

UNIVERSIDAD AMAZÓNICA DE PANDO
ÁREA DE CIENCIAS Y TECNOLOGÍA
CARRERA DE INGENIERÍA INFORMÁTICA



PROYECTO DE GRADO

**“SISTEMA INFORMÁTICO DE GESTIÓN HOTELERA
PARA LA RESIDENCIAL OPI”**

**INFORME FINAL DE PROYECTO DE GRADO PARA OPTAR AL TÍTULO DE
LICENCIADO EN INGENIERÍA DE SISTEMAS INFORMÁTICOS**

Postulante: Univ. Cesar Roly Crispin Condorceth
Tutor: Lic. Florencio Milton Ramírez Linares
Asesor: Lic. Humberto Fernández Calle

Cobija - Pando - Bolivia

2011

DEDICATORIA

La fe, el esfuerzo y el optimismo dedicado a lo largo de los años de estudio, son el fruto de la gente que ha creído en mi persona, apoyándome en todo sentido. Es por ello que este trabajo está dedicado a las personas que a lo largo de mi vida me han dado la formación de ser persona.

Con mucho cariño a mis padres Paulino e Irma por haber sido el principal apoyo a lo largo de mi vida.

AGRADECIMIENTOS

Sin duda alguna, para llegar a esta etapa hubo muchas personas que me han brindado su apoyo, entre docentes, familia, amigos y compañeros, a los cuales hago llegar mi sincero agradecimiento.

Pero específicamente, deseo expresar mi agradecimiento al M.Sc. Lic. Juan Carlos Huanca Guanca, por haberme transmitido sus conocimientos tanto a nivel académico, con sus observaciones y aportes con el seguimiento del trabajo, a nivel moral, con sus consejos y ejemplo.

A la vez deseