metodología ágil XP, utilizara el cifrado de datos para la seguridad, además del ISO/9126 para asegurar la calidad del software.



2. MARCO TEÓRICO

2.1 INTRODUCCIÓN

En este Capítulo se hace referencia al marco teórico necesario para implementar el “Sistema de Control y Gestión Hotelera”.

El marco teórico, incluye lo relacionado a los antecedentes de la organización, después se introduce un poco de teoría sobre la metodología, seguidamente se describe el Paradigma Orientado a Objetos y La Programación Orientada a Objetos, también se menciona el lenguaje de programación y los Sistemas de Manejo de Base de Datos, seguidamente se ve las Estructuras de Datos que entraran en el desarrollo del proyecto y se da a conocer un poco sobre La Encriptación.

2.2 ANTECEDENTES DE LA ORGANIZACIÓN

Los Hoteles desempeñan un papel muy importante pues generan ingresos directos e indirectos, además incentivan el turismo en el País.

2.2.1 HOTEL SAYARI

El Hotel Sayari es un establecimiento hotelero, catalogado como “Hotel 3 Estrellas”, que trabaja de manera integrada, con las empresas Turísticas de la ciudad de La Paz y de Bolivia además de redes Internacionales.

El Hotel Sayari ofrece una gran variedad de servicios:

- Habitaciones, ofrece 47 habitaciones, Simples, Dobles, Matrimoniales, Cuádruples y suites, todas con baño privado;
- Televisión por cable;
- Teléfono, llamadas nacionales e internacionales;
- Servicio, de fax, lavandería, room service ,etc.;
- Restaurante, con menú nacional e internacional;
- Salón, de conferencia, reuniones, etc.;

Cuyo objetivo es brindar la mayor comodidad a los huéspedes que están alojados.

En cuanto al personal, para el funcionamiento el hotel cuenta con; el recepcionista el cual se encarga de realizar las reservas y registros de los huéspedes además de anotar los servicios solicitados; los botones que son los encargados de acompañar a los huéspedes a sus habitaciones con las maletas además de proporcionar todos los servicios que solicite este, el departamento de gobernanza que se encarga de la limpieza de las habitaciones, piso etc.., el restaurante que absorbe todo lo referente al consumo, finalmente la seguridad.

2.3 METODOLOGIA

2.3.1 Programación Extrema (XP)

La XP, es una metodología para el desarrollo de proyectos software que trata de dar solución a los problemas de la ingeniería del software desde un enfoque completamente distinto al tradicional.

La XP, recibe este calificativo porque defiende un enfoque radical.

Reconoce las bondades de las prácticas de las metodologías tradicionales y propone llevarlas hasta su extremo:

“Si diseñar es bueno, diseñemos todo el tiempo”

“Si las pruebas son buenas, probemos todo el tiempo”

La XP utiliza un enfoque OO, como su paradigma de desarrollo preferido. La XP abarca un conjunto de reglas y prácticas que ocurren en el contexto de 4 actividades del marco de trabajo:

- Planeación;
- Diseño;
- Codificación;
- Prueba;

Proceso de desarrollo extremo;

Figura 2.1 Vista del Paradigma XP



2.3.1.1 Valores de XP

- **Comunicación.** Para ser efectiva, debe involucrar a todos los participantes en el proyecto, y debe ser libre y sincera;
- **Simplicidad.** Nunca debe perderse de vista que el objetivo de un proyecto es proporcionar valor al cliente, no es demostrar la pericia técnica del equipo ni construir una aplicación que resuelva más problemas que los del cliente;
- **Refabricación.** No se puede dirigir adecuadamente un proceso si no se dispone de realimentación permanente sobre su progreso. La realimentación puede provenir del cliente, de los programadores, de herramientas automáticas y otros;
- **Coraje.** A veces, hacer lo que es correcto requiere valor. Por ejemplo, hay que tener coraje para reescribir código que funciona pero ha alcanzado su límite de escalabilidad;

2.3.1.2 Principios de la XP

Los valores mencionados dan origen a cinco principios básicos:

- Conseguir realimentación rápida;
- No complicar las cosas con suposiciones (asumir que las cosas son simples);
- Realizar cambios incrementales;
- Abrazar el cambio;
- Generar productos de calidad;

2.3.1.3 Prácticas de la XP

Los principios se manifiestan a través de las prácticas de XP. Son actividades que el equipo de un proyecto lleva a cabo cada día. Las 12 prácticas de la XP tienen su origen en prácticas conocidas en la Ingeniería de Software y en el sentido común. Sin embargo, lo que caracteriza a este conjunto es la cohesión de todos los elementos, y que cada práctica ha sido llevada al extremo:

- 1. El juego de la planificación.** Esta práctica busca dividir la funcionalidad de un proyecto en pequeños fragmentos auto contenidos, c/u de los cuales se denomina historia de usuario;
- 2. Entregas frecuentes.** Se trata de publicar una nueva versión en cuanto sea posible aportar algún nuevo valor al cliente;
- 3. Diseño simple.** El sistema debe ser el más simple posible que cumpla especificaciones (pruebas de aceptación). En un entorno donde los requisitos del cliente y sus prioridades cambian continuamente, no tiene sentido realizar un diseño sofisticado. La mejor forma de obtener una idea de los futuros requisitos de un sistema es proporcionar cuanto antes un prototipo al cliente y obtener realimentación;
- 4. Pruebas automáticas.** ¿Cómo puede saber un programador que el código que ha escrito funciona realmente? ¿Cómo puede saber que seguirá funcionando siempre, incluso aunque lo refactorice?. La única manera de asegurarlo con cierta confianza es escribiendo pruebas automáticas con las que pueda comprobar el código en cualquier momento y sin esfuerzo;
- 5. Integración continua.** Nuevamente se lleva al extremo una práctica convencional de la ingeniería del software. Si la integración es una fase crucial, en la que pueden aparecer errores, ¿por qué dejarla para el final, cuando el riesgo es mayor? Resulta más conveniente realizar integración continua (cada hora, cada día). Para poder hacerlo, es imprescindible que el proceso de integración esté automatizado y pueda verificarse mediante pruebas;
- 6. Refactorización.** La refactorización es un proceso disciplinado por el cual se modifica el diseño de un módulo sin afectar a su comportamiento externo. Gracias a la refactorización, es posible compatibilizar el diseño simple con la flexibilidad. El coraje para refactorizar proviene de la disponibilidad de pruebas automáticas;
- 7. Programación por parejas.** Llevar al extremo una práctica habitual: las revisiones formales de código. Si revisar el código es bueno, ¿por qué no revisarlo continuamente, incluso desde el mismo momento en el que se escribe por primera vez? En la programación por parejas, dos programadores comparten un único ordenador y colaboran para escribir el código o las pruebas;

8. Propiedad colectiva del código. En el transcurso de una refactorización, o mientras se corrige un defecto, algún programador va a tener que modificar líneas de código escritas por otro programador. XP invita a llevar a cabo esa modificación con coraje, y el coraje procede de las pruebas automáticas. Esta práctica permite que funcionen bien los equipos dinámicos, cuya composición puede variar durante el proyecto;

9. Semana de 40 horas. Los programadores cansados se equivocan más. Si las semanas de más de 40 horas son la norma, algo no funciona bien en el proyecto o en la empresa;

10. Metáfora. El objetivo de esta práctica es encontrar una metáfora que ayude al equipo del proyecto y al cliente a hablar en los mismos términos, compartiendo una visión común del sistema;

11. Cliente en el equipo. Para lograr una realimentación ágil, el cliente no puede estar muy lejos del equipo, en una situación ideal, el cliente debe estar dentro del equipo;

12. Estándares de codificación. Es una necesidad cuando se trata de escribir código que otros programadores puedan entender y modificar, [Aguilar 2009].

2.4 PARADIGMA ORIENTADO A OBJETOS

Los conceptos y herramientas orientadas a objetos son tecnologías que permiten que problemas del mundo real sean expresados de modo fácil y natural. Las técnicas orientadas a objetos proporcionan mejoras y metodologías para construir sistemas de software complejos a partir de unidades de software modularizado reutilizable.

Se necesita de un nuevo enfoque para construir software en la actualidad.

Este nuevo enfoque debe ser capaz de manipular tanto sistemas grandes como pequeños y debe crear sistemas fiables que sean flexibles, mantenibles y capaces de evolucionar para cumplir las necesidades de cambio.

El soporte fundamental es el modelo objeto, los cuatro elementos más importantes de este modelo son:

- **Abstracción**, que es la propiedad que nos permite representar las características esenciales de un objeto sin preocuparse de las restantes características no esenciales. Definir una abstracción significa describir una entidad del mundo real, no importa lo compleja que pueda ser y a continuación, utilizar esta descripción en un programa;
- **Encapsulamiento o encapsulación**, es la propiedad que permite asegurar que el contenido de la información de un objeto esta oculta al mundo exterior, el objeto A no

conoce lo que hace el objeto B y viceversa. La encapsulación también se conoce como ocultación de la información;

- **Modularidad**, es la propiedad que permite subdividir una aplicación en partes más pequeñas llamadas módulos, cada una de las cuales debe ser tan independiente como sea posible de la aplicación en sí y de las restantes partes;
- **Jerarquía**, es la propiedad que nos permite una ordenación de abstracciones. Las dos jerarquías más importantes de un sistema complejo son: estructura de clases y estructura de objetos;

El modelo ideal no solo tiene las propiedades anteriormente citadas, sino que además es conveniente que soporte estas otras propiedades:

- **Concurrencia**, Es la propiedad que distingue un objeto activo de uno no activo. Concurrencia permite que diferentes objetos actúen al mismo tiempo, usando distintos hilos de control.
- **Persistencia**, Es la propiedad por la cual la existencia de un objeto trasciende en el tiempo (esto es, el objeto sigue existiendo después de que su creador deja de existir) o en el espacio (esto es, la localización del objeto cambia respecto a la dirección en la que fue creado).
- **Generalidad y manejo de excepciones**, Es la capacidad de poder manejar distintos errores, en cualquier parte del programa,[Joyanes 2000].

2.4.1 Características de la Programación Orientación a Objeto

La Programación Orientada a Objetos (POO) es un importante conjunto de técnicas que pueden utilizarse para hacer el desarrollo de programas más eficiente.

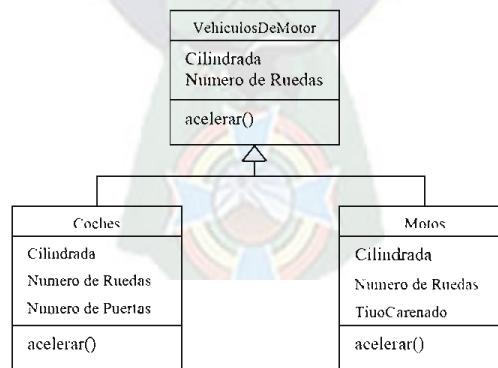
En la POO los objetos son los elementos principales de construcción,[Joyanes 2000].

A continuación los conceptos fundamentales de la POO:

- **Clase**: Es una descripción de un conjunto de objetos similares. Por ejemplo la clase Coches. Una clase contiene los atributos y las operaciones sobre esos atributos que hacen que una clase tenga la entidad que se desea;
- **Objeto**: Un objeto es una cosa, generalmente extraída del vocabulario del espacio del problema o del espacio de la solución. Todo objeto tiene un nombre (se le puede identificar), un estado (generalmente hay algunos datos asociados a él) y un comportamiento (se le pueden hacer cosas a objeto y él puede hacer cosas a otros objetos). Un objeto de la clase Coches puede ser un Ford Mustang;

- **Atributo:** Es una característica concreta de una clase. Por ejemplo atributos de la clase Coches pueden ser el Color, el Numero de Puertas;
- **Método:** Es una operación concreta de una determinada clase. Por ejemplo de la clase Coches podríamos tener un método arrancar() que lo que hace es poner en marcha el coche;
- **Instancia:** Es una manifestación concreta de una clase (un objeto con valores concretos). También se le suele llamar ocurrencia. Por ejemplo una instancia de la clase Coches puede ser: Un Ford Mustang, de color Gris con 3 puertas;
- **Herencia:** Es un mecanismo mediante el cual se puede crear una nueva clase partiendo de una existente, se dice entonces que la nueva clase hereda las características de la clase existentes aunque se le puede añadir más capacidades (añadiendo datos o capacidades) o modificar las que tiene. Por ejemplo supongamos que tenemos la clase VehiculosDeMotor. En esta clase tenemos los siguientes atributos: Cilindrada y Numero de Ruedas, y el método acelerar(). Mediante el mecanismo de herencia podemos definir la clase Coches y la clase Motos. Estas dos clases heredan los atributos Cilindrada y Numero de Ruedas de la clase VehículosDeMotor pero a su vez tendrán atributos propios (como hemos dicho antes el Numero de Puertas es un atributo propio de la clase Coches que no tienen sentido en la clase Motos). Se puede decir que Coches extiende la clase VehículosDeMotor, o que VehículosDeMotor es una **generalización** de las clases Coches y Motos;

Figura 2.2 Ejemplo de Herencia



Fuente:[Alarcón, 2000]

- **Polimorfismo:** Hace referencia a la posibilidad de que dos métodos implementen distintas acciones, aun teniendo el mismo nombre, dependiendo del objeto que lo

ejecuta o de los parámetros que recibe. En el ejemplo anterior teníamos dos objetos que heredaban el método acelerar() de la clase VehiculosDeMotor. De hecho en clase VehiculosDeMotor al ser general no tiene sentido que tenga una implementación concreta de este método. Sin embargo, en las clases Coches y Motos si que hay una implementación clara y distinta del método acelerar, [Alarcón 2000].

Ejemplo:

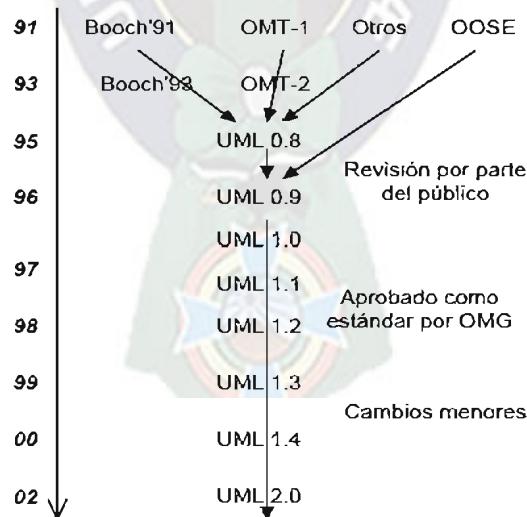
```
Coches::acelerar () {
    Pisar más el pedal derecho
}
Código fuente 1. Posible implementación para coches
```

```
Motos::acelerar () {
    Girar más el puño derecho
}
Código fuente 2. Implementación para motos
```

2.5 UML

UML fue desarrollado en un esfuerzo para simplificar y consolidar el gran número de métodos de desarrollo Orientado a Objetos que habían surgido.

Figura 2.3 Vista general de UML



Fuente: [Niosky, 2009]

El Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar

UML es un lenguaje estándar que sirve para escribir los *planos del software*, puede utilizarse para visualizar, especificar, construir y documentar todos los artefactos que componen un sistema con gran cantidad de software. UML puede usarse para modelar desde sistemas de información hasta aplicaciones distribuidas basadas en Web, pasando por sistemas empotrados de tiempo real. UML es solamente un lenguaje por lo que es sólo una parte de un método de desarrollo software, es independiente del proceso aunque para que sea óptimo debe usarse en un proceso dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental, [Alarcón, 2000].

2.5.1 Características de UML

El lenguaje UML se compone de tres elementos básicos, los bloques de construcción, las reglas y algunos mecanismos comunes. Estos elementos interaccionan entre sí para dar a UML el carácter de completitud y no-ambigüedad.

Los **bloques de construcción** se dividen en tres partes: **Elementos**, que son las abstracciones de primer nivel, **Relaciones**, que unen a los elementos entre sí, y los **Diagramas**, que son agrupaciones interesantes de elementos.

Existen cuatro tipos de elementos en UML, dependiendo del uso que se haga de ellos: elementos estructurales, elementos de comportamiento, elementos de agrupación y elementos de anotación.

Las relaciones, a su vez se dividen para abarcar las posibles interacciones entre elementos que se nos pueden presentar a la hora de modelar usando UML, estas son: relaciones de dependencia, relaciones de asociación, relaciones de generalización y relaciones de realización.

Se utilizan diferentes diagramas dependiendo de qué, nos interese representar en cada momento, para dar diferentes perspectivas de un mismo problema, para ajustar el nivel de detalle..., por esta razón UML soporta un gran número de diagramas diferentes aunque, en la práctica, sólo se utilicen un pequeño número de combinaciones, [Alarcón, 2000].

En UML 2.0 hay 13 tipos diferentes de diagramas. Para comprenderlos de manera concreta, a veces es útil categorizarlos jerárquicamente.

Los **Diagramas de Estructura** enfatizan en los elementos que deben existir en el sistema modelado:

- **Diagrama de clases.** Es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos;

- **Diagrama de componentes.** Representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes;
- **Diagrama de objetos.** Son utilizados durante el proceso de Análisis y Diseño de los sistemas informáticos en la metodología UML;
- **Diagrama de estructura compuesta (UML 2.0).** Un Diagrama de estructura compuesta es un tipo de *diagrama de estructura estática* (UML);
- **Diagrama de despliegue.** Es un tipo de diagrama del UML que se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes;
- **Diagrama de paquetes.** En UML, un diagrama de paquetes muestra como un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones;

Los *Diagramas de Comportamiento* enfatizan en lo que debe suceder en el sistema modelado:

- **Diagrama de actividades.** Un diagrama de actividades representa los flujos de trabajo paso a paso de negocio y operacionales de los componentes en un sistema;
- **Diagrama de casos de uso.** En el UML, un diagrama de casos de uso es una especie de diagrama de comportamiento;
- **Diagrama de estados.** Se usan para representar gráficamente máquinas de estados finitos. Las Tablas de Transiciones son otra posible representación;

Los *Diagramas de Interacción* son un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado:

- **Diagrama de secuencia.** Es uno de los diagramas más efectivos para modelar interacción entre objetos en un sistema;
- **Diagrama de comunicación,** Que es una versión simplificada del Diagrama de colaboración (UML 1);
- **Diagrama de tiempos (UML 2.0).** Un diagrama de tiempos o cronograma es una gráfica de formas de onda digitales que muestra la relación temporal entre varias señales, y cómo varía cada señal en relación a las demás;
- **Diagrama global de interacciones o Diagrama de vista de interacción (UML 2.0).** Un diagrama de comportamiento, más precisamente, uno de los cuatro diagramas de interacción. Muestra una cierta vista sobre los aspectos dinámicos de los sistemas modelados, [Rumbaugh, 2000].

2.6 LENGUAJE DE PROGRAMACION

2.6.1 PHP

PHP es un lenguaje de programación, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica.

El código PHP se incluye entre etiquetas especiales de comienzo y final que nos permitirán entrar y salir del modo PHP.

Lo que distingue a PHP de la tecnología Javascript, la cual se ejecuta en la máquina cliente, es que el código PHP es ejecutado en el servidor. Si tuviésemos un script similar al de nuestro ejemplo en nuestro servidor, el cliente solamente recibiría el resultado de su ejecución en el servidor, sin ninguna posibilidad de determinar qué código ha producido el resultado recibido. El servidor web puede ser incluso configurado para que procese todos los ficheros HTML con PHP.

-Características de PHP

Al nivel más básico, PHP puede hacer cualquier cosa que se pueda hacer con un script CGI, como procesar la información de formularios, generar páginas con contenidos dinámicos, o mandar y recibir cookies.

Quizás la característica más potente y destacable de PHP es su soporte para una gran cantidad de bases de datos. Escribir un interfaz vía web para una base de datos es una tarea simple con PHP. Las siguientes bases de datos están soportadas actualmente:

Adabas D Ingres Oracle (OCI7 and OCI8)

dBase InterBase PostgreSQL;

Empress FrontBase Solid;

FilePro mSQL Sybase;

IBM DB2 MySQL Velocis;

Informix ODBC Unix dbm;

PHP también soporta el uso de otros servicios que usen protocolos como IMAP, SNMP, NNTP, POP3, HTTP y derivados.

También se pueden abrir sockets de red directos (raw sockets) e interactuar con otros protocolos, [Saether, 2001].

2.6.2 JavaScript

JavaScript es un lenguaje, utilizado para acceder a objetos en aplicaciones. Es utilizado, principalmente, integrado en un navegador web permitiendo el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas. JavaScript ha tenido influencia de múltiples lenguajes y fue diseñado para tener una sintaxis similar a Java, aunque más fácil de utilizar para programadores principiantes.

Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas web. El lenguaje fue inventado por Brendan Eich en la empresa Netscape Communications, que es la que desarrolló los primeros navegadores web comerciales.

Apareció por primera vez en el producto de Netscape llamado Netscape Navigator 2.0, [Camilo 2009].

2.6.2.3 jQuery

jQuery es una biblioteca de Javascript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web. Fue presentada en enero de 2006.

jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en Javascript que de otra manera requerirían de mucho más código. Es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio, [ArthurBot, 2009].

2.7 BASES DE DATOS RELACIONALES.

Una base de datos Relacionales es una Base de Datos en donde todos los datos visibles al usuario están organizados como tablas de valores, en donde todas las operaciones se realizan sobre estas tablas. Estas Bases de Datos son percibidas por el usuario como una colección de relaciones normalizadas de diversos grados que varían con

el tiempo. El modelo Relacional representa un sistema de Base de Datos en un nivel de abstracción un tanto alejado de los detalles de la maquina subyacente.

De hecho, el modelo relacional puede considerarse como un lenguaje de programación más bien abstracto orientado de manera específica hacia las aplicaciones de base de datos, [Data, 2001].

2.9 SISTEMAS DE MANEJO DE BASE DE DATOS (DBMS)

El (DBMS) es considerado la parte modular de la Base de Datos que permite la creación, modificación y actualización de la base de datos, la recuperación de datos y la generación de reportes.

Entre los objetivos de una DBMS sobresalen:

- Asegurarse de que la base de datos pueda ser compartida entre los usuarios de una diversidad de aplicaciones;
- Asegurar que todos los datos requeridos para las aplicaciones actuales y futuras estén fácilmente disponibles;
- Permitir que los usuarios construyan su vista personal de los datos sin preocuparse de la forma en que estén físicamente guardados los datos;
- Manejar datos que sean precisos y consistentes;
- Respaldo, los DBMS deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder, [Data, 2001].

2.9.3 MySQL

Es un sistema de gestión de base de datos relacional, multiusuario con más de seis millones de instalaciones. MySQL desde enero de 2008 una subsidiaria de Sun Microsystems desarrolla MySQL como software libre en un esquema de licenciamiento dual.

Por un lado se ofrece bajo la **GNU GPL** para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y el copyright del código está en poder del autor individual, MySQL es

propietario y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet, MySQL AB fue fundado por David Axmark, Allan Larsson y Michael Widenius.

SQL (*Lenguaje de Consulta Estructurado*) fue comercializado por primera vez en 1981 por IBM, el cual fue presentado a ANSI y desde ese entonces ha sido considerado como un estándar para las bases de datos relacionales. Desde 1986, el estándar SQL ha aparecido en diferentes versiones como por ejemplo: SQL:92, SQL:99, SQL:2003. MySQL es una idea originaria de la empresa opensource MySQL AB establecida inicialmente en Suecia en 1995 y cuyos fundadores son David Axmark, Allan Larsson, y Michael "Monty" Widenius.

El objetivo que persigue esta empresa consiste en que MySQL cumpla el estándar SQL, pero sin sacrificar velocidad, fiabilidad o usabilidad.

La procedencia del nombre de MySQL no es clara. Desde hace más de 10 años, las herramientas han mantenido el prefijo My. También, se cree que tiene relación con el nombre de la hija del cofundador Monty Widenius quien se llama *My*.

Por otro lado, el nombre del delfín de MySQL es Sakila y fue seleccionado por los fundadores de MySQL AB en el concurso “Name the Dolphin”. [Isha, 2009]

2.10 ESTRUCTURA DE DATOS

2.10.1 Arrays Unidimensionales: Los Vectores

Un array (matriz o vector) es un conjunto finito y ordenado de elementos homogéneos. La propiedad “ordenado” significa que el elemento primero, segundo, tercero,... enésimo de un array puede ser identificado. Los elementos de un array son homogéneos, es decir del mismo tipo de datos.

El tipo más simple de array es el array unidimensional o vector (matriz de una dimensión). Un vector de una dimensión denominado NOTAS que consta de n elementos se puede representar por la Figura (2.4), [Joyanes 2003].

Figura 2.4 Vectores

NOTAS (1)	NOTAS(2)	NOTAS(I)	NOTAS(N)
-----------	----------	-------	----------	-------	----------

Fuente: [Joyanes, 2003]

2.10.2 Array Bidimensionales (Tablas/Matrices)

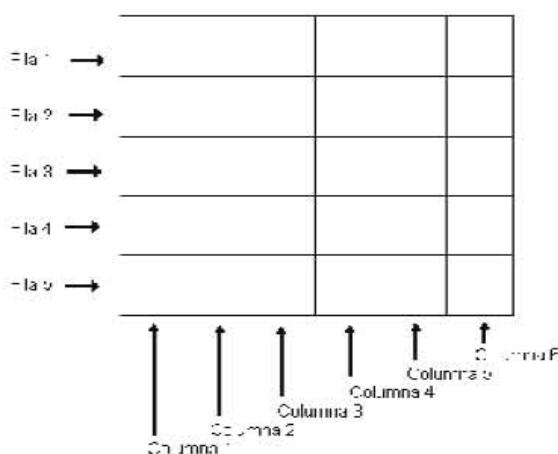
El array bidimensional se puede considerar como un vector de vectores. Es, por consiguiente, un conjunto de elementos, todos del mismo tipo, en el cual el orden de los componentes es significativo, y en el que se necesita especificar dos subíndices para poder identificar cada elemento del array.

Si se visualiza un array unidimensional, se puede considerar como una columna de datos; un array bidimensional es un grupo de columnas, como se ilustra en la Figura(2.5).

El diagrama representa una tabla o matriz de treinta elementos (5×6) con 5 filas y 6 columnas.

Como en un vector de treinta elementos, cada uno de ellos tiene el mismo nombre. Sin embargo, un subíndice no es suficiente para especificar un elemento de un array bidimensional; por ejemplo, si el nombre del array es M, no se puede indicar M[3]. Para evitar la ambigüedad, los elementos de un array bidimensional se referencia con dos subíndices, el primer subíndice se refiere a la fila y el segundo subíndice se refiere se refiere a la columna. Por consiguiente, M[2,3] se refiere al elemento de la segunda fila, tercera columna [Joyanes 2003].

Figura 2.5 Matriz



Fuente: [Joyanes, 2003]

Un array bidimensional M, también denominado matriz (términos matemáticos) o tabla (términos financieros), se considera que tiene dos dimensiones (una dimensión por cada subíndice) y necesita un valor para cada subíndice para poder identificar un elemento individual. En notación estándar, normalmente el primer subíndice se refiere a la fila del array, mientras que el segundo subíndice se refiere a la columna del array.

Es decir $B[I, J]$ es el elemento de B que ocupa la I^a fila y la J^a columna como se indica en la Figura (2.6).

Figura 2.6 Matrices Ejemplo (subíndices)

	1	2	3	4	...	J	...	N
1								
2								
...								
I						B[I, J]		
...								
M								

Fuente: [Joyanes, 2003]

2.10.3 Listas

Una lista es un conjunto de elementos de un tipo dado que pueden variar en número y donde cada elemento tiene un único predecesor y un único sucesor o siguiente, excepto el primero y último de la lista. Esta es una definición muy general que incluye los ficheros y vectores. Los elementos de una lista lineal se almacenan contiguos (un elemento detrás de otro) en posiciones consecutivas de la memoria, [Joyanes 2003].

Las operaciones que se pueden realizar con listas lineales son:

1. Insertar eliminar o localizar un elemento;
2. determinar el tamaño –número de elementos - de la lista;
3. Recorrer la lista para poder localizar un determinado elemento;
4. Clasificar los elementos de la lista en forma ascendente o descendente;
5. Unir dos o más listas en una sola.

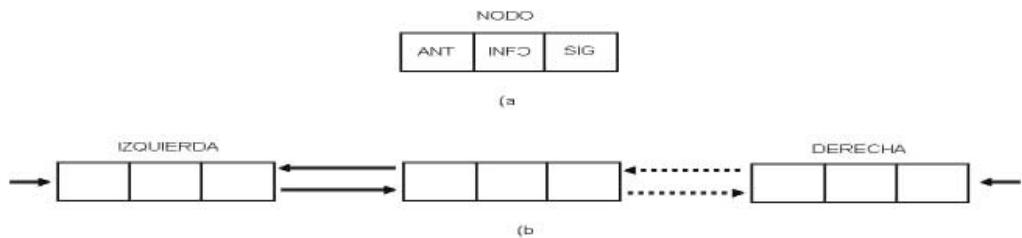
2.10.4 Listas Dblemente Enlazadas

Las listas que pueden recorrerse en ambas direcciones se denominan listas doblemente enlazadas.

En estas listas cada nodo consta del campo INFO de datos y dos campos de enlace o punteros: ANTERIOR(ANT) y SIGUIENTE(SIG) que apuntan hacia delante y hacia atrás Figura (2.7). Como cada elemento tiene dos punteros, una lista doblemente enlazada ocupa más espacio en memoria que una lista simplemente enlazada para una misma cantidad de información. La lista necesita dos punteros CABECERA y FIN que apunta hacia el primero y ultimo nodo.

La variable CABECERA y el puntero SIG permiten recorrer la lista en el sentido normal y la variable FIN y el puntero ANT permiten recorrerla en el sentido inverso, [Joyanes 2003].

Figura 2.7 Lista Dblemente Enlazada



Fuente: [Joyanes 2003]

2.11 INTERFAZ GRAFICA DE USUARIO

2.11.1 Diseño de la Interfaz de Usuario

El diseño de la interfaz se centra en tres áreas de interés: (1) el diseño de la interfaz entre los componentes del software; (2) el diseño de las interfaces entre el software y los otros productores y consumidores de información no humanos (esto es, otras entidades externas) y (3) el diseño de la interfaz entre el hombre (esto es, el usuario) y la computadora, [Pressman, 2002].

- **El diseño de la interfaz entre los componentes del software.** Este diseño consiste en convertir el diseño de datos, interfaces y arquitectura en un software operacional. Para poderlo llevar a cabo, el diseño se deberá representar a un nivel de abstracción

cercano a un código. El diseño a nivel de componentes establece los datos algorítmicos que se requieren para manipular las estructuras de datos, efectuar la comunicación entre los componentes del software por medio de las interfaces. e implementar los algoritmos signados a cada componente.

- **El diseño de las interfaces entre el software y los otros productores y consumidores de información no humanos.** Este punto se refiere a los distintos actores que entran en contacto con el sistema de manera directa o indirecta en función a diferentes requisitos necesarios (no humanos).
- **El diseño de la interfaz entre el hombre y la computadora.** Se puede argumentar que la interfaz de usuario es el elemento más importante de un sistema o producto basado en computadora. Si la interfaz tiene un diseño pobre, la capacidad que tiene el usuario de aprovecharse de la potencia de proceso de una aplicación se puede dificultar gravemente. En efecto, una interfaz débil puede llevar al fracaso de una aplicación con una implementación sólida y un buen diseño. Existen tres principios importantes que dirigen el diseño de interfaces de usuario eficaces: (1) poner el control en manos del usuario; (2) reducir la carga de la memoria del usuario; (3) construir una interfaz consecuente. Para lograr que una interfaz se atenga a estos principios, se deberá desarrollar un proceso de diseño organizado. El diseño de la interfaz de usuario comienza con la identificación de los requisitos del usuario, de las tareas y del entorno. El análisis de tareas es una actividad de diseño que define las tareas y acciones del usuario empleando un enfoque elaborativo u orientado a objetos, [Pressman, 2002].

- Evaluación del Diseño de Interfaz

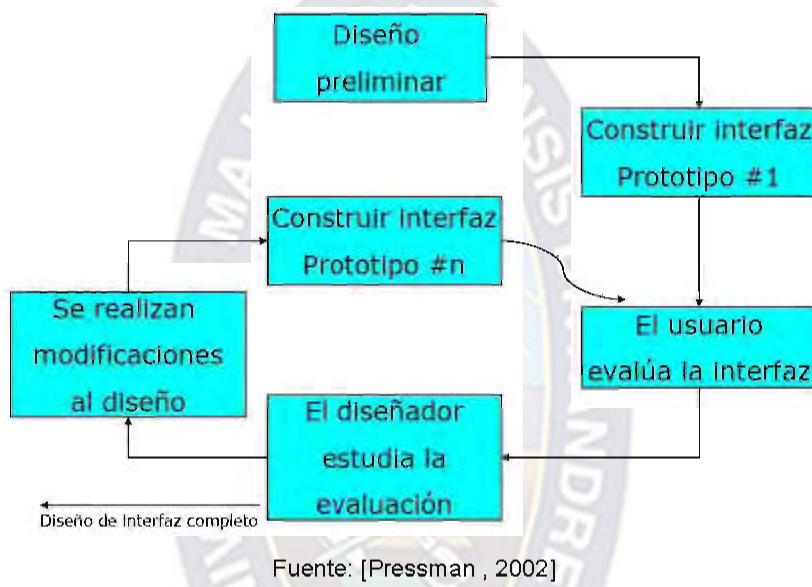
Una vez que se ha creado un prototipo de interfaz de usuario, deberá sufrir una evaluación para determinar si cumple las necesidades del usuario. La evaluación podrá abarcar un espectro de formalidad: desde «pruebas» informales en donde el usuario proporciona respuestas espontáneas hasta un estudio formalmente diseñado que utilizará métodos estadísticos para la evaluación de cuestionarios cumplimentados por un grupo de usuarios finales.

El ciclo de evaluación de la interfaz adquiere forma en la Figura 2.8. Una vez finalizado el modelo de diseño, se crea un prototipo de primer nivel. Este prototipo es

evaluado por el usuario, que es quien proporcionará al diseñador los comentarios directos sobre la eficacia de la interfaz.

Las modificaciones que se realicen sobre el diseño se basarán en la entrada del usuario y entonces se creará el prototipo de segundo nivel. El ciclo de evaluación continúa hasta que ya no sean necesarias más modificaciones del diseño de la interfaz,[Pressman, 2002].

Figura 2.8 Evaluación del diseño de interfaz



- Métricas de Diseño de Interfaz

Aunque existe una significativa cantidad de literatura sobre el diseño de interfaces hombre-máquina, se ha publicado relativamente poca información sobre métricas que proporcionen una visión interna de la calidad y facilidad de empleo de la interfaz.

Sears [SEA 93] sugiere la conveniencia de la representación (CR) como una valiosa métrica de diseño para interfaces hombre-máquina. Una IGU (Interfaz Gráfica de Usuario) típica usa entidades de representación -iconos gráficos, texto, menús, ventanas y otras- para ayudar al usuario a completar tareas. Para realizar una tarea dada usando una IGU, el usuario debe moverse de una entidad de representación a otra. Las posiciones absolutas y relativas de cada entidad de representación, la frecuencia con que se utilizan y el «coste» de la transición de una entidad de representación a la siguiente contribuirán a la conveniencia de la interfaz.

Para una representación específica (por ejemplo, un diseño de una IGU específica), se pueden asignar costes a cada secuencia de acciones de acuerdo con la siguiente relación:

$$\text{Costes} = \sum [\text{frecuencia de transición } (k) \times$$

$$X \text{ coste de transición } (k)]$$

Donde k es la transición específica de una entidad de representación a la siguiente cuando se realiza una tarea específica. Esta suma se da con todas las transiciones de una tarea en particular o conjunto de tareas requeridas para conseguir alguna función de la aplicación. El coste puede estar caracterizado en términos de tiempo, retraso del proceso o cualquier otro valor razonable, tal como la distancia que debe moverse el ratón entre entidades de la representación.

La conveniencia de la representación se define como:

$$CR = 100 \times [(\text{coste de la representación Óptima CR}) /$$

$$/ (\text{coste de la representación propuesta})]$$

donde $CR = 100$ para una representación Óptima.

Para calcular la representación óptima de una IGU, la superficie de la interfaz (el área de la pantalla) se divide en una cuadrícula. Cada cuadro de la cuadricula representa una posible posición de una entidad de la representación. Para una cuadrícula con N posibles posiciones y K diferentes entidades de representación para colocar, el número posible de distribuciones se representa de la siguiente manera [SEA93]:

$$\text{número posible de distribuciones} = [N! / (K! \times (N - K)!)] \times K!$$

A medida que crece el número de posiciones de representación, el número de distribuciones posibles se hace muy grande. Para encontrar la representación óptima (menor coste). Sears [SEA93] propone un algoritmo de búsqueda en árbol.

La **CR** se emplea para valorar diferentes distribuciones propuestas de IGU y la sensibilidad de una representación en particular a los cambios en las descripciones de tareas (por ejemplo, cambios en la secuencia y/o frecuencia de transiciones). El diseñador de interfaces puede emplear el cambio en la conveniencia de la representación, **ACR**, como guía en la elección de la mejor representación de IGU para una aplicación en particular. Es importante apuntar que la selección de un diseño de IGU puede guiarse con métricas tales como **CR**, pero el árbitro final debería ser la respuesta del usuario basada en prototipos de IGU. Nielsen y Levy [NIE94] informan que «existe una gran posibilidad de éxito si se elige una interfaz basándose solamente en la opinión del usuario. El rendimiento medio de tareas de usuario y su satisfacción con la IGU están altamente relacionadas.», [Pressman, 2002].

2.12 ESTRATEGIA DE PRUEBA

Las pruebas del software son un elemento crítico para la garantía de calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación.

La creciente percepción del software como un elemento del sistema y la importancia de los «costes» asociados a un fallo del propio sistema, están motivando la creación de pruebas minuciosas y bien planificadas. No es raro que una organización de desarrollo de software emplee entre el 30 y el 40 por ciento del esfuerzo total de un proyecto en las pruebas. En casos extremos, las pruebas del software para actividades críticas (por ejemplo, control de tráfico aéreo, control de reactores nucleares) puede costar ide tres a cinco veces más que el resto de los pasos de la ingeniería del software juntos! ,[Pressman, 2002].

2.12.1 Pruebas de Caja Negra

Las pruebas de caja negra, también denominada prueba de comportamiento, se centran en los requisitos funcionales del software. O sea, la prueba de caja negra permite al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa.

La prueba de caja negra intenta encontrar errores de las siguientes categorías: (1) funciones incorrectas o ausentes, (2) errores de interfaz, (3) errores en estructuras de datos o en accesos a bases de datos externas,(4) errores de rendimiento y (5) errores de inicialización y de terminación. Ya que la prueba de caja negra ignora intencionadamente la estructura de control, centra su atención en el campo de la información,[Pressman, 2002].

2.12.2 Pruebas de Caja Blanca

La prueba de Caja Blanca, denominada a veces *prueba de caja de cristal* es un método de diseño de casos de prueba, para obtener los casos de prueba. Mediante los métodos de prueba de caja blanca, el ingeniero del software puede obtener casos de prueba que (1) garanticen que se ejercita por lo menos una vez todos los caminos independientes de cada módulo; (2) ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa; (3) ejecuten todos los bucles en sus límites y con sus límites operacionales; y (4) ejerciten las estructuras internas de datos para asegurar su validez.

En este momento, se puede plantear un pregunta razonable: ¿Por qué emplear tiempo y energía preocupándose de (y probando) las minuciosidades lógicas cuando podríamos emplear mejor el esfuerzo asegurando que se han alcanzado los requisitos del programa? O, dicho de otra forma, ¿por qué no empleamos todas nuestras energías en la prueba de caja negra? La respuesta se encuentra en la naturaleza misma de los defectos del software, por ejemplo:

- Los errores lógicos y las suposiciones incorrectas son inversamente proporcionales a la probabilidad de que se ejecute un camino del programa. Los errores tienden a introducirse en nuestro trabajo cuando diseñamos e implementamos funciones, condiciones o controles que se encuentran fuera de lo normal. El procedimiento habitual tiende a hacerse más comprensible (y bien examinado), mientras que el procesamiento de «casos especiales» tiende a caer en el caos;
- A menudo creemos que un camino lógico tiene pocas posibilidades de ejecutarse cuando, de hecho, se puede ejecutar de forma normal. El flujo lógico de un programa a veces no es nada intuitivo, lo que significa que nuestras suposiciones intuitivas sobre el flujo de control y los datos nos pueden llevar a tener errores de diseño que sólo se descubren cuando comienza la prueba del camino;
- Los errores tipográficos son aleatorios. Cuando se traduce un programa a código fuente en un lenguaje de programación, es muy probable que se den algunos errores de escritura. Muchos serán descubiertos por los mecanismos de comprobación de sintaxis, pero otros permanecerán sin detectar hasta que comience la prueba. Es igual de probable que haya un error tipográfico en un oscuro camino lógico que en un camino principal, [Pressman, 2002].

2.13 CRIPTOGRAFIA

Las raíces etimológicas de la palabra criptografía son *Kryptos*, que significa oculto, y *graphos*, que se traduce como escribir, lo que da una clara idea de su definición clásicamente de escribir mensajes en clave secreta o enigmáticamente.

Fue considerada un arte hasta que Shannon publicó en 1949 la *Teoría de las comunicaciones secretas*. Entonces la Criptografía fue considerada una ciencia aplicada, debido a su relación con otras ciencias, como la estadística, la teoría de los números, la teoría de la información y la teoría de la complejidad computacional.

Ahora bien la criptografía corresponde solo a una parte de la comunicación secreta. Si se requiere secreto para la comunicación es porque existe desconfianza y peligro de que el mensaje transmitido sea interceptado por un enemigo. Este enemigo, si existe, utilizará todos los medios a su alcance para descifrar esos mensajes secretos mediante un conjunto de técnicas y métodos que constituyen una ciencia conocida como criptoanálisis. Al conjunto de ambas ciencias criptografía y criptoanálisis, se le denomina criptología.

Como en muchas otras áreas científicas el mayor desarrollo de la criptología tuvo lugar durante las dos guerras mundiales. En este caso se debió a la necesidad de establecer comunicaciones secretas militares y diplomáticas utilizando nuevas tecnologías, como la telegráfica y la radiotécnica, [Caballero 2003].

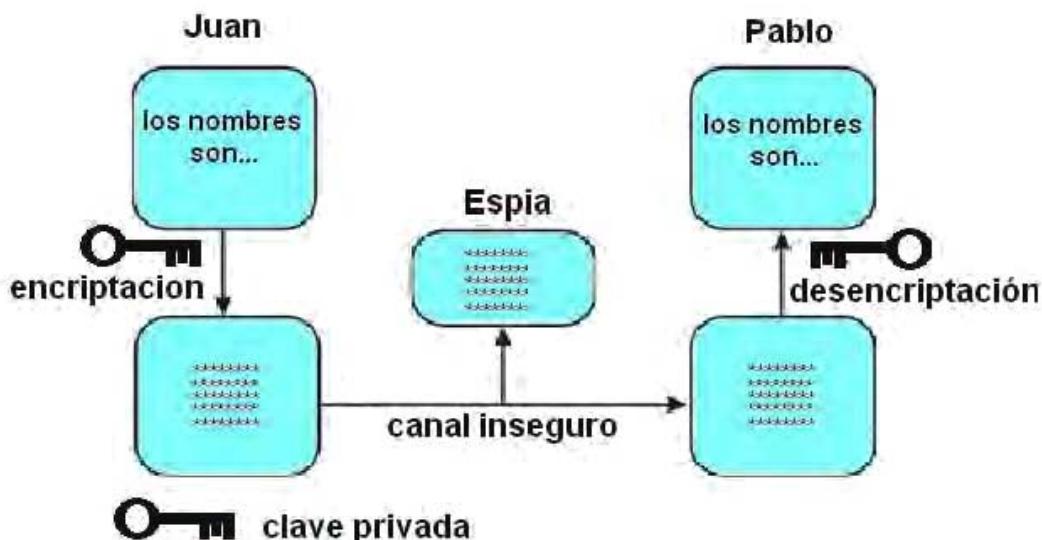
Los métodos de encriptación pueden dividirse en dos grupos:

- Clave secreta (simétrica)
- Clave pública (asimétrica)

2.13.1 Criptografía de Clave Secreta (simétrica)

En los criptosistemas de clave secreta, la clave de cifrado es un secreto compartido exclusivamente por emisor y receptor, pero también conocida por el enemigo, [Caballero 2003].

Figura 2.9 Clave Privada



2.13.1.1 Cifrado de Vernam

Es una sustitución polialfabética, dada una sucesión finita de variables aleatorias $\{K_i\}_{i=0,1,2,\dots,n-1}$ independientes e idénticamente distribuidas según una distribución equiprobable sobre Z_m , el cifrado Vernam o de cinta aleatoria cifra el mensaje $X=(X_0,\dots,X_{n-1})$ según la transformación $Y_i=T_i(X_i)=K_i+X_i \pmod m$ $0 \leq i < n$.

Las variables aleatorias Y_0,\dots,Y_{n-1} tambien son independientes e idénticamente distribuidas según una distribución equiprobable sobre Z_m , [Caballero 2003].

2.13.1.2 Algoritmo Rijndael

En el año 2000 el NIST (National Institute for Standards and Technology) estadounidense declaro como nuevo estándar avanzado de cifrado el conocido Rijndael, por las iniciales de sus autores Rijmen y Daemen.

La adopción del nuevo estándar fue realizada después de una convocatoria pública en el que se propusieron y analizaron abiertamente varios algoritmos.

Este cifrado trabaja con longitudes de clave y bloque, variables en ambos casos entre 128 y 256, de ahí que el criptoanálisis exhaustivo quede descartado. También el número de iteraciones es flexible.

Todas las operaciones realizadas en el cifrado son inversibles, por lo que el descifrado consiste en utilizar el mismo algoritmo con las inversas de las operaciones originales aplicadas en orden inverso, [Caballero 2003].

2.13.2 Criptografía de Clave Pública (asimétrica).

Con un sistema asimétrico cualquier usuario puede enviar un mensaje cifrado a otro usando la clave publica de este ultimo, pero solo aquellos que conozcan la clave secreta correspondiente pueden descifrar correctamente, [Caballero 2003].

2.13.2.1 Sistema de ElGamal

El sistema de ElGamal tiene un funcionamiento distinto al resto de los sistemas de clave pública, ya que el cifrado se realiza utilizando además de la clave pública del receptor la clave secreta del emisor.

Una descripción del algoritmo:

-Serán de información pública un numero primo grande p y a, una raíz primitiva de un solo módulo p (es decir, tal que $a^{p-1} \equiv 1 \pmod{p}$ y $a^{d-1} \not\equiv 1 \pmod{p}$ para todo d tal que $1 < d < p$);

- La clave privada del usuario B es un entero K_b escogido dentro del intervalo $[1, p-1]$;

-La clave pública de B es un entero $K_b \equiv a^{K_b} \pmod{p}$;

-Se supone que el emisor A quiere enviar un mensaje $M (1 \leq M \leq p-1)$ al receptor B;

El proceso de cifrado que se debe llevar a cabo consiste en:

1. Escoger aleatoriamente un entero K_a tal que $1 \leq K_a \leq p-1$, que constituye su clave secreta;
2. Calcular la clave de cifrado a partir de su propia clave privada y la clave pública de B, $Q \equiv (K_b^{(K_a)}) \pmod{p}$;
3. Cifrar el mensaje M según la expresión $C \equiv QM \pmod{p}$;

El proceso de descifrado que lleva B, consiste en:

1. Obtener Q gracias a la clave pública de A, K_a , y a su propia clave secreta mediante la fórmula $Q \equiv (K_a^{(K_b)}) \pmod{p}$;
2. Recuperar M a partir de $M \equiv (Q^{p-1})^{-1} pC \pmod{p}$ donde $(Q^{p-1})^{-1}$ denota el inverso de Q en módulo p; [Caballero 2003].

2.13.2.2 MD5

MD5 (abreviatura de *Message-Digest Algorithm 5*, Algoritmo de Resumen del Mensaje 5) es uno de los algoritmos de reducción criptográficos diseñados por el profesor Ronald Rivest del MIT (Massachusetts Institute of Technology, Instituto Tecnológico de Massachusetts). Fue desarrollado en 1991 como reemplazo del algoritmo MD4.

El md5 se utiliza extensamente en el mundo del software para proporcionar la seguridad de que un archivo descargado de Internet no se ha alterado. Comparando una suma MD5 publicada con la suma de comprobación del archivo descargado, un usuario puede tener la confianza suficiente de que el archivo es igual que el publicado por los desarrolladores. Esto protege al usuario contra los 'Caballos de Troya' o 'Troyanos' y virus que algún otro usuario malicioso pudiera incluir en el software. La comprobación de un archivo descargado contra su suma MD5 no detecta solamente los archivos alterados de una manera maliciosa, también reconoce una descarga corrupta o incompleta, [Aloriel, 2009].

2.14 ARQUITECTURA CLIENTE/SERVIDOR

Esta arquitectura consiste básicamente en un cliente que realiza peticiones al servidor, el cual le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema multiusuario distribuido a través de una red de computadoras.

2.14.1 Arquitectura cliente-servidor de tres niveles

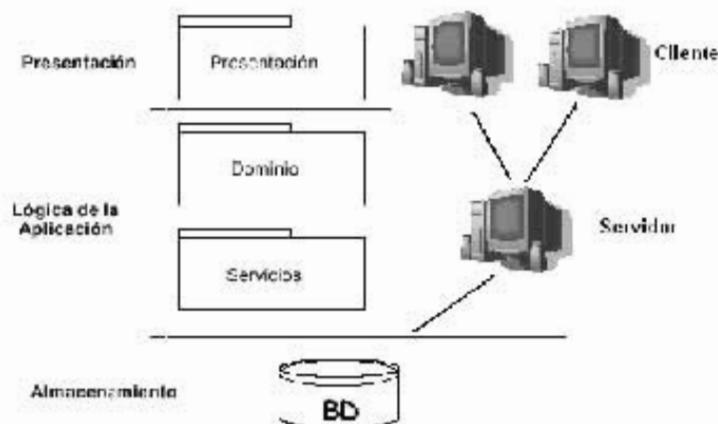
La llamada "Arquitectura en Tres Niveles", es la más común en sistemas de información.

Una descripción de los tres niveles sería la siguiente:

- Nivel 1: Presentación, ventanas, informes, etc.
- Nivel 2: Lógica de la Aplicación, tareas y reglas que gobiernan el proceso.
- Nivel 3: Almacenamiento, mecanismo de almacenamiento.

El nivel 1 tiene relación directa con el cliente y el nivel 2 corresponde al servidor por su parte el nivel 3 tiene relación con la Base de Datos, [Ferre, 2005]

Figura 2.10: Arquitectura de tres niveles



Fuente: [Ferre, 2005]

2.15 CALIDAD

ISO/9126 es un estándar internacional para la evaluación del Software.

El estándar está dividido en cuatro partes las cuales dirigen, respectivamente, lo siguiente: modelo de calidad, métricas externas, métricas internas y calidad en las métricas de uso.

El modelo de calidad establecido en la primera parte del estándar, ISO 9126-1, clasifica la calidad del software en un conjunto estructurado de características y subcaracterísticas de la siguiente manera:

- Funcionalidad - Un conjunto de atributos que se relacionan con la existencia de un conjunto de funciones y sus propiedades específicas. Las funciones son aquellas que satisfacen lo indicado o implica necesidades.
 - Idoneidad
 - Exactitud
 - Interoperabilidad
 - Seguridad
 - Cumplimiento de normas.
- Fiabilidad - Un conjunto de atributos relacionados con la capacidad del software de mantener su nivel de prestación bajo condiciones establecidas durante un período de tiempo establecido.
 - Madurez
 - Recuperabilidad
 - Tolerancia a fallos
- Usabilidad - Un conjuntos de atributos relacionados con el esfuerzo necesario para el uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios.
 - Aprendizaje
 - Comprensión
 - Operatividad
 - Atractividad
- Eficiencia - Conjunto de atributos relacionados con la relación entre el nivel de desempeño del software y la cantidad de recursos necesitados bajo condiciones establecidas.
 - Comportamiento en el tiempo

- Comportamiento de recursos
- Mantenibilidad - Conjunto de atributos relacionados con la facilidad de extender, modificar o corregir errores en un sistema software.
 - Estabilidad
 - Facilidad de análisis
 - Facilidad de cambio
 - Facilidad de pruebas
- Portabilidad - Conjunto de atributos relacionados con la capacidad de un sistema software para ser transferido desde una plataforma a otra.
 - Capacidad de instalación
 - Capacidad de reemplazamiento
 - Adaptabilidad
 - Co-Existencia

El estándar provee un entorno para que las organizaciones definan un modelo de calidad para el producto software. Haciendo esto así, sin embargo, se lleva a cada organización la tarea de especificar precisamente su propio modelo. Esto podría ser hecho, por ejemplo, especificando los objetivos para las métricas de calidad las cuales evalúan el grado de presencia de los atributos de calidad.

Métricas internas son aquellas que no dependen de la ejecución del software (medidas estáticas).

Métricas externas son aquellas aplicables al software en ejecución.

La calidad en las métricas de uso están sólo disponibles cuando el producto final es usado en condiciones reales.

Idealmente, la calidad interna determina la calidad externa y esta a su vez la calidad en el uso.

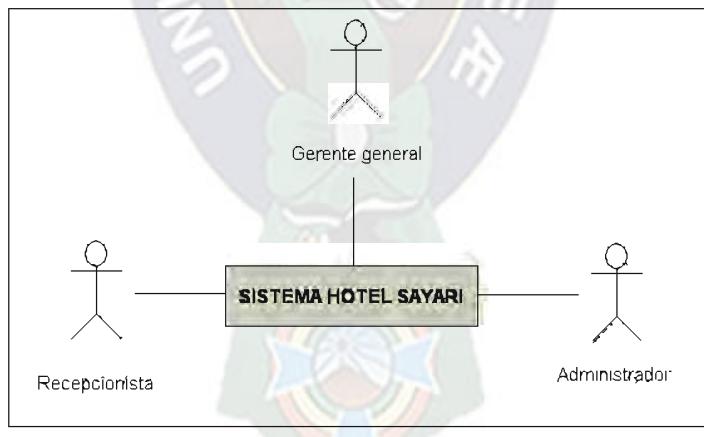
ISO 9126 distingue entre fallo y no conformidad. Un fallo es el incumplimiento de los requisitos previos, mientras que la no conformidad es el incumplimiento de los requisitos especificados. Una distinción similar es la que se establece entre validación y verificación, [Pressman, 2002].

3. MARCO PRÁCTICO

3.1 SISTEMA FÍSICO ACTUAL

En la Figura 3.1 se presenta la estructura Orgánica del Hotel, realizando un análisis de los distintos departamentos, Recepción y Gerencia a continuación se detalla los actores que intervienen en el Sistema actual.

Figura 3.1. Actores del Sistema



A continuación se Realiza la descripción de cada una de las funciones:

Actor: Recepcionista

Descripción, Es el encargado de controlar todo el movimiento de los huéspedes durante su hospedaje, cumple las siguientes funciones:

- Recibe a los Huéspedes;

- Ofrece información acerca de las Promociones y Servicios que ofrece el Hotel;
- Registra a los Huéspedes (con o sin reserva) que llegan al hotel, adjuntándolos a la tarjeta de registro de huéspedes;
- Realiza la Reserva que solicita algún Huésped, o Empresa Turística;
- Recepción de llamadas;
- Control de las llaves de la habitación;
- Llena la tarjeta de servicios, con los servicios que adquirió el huésped;
- Efectúa la facturación y realiza los reportes;
- Genera Reportes;

Actor: Administrador

Descripción, Es el encargado del manejo control y administración del hotel, realiza las siguientes funciones:

- Coordina y supervisa las labores del personal;
- Mantiene informado al personal de las actividades y disposiciones de la gerencia;
- Administra los reportes del material de consumo, servicios, caja y habitación;
- Administra la Caja de Seguridad;
- Contrata Personal;

Actor: Gerente General

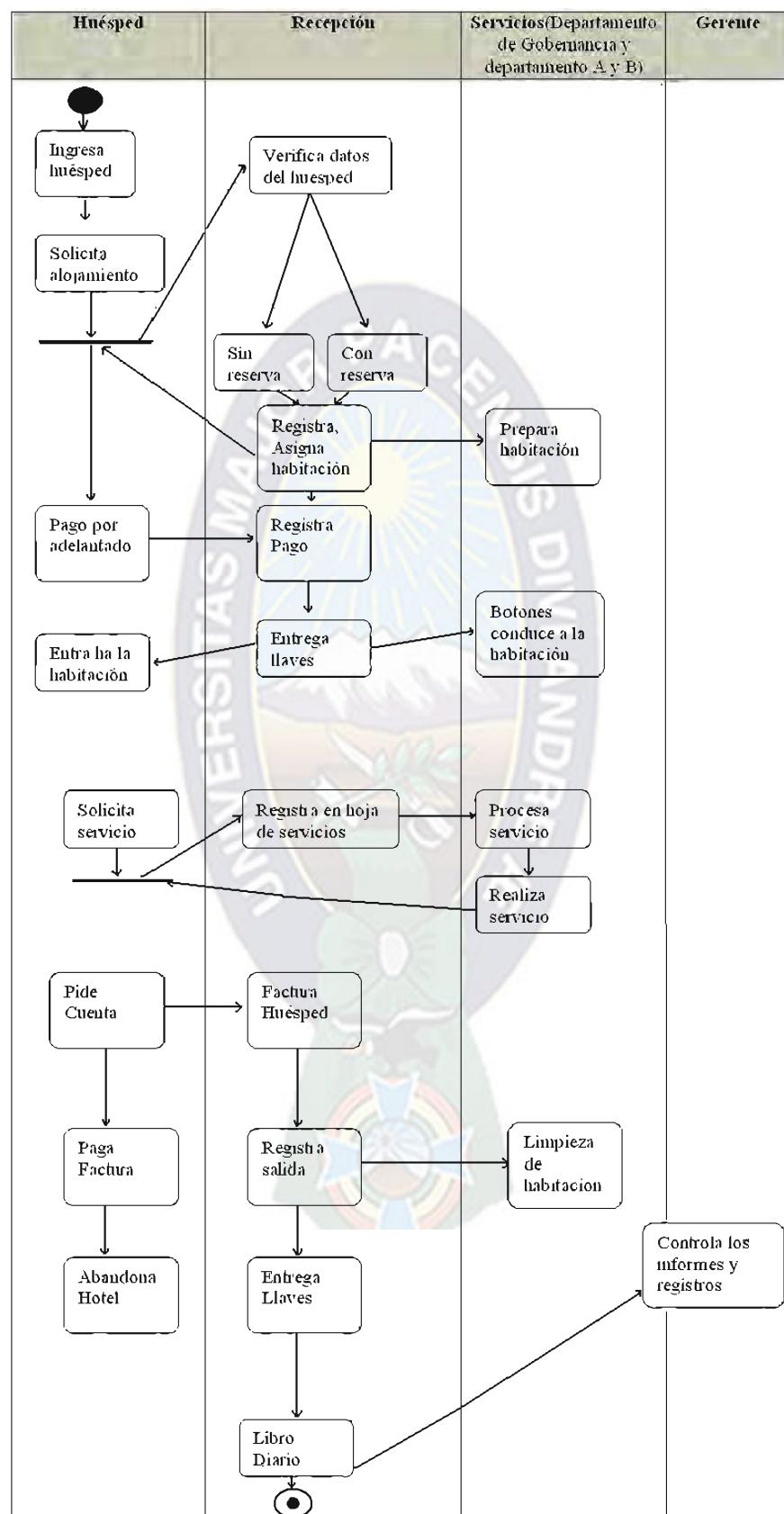
Descripción, Es el encargado de la administración general del Hotel, realizando las siguientes funciones:

- Define las funciones de los puestos;
- Supervisa el trabajo de los subalternos;
- Participa en las reuniones con los encargados de Área;
- Participa en las reuniones con los Administradores;
- Administra Reportes de Caja y Liquidación;
- Define Estrategias a Futuro;

Actualmente todo el proceso que se genera cuando un huésped visita el hotel, se lo realiza de manera manual.

A continuación se muestra el Diagrama de Actividades el cual representa los flujos de trabajo paso a paso de negocio Actual, Figura 3.2.

Figura 3.2. Diagrama de Actividades Actual

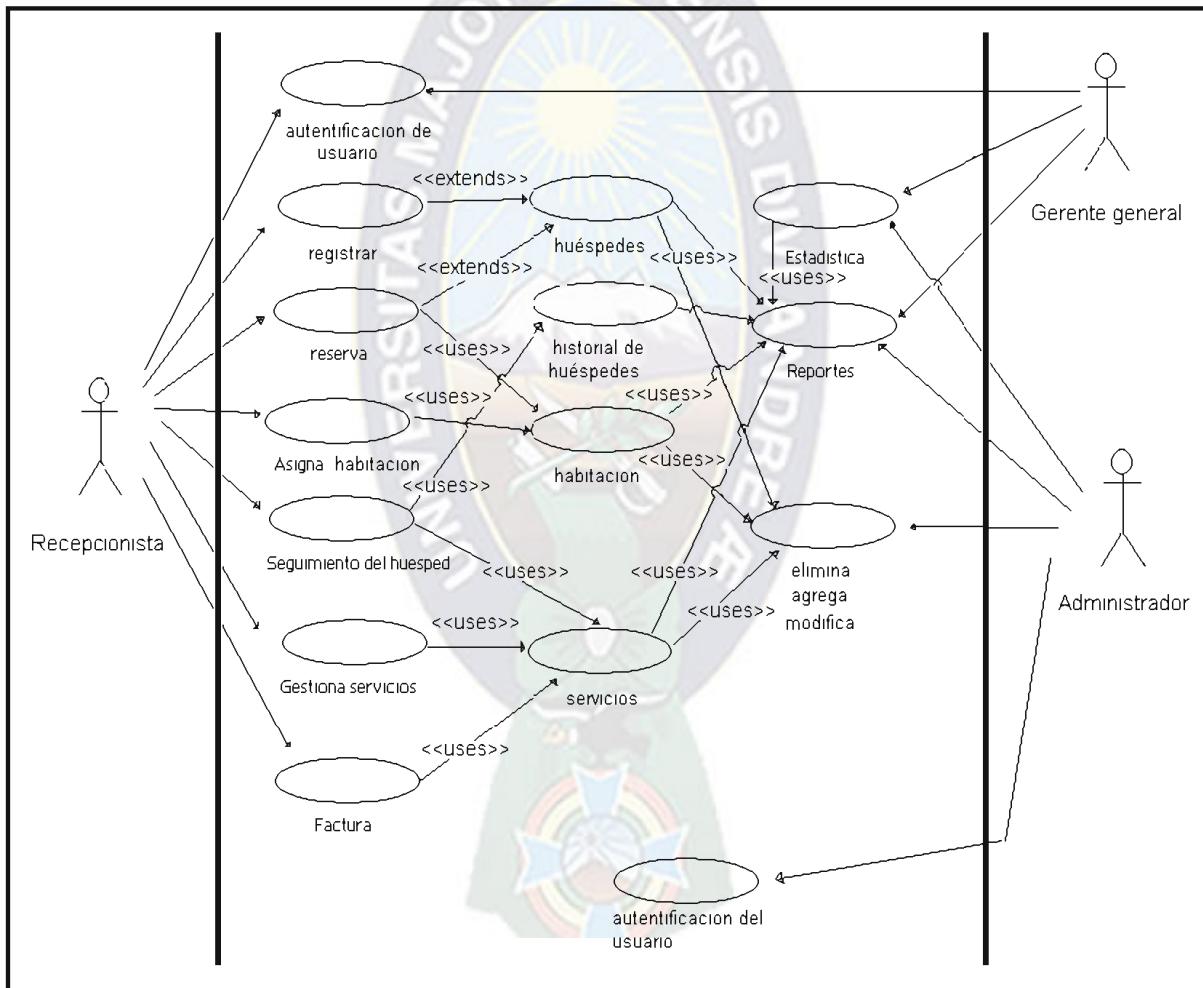


3.2 SISTEMA LÓGICO

3.2.1 Diagrama de Casos de Uso

El diagrama de Casos de Uso muestra una parte de la funcionalidad general que tendrá el nuevo sistema, la Figura 3.3 muestra el modelo de casos de uso general.

Figura 3.3 Diagrama de casos e Uso



3.3 Requerimientos del Ingeniero de Software.

Los requerimientos que serán necesarios para el desarrollo del Proyecto se destacan en la siguiente página.

3.3.1 En Cuanto a Hardware

Se necesita lo siguiente:

- Procesador Pentium III o superior;
- 64 MB de memoria ram o Superior;
- Disco duro de 10 GB;
- Monitor de 14 pulgadas o superior;
- Impresora

3.3.2 En cuanto a Software

Se necesita lo siguiente:

- Microsoft Windows XP;
- Microsoft Office 2000;
- PHP Script Language Version 5.1.1;
- MySQL Database Version 5.0.16;
- Navegador Internet Explorer 6 /adelante o firefox 1.5 /adelante

4. PROCESO DE INVESTIGACIÓN

En el Capítulo 2, se describió el ciclo del vida de la Metodología Programación Extrema, Primero se realiza la Planeación en función de las Historias de Usuario.

4.1 ELABORACIÓN DE LAS HISTORIAS DE USUARIO

Historias de usuario
Número : 1
Nombre: Diseñar el Sistema de Control y Gestión Hotelera
Descripción: Actualmente el Hotel Sayari funciona con un entorno de trabajo manual en consecuencia los procesos son lentos no existe un sistema informático que pueda solucionar los problemas de reservas, registros, servicios, reportes, caja y auditoria nocturna de forma rápida.
Debido ha esta situación se implementara el Sistema de Control y Gestión Hotelera.
Observaciones: Sin observaciones.

Tarea	
Número tarea: 1	Número Historia: 1
Nombre tarea: Diseñar el Diagrama de Casos de Uso	
Descripción: El Diagrama de Casos de Uso nos dará una idea de la funcionalidad que tendrá el nuevo Sistema.	

Tarea	
Número tarea: 2	Número Historia: 1
Nombre tarea: Diseñar el Diagrama de Clases	
Descripción: El Diagrama de Clases es fundamental en el diseño y análisis de un sistema, en el se hace una descripción de clases, atributos y las relaciones entre ellos.	

Tarea	
Número tarea: 3	Número Historia: 1
Nombre tarea: Seleccionar el Software de Uso	
Descripción: La elección del software es muy importante para la ejecución del sistema, eligiendo las últimas versiones.	

Las siguientes historias de usuario se detallan en el Anexo A:

Historia #2: Registro de Huéspedes y Reserva de habitación;

Tarea 1: Diagramar el modelo Entidad-Relación;

Tarea 2: Diagramar el modelo Jerárquico;

Tarea 3: Diseñar Esquema de Ubicación de Ordenadores;

Tarea 4: Desarrollar modulo Registro;

Tarea 5: Realizar modulo Reserva;

Historia #3: Interfaz Grafica de Usuario y validaciones;

Tarea 1: Crear la Interfaz Grafica;

Tarea 2: validar campos;

Historia #4: Modulo que Realice Seguimiento de los Huéspedes Registrados;

Tarea 1: Desarrollar un modulo que obtenga la información de los Huéspedes que ocupan una determinada habitación;

Tarea 2: Realizar un modulo que pueda mostrar y cambiar la habitación del huésped;

Tarea 3: Implementar un modulo que pueda adicionar, modificar y eliminar los servicios a una determinada habitación además dar la cuenta;

Historia #5: Facturación;

Tarea 1: Desarrollar una Factura por habitación;

Tarea 2: Desarrollar una Factura Global (varias habitaciones);

Tarea 3: Implementar el Botón que Genere el documento para Impresión;

Historia #6: Modulo Caja;

Tarea 1: Desarrollar un modulo que pueda dar información a cerca de las Facturas Emitidas;

Tarea 2: Implementar un modulo que cotice la moneda;

Tarea 3: realizar un modulo que sea capaz de dar los datos acerca de los ingresos del día, así como de una fecha anterior;

Historia #7: Modulo Reportes;

Tarea 1: Desarrollar un modulo que genere informes;

Tarea 2: Implementar un modulo de Estadísticas;

Historia #8: Modulo de Configuración;

Tarea 1: Desarrollar un modulo que pueda configurar habitaciones;

Tarea 2: Implementar un modulo que pueda configurar servicios;

Tarea 3: Realizar la opción que configure a los administradores del sistema;

Historia #9: Aplicación de Contraseñas;

Tarea 1: Implementar Cifrado de datos, MD5;

Tarea 2: Implementar Ayudas;

Historia #10: Capacitación;

Tarea 1: Políticas de Seguridad;

Tarea 2: Resguardo de la Base de Datos

4.2 PLANIFICACIÓN

En función de las historias de usuario se realiza la Planificación en 10 incrementos basándose en el tiempo, Tabla 4.1.

Tabla 4.1 Cronograma del desarrollo de Incrementos

Tiempo \ Actividad	SEP	OCT	NOV	DIC
H1	■			
H2		■■■		
H3		■■		
H4			■■	
H5			■	
H6			■	
H7			■■	
H8			■■	
H9				■■
H10				■

4.3 DISEÑO Y DESARROLLO

Esta fase incluye el Diseño y el Desarrollo de las historias de usuario, además contiene las respectivas pruebas.

4.3.1 PRIMER INCREMENTO

Historia de Usuario N° 1: Diseñar un Sistema de Control y Gestión Hotelera.

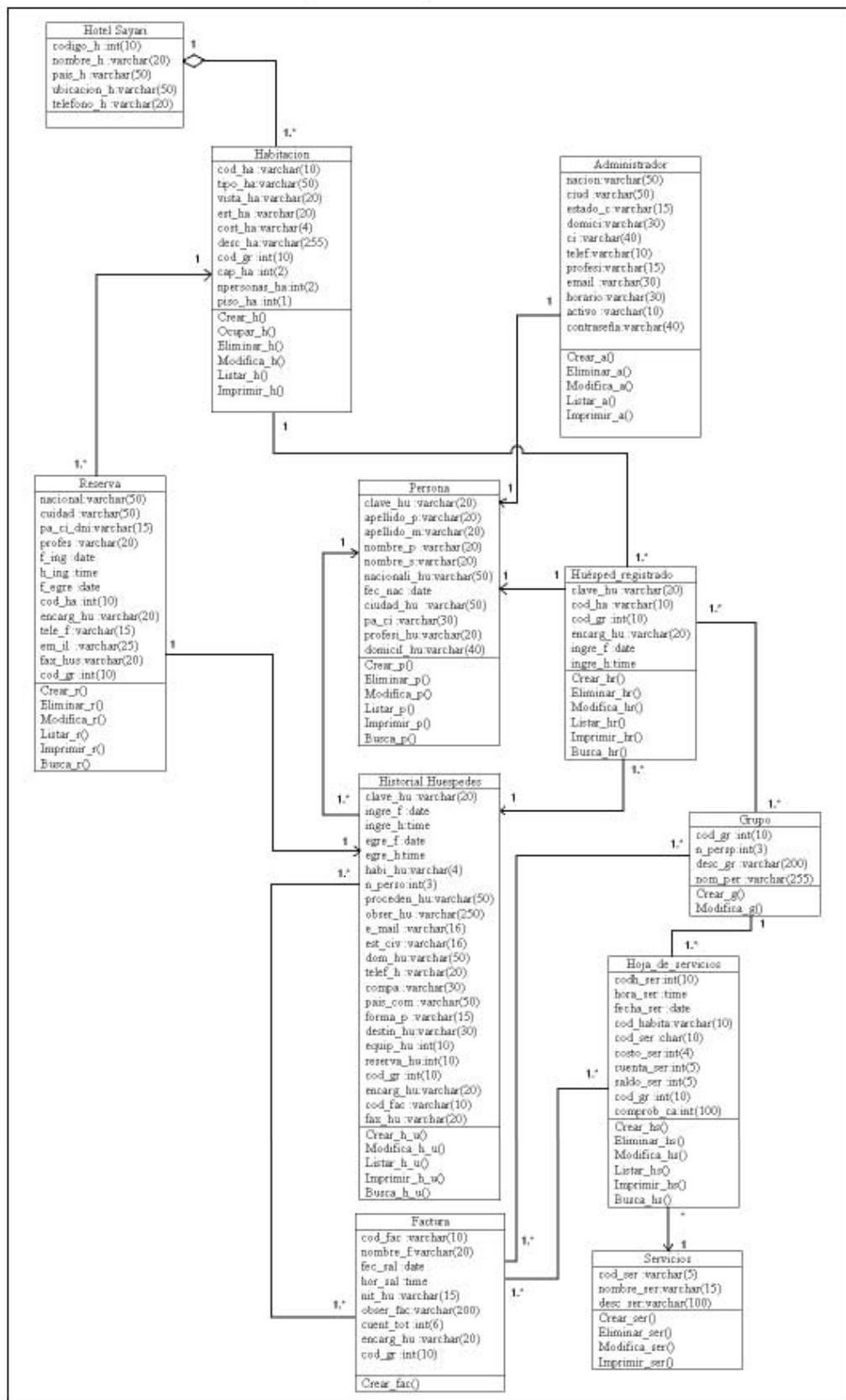
Tarea #1: Diseñar el Diagrama de Casos de Uso.

El Diagrama de casos de uso se muestra en la Figura, 3.3, Pág. (40) el cual muestra una parte de la funcionalidad general.

Tarea #2: Diseñar el Diagrama de Clases.

El Diagrama de Clases es el diagrama principal de Análisis y Diseño de un Sistema. En él, se especifica, las relaciones entre clases. Durante el análisis del sistema, el diagrama se desarrolla buscando una solución ideal. Durante el diseño se usa el mismo diagrama, y se modifica para satisfacer los detalles de las implementaciones

Figura 4.1 Diagrama de Clases



Tarea #3: Seleccionar el Software de Uso.

Se Tiene la selección de Software y hardware en la Pagina 41.

4.3.2 SEGUNDO INCREMENTO

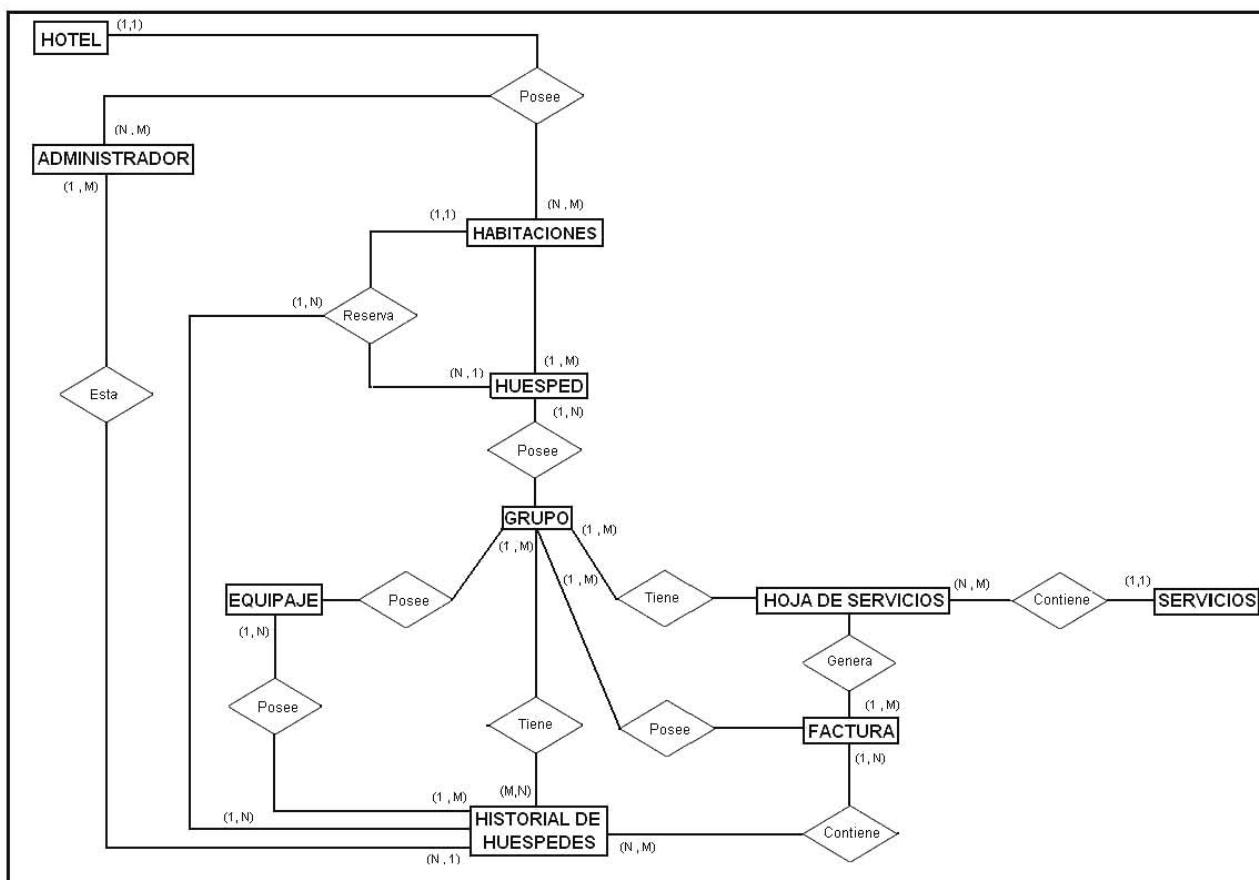
Historia de Usuario Nº 2: Registro de Huéspedes y Reserva de habitación.

Antes del Registro y Reserva de habitación es indispensable realizar las tareas 1,2 y 3.

Tarea 1: Diagramar el modelo Entidad-Relación;

Como se esta utilizando un Sistemas de Manejo de Base de datos (dbms), MySQL Database Versión 5.0.16, que es relacional entonces debemos utilizar una base de datos relacional, así se realiza la transformación del diagrama de Clases al Modelo Entidad Relación.

Figura 4.2 Modelo Entidad - Relación



A continuación se hace una descripción de las tablas que tienen la Base de Datos del Sistema:

Tabla Hotel

Información general del hotel.

codigo_h, nombre_h, pais_h, ubicacion_h, telefono_h

Clave: codigo_h.

Tabla Administrador

En esta tabla se almacenan los datos de los usuarios que manejarán el sistema, Gerente , Administrador y recepción.

ap_p , ap_m , nom_p, nom_s, nacion, ciud,estado_c,domici, ci, telef, profesi, email, horario, activo, contrasena.

Clave: ap_p, contrasena.

Tabla Huéspedes

Se almacena los datos fundamentales de los huéspedes.

clave_hu, apellido_p, apellido_m, nombre_p, nombre_s, nacionali_hu, fec_nac, ciudad_hu, pa_ci, profesi_hu, domicil_hu.

Clave: clave_hu.

Tabla Historial de Huéspedes (histo_hues)

Tiene toda la información del historial del huésped.

clave_hu, ingre_f, ingre_h, egre_f, egre_h, habi_hu, n_perso, proceden_hu, obser_hu, e_mail, est_civ, dom_hu, telef_h, compa, pais_com, forma_p, destin_hu, equip_hu, reserva_hu, cod_gr, encarg_hu, cod_fac, fax_hu

Clave: clave_hu, cod_gr , cod_fac .

Tabla Reserva de Huéspedes (reser_huesp)

Almacena los datos de los huéspedes que realizan reserva.

a_p, a_m, n_p, n_s, nacional, ciudad, pa_ci_dni, profes_f_ing, h_ing, f_egre, cod_ha, encarg_hu, tele_f, em_il, fax_hus, cod_gr

Clave: pa_ci_dni, cod_ha, cod_gr

Tabla Huéspedes Registrados (huesped_reg)

Esta tabla contiene los datos de los huéspedes que están Alojados en el hotel.

clave_hu, cod_ha, cod_gr, encarg_hu, ingre_f, ingre_h.

Clave: clave_hu, cod_ha, cod_gr.

Tabla Habitación

Contiene los atributos referentes a las habitaciones.

cod_ha, tipo_ha, vista_ha, est_ha, cost_ha, desc_ha, cod_gr, cap_ha, npersonas_ha, piso_ha.

Clave: cod_ha.

Tabla Equipaje

Almacena la información del equipaje de los huéspedes.

clave_hu, desc_eq, ingre_f, ingre_h

Clave: clave_hu.

Tabla Grupo

En esta tabla se almacena los grupos que ingresaron al hotel.

cod_gr, n_persp, desc_gr, Comenta_per

Clave: cod_gr.

Tabla Hoja de Servicios (hoja_ser)

Almacena los servicios que adquirieron los huéspedes.

codh_ser, hora_ser, fecha_ser, cod_habita, cod_ser, costo_ser, cuenta_ser, saldo_ser, cod_gr, comprob_ca.

Clave: codh_ser, cod_habita, cod_ser, cod_gr.

Tabla Servicios

En esta tabla se almacena los servicios que ofrece el hotel.

cod_ser, nombre_ser, desc_ser.

Clave: cod_ser.

Tabla Moneda

Contiene los datos de la cotización de las distintas monedas en circulación.

Moneda, cotizacion, fecha.

Tabla Factura

Describe información acerca de la factura.

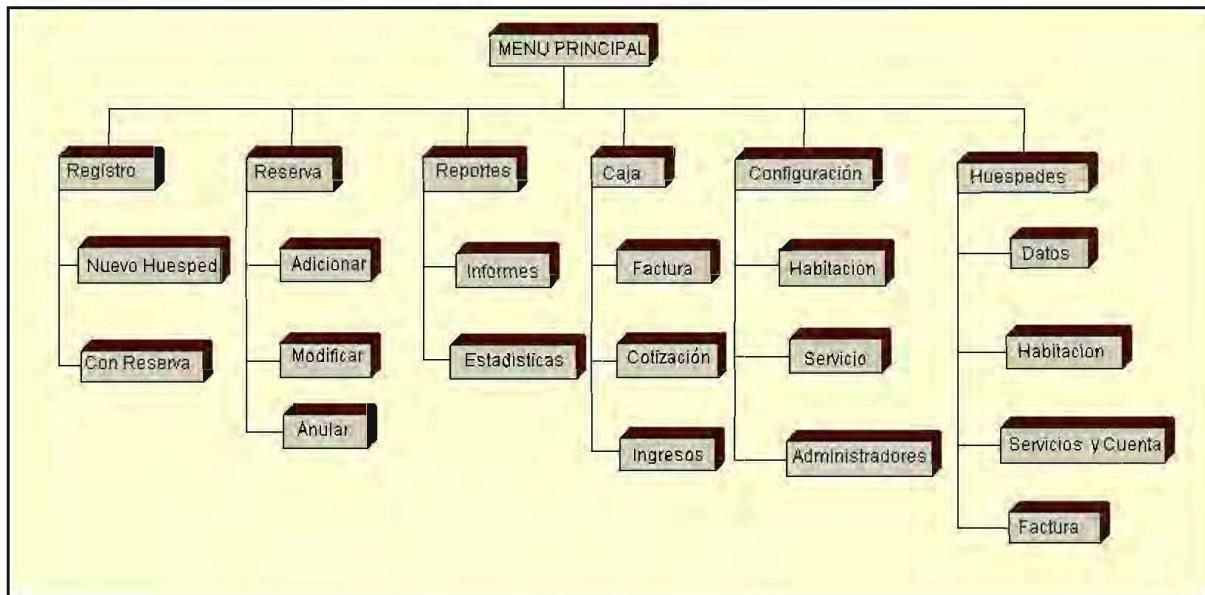
`cod_fac, nombre_f, fec_sal, hor_sal, nit_hu, obser_fac, cuent_tot, encarg_hu, cod_gr`

Clave: `cod_fac, cod_gr`.

Tarea 2: Diagramar el modelo Jerárquico;

La descomposición de las funciones complejas en subfunciones es el objetivo del Diagrama Jerárquico, Figura 4.3.

Figura 4.3 Diagrama Jerárquico



Tarea 3: Diseñar Esquema de Ubicación de Ordenadores;

En la Figura 4.4 se muestra la ubicación de los ordenadores en el Hotel Sayari.

Figura 4.4 Estructura Cliente Servidor Hotel Sayari

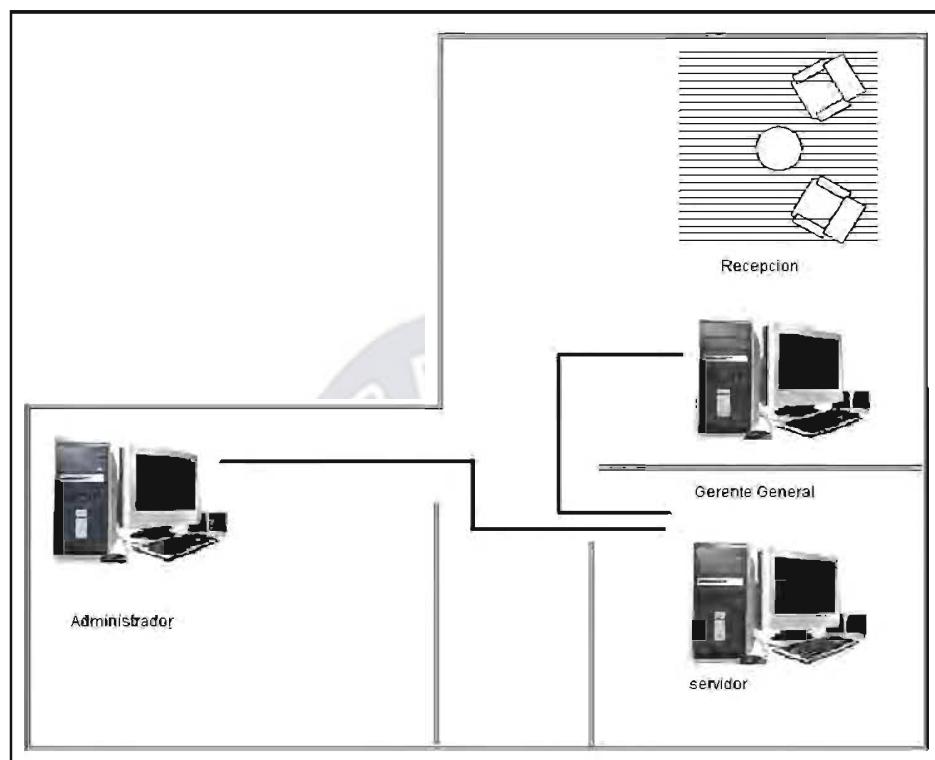


Figura 4.5 Comportamiento de la arquitectura tres capas



- Capa Presentación.** Sistema Operativo Windows xp/ adelante, navegador Internet Explorer 6/ adelante o Firefox 1.5 / adelante.
- Capa Lógica de la Aplicación.** Plataforma Windows, Php, Mysql y Servidor Apache.
- Capa de Almacenamiento.** Sistema gestor de base de datos MySql.

Tarea 4: Desarrollar modulo Registro;

A continuación se detalla el modulo Registro, cabe notar que en esta etapa no se tiene definida la interfaz de Usuario.

Figura 4.6 Registro

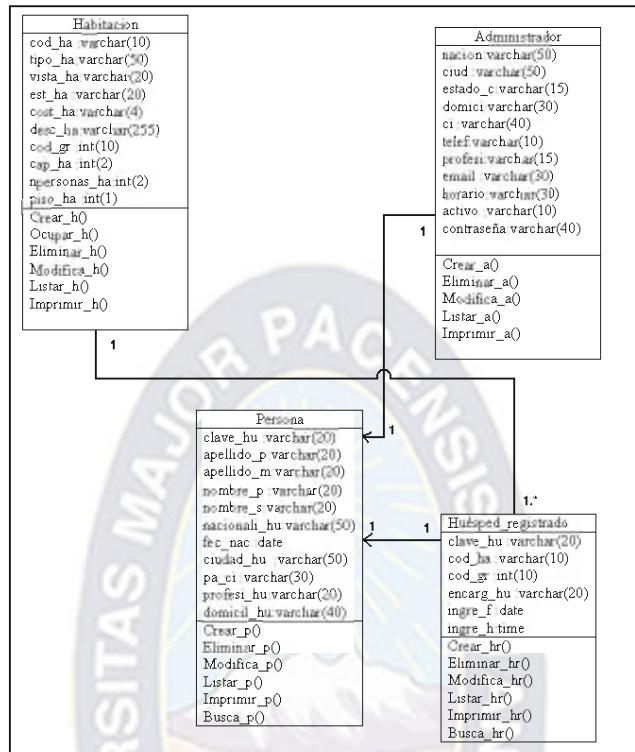
The screenshot shows a Mozilla Firefox browser window with the title bar 'REGISTRO - Mozilla Firefox'. The address bar displays the URL 'http://localhost/maya1.0/Registro/registrar/registrar.php'. The main content area contains a form titled 'REGISTRO' with the following fields:

- Apellidos Last Name: [text input]
- Nombres First Name: [text input]
- Nacionalidad Nationality: [dropdown menu] Ciudad City: [text input]
- E-mail: [text input] Estado Civil Marital Status: Soltero Casado
- Domicilio Address: [text input] Fecha de nacimiento Birth date: [date picker set to 1 Ene 1920]
- Paseport Passport Cartera de identidad: [text input] Teléfono Phone: [text input]
- Profesion Profession: [text input] Compañía Firm: [text input] País/Country: [dropdown menu] Escoge una opción
- Fax: [text input] Forma de pago/Form of payment: [dropdown menu] DNI
- Procedencia/Coming From: [text input] Desuno Going To: [text input]
- Nº personas: [text input] Equipaje: si no
- Observacion del huésped: [text area with character limit 'máx de 255 caracteres']
- Fecha de ingreso: 2009-10-25 Egreso: [date picker set to 1 Ene 2005]
- Buttons: Siguiente, SALIR, and Terminado.

Esta es la ventana donde se realiza el registro de los huéspedes que ingresan al hotel, después se escoge la habitación que va ocupar el Huésped.

Para realizar estos procesos intervinieron las siguientes clases:

Figura 4.7 Clases del modulo Registro



Prueba

Descripción

El encargado de recepción, tras seleccionar los campos empieza ha llenar la información, después de comprobar la sintaxis pulsa el botón siguiente, escoge la habitación a ocupar y presiona finalizar

Condiciones de ejecución

El administrador deberá conocer el funcionamiento de este módulo.

- Se introduce los datos.
- Se pulsa el botón siguiente.
- En esa ventana se mostrará un botón para seleccionar la habitación a ocupar.
- Tras la ejecución, se procesará internamente y se mostrará el estado un mensaje indicando que el pedido ha sido correctamente procesado.
- El Registro se considera como finalizado.

Resultado esperado

Tras la introducción de los datos, si el procesado ha sido correcto, en la Base de Datos aparecerán los datos del nuevo huésped.

Evaluación de la prueba

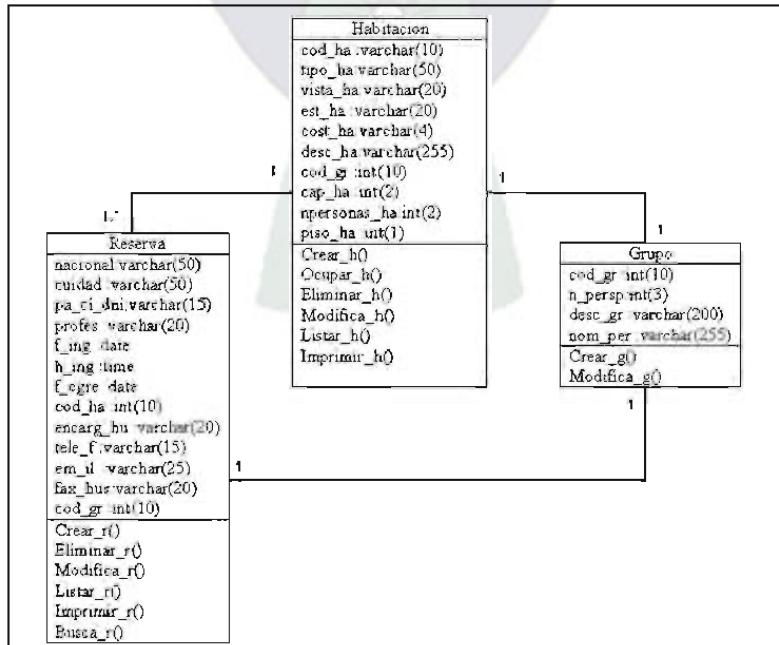
Prueba satisfactoria.

Tarea 5: Realizar modulo Reserva;

Figura 4.8 Reserva

La Figura 4.8, describe como se realiza la reserva teniendo especial cuidado fecha de ingreso, se utilizan las siguientes Clases:

Figura 4.9 Clases del modulo Reserva



Prueba

Descripción

El encargado de recepción, tras seleccionar el modulo Reservas empieza ha llenar la información, después de comprobar la sintaxis y teniendo especial cuidado con la fecha de ingreso pulsa el botón siguiente, escoge las habitaciones a Reservar y presiona finalizar

Condiciones de ejecución

El encargado de recepción debe conocer el funcionamiento de este módulo.

- Se introduce los datos.
- Se pulsa el botón siguiente.
- En esa ventana se mostrará un botón para seleccionar la habitación a Reservar.
- Tras la ejecución, se procesará internamente y se mostrará el estado un mensaje indicando que el pedido ha sido correctamente procesado.
- La reserva se considera como finalizado.

Resultado esperado

Tras la introducción de los datos, si el procesado de los datos ha sido correcto, en la Base de Datos aparecerán los datos de la Reserva, Las Habitaciones figurarán como reservadas y se actualizará la tabla Grupos.

Evaluación de la prueba

Prueba satisfactoria.

4.3.3 TERCER INCREMENTO

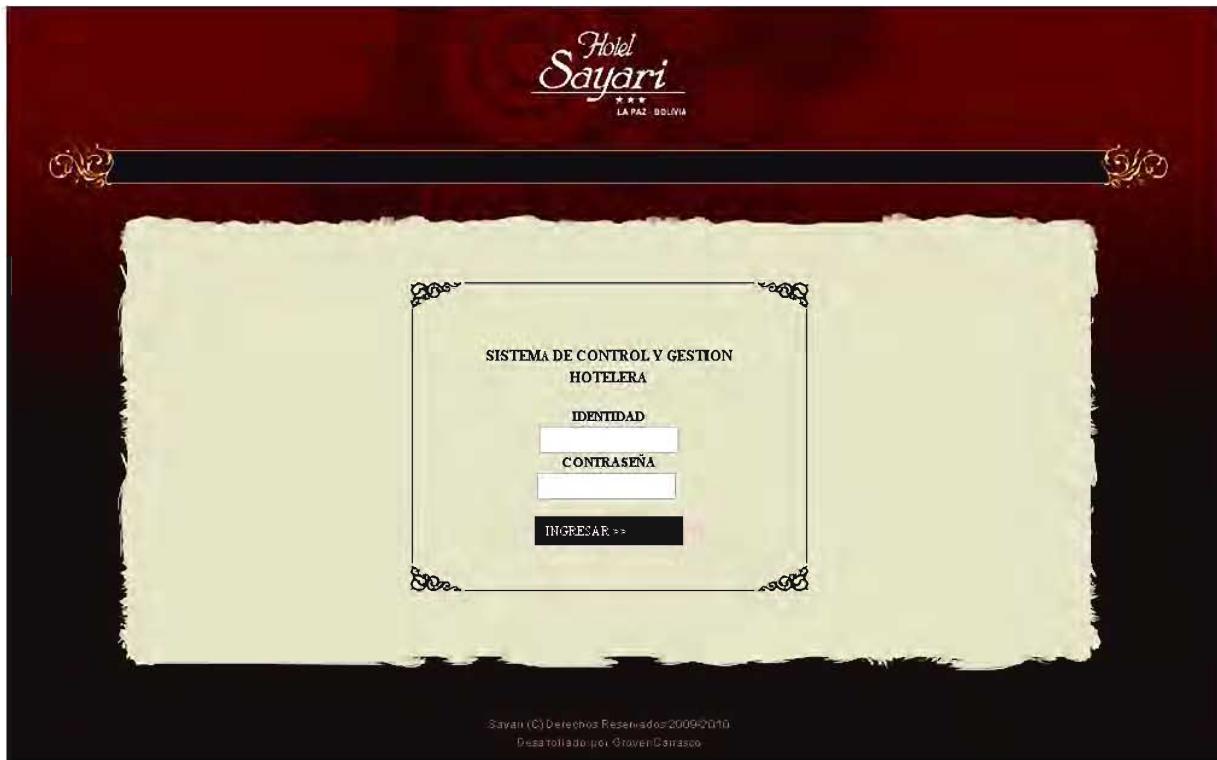
Historia de Usuario Nº 3: Interfaz Grafica de Usuario y validaciones;

Tarea 1: Crear la interfaz Grafica.

Después de entrevistar al administrador del Hotel Sayari, se puso en claro el estilo que quería para la interfaz, se enfatizo en utilizar distintos tonos de color guindó, amarillo y negro, esta combinación da como resultado un aspecto elegante además son los colores que sobresalen en la arquitectura del Hotel.

El resultado obtenido se muestra en la Figura 4.10, el mismo satisfizo al Administrador.

Figura 4.10 Interfaz



Tarea 2: validar campos;

Mediante las instrucciones JavaScript, que es un lenguaje de Programación de lado del cliente se realizo las validaciones de los distintos campos que se deben llenar. En la Figura 4.11 se muestra una validación en la ventana Registro, esta ya cuenta con la interfaz de usuario, al igual se realiza la validación de todos los campos del Sistema.

Figura 4.11 Validación

4.3.4 CUARTO INCREMENTO

Historia de Usuario N° 4: Módulo que Realice Seguimiento de los Huéspedes Registrados;

Tarea 1: Desarrollar un módulo que obtenga la información de los Huéspedes que ocupan una determinada habitación;

Se crea una ventana Menú (Figura 4.12) la cual reparte a los demás módulos del sistema, uno de esos módulos es el de huéspedes, al ingresar a este modulo se presenta una ventana que contiene los datos de las personas que están hospedadas en el hotel, al escoger una de ellas se entra a otra ventana, la cual presenta otro menú en la que esta la información del huésped, junto con su habitación además del grupo que ingreso con el huésped (Figura 4.13).

Figura 4.12 Ventana Menú Principal

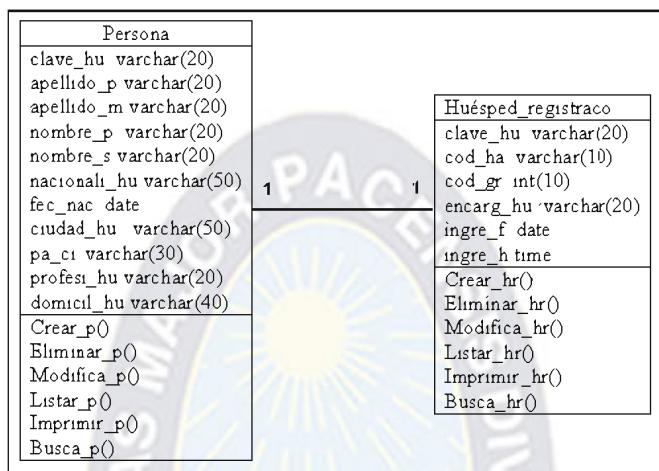


Figura 4.13 Ventana datos del huésped.

Primer Apellido	Primer Nombre
Pachero	Angel
yuca	javier
Rodriguez	Alvarado

Además de mostrar información del huésped también tiene la opción de modificar algún dato que haya sido introducido erróneamente, solamente se debe escoger del menú desplegado. Para realizar este modulo intervinieron las siguientes Clases.

Figura 4.14 Clases del modulo, datos del huésped



Prueba

Descripción

Una vez que el Recepcionista ha entrado en el sistema, seleccionará la opción Huéspedes. Se le mostrará un listado con todos los huéspedes que han sido registrados previamente, examinara los datos y si se da el caso de algún error realizara la modificación de los datos.

Condiciones de ejecución

Debe existir por lo menos un huésped registrado en la Base de Datos. El recepcionista deberá conocer el uso del Sistema.

Entrada

- Del menú principal seleccionará “Huéspedes”.
- Se mostrará un listado con todos los huéspedes registrados.
- Elegirá un huésped.
- En la ventana “Datos del Huésped”, verificará los datos.
- En la ventana “Datos del Huésped”, si los datos son incorrectos entonces elegirá el huésped, pulsara “Siguiente” e introducirá los nuevos datos para realizar la Modificación.

Resultado esperado

En la Base de Datos, los datos del huésped deben ser los correctos, además debe mostrar los datos nuevos si es que se los modifíco.

Evaluación de la prueba

Prueba satisfactoria.

Tarea 2: Realizar un módulo que pueda mostrar y cambiar la habitación del huésped;

Como se describió en la tarea anterior ya se tiene el menú donde está los Datos de los huéspedes (ver. Figura 4.13), también podemos apreciar que ya está creada la aplicación de habitación, pulsando esta opción tenemos. Figura 4.15,

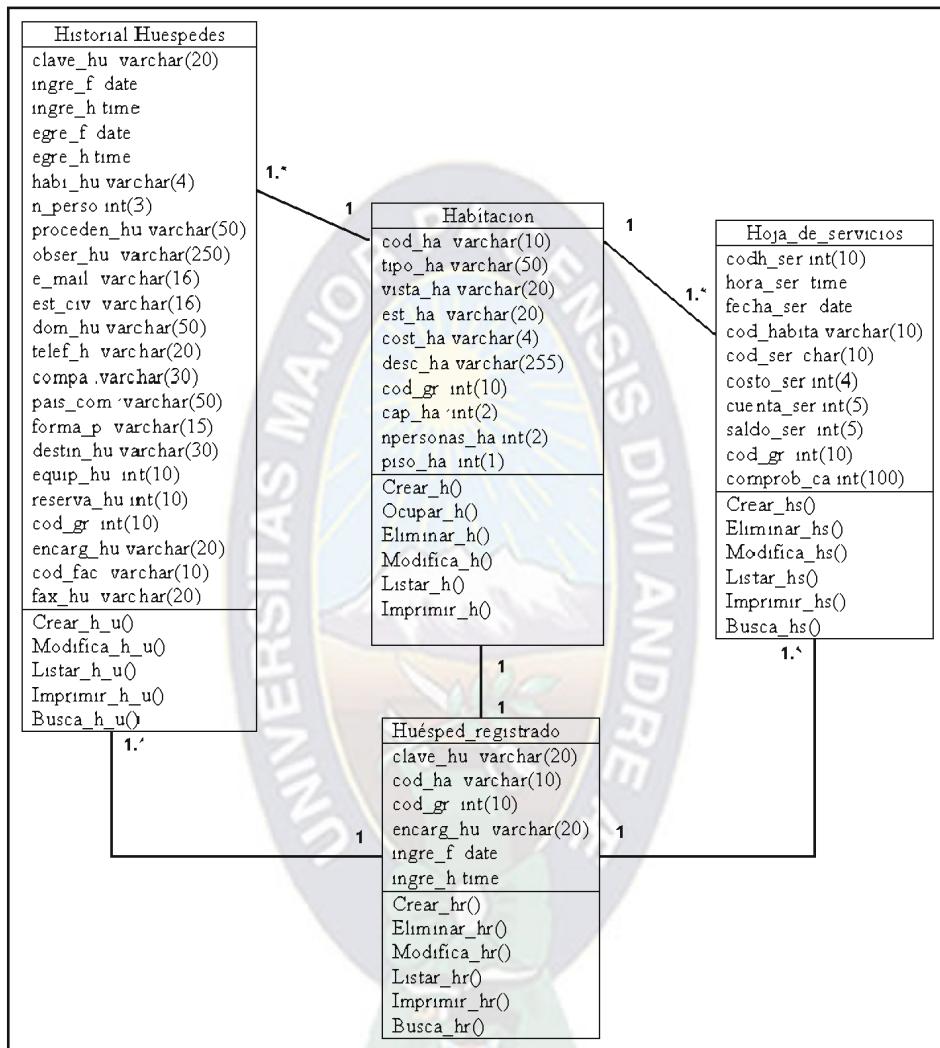
En la Figura se puede ver un grupo de tres personas que entraron juntas al Hotel y pidieron habitaciones separadas, la 201, 203 y 211, si alguno de ellos decide realizar el cambio de habitación entonces escogiendo el número de la Habitación del menú desplegable y presionando siguiente se puede realizar el cambio de habitación.

Figura: 4.15 Ventana Habitación



Además de mostrar de las habitaciones del grupo, también tiene la opción de realizar el cambio de habitación, para realizar este modulo intervinieron las siguientes Clases.

Figura 4.16 Clases del modulo, Habitación



Prueba

Descripción

Una vez que el Recepcionista ha entrado en el sistema, seleccionará la opción Huéspedes. Se le mostrará un listado con todos los huéspedes que han sido registrados previamente, escogerá al huésped y entrara en la opción habitación Figura 4.15, ahí podrá realizar el cambio de habitación si es que el huésped lo desea.

Condiciones de ejecución

Debe existir por lo menos un huésped registrado en la Base de Datos, debe existir al menos una habitación disponible. El recepcionista deberá conocer el uso del Sistema.

Entrada

- Del menú principal seleccionará “Huéspedes”;
- Se mostrará un listado con todos los huéspedes registrados;
- Elegirá un huésped;
- En la ventana “Habitación”, verificará los datos;
- En la ventana “Habitación”, se puede ver los datos de quien ocupa la habitación, además se realiza el cambio eligiendo la habitación a cambiar y presionando el botón “Siguiente”;

Resultado esperado

En la ventana se debe mostrar la habitación correcta que ocupa el huésped, en la Base de Datos, los datos de la habitación deben ser correctos, en caso de que se haya cambiado debe cambiar también los datos.

Evaluación de la prueba

Prueba satisfactoria

Tarea 3: Implementar un modulo que pueda adicionar, modificar y eliminar los servicios a una determinada habitación además dar la cuenta;

La ventana que se muestra en la Figura 4.17, lo que hace es mostrar los servicios por habitación, que adquirió el huésped, también muestra la cuenta y por último tiene tres botones, nuevo, modificar y eliminar servicio.

Botón “Nuevo” sirve para poder adicionar un servicio a la habitación que ocupa un determinado huésped, previamente se debe elegir la habitación que quiere adquirir el servicio.

Botón “Modificar”, realiza la modificación de algún servicio adquirido por el huésped.

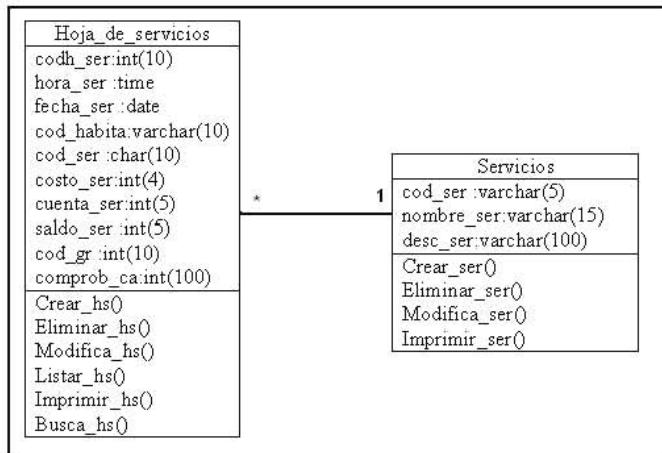
Botón “Eliminar”, elimina algún servicio adquirido por el huésped.

Figura 4.17 Ventana Servicios y Cuenta

DATOS DEL HUESPED	HABITACION	SERVICIOS Y CUENTA	FACTURA																																																				
SERVICIOS ADQUIRIDOS POR EL GRUPO 1																																																							
La Habitacion :201																																																							
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Habitacion</th> <th>Hora</th> <th>Fecha</th> <th>Servicio</th> <th>Costo</th> <th>Pago</th> <th>Debe</th> </tr> </thead> <tbody> <tr> <td>201</td> <td>21:00:40</td> <td>2009-10-25</td> <td>alojamiento</td> <td>180</td> <td>0</td> <td>0</td> </tr> <tr> <td>201</td> <td>23:35:39</td> <td>2009-10-25</td> <td>garaje</td> <td>40</td> <td>0</td> <td>40</td> </tr> <tr> <td colspan="6">Total Cuenta:</td> <td>220</td> </tr> <tr> <td colspan="6">Cancelo La Suma:</td> <td>0</td> </tr> <tr> <td colspan="6">Total a Cancelar:</td> <td>220</td> </tr> </tbody> </table>							Habitacion	Hora	Fecha	Servicio	Costo	Pago	Debe	201	21:00:40	2009-10-25	alojamiento	180	0	0	201	23:35:39	2009-10-25	garaje	40	0	40	Total Cuenta:						220	Cancelo La Suma:						0	Total a Cancelar:						220							
Habitacion	Hora	Fecha	Servicio	Costo	Pago	Debe																																																	
201	21:00:40	2009-10-25	alojamiento	180	0	0																																																	
201	23:35:39	2009-10-25	garaje	40	0	40																																																	
Total Cuenta:						220																																																	
Cancelo La Suma:						0																																																	
Total a Cancelar:						220																																																	
La Habitacion :203																																																							
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Habitacion</th> <th>Hora</th> <th>Fecha</th> <th>Servicio</th> <th>Costo</th> <th>Pago</th> <th>Debe</th> </tr> </thead> <tbody> <tr> <td>203</td> <td>21:40:00</td> <td>2009-10-25</td> <td>alojamiento</td> <td>160</td> <td>0</td> <td>0</td> </tr> <tr> <td>203</td> <td>23:35:58</td> <td>2009-10-25</td> <td>Lavanderia</td> <td>50</td> <td>0</td> <td>50</td> </tr> <tr> <td colspan="6">Total Cuenta:</td> <td>210</td> </tr> <tr> <td colspan="6">Cancelo La Suma:</td> <td>0</td> </tr> <tr> <td colspan="6">Total a Cancelar:</td> <td>210</td> </tr> </tbody> </table>							Habitacion	Hora	Fecha	Servicio	Costo	Pago	Debe	203	21:40:00	2009-10-25	alojamiento	160	0	0	203	23:35:58	2009-10-25	Lavanderia	50	0	50	Total Cuenta:						210	Cancelo La Suma:						0	Total a Cancelar:						210							
Habitacion	Hora	Fecha	Servicio	Costo	Pago	Debe																																																	
203	21:40:00	2009-10-25	alojamiento	160	0	0																																																	
203	23:35:58	2009-10-25	Lavanderia	50	0	50																																																	
Total Cuenta:						210																																																	
Cancelo La Suma:						0																																																	
Total a Cancelar:						210																																																	
La Habitacion :211																																																							
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Habitacion</th> <th>Hora</th> <th>Fecha</th> <th>Servicio</th> <th>Costo</th> <th>Pago</th> <th>Debe</th> </tr> </thead> <tbody> <tr> <td>211</td> <td>21:00:40</td> <td>2009-10-25</td> <td>alojamiento</td> <td>160</td> <td>0</td> <td>0</td> </tr> <tr> <td>211</td> <td>23:36:16</td> <td>2009-10-25</td> <td>restaurant</td> <td>150</td> <td>0</td> <td>150</td> </tr> <tr> <td>211</td> <td>23:36:38</td> <td>2009-10-25</td> <td>service room</td> <td>50</td> <td>0</td> <td>50</td> </tr> <tr> <td colspan="6">Total Cuenta:</td> <td>360</td> </tr> <tr> <td colspan="6">Cancelo La Suma:</td> <td>0</td> </tr> <tr> <td colspan="6">Total a Cancelar:</td> <td>360</td> </tr> </tbody> </table>							Habitacion	Hora	Fecha	Servicio	Costo	Pago	Debe	211	21:00:40	2009-10-25	alojamiento	160	0	0	211	23:36:16	2009-10-25	restaurant	150	0	150	211	23:36:38	2009-10-25	service room	50	0	50	Total Cuenta:						360	Cancelo La Suma:						0	Total a Cancelar:						360
Habitacion	Hora	Fecha	Servicio	Costo	Pago	Debe																																																	
211	21:00:40	2009-10-25	alojamiento	160	0	0																																																	
211	23:36:16	2009-10-25	restaurant	150	0	150																																																	
211	23:36:38	2009-10-25	service room	50	0	50																																																	
Total Cuenta:						360																																																	
Cancelo La Suma:						0																																																	
Total a Cancelar:						360																																																	
Escoja el codigo de la habitacion:																																																							
<input style="border: 1px solid black; padding: 2px; width: 100%;" type="button" value="201"/> <input style="border: 1px solid black; background-color: #333; color: white; padding: 2px; width: 100%;" type="button" value="NUEVO SERVICIO>>"/>	<input style="border: 1px solid black; padding: 2px; width: 100%;" type="button" value="201"/> <input style="border: 1px solid black; background-color: #333; color: white; padding: 2px; width: 100%;" type="button" value="ELIMINAR SERVICIO>>"/>	<input style="border: 1px solid black; padding: 2px; width: 100%;" type="button" value="201"/> <input style="border: 1px solid black; background-color: #333; color: white; padding: 2px; width: 100%;" type="button" value="MODIFICA SERVICIO>>"/>																																																					

Además de mostrar la cuenta de las habitaciones que del grupo, genera automáticamente las opciones desplegables, para realizar este modulo intervinieron las siguientes Clases.

Figura 4.18 Clases del modulo, servicios y cuenta



Prueba

Descripción

Una vez que el Recepcionista ha entrado en el sistema, seleccionará la opción Huéspedes. Se le mostrará un listado con todos los huéspedes que han sido registrados previamente, escogerá al huésped y entrara en la opción “Servicios y Cuenta” Figura 4.17, ahí podrá ver los servicios que adquiridos por habitación, además el Total a cancelar, y deberá ejecutar las opciones “Nuevo”, “Eliminar” y “Modificar”.

Condiciones de ejecución

Debe existir por lo menos un huésped registrado en la Base de Datos, el recepcionista deberá conocer el uso del Sistema.

Entrada

- Del menú principal, seleccionará “Huéspedes”.
- Se mostrará un listado con todos los huéspedes registrados.
- Elegirá un huésped.
- En la opción “Servicios y Cuenta”, verificará los datos.
- Ejecutar un “Nuevo Servicio” al azar, de cualquier habitación del menú desplegable.

- Ejecutar “Eliminar Servicio” al azar, de cualquier habitación del menú desplegable.
- Ejecutar “Modificar Servicio” al azar, de cualquier habitación del menú desplegable.

Resultado esperado

En la ventana se debe mostrar la habitación correcta que ocupa el huésped, los servicios que adquirió la habitación y la cuenta total, en la Base de Datos, los datos en la tabla “Hoja de servicios” deben ser correctos, en caso de que se haya ejecutado Nuevo, Elimina y/o Modificar deben cambiar también los datos en la tabla “Hoja de servicios”.

Evaluación de la prueba

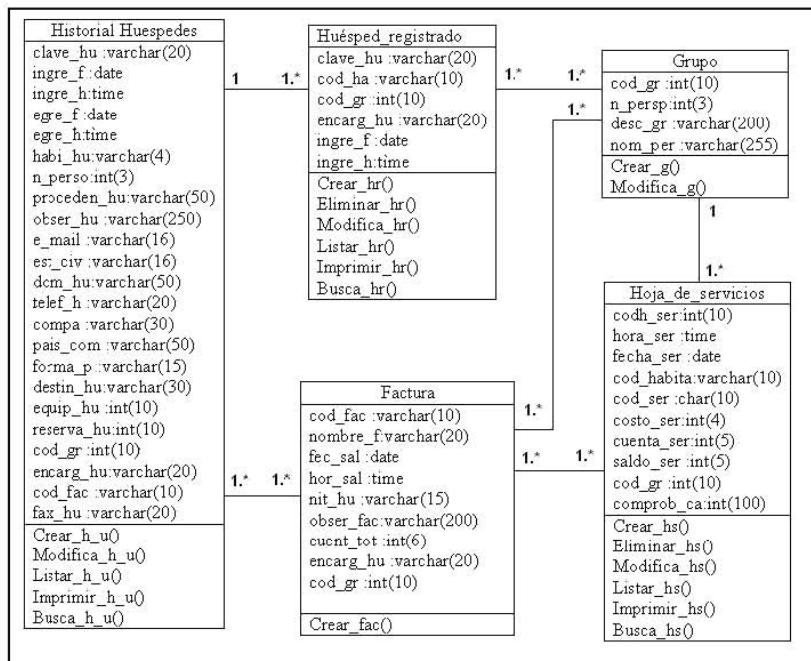
Prueba satisfactoria.

4.3.5 QUINTO INCREMENTO

Historia de Usuario Nº 5: Facturación;

Tanto la tarea #1 y tarea # 2, para su desarrollo utilizaron las siguientes Clases.

Figura 4.19 Clases del modulo ,factura



Tarea 1: Desarrollar una Factura por habitación;

En la Figura 4.20 se puede apreciar la ventana factura, muestra la información de la cuenta total de cada una de las habitaciones que ocupa un determinado grupo además de una cuenta total de todas las habitaciones.

Figura 4.20 Ventana Factura

The screenshot shows a software interface for generating invoices. At the top, there is a navigation bar with four tabs: 'DATOS DEL HUESPED' (Guest Data), 'HABITACION' (Room), 'SERVICIOS Y CUENTA' (Services and Account), and 'FACTURA' (Invoice). The 'FACTURA' tab is currently selected.

The main area is titled 'FACTURA' and displays the following information:

- EL CODIGO ES 1
- LA HABITACION : 201 TOTAL A FACTURAR : 220
- LA HABITACION : 203 TOTAL A FACTURAR : 210
- LA HABITACION : 211 TOTAL A FACTURAR : 360
- CUENTA DE TODO EL GRUPO : 1 TOTAL A FACTURAR : 790

Below this, there are two buttons:

- A dropdown menu labeled 'LIQUIDAR' (Liquidate) containing the option 'FACTURA POR HABITACION' (Invoice by Room).
- A button labeled 'Escoja el codigo habitacion:' (Select room code:) with a dropdown menu showing '201'.

At the bottom, there is a button labeled 'FACTURAR >>' (Generate Invoice) and another button labeled 'FACTURA GRUPAL >>' (Group Invoice).

Escogiendo la opción facturar habitación 203, se habré otra ventana Figura 4.21, que presenta un formulario, el cual se llena con los datos característicos de una factura.

Figura 4.21 Ventana Factura por habitación

The screenshot shows a software interface for generating a guest bill. At the top, it displays the date 'La Paz, 2009-10-26', NIT/CI '5940666', and the bill number '00150'. Below this, the guest's name 'Señor(es)/Nombre Empresa: Ortiz' is entered. A section labeled 'Detalle precio total' contains a table with two items: 'alojamiento' at 160 and 'Lavanderia' at 50. The total amount, 'Total 210 Bs', is shown below the table. A note states 'Son: doscientos diez Bolivianos'. There is a text input field for 'Comentario de la factura...' which is currently empty. At the bottom left is a dark button labeled 'FACTURAR >>'. The entire window has decorative scrollwork borders.

DETALLE	PRECIO TOTAL
alojamiento	160
Lavanderia	50

Una vez llenado los datos, se presenta otra pantalla donde se realiza la impresión de la factura (tarea 3).

Prueba

Descripción

Una vez que el Recepcionista ha entrado en el sistema, seleccionará la opción Huéspedes. Se le mostrará un listado con todos los huéspedes que han sido registrados previamente, escogerá al huésped y entrara en la opción "Factura" Figura 4.20, ahí podrá ver la cuenta Total de las habitaciones, seguidamente elegirá la habitación que desee facturar, llenara el formulario correspondiente y mostrara la factura para imprimir.

Condiciones de ejecución

Debe existir por lo menos un huésped registrado en la Base de Datos, el recepcionista deberá conocer el uso del Sistema.

Entrada

- Del menú principal seleccionará “Huéspedes”;
- Se mostrará un listado con todos los huéspedes registrados;
- Elegirá un huésped;
- En la opción “Facturar”, verificará los datos;
- Elegir habitación ha facturar;
- Ejecutar la opción “Facturar por habitación”;
- Llenara los datos de la factura;
- Pulsará el botón “Factura”;
- Se genera la factura correspondiente ha la Habitación;

Resultado esperado

Se generará la factura correspondiente a la habitación, también se almacena en la Base de Datos, en la tabla Factura.

Evaluación de la prueba

Prueba satisfactoria.

Tarea 2: Desarrollar una Factura Global (varias habitaciones);

Partiendo de la Ventana Factura (Figura 4.20), escogemos la opción factura global, esta nos muestra una pantalla parecida a la ventana facturación por habitación (Figura 4.21), pero ha diferencia de esta muestra la sumatoria de todas las cuentas de las habitaciones que ocupo el grupo, Figura 4.22.

Figura 4.22 Ventana Factura Grupal

The screenshot shows a Windows application window titled "FACTURA". At the top, there are input fields for "La Paz, 2009-10-26; NIT/CI" (954681) and "Número de Factura" (000155). Below these is a field for "Señor(es)/Nombre Empresa" containing "Soboce". A section labeled "Detalle precio total" contains a table:

DETALLE	PRECIO TOTAL
alojamiento	180
alojamiento	160
alojamiento	160
garaje	40
Lavandería	50
restaurant	150
service room	50

The total amount is listed as "Total 790 Bs" and "Son setecientos noventa Bolivianos". There is a text area for "Comentario de la factura..." which is currently empty. At the bottom is a black button labeled "FACTURAR >>".

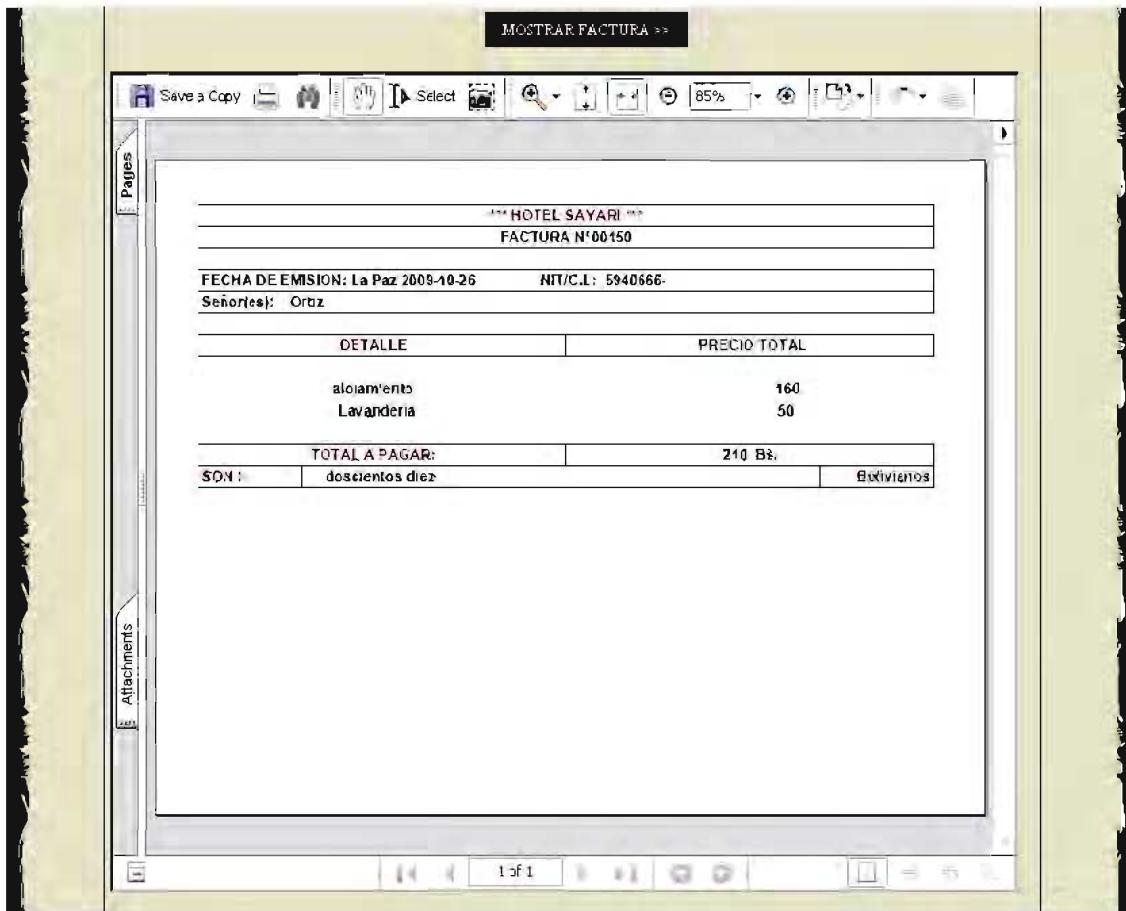
Prueba

La prueba es parecida a la, Tarea 1, y el resultado es satisfactorio, cabe notar que la factura global o por habitaciones una vez que se ejecuta significa que el huésped esta desocupando el hotel, y si desea volver debe registrarse nuevamente.

Tarea 3: Implementar el Botón que Genere el documento para Impresión;

Una vez que se llena el formulario de Factura, muestra la factura en formato pdf esta tiene los datos reales de la Figura 4.22, se muestra en la Figura 4.23.

Figura 4.23 Factura, Impresión



Una vez generado el documento pdf se puede mandar a impresión.

4.3.6 SEXTO INCREMENTO

Historia de Usuario N° 6: Modulo Caja;

Tarea 1: Desarrollar un modulo que pueda dar información a cerca de las Facturas Emitidas;

Partiendo la ventana menú principal (Fig. 4.12) escogemos las opción caja, aparece un menú desplegable, del que elegimos factura, Figura 4.24, en esta ventana tenemos dos opciones.

Para desarrollar este modulo se utilizo sentencias Sql.

La primera opción nos permite realizar la búsqueda de una factura por nombre y numero de factura, en el ejemplo se realiza una búsqueda por nombre.

La segunda opción sirve para mostrar las facturas en un determinado tiempo, las casillas para llenar son por orden cronológico, la fecha inicial y la fecha final, entonces busca todas las facturas emitidas en ese intervalo.

Figura 4.24 Busca factura



En la Figura 4.25 se muestra la búsqueda de la factura cuyo nombre era Ortiz.

Figura 4.25 Ejemplo de busca factura

The screenshot displays the search results page with the heading 'LOS RESULTADOS SON'. Below this, a table lists the found invoice. The table has columns: Nro. Factura, Nombre, Fecha, NIT, and Total Facturado. There is one row of data: Nro. Factura 10150, Nombre Ortiz, Fecha 2009-10-26, NIT 19240665, and Total Facturado 2.10.

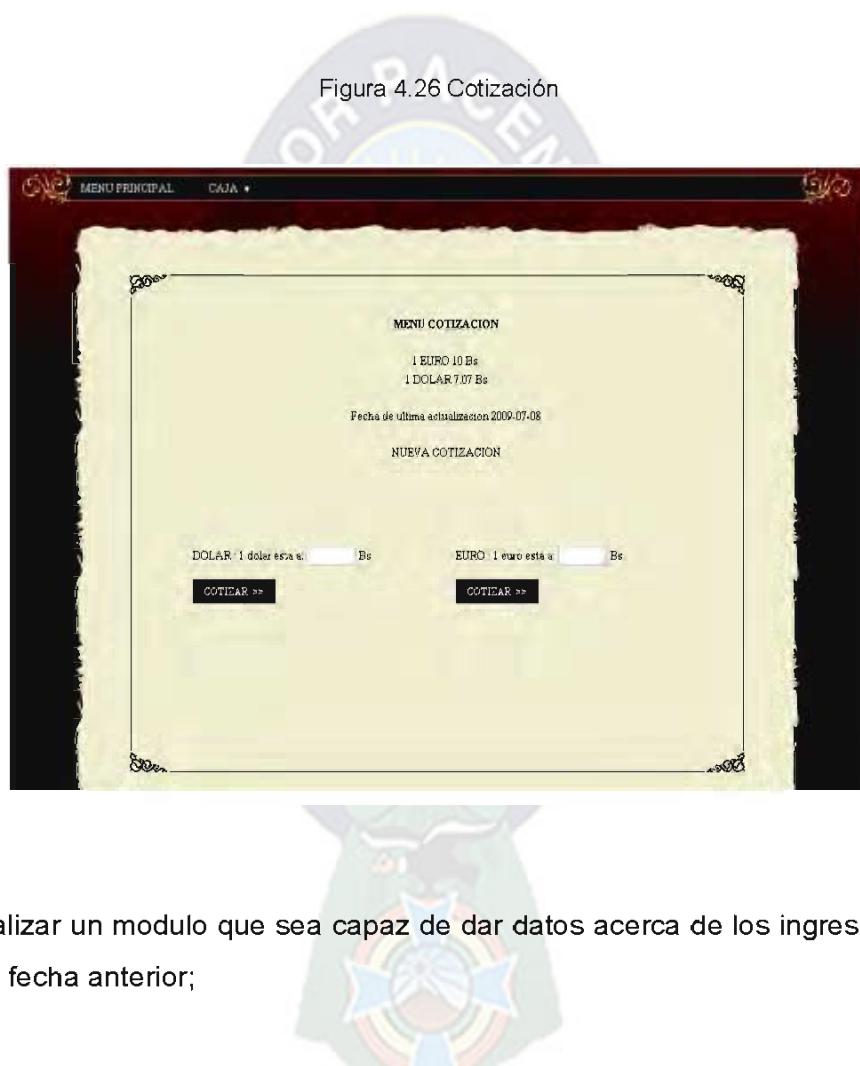
Nro. Factura	Nombre	Fecha	NIT	Total Facturado
10150	Ortiz	2009-10-26	19240665	2.10

Tarea 2: Implementar un modulo que cotice la moneda;

Partiendo del menú desplegable caja escogemos la opción cotizar, la Figura 4.26 muestra la ventana que realiza una cotización actual del cambio de moneda, en relación a las dos monedas que son más utilizados en este momento (Dólar y Euro).

Para tal fin primero se debe saber la relación que existe entre las monedas y la moneda boliviana.

Figura 4.26 Cotización



Tarea 3: Realizar un modulo que sea capaz de dar datos acerca de los ingresos del día, así como de una fecha anterior;

La Figura 4.27 ventana ingresos, muestra los ingresos del día, este proceso esta en función de la factura, da una cuenta total de los ingresos, estos datos son muy importantes para el administrador del hotel ya que sirven para realizar la auditoria nocturna además de el cierre del día. La ventana también cuenta con la opción buscar ingreso, que esta en función del la fecha.

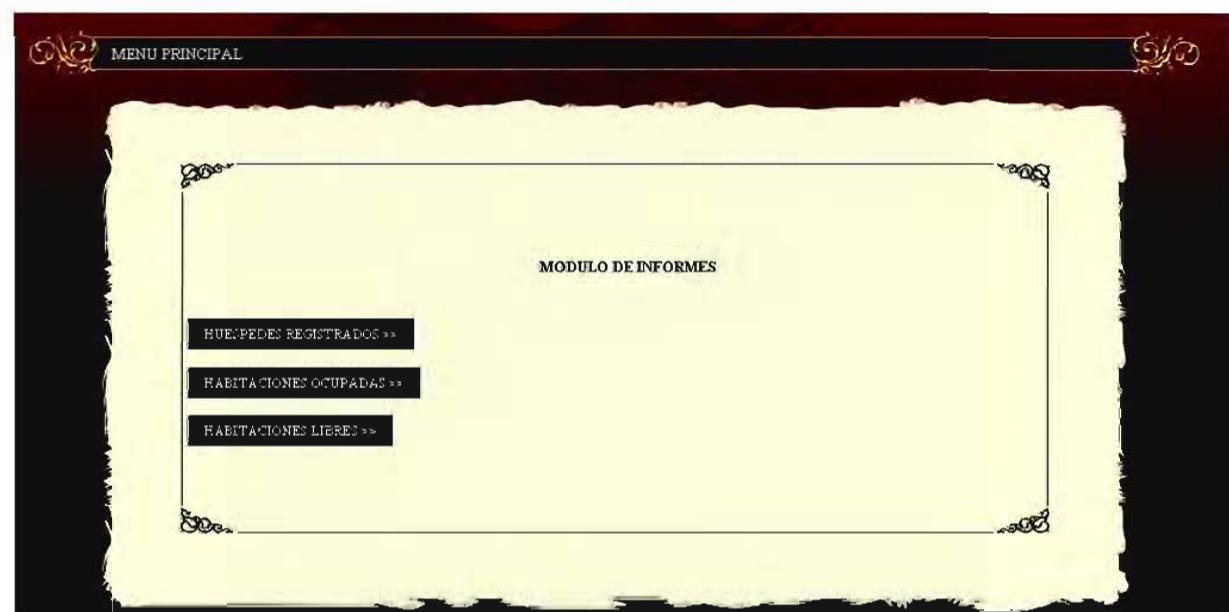
Figura 4.27 ventana ingresos



4.3.7 SÉPTIMO INCREMENTO

Historia de Usuario N° 7: Modulo Reportes;
Tarea 1: Desarrollar un modulo que genere informes;

Figura 4.28 ventana informes



En la Figura 4.28 se puede ver los tres informes que genera el sistema.

Huéspedes registrados, esta opción despliega un listado de todos los huéspedes que están alojados en el hotel, sirve para poder dar un informe diario a la policía turística, en la Figura 4.29 se puede ver un ejemplo del listado, también muestra la opción impresión.

Habitaciones libres, esta opción permite desplegar un listado de las habitaciones libres que tiene el hotel.

Habitaciones Ocupadas, esta opción permite desplegar un listado de las habitaciones Ocupadas en el hotel.

Figura 4.29 listado de huéspedes

HUESPEDES										
Apellido P.	Apellido S.	Nombre	Fecha ingreso	Hora	Habitación	Fecha de Nacimiento	Nacionalidad	Profesión	Documento	
Pacheco	Villaruel	Angel	2009-10-25	21:00:40	201	1984-04-01	Bolivia	constructor	456123	
Rodrigues	Rodrigues	Alvarado	2009-10-25	21:04:29	211	1982-05-06	Bolivia	informatico	9514683	

[Impresión >>](#)
[Atrás](#)

Tarea 2: Implementar el modulo de Estadísticas;

Figura 4.30 Estadísticas

MODULO DE ESTADISTICAS

ESTADISTICA POR FECHA

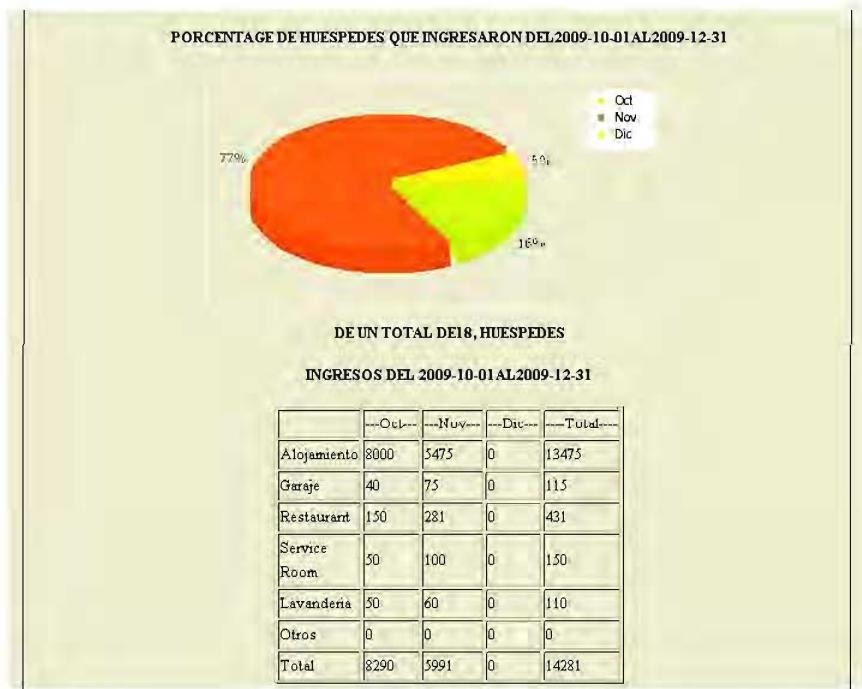
Fecha: Oct 2009

MOSTRAR >>

En la Figura 4.30 podemos ver la opción estadísticas, solo hay que introducir la fecha, entonces nos muestra dos tipos de estadísticas, el porcentaje de huéspedes que ingresaron

al hotel a lo largo de tres meses, expresados en una torta y también muestra una tabla que proporciona la información acerca de ingresos y servicios solicitados.

Figura 4.31 Estadísticas, gráficos



El porcentaje de huéspedes que ingresaron al hotel a lo largo de tres meses se realizó siguiendo las siguientes fórmulas:

$$\text{Porcentaje } i = (n_i * 100) / \text{Total}$$

Donde:

n_i = numero de personas del mes i

Total = Sumatoria n_i

Así, se saca el porcentaje de los 3 meses, que cumple la siguiente condición:

$$\text{Sumatoria porcentaje } i = 100\%$$

Los ingresos y servicios solicitados, se representa en una tabla en la que se obtiene la información por meses, en la ventana de la Figura 4.31, se puede apreciar los ingresos de octubre, noviembre y diciembre, detallando la sumatoria de cada uno de los servicios, esta información es útil para los administradores del Hotel, para determinar que servicio fue el más solicitado.

4.3.8 OCTAVO INCREMENTO

Historia de Usuario Nº 8: Modulo de Configuración;

Cabe notar que el modulo de configuración solo puede ser utilizado por un usuario que figure en el Sistema como Súper Administrador, no así recepción.

Tarea 1: Desarrollar un modulo que pueda configurar habitaciones;

En al figura 4.32 se puede apreciar la ventana configuración de habitación, se despliega un formulario con los campos a ser llenados por el administrador del sistema, también tiene la opción de eliminar habitación.

Figura 4.32 Configuración, Habitación

The screenshot shows a software interface titled "ADICIONAR HABITACION". It features several input fields and a text area. At the top, there are two input fields: "Codigo de habitación" and "Tipo de habitación". Below them are two more fields: "Vista de habitación" and "Costo de la habitación". Underneath these are two small input fields: "Capacidad" and "Piso". Below these is a text area labeled "Descripción de la habitación:" with the instruction "no mas de 255 caracteres". At the bottom right of the form is a button labeled "ADICIONAR >>". The overall design has a vintage, parchment-like aesthetic.

Para desarrollar este modulo se utilizo la clase habitación:

Figura 4.33 Clases del modulo Registro

Habitacion
cod_ha varchar(10)
tipo_ha varchar(50)
vista_ha varchar(20)
est_ha varchar(20)
cost_ha varchar(4)
desc_ha varchar(255)
cod_gr int(10)
cap_ha int(2)
npersonas_ha int(2)
piso_ha int(1)
Crear_h()
Ocupar_h()
Eliminar_h()
Modifica_h()
Listar_h()
Imprimir_h()

Prueba

Descripción

Una vez que el Administrador ha ingresado al Sistema, seleccionará la opción Configuración y elegirá “Habitación”, ingresa ha la venta donde puede adicionar una nueva habitación, realizara el llenado del formulario con cuidado, y presionará el Botón “Siguiente” y luego “Finalizar”.

Por otro lado si es que desea eliminar alguna habitación, escogerá el código y presionará el Botón “Siguiente” y luego “Finalizar”.

Condiciones de ejecución

Debe existir por lo menos una habitación en la Base de Datos, el administrador debe conocer el uso del Sistema.

Entrada

- Del menú principal seleccionará “Configuración”;
- Ingresar a Habitación;
- Decide entre Adicionar o Eliminar Habitación;

- En caso de Adicionar llenar el formulario;
- Presionar el Botón “Siguiente”;
- Presionar el Botón “Finalizar”;
- En caso de Eliminar Escoger el código de la Habitación;
- Presionar el Botón “Siguiente”;
- Presionar el Botón “Finalizar”;

Resultado esperado

En la Base de Datos, debe aparecer los datos de la habitación nueva si se adiciona una nueva habitación, si se elimina debe desaparecer la habitación que se selecciono.

Evaluación de la prueba

Prueba satisfactoria.

Tarea 2: Implementar un modulo que pueda configurar servicios;

Tarea 3: Realizar la opción que configure a los administradores del sistema;

Las tareas 2 y 3 son análogas, ver Anexos B.

4.3.9 NOVENO INCREMENTO

Historia de Usuario Nº 9: Aplicación de Contraseñas;

Tarea 1: Implementar Cifrado de datos, MD5;

En el sistema de Control y Gestión Hotelera, el algoritmo de cifrado se encuentra cuando se adhiere un nuevo administrador, o recepcionista, como se muestra en la Figura 4.34 esta encriptación se hace con PHP, el algoritmo de encriptación también se encuesta al ingresar al sistema cabe notar que esta encriptación se la realiza con JavaScript, (encriptación de lado del cliente), se hace una comparación entre código hash en la base de datos para el ingreso al Sistema.

Figura 4.34 Encriptación

NUEVO ADMINISTRADOR	
Apellidos:	Rodrigues
Caballero:	
Nombres:	Juan
Carlos	
Nacionalidad:	Bolivia
Ciudad:	La Paz
Estado Civil:	<input checked="" type="radio"/> Soltero <input type="radio"/> Casado
Domicilio:	villa Fatima, Nº 2130
Carnet de identidad	5840669
Teléfono:	73021144
Tipo de Usuario:	Recepción
Email:	donjuan@hotmail.com
Contraseña:	*****
Horario:	Tarde

Tarea 2: Implementar Ayudas, se desarrolla en el Anexo B

4.3.10 DÉCIMO INCREMENTO

Historia de Usuario Nº 10: Capacitación;

Tarea 1: Políticas de Seguridad;

Las políticas de seguridad a proponer tienen como función salvaguardar la integridad de la información que se almacenará en el sistema de control y gestión hotelera.

Para definir la seguridad del Sistema se describe el grado de actuación de los usuarios, los usuarios se definen a continuación.

- Recepcionista. Esta autorizado para realizar el registro, reserva, servicios, facturación e informes;
- Administrador. Puede realizar las tareas del recepcionista, además de realizar configuración, reportes y caja;
- Gerente general. Puede realizar las tareas del administrador en el sistema;
- Los botones, seguridad, mucamas y otros eventuales, no están autorizados a manipular el sistema;

Las políticas de seguridad son:

- Restringir el acceso a los Botones, seguridad y mucamas;
- Restringir el acceso al sistema para técnicos, servicios especiales y otros eventuales;

En cuanto a control de acceso y administración de contraseñas

- El sistema debe verificar que el usuario tenga una contraseña valida;
- El Cifrado de la contraseña;
- Cierre de sesión de Windows en caso de abandono e maquina;
- Cambio de contraseña cada tres meses;

Las políticas de seguridad en cuanto a controles de Software específico.

- El acceso al sistema operativo debe ser restringido;
- El acceso a archivos de programa debe ser restringido;

Las políticas en cuanto a seguridad física son las siguientes:

- Uso de estabilizadores para protección de equipos;
- Uso de extintores para equipo eléctrico;
- Actualización de antivirus;
- Obtención de copias periódicas de la Base de Datos;

Tarea 2: Resguardo de la Base de Datos

Es fundamental controlar con una política correcta las copias de seguridad para el resguardo de la Base de datos.

“Back-up”: hacer copias de los archivos periódicamente.

“Archiving”: copiar los archivos para almacenar por un largo periodo de tiempo, sea por cuestiones legales o por motivos de espacio.

4.4 DEMOSTRACIÓN DE OBJETIVOS

A continuación se desarrolla los objetivos secundarios que fueron propuestos en el Primer Capítulo.

- Desarrollar un módulo que pueda facturar, integrando procesos.

El modulo de facturación, se desarrolla en la historia de usuario N° 5, esta aplicación tiene la opción de generar la Factura para una habitación o mas, dependiendo del numero de habitaciones que haya tomado un grupo de huéspedes, para desarrollar este modulo intervienen los procesos de reserva, registro, historial y administrador, cabe notar que una vez realizada la facturación se presume que el huésped abandona el Hotel.

- Implementar una interfaz gráfica de las habitaciones del hotel, utilizando Matrices.

Figura 4.35 Interfaz utilizando matrices

SELECCIONE LA HABITACION ***											
Disponible			Ocupado			Reservado			Fuera de Servicio		
201 DOBLE AVENID	202 MATRIN AVENID	203 SIMPLE AVENID	204 SIMPLE AVENID	205 SUITE JUNIOR AVENID	206 DOBLE HALL	207 DOBLE HALL	208 DOBLE CALLE	209 MATRIN CALLE	210 CUADRU CALLE		
211 SIMPLE PATIO	212 SIMPLE PATIO	213 DOBLE PATIO	214 TRIPLE PATIO	215 DOBLE PATIO							
301 DOBLE AVENID	302 MATRIN AVENID	303 SIMPLE AVENID	304 SIMPLE AVENID	305 SUITE JUNIOR AVENID	306 DOBLE HALL	307 DOBLE HALL	308 DOBLE CALLE	309 DOBLE CALLE	310 CUADRU CALLE		
311 SIMPLE PATIO	312 SIMPLE CALLE	313 DOBLE PATIO	314 TRIPLE PATIO	315 MATRIN PATIO	316 DOBLE PATIO	317 DOBLE PATIO					
401 DOBLE AVENID	402 MATRIN AVENID	403 SIMPLE AVENID	404 SIMPLE AVENID	405 SUITE JUNIOR AVENID	406 DOBLE AVENID	407 DOBLE HALL	408 DOBLE CALLE	409 DOBLE CALLE	410 CUADRU CALLE		
411 SIMPLE PATIO	412 SIMPLE PATIO	413 DOBLE PATIO	414 TRIPLE PATIO	415 DOBLE PATIO	416 MATRIN PATIO	417 DOBLE PATIO					
501A TRIPLE CALLE	501B MATRIN CALLE	502A DOBLE PATIO	502B MATRIN PATIO	503A DOBLE PATIO	503B MATRIN PATIO	504A DOBLE PATIO	504B MATRIN PATIO	504C SIMPLE PATIO			

En el gráfico se puede apreciar las habitaciones que tiene el hotel representado, por pisos esta estructura se asemeja a la de una matriz.

En el desarrollo de la interfaz se utilizo tres matrices de 7x10.

Figura 4.36 Matriz 7x10

1,1									
									7,10

En la primera Matriz, cada una de las celdas se guarda el botón a ser ejecutado, este código esta echo en PHP.

En la segunda matriz se guarda los estados de la habitación, reservado, ocupado, disponible y fuera de servicio, en función a los estados podemos guardar también el color.

En la tercera matriz se guarda la descripción del la habitación, así como la fecha de reserva si es que esta estuviera reservada.

Otra solución para poder escoger una habitación es introducir el código de la habitación en un formulario para que se pueda procesar, lamentablemente esta solución no es segura porque el usuario puede introducir mal los datos, en caso de estar validado el formulario, el usuario tendría que escoger de una lista de mas de 50 opciones, de esta forma perdería tiempo.

La solución con matrices es eficiente, el tiempo de ejecución es menor a 1 segundos además es sencillo de usar.

- Realizar la Interfaz Grafica de Usuario, aplicando métricas.

La interfaz de Usuarios se realiza en la Historia de usuario N° 3, cabe notar que esta solo tiene 3 pantallas, a continuación se aplica la métrica propuesta ya con la interfaz de usuario completa.

$$CR = 100 \times [(\text{coste de la representación Óptima CR}) / (\text{coste de la representación propuesta})]$$

CR es la representación óptima de IGU.

Coste de representación propuesta:

$$\text{Costes} = \sum [\text{frecuencia de transición } (k) \times \text{coste de transición } (k)]$$

K representa las tareas en general como registro, reserva, reportes, caja y huéspedes.

No se toma la tarea de configuración pues es poco usado.

Coste propuesto:

Se toma una frecuencia de 100.

Y el costo de transición se mide en segundos

$$\text{Costes} = 15 \times 0.5 + 10 \times 0.5 + 25 \times 0.4 + 20 \times 0.5 + 30 \times 0.9$$

$$= 59.5 \quad \text{El coste propuesto}$$

El coste óptimo debe tener la misma frecuencia para cada k y el mismo tiempo para cada coste.

Así

$$\text{Coste Optimo} = 5 \times 20 \times 0.5$$

$$= 50$$

Por lo tanto:

$$\text{CR} = 100 \times [(\text{coste de la representación Óptima CR}) / (\text{coste de la representación propuesta})]$$

$$= (100 \times 50) / 59.5$$

$$= 84.7$$

$$\text{CR optimo} = 100$$

Entonces teniendo $\text{CR} = 84.7\%$, el CR del Sistema es Bueno.

- Implementar contraseñas, cifrando datos.

Se desarrolla en la Historia de Usuario Número 9, a continuación se hace una descripción del algoritmo MD5, cuya operación fundamental se basa en la multiplicación matricial.

Descripción del Algoritmo de Encriptación MD5

Código del Algoritmo MD5

```

/* Procesar cada bloque de 16 palabras. */
para i = 0 hasta N/16-1 hacer
    /* Copiar el bloque 'i' en X. */
    para j = 0 hasta 15 hacer
        hacer X[j] de M[i*16+j].
    fin para /* del bucle 'j' */
    /* Guardar A como AA, B como BB, C como CC, y D como DD. */
    AA = A
    BB = B
    CC = C
    DD = D
    /* Ronda 1. */
    /* [abcd k s i] denotarán la operación
       a = b + ((a + F(b, c, d) + X[k] + T[i]) <<< s). */
    /* Hacer las siguientes 16 operaciones. */
    [ABCD 0 7 1] [DABC 1 12 2] [CDAB 2 17 3] [BCDA 3 22 4]
    [ABCD 4 7 5] [DABC 5 12 6] [CDAB 6 17 7] [BCDA 7 22 8]
    [ABCD 8 7 9] [DABC 9 12 10] [CDAB 10 17 11] [BCDA 11 22 12]
    [ABCD 12 7 13] [DABC 13 12 14] [CDAB 14 17 15] [BCDA 15 22 16]
    /* Ronda 2. */
    /* [abcd k s i] denotarán la operación
       a = b + ((a + G(b, c, d) + X[k] + T[i]) <<< s). */
    /* Hacer las siguientes 16 operaciones. */
    [ABCD 1 5 17] [DABC 6 9 18] [CDAB 11 14 19] [BCDA 0 20 20]
    [ABCD 5 5 21] [DABC 10 9 22] [CDAB 15 14 23] [BCDA 4 20 24]
    [ABCD 9 5 25] [DABC 14 9 26] [CDAB 3 14 27] [BCDA 8 20 28]
    [ABCD 13 5 29] [DABC 2 9 30] [CDAB 7 14 31] [BCDA 12 20 32]

    /* Ronda 3. */
    /* [abcd k s t] denotarán la operación
       a = b + ((a + H(b, c, d) + X[k] + T[i]) <<< s). */
    /* Hacer las siguientes 16 operaciones. */
    [ABCD 5 4 33] [DABC 8 11 34] [CDAB 11 16 35] [BCDA 14 23 36]
    [ABCD 1 4 37] [DABC 4 11 38] [CDAB 7 16 39] [BCDA 10 23 40]
    [ABCD 13 4 41] [DABC 0 11 42] [CDAB 3 16 43] [BCDA 6 23 44]
    [ABCD 9 4 45] [DABC 12 11 46] [CDAB 15 16 47] [BCDA 2 23 48]
    /* Ronda 4. */
    /* [abcd k s t] denotarán la operación
       a = b + ((a + I(b, c, d) + X[k] + T[i]) <<< s). */
    /* Hacer las siguientes 16 operaciones. */
    [ABCD 0 6 49] [DABC 7 10 50] [CDAB 14 15 51] [BCDA 5 21 52]
    [ABCD 12 6 53] [DABC 3 10 54] [CDAB 10 15 55] [BCDA 1 21 56]
    [ABCD 8 6 57] [DABC 15 10 58] [CDAB 6 15 59] [BCDA 13 21 60]
```

[ABCD 4 6 61] [DABC 11 10 62] [CDAB 2 15 63] [BCDA 9 21 64]
 /* Ahora realizar las siguientes sumas. (Este es el incremento de cada uno de los cuatro registros por el valor que tenían antes de que este bloque fuera inicializado.) */

A = A + AA

B = B + BB

C = C + CC

D = D + DD

fin para /* del bucle en 'l' */ Fuente:[Aloriel, 2009]

A continuación se puede apreciar el cifrado de la contraseña de un administrador en la Base de Datos, longitud de 32 bits.

Figura 4.37 Base de Datos, MD5

email	horario	activo	contraseña
Sayoja@hotmail.com	Noche	0	a45fdb1e4ac646c9e65a1769663e5704
edcars@hotmail.com	Mañana	1	d964173dc44da83eeafa3aebbee9a1a0

- Desarrollar un sistema multiusuario, utilizando el modelo Cliente Servidor.

La arquitectura Cliente servidor conciente a la aplicación, se desarrolla en Historia de Usuario Nº 2, cabe añadir que el protocolo utilizado es el TCP/IP.

- Implementar el proceso de Reservas mediante Listas Dobles.

Características de las listas doblemente enlazadas

Crear, Adicionar, Eliminar;

A continuación el código fuente de Adicionar un valor a la lista doblemente enlazada.

```
function crea_listad($contador,$valor)
{
If ($contador==1)
$listad=array("posicion"=> "$contador","anterior" => "0","siguiente" => "$contador +1 ","valor"
=> "$valor, );
Else
$listad=array("posicion"=> "$contador","anterior" => "$contador -1","siguiente" => "$contador
+1","valor" => "$valor, );
return($listad);}
```

La siguiente función se utiliza en la ventana de reserva de Huéspedes, una vez llenado el formulario se procede ha elegir la habitación, es ahí donde se almacena los datos de la habitación reservada y se van creando las listas dobles en función de las habitaciones reservadas.

Función de asignación de habitación

```
$codhab=$_POST['codigoh'];  
$cont=1;  
While($codhab<>NULL)  
{crea_listad($contador,$listadoble);  
reduce($codhab,$valor);  
adiciona_listad($cont,$valor,);  
$cont++;  
}  
}
```

Una vez que se escogen las habitaciones reservadas, se pasan los datos de la lista Dblemente enlazada, a la Base de Datos.

4.5 CALIDAD

Se han desarrollado varios modelos de calidad para diferentes productos y procesos software, donde la mayor parte de ellos tiene como base la norma ISO 9126, por lo que se aplica esta norma para medir la calidad del sistema; es decir, medir los atributos mediante la utilización de métricas.

4.5.1 FUNCIONALIDAD

La métrica del punto función, es un método para medir el tamaño del software. Pretende medir la funcionalidad entregada al usuario independientemente de la tecnología utilizada.

La Tabla 4.1 y 4.2 muestran las características del dominio e la información del sistema.

Tabla 4.1: Matriz de Punto Función

Parámetros de medición	Factor de ponderación				
	Cuenta	Simple	Medio	Complejo	Total
Numero de entradas de Usuario	2	3	4	6	8
Numero de salidas de Usuario	23	4	5	7	92
Numero de peticiones de usuario	14	3	4	6	42
Numero de archivos	13	7	10	15	91
Numero de interfaces externas	2	5	7	10	10
Cuenta Total					243

La escala se mide de 0 a 5, donde 0 significa sin importancia hasta llegar a 5 que significa esencial, Tabla 4.2.

Tabla 4.2 Punto Función

Escala	0	1	2	3	4	5
Factor						
Requiere el sistema copia de seguridad y de recuperación fiable						X
Se Requiere comunicación de datos					X	
Existen funciones de procesos distribuidos					X	
Es crítico el rendimiento		X				
Sera ejecutado el sistema en un S.O. existente						X

Requiere el Sistema entrada interactiva					X	
Requiere entrada de datos interactiva sobre múltiples ventanas		X				
Se actualizan los archivos maestros de forma interactiva					X	
Son complejas las salidas, los archivos a la petición			X			
Es complejo el procesamiento interno				X		
Se ha diseñado el código para ser reutilizable.				X		
Están incluidas en el diseño la conversión y la instalación.				X		
Se ha diseñado el sistema para soportar múltiples instalaciones.		X				
Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario						X
TOTAL						46

Los resultados obtenidos con $i=14$, y los valores de la Tabla 4.2 se tiene el siguiente valor:

$\sum F_i = 46$. Remplazando en la Formula de punto Función.

$$PF = \text{cuenta-total} * (0.65 + 0.01 * \sum [F_i])$$

Donde:

0.65: Valor mínimo de ajuste de complejidad

0.01: Factor de conversión, asumiendo un error de 1%

$\sum [F_i]$: Valores de ajuste de complejidad

$$PF = 243 * (0.65 + 0.01 * 46)$$

$$PF = 269.73$$

$$\text{Sacando } PF_{\max}, \text{ con } \sum [F_i] = 70$$

$$PF_{\max} = 243 * (0.65 + 0.01 * 70)$$

$$PF_{\max} = 328.05$$

Por tanto la funcionalidad esta dada por:

$$(PF / PF_{\max}) * 100\% = (269.73 / 328.05) * 100\% = 82\%$$

Así se concluye que el sistema tiene una funcionalidad es del 82%.

- **Exactitud**

Toda la información referente al modulo reportes esta almacenada en la Base de Datos y es exacta, tiene que ver con las habitaciones ocupadas, libres y Huéspedes Registrados, gracias a esto el administrador cuenta con información, con un gran nivel de confianza.

- **Seguridad de Acceso**

Se describe claramente la seguridad que tiene el sistema en cuanto al acceso de los usuarios haciendo el cifrado de los datos, se especifica en la Historia de Usuario Nº 9.

4.5.2 FIABILIDAD

La fiabilidad o confiabilidad de un producto software se define como la probabilidad de operación libre de fallos del sistema en un entorno determinado y durante un tiempo específico. La confiabilidad del sistema se calcula a partir de los subsistemas o módulos.

Se considera dos casos:

CASO 1: Si n componentes que funcionan independientemente, están conectados en serie, y si la i-ésima componente tiene la confiabilidad $R_i(t)$, la confiabilidad $R(t)$ del sistema completo esta dada por:

$$R(t) = R_1(t) R_2(t) R_3(t) \dots R_n(t)$$

CASO 2 : Si n componentes que funcionan independientemente están conectados en paralelo, y si la i-ésima componente tiene la confiabilidad $R_i(t)$, la confiabilidad $R(t)$ del sistema completo esta dada por:

$$R(t) = 1 - \{1 - R_1(t)\} \{1 - R_2(t)\} \{1 - R_3(t)\} \dots \{1 - R_n(t)\}$$

Como $R(t)$ se halla en un intervalo de tiempo, se puede aplicar la distribución exponencial para su cálculo como se muestra a continuación:

$$R(t) = e^{-\lambda t}$$

Donde: λ

$R(t)$: Confiability del subsistema.

λ : Error o tasa constante de fallas.

t : tiempo de operación del sistema.

La Figura muestra el análisis del modelo planteado anteriormente donde se muestran las combinaciones.

Figura 4.38 Modelo del sistema

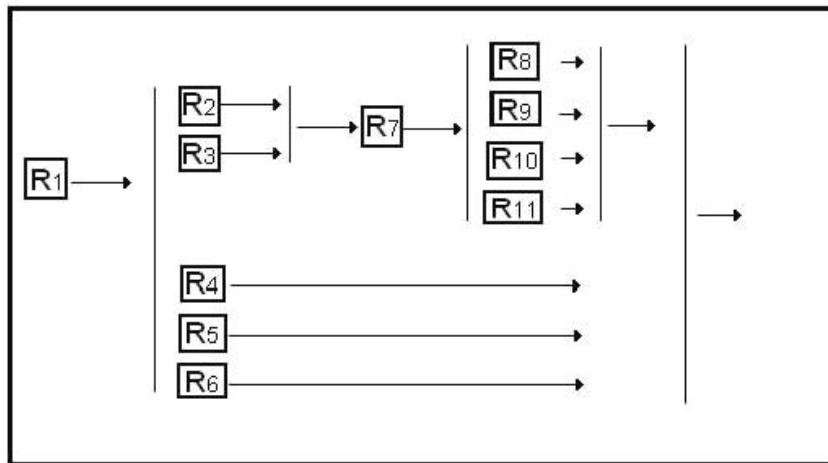


Tabla 4.3 Confiabilidad por componente del Sistema

Descripción	R _i	λ	P(t)	P(t)- λ
Autentificación	R ₁	0.11	1	0.89
Registro	R ₂	0.1	1	0.9
Reserva	R ₃	0.1	1	0.9
Reportes	R ₄	0.05	1	0.95
Caja	R ₅	0.07	1	0.93
Configuración	R ₆	0.1	1	0.9
Huéspedes	R ₇	0.05	1	0.95
Datos	R ₈	0.1	1	0.9
Habitación	R ₉	0.1	1	0.9
Servicios	R ₁₀	0.05	1	0.95
Factura	R ₁₁	0.07	1	0.93

$$R_{12} = 1 - (1 - R_2)(1 - R_3) = 0.99$$

$$R_{13} = 1 - (1 - R_8)(1 - R_9)(1 - R_{10})(1 - R_{11}) = 0.99$$

$$R_{14} = R_{12} \cdot R_7 \cdot R_{13} = 0.93$$

$$R_{15} = 1 - (1 - R_{14})(1 - R_4)(1 - R_5)(1 - R_6) = 0.99$$

$$R_{16} = R_1 \cdot R_{15} = 0.88$$

Por lo tanto podemos Concluir que la fiabilidad del sistema es del 88%.

- **Recuperabilidad**

Para prever perdida de información en caso de falla eléctrica, falla del equipo, perdida de la conexión, etc. toda acción de registro, modificación y eliminación se la realiza directamente en la base de datos, además cuenta con "Back-up".

- **Tolerancia a Fallas**

El software desarrollado presenta funcionalidad parcial cuando se produce un error de funcionamiento, siempre y cuando el error no se presente con el servidor de base de datos.

4.5.3 USABILIDAD

Responde a la pregunta ¿Es fácil entender y reconocer a la estructura de y la lógica de su aplicabilidad?

El sistema fue desarrollado bajo la supervisión del usuario final, tomando en cuenta todos los requerimientos, así mismo fue aprobando cada una de sus iteraciones y modificaciones echas al sistema de modo que se asegura la comprensibilidad, es atractivo y fácil de entender esto no discrimina el realizar una capacitación para el correcto uso del sistema y para los posibles fallos que pudiesen existir o los casos especiales que existen en el modelamiento.

4.5.4 EFICIENCIA

La eficiencia esta relacionado con el comportamiento del sistema en el tiempo y de los recursos utilizados, dado que el sistema tarda menos de 2 segundos en realizar cualquier proceso y sus resultados son confiables entonces el sistema es eficiente.

- **Comportamiento con Respecto al Tiempo**

El Sistema desarrollado utiliza el Lenguaje Estructurado de Consulta (SQL) para realizar solicitar información al servidor de base de datos, de esta forma la transferencia de información es rápida, además que solo se solicitan los datos necesarios para trabajar, de

esta manera el intercambios de información entre el sistema y el servidor no contempla mucho tiempo.

- **Comportamiento con Respecto a Recursos**

El sistema desarrollado realiza las consultas necesarias a la Base de Datos para un mejor aprovechamiento de los recursos del servidor.

4.5.5 MANTENIBILIDAD

- **Índice de madurez del software**

Se representa la cantidad de esfuerzo, requerida para conservar el funcionamiento normal o para restablecerlo una vez que se presenta un fallo. A continuación se establece el índice de madurez del software (IMS), que proporciona indicación de la estabilidad del software basada en cambios que ocurren en cada versión.

Esta dado por:

$$IMS = [Mt (Fa +Fc +Fd)] / Mt$$

Donde:

Mt : Número de módulos en la versión actual

Fc : Número de módulos en la versión actual que se han cambiado

Fa : Número de módulos en la versión actual que se han añadido

Fd : Número de módulos en la versión anterior que se han borrado en la versión actual

Aplicando al sistema:

$$IMS = [33 - (2 + 3 +1) / 33] = 0,81$$

A medida que el IMS se aproxima a 1.0, el sistema se empieza a estabilizar.

- **Facilidad de Análisis**

Frente a una falla de código, ofrecen mensajes de ubicación de los errores propios del lenguaje de programación, De este modo se pueden identificar la ubicación de los errores para una posterior modificación y rectificación de los mismos.

- **Facilidad de Cambio**

Gracias a la estructura orientada a objetos del lenguaje de desarrollo del software (PHP) la modificación del código fuente del software es sencilla e intuitiva. Además todas las modificaciones del código fuente se adaptan fácilmente al entorno del sistema, siempre y cuando se respeten los protocolos de comunicación con el servidor de base de datos y la estructura de la base de datos.

4.5.6 PORTABILIDAD

Consiste en el esfuerzo necesario para transferir el programa de un entorno de hardware y software a otro.

En cuanto a hardware, el sistema no usa demasiados recursos, por lo tanto el cambio no afecta en gran medida.

En cuanto a la Plataforma, al ser un sistema hecho en PHP y MySql, puede ser soportado por las plataformas LINUX y WINDOWS.

Gracias a que la base de datos se encuentra separa del software, el sistema puede compartir este recurso con otras aplicaciones.

5. PARTE FINAL

En este capítulo se presenta las conclusiones del proyecto, el cual demuestra el alcance de los objetivos planteados en el capítulo I. En las recomendaciones se propone el desarrollo de trabajos que se pueden desarrollar posteriormente.

5.1 Conclusiones

Una vez terminado el proceso de construcción del sistema se concluye que:

- Se ha desarrollado e implementado el Sistema de Control y Gestión Hotelera, obteniendo un producto de calidad, cumpliendo con los puntos requeridos por el usuario;
- Se ha implementado una herramienta de Software que realiza la Reserva y Registro de los huéspedes en forma rápida;
- El Sistema realiza un seguimiento acerca de los Servicios adquiridos por el huésped, de forma confiable, además se tiene acceso a los informes, estadísticas e ingresos de manera rápida;
- Se desarrollo e implemento el modulo de Facturación de manera satisfactoria;
- La interfaz de usuario satisfizo a los Administradores del Hotel;

Por lo tanto se ha cumplido con los objetivos propuestos.

5.2 Recomendaciones

- Desarrollar un subsistema de restaurante;
- Realizar un subsistema que realice el seguimiento y control de inventarios;
- Si bien el sistema puede manejar grupos de huéspedes, es necesario ampliar un subsistema que controle los convenios y eventos que se tiene con empresas Turísticas.



REFERENCIAS BIBLIOGRAFICAS

Pressman Roger. 2002, "Ingeniería de Software", Quinta edición, Editorial McGraw Hill. , México D.F.

C. J. Data, 2001, "Introducción a los Sistemas de Base de Datos", Séptima Edición, Editorial PEARSON EDUCACION. , México D.F.

Hernández Sampieri Roberto, Fernández Collao Carlos, Baptista Lucio Pilar. 1999, "Metodología de la Investigación", Segunda edición, Editorial McGRAW-HILL INTERAMERICANA EDITORES. México D.F.

Téllez Flores José Rodolfo. 2002, "Legislación Turística de Bolivia", G.B.T. Editores, La Paz Bolivia.

Luís Joyanes Aguilar, 2003, "Fundamentos de Programación, Algoritmos, Estructura de Datos y Objetos", Tercera Edición, Editorial McGraw Hill., España.

Raúl Alarcón. 2000, "Diseño Orientado a Objetos con UML", Editorial Grupo EIDOS., España.

Stig Saether Bakken, Alexander Aulbach, Egon Schmid, Jim Winstead, Lars Torben Wilson, Rasmus Lerdorf, Andrei Zmievski, Jouni Ahto. 2001, "MANUAL DE PHP", Editado por Rafael Martínez, Grupo de documentación de PHP.

Pino Caballero Gil. 2003, "Introducción a la Criptografía", 2^a Edicion actualizada, RA-MA Editorial, Madrid España.

Luis Joyanes Aguilar, 2000, "Programación Orientada a Objetos", Segunda Edición, Editorial McGraw Hill., España.

James Rumbaugh, Ivar Jacobson, Grady Booch, 2000,"El Lenguaje Unificado de Modelado. Manual de Referencia", PEARSON EDUCACIÓN, S.A., Madrid.

Richard Cuela. 2007, "Sistema de Seguimiento y Control Caso: Hotel Sagarnaga", La Paz Bolivia.

Cabero, 2000,"Sistema de Información Hotelera", La Paz Bolivia.

Loza, 2001,"Sistema de Seguimiento y Control para la Hotelería Boliviana", La Paz Bolivia.

REFERENCIAS WEB

RoyFoker.(2009). Hotel.En wikipedia [Online]. Disponible: <http://es.wikipedia.org/wiki/Hotel> [2009, marzo 15]

Isha.(2009). MySQL.En wikipedia [Online]. Disponible: <http://es.wikipedia.org/wiki/MySQL> [2009, octubre 11]

Juan López E. (24 de noviembre 2007). Hoteles. Consultado el día 15 de marzo de 2009 de la World Wide Web, <http://www.arqphys.com/contenidos/hoteles-historia.html>.

Alejandro Aguilar Sierra.(22 de diciembre 2007) . XP. Consultado el día 15 de marzo de 2009 de la World Wide Web, <http://www.programacionextrema.org/>.

Xavier Ferré Grau, María Isabel Sánchez Segura. (20 de septiembre 2007). Desarrollo Orientado a Objetos con UML. Consultado el día 16 de diciembre de 2008 de la World Wide Web, <http://www.itox.mx/Comunidad/Librero/>

Aloriel.(2009). MD5.En wikipedia [Online]. Disponible: <http://es.wikipedia.org/wiki/MD5> [2009, marzo 15]

Camilo.(2009).JavaScript.En wikipedia [Online].
Disponible: <http://es.wikipedia.org/wiki/JavaScript> [2009, julio 15]

ArthurBot.(2009). jQuery.En wikipedia [Online]. Disponible: <http://es.wikipedia.org/wiki/Jquery> [2009, julio 15]

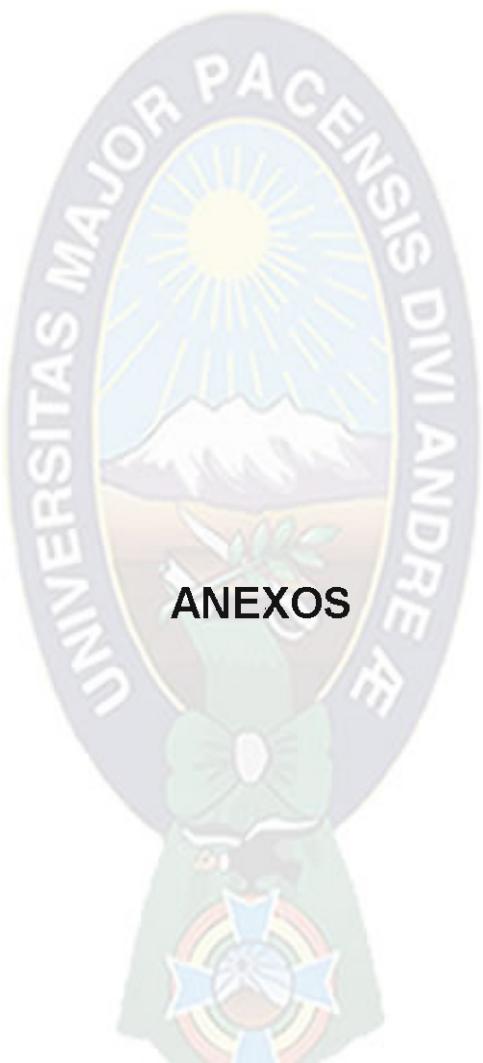
GLOSARIO DE TERMINOS

Hotel: Establecimiento que presta en forma permanente el servicio de hospedaje en unidades habitacionales.

Habitación: Es el ambiente especialmente diseñado y amoblado para que el huésped pueda encontrar descanso. En ella debe existir una o más camas de acuerdo al tipo de habitación Simple, Doble, Triple etc...

Suite: Es aquella unidad habitacional que consta por lo menos de dos áreas claramente diferenciadas una habitacional y una zona social, intercomunicadas entre si.

Huésped: Es el cliente de un establecimiento hotelero.



ANEXOS

ANEXO A. HISTORIAS DE USUARIO

Historia de Usuario Nº2

Historias de usuario
Número :2
Nombre: Registro de Huéspedes y Reserva de habitación.
Descripción: Como segunda Historia de usuario se decide realizar el Registro y la Reserva de los huéspedes, para tal efecto se detalla la información que debe ser rellenada en los formularios teniendo mucho cuidado en la fecha de ingreso y egreso del huésped además de el llenado de todas las habitaciones en la Base de Datos.
Observaciones: Sin observaciones.

Tarea	
Número tarea: 1	Número Historia: 2
Nombre tarea: Diagramar el modelo Entidad-Relación.	
Descripción: Debido a que el gestor de base de datos es relacional, se pasa a convertir el diagrama de clases al modelo relacional mediante el modelo objeto relación que sirve para tal propósito. El modelo entidad relación describe los datos como entidades las cuales se relacionan entre si cuya característica fundamental es la de poder representar sus relaciones mediante axiomas matemáticos.	

Tarea	
Número tarea: 2	Número Historia: 2
Nombre tarea: Diagramar el modelo Jerárquico;	
Descripción: Es el diagrama que nos muestra los módulos que poseerá el sistema de Control y Gestión Hotelera.	

Tarea	
Número tarea: 3	Número Historia: 2
Nombre tarea: Diseñar Esquema de Ubicación de Ordenadores;	
Descripción: La ubicación de los ordenadores es importante para los usuarios, que interactuarán con el sistema y la ubicación es primordial.	

Tarea	
Número tarea: 4	Número Historia: 2
Nombre tarea: Desarrollar modulo Registro;	
Descripción: Registrar los datos de los huéspedes que ingresan al hotel, nombres, apellidos, nacionalidad, e-mail, domicilio, pasaporte, profesión, compañía, procedencia, destino, fecha de ingreso y egreso, etc.., son datos que deben ser llenados en la base de datos.	

Tarea	
Número tarea: 5	Número Historia: 2
Nombre tarea: Realizar modulo Reserva;	
Descripción: Este modulo debe ser capaz de registrar los datos mínimos del huésped además de las habitaciones que reserva para una fecha determinada.	

Historia de Usuario Nº3

Historias de usuario
Numero :3
Nombre: Interfaz Grafica de Usuario y validaciones.
Descripción: La Interfaz Grafica de usuario, la mediación entre el usuario y la maquina, es fundamental para seguir con el desarrollo el sistema, pues hasta el momento es sistema por

el mismo echo de no tener interfaz no es agradable, además es necesario implementar las validaciones para poder controlar los datos introducidos.

Observaciones: Sin observaciones.

Tarea

Número tarea: 1

Número Historia: 3

Nombre tarea: Crear la Interfaz Gráfica.

Descripción: Se Realiza el Diseño preliminar de la interfaz gráfica de Usuario, tomando en cuenta los colores preferidos del Administrador del Hotel, teniendo gran cuidado en la manera de combinar.

Tarea

Número tarea: 2

Número Historia: 3

Nombre tarea: Validar campos.

Descripción: La información de los campos, que se introduce de forma manual tiende a ser incorrecta se desarrolla la validación tanto para caracteres tipo int y char.

Historia de Usuario Nº4

Historias de usuario

Numero :4

Nombre: Modulo que Realice Seguimiento de los Huéspedes Registrados.

Descripción: El modulo debe ser capas de obtener la información de un grupo determinado de huéspedes, las habitaciones que ocupan, los servicios que adquirieron y la cuenta.

Observaciones: Sin observaciones.

Tarea

Número tarea: 1

Número Historia: 4

Nombre tarea: Desarrollar un modulo que obtenga la información de los Huéspedes que ocupan una determinada habitación.

Descripción: Debido a que más de un huésped puede ocupar una habitación (no simple) se hace imprescindible saber quienes ocupan una determinada habitación.

Tarea	
Número tarea: 2	Número Historia: 4
Nombre tarea: Realizar un modulo que pueda mostrar y cambiar la habitación del huésped.	
Descripción: Debido a que en algunas ocasiones el huésped puede quedar insatisfecho con la habitación. Este módulo tiene que realizar el cambio de habitación a una con características similares, esto no debe afectar los servicios que adquirió la antigua habitación	

Tarea	
Número tarea: 3	Número Historia: 4
Nombre tarea: Implementar un modulo que pueda adicionar, modificar y eliminar los servicios a una determinada habitación además dar la cuenta.	
Descripción: Debido a que no se puede saber que huésped pidió tal o cual servicio, se recarga la cuenta por habitación. Este modulo debe ser capaz de dar la información de los servicios que adquirió una habitación además de adicionar, modificar y eliminar servicios.	

Historia de Usuario Nº5

Historias de usuario
Numero :5
Nombre: Facturación.
Descripción: La Factura que recibe el huésped debe tener información rápida y confiable, además debe poder realizar la facturación ya sea por habitación o grupal (grupo que ocupa varias habitaciones).
Observaciones: Sin observaciones.

Tarea	
Número tarea: 1	Número Historia: 5
Nombre tarea: Desarrollar una Factura por habitación.	

Descripción: El modulo debe realizar la facturación de la habitación escogida, presentando un formulario con los datos característicos de una factura, efectuando la sumatoria de los servicios obtenidos, además de sacar el literal del valor total.

Tarea	
Número tarea: 2	Número Historia: 5
Nombre tarea: Desarrollar una Factura Global (varias habitaciones).	
Descripción: El modulo debe realizar la facturación de las habitaciones ocupadas por un grupo, presentando un formulario con los datos característicos de una factura, efectuando la sumatoria de los servicios obtenidos, además de sacar el literal del valor total, se asume que una vez se haya facturado el huésped abandona el hotel.	

Tarea	
Número tarea: 3	Número Historia: 5
Nombre tarea: Implementar el Botón que Genere el documento para Impresión;	
Descripción: Realizar un botón que genere un documento con la información de la Factura (simple o grupal), en formato .PDF, para su posterior impresión.	

Historia de Usuario Nº6

Historias de usuario
Numero :6
Nombre: Modulo Caja.
Descripción: Desarrollar un modulo capaz de dar información acerca de las facturas emitidas, implementar una herramienta apta para realizar la cotización de las monedas más utilizadas (Dólar y Euro), y los ingresos.
Observaciones: Sin observaciones.

Tarea	
Número tarea: 1	Número Historia: 6

Nombre tarea: Desarrollar un modulo que pueda dar información a cerca de las Facturas Emítidas.

Descripción: Implementar un modulo que Busque facturas por nombre, Nit y por fecha (un rango).

Tarea	
Número tarea: 2	Número Historia: 6
Nombre tarea: Implementar un modulo que cotice la moneda.	
Descripción: Realizar una herramienta que pueda cotizar de forma confiable el Dólar y Euro.	

Tarea	
Número tarea: 3	Número Historia: 6
Nombre tarea: Realizar un modulo que sea capaz de dar los datos acerca de los ingresos del día, así como de una fecha anterior.	
Descripción: Implementar un modulo que Busque los ingresos por fecha (un rango), que genere un listado además de la información de los ingresos del día.	

Historia de Usuario Nº7

Historias de usuario	
Numero :7	
Nombre: Modulo Reportes.	
Descripción: Implementar los informes acerca de los huéspedes registrados, habitaciones ocupadas y habitaciones libres, además del modulo de estadísticas.	
Observaciones: Sin observaciones.	

Tarea	
Número tarea: 1	Número Historia: 7
Nombre tarea: Desarrollar un modulo que genere informes.	

Descripción: El modulo informes debe generar la lista de huéspedes registrados para el informe a la policía turística, habitaciones ocupadas y libres para enviar a las empresas turísticas.

Tarea	
Número tarea: 2	Número Historia: 7
Nombre tarea: Implementar un modulo de Estadísticas.	
Descripción: Desarrollar el modulo que pueda mostrar el porcentaje de ingresos en tres meses y los ingreso en función a los servicios que ofrece el Hotel.	

Historia de Usuario Nº8

Historias de usuario
Numero :8
Nombre: Modulo de Configuración.
Descripción: Implementar un modulo que pueda configurar tanto habitaciones, servicios y administradores.
Este modulo solo podrá ser usado por el Administrador o Gerente General del Hotel.
Observaciones: Sin observaciones.

Tarea	
Número tarea: 1	Número Historia: 8
Nombre tarea: Desarrollar un modulo que pueda configurar habitaciones.	
Descripción: Implementar en el modulo configuración la opción de adicionar y eliminar habitación, con el cuidado de introducir el código de la habitación de acuerdo al piso de la misma	

Tarea	
Número tarea: 2	Número Historia: 8

Nombre tarea: Implementar un modulo que pueda configurar servicios.

Descripción: Implementar en el modulo configuración la opción de adicionar y eliminar Servicios, con el cuidado de introducir el código de servicio en función a la secuencia de estos.

Tarea	
Número tarea: 3	Número Historia: 8
Nombre tarea: Realizar que configure a los administradores del sistema.	
Descripción: Implementar en el modulo configuración la opción de adicionar y eliminar administrador, con el cuidado de escoger entre un usuario tipo “recepción” o tipo “Súper administrador”, además de una contraseña de por lo menos 6 caracteres.	

Historia de Usuario Nº9

Historias de usuario
Numero : 9
Nombre: Aplicación de Contraseñas.
Descripción: Es importante aplicar contraseñas, para restringir el acceso al sistema de personas ajenas que puedan utilizar la información de manera inadecuada.
Observaciones: Sin observaciones.

Tarea	
Número tarea: 1	Número Historia: 9
Nombre tarea: Implementar Cifrado de datos, MD5.	
Descripción: Implementar el algoritmo MD5 al ingresar al sistema y al adicionar un nuevo administrador.	

Historia de Usuario Nº10

Historias de usuario
Numero : 10

Nombre: Capacitación.

Descripción: Tiene como objeto que los usuarios aprendan el adecuado manejo del sistema.

Observaciones: Sin observaciones.

Tarea	
Número tarea: 1	Número Historia: 10
Nombre tarea: Políticas de Seguridad.	
Descripción: Las políticas de seguridad reflejan una serie de normas, reglas metas y protocolos a seguir, donde se definen los distintos modelos a tomar para proteger la seguridad del sistema con las funciones y responsabilidades de los distintos componentes de la institución y los mecanismos para controlar su correcto funcionamiento.	

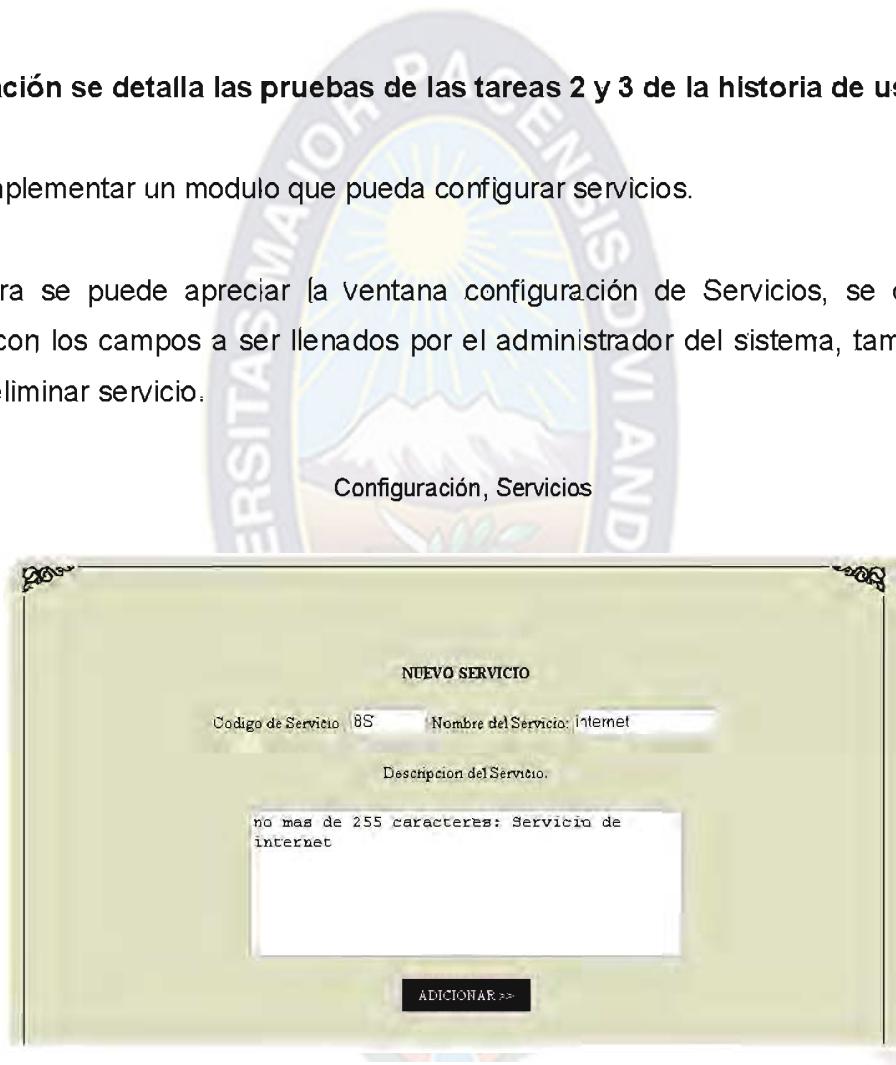
Tarea	
Número tarea: 2	Número Historia: 10
Nombre tarea: Resguardo de la Base de Datos.	
Descripción: Se trata de los métodos que existen para poder resguardar la base de datos en caso que surja alguna contingencia.	

ANEXO B. Tareas

A continuación se detalla las pruebas de las tareas 2 y 3 de la historia de usuario Nº8

Tarea 2: Implementar un modulo que pueda configurar servicios.

En al Figura se puede apreciar la ventana configuración de Servicios, se despliega un formulario con los campos a ser llenados por el administrador del sistema, también tiene la opción de eliminar servicio.



Para desarrollar este modulo se utilizo la Clase servicios:

Clase servicios

Servicios
cod_ser:varchar(5)
nombre_ser:varchar(15)
desc_ser:varchar(100)
Crear_ser()
Eliminar_ser()
Modifica_ser()
Imprimir_ser()

Prueba

Descripción

Una vez que el Administrador ha ingresado al Sistema, seleccionará la opción Configuración y elegirá "Servicios", ingresa ha la venta donde puede adicionar un nuevo Servicio, realizara el llenado del formulario con cuidado, y presionará el Botón "Siguiente" y luego "Finalizar".

Por otro lado si es que desea eliminar algún servicio, escogerá el código y presionará el Botón "Siguiente" y luego "Finalizar".

Condiciones de ejecución

Debe existir por lo menos un servicio en la Base de datos, tabla servicios, el administrador debe conocer el uso del Sistema.

Entrada

- Del menú principal seleccionará "Configuración";
- Ingresar a Servicios;
- Decide entre Adicionar o Eliminar Servicio;
- En caso de Adicionar llenar el formulario;
- Presionar el Botón "Siguiente";
- Presionar el Botón "Finalizar";
- En caso de Eliminar Escoger el código del servicio;
- Presionar el Botón "Siguiente";
- Presionar el Botón "Finalizar";

Resultado esperado

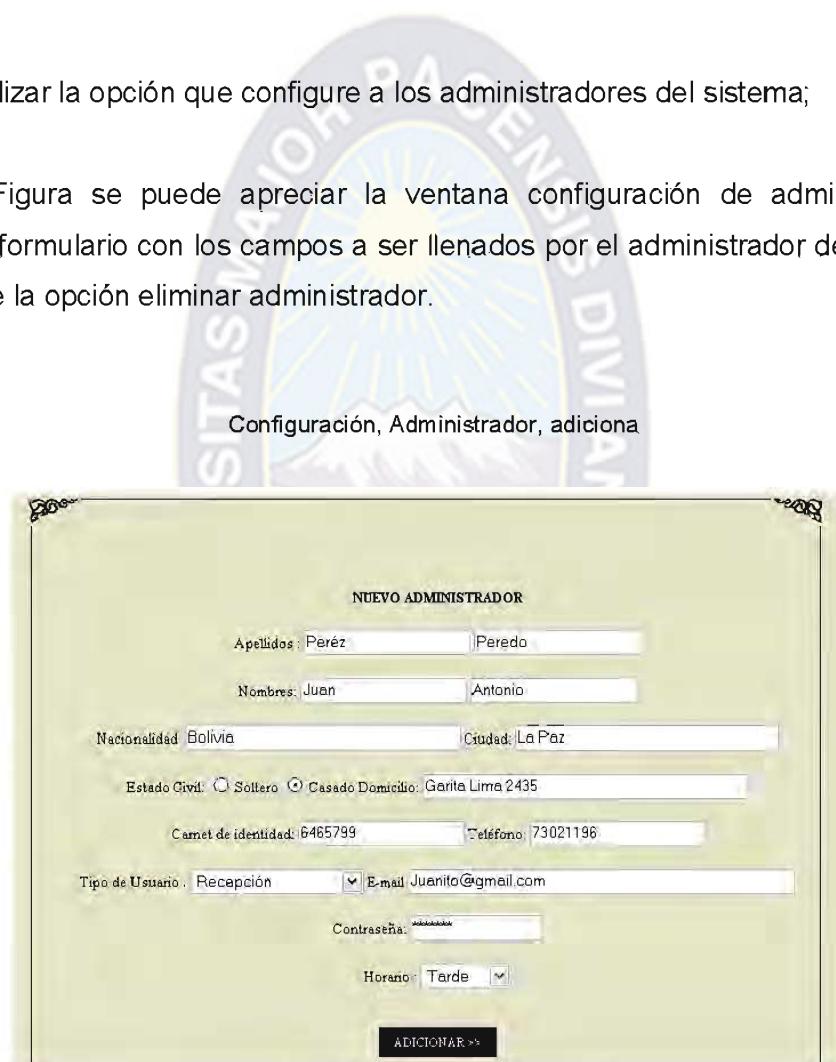
En la Base de Datos, debe aparecer los datos del servicio nuevo si se realizo la adición, si se elimino debe desaparecer el servicio que selecciono.

Evaluación de la prueba

Prueba satisfactoria.

Tarea 3: Realizar la opción que configure a los administradores del sistema;

En la Figura se puede apreciar la ventana configuración de administradores, se despliega un formulario con los campos a ser llenados por el administrador del sistema. Mas abajo se tiene la opción eliminar administrador.



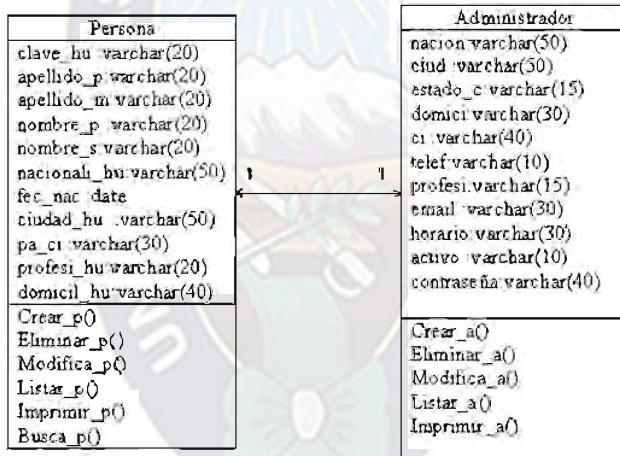
Configuración, Administrador, elimina

ELIMINAR ADMINISTRADOR

Primer Apellido	Segundo Apellido	Primer Nombre	C.I.	Estado
Cayoja	Salvador	Mansol	789456	0
Salinas	Baldivieso	Edgar	387946	1

Para desarrollar este modulo intervinieron las siguientes clases:

Clases del modulo administrador



Prueba

Descripción

Una vez que el Administrador ha ingresado al Sistema, seleccionará la opción Configuración y elegirá “Administradores”, ingresa ha la venta donde puede adicionar un nuevo administrador, realizara el llenado del formulario teniendo mucho cuidado en la opción tipo de usuario y en la contraseña pues el nuevo administrador debe llenar este dato, y presionará el Botón “Siguiente” y luego “Finalizar”.

Por otro lado si es que desea eliminar algún servicio, escogerá el código y presionará el Botón “Siguiente” y luego “Finalizar”.

Condiciones de ejecución

Debe existir por lo menos un administrador en la Base de Datos, tabla “administrador”, el administrador debe conocer el uso del Sistema.

Entrada

- Del menú principal seleccionará “Configuración”;
- Ingresar a Administradores;
- Decide entre Adicionar o Eliminar Administrador;
- En caso de Adicionar llenar el formulario;
- Presionar el Botón “Siguiente”;
- Presionar el Botón “Finalizar”;
- En caso de Eliminar Escoger el Apellido del Administrador;
- Presionar el Botón “Siguiente”;
- Presionar el Botón “Finalizar”;

Resultado esperado

En la Base de Datos, debe aparecer los datos del nuevo administrador si se realizo la adición, si se elimino debe desaparecer el administrador de la Base de Datos.

Evaluación de la prueba

Prueba satisfactoria.

A continuación se detallas la tarea Nº 2 de la Historia de Usuario Nº 9

Tarea 2: Implementar Ayudas;

Se decidió implementar ayudas sencillas en el sistema, links de nombre “ayuda” dispuestos en cada una de las ventanas, la idea es poder solucionar cualquier duda que surja al utilizar el sistemas, estas ayudas están programadas en JavaScript, y son ventanas veloces pues se ejecutan del lado del cliente.

[Link Ayuda](#)



Ejecutando la opción ayuda se genera una ventana emergente donde nos describe la función de cada uno de los “botones”

En la siguiente figura se puede ver la ventana ayuda del menú principal:



Cabe notar que se entregó el manual de usuario que es la mejor ayuda que se puede proporcionar.



DOCUMENTOS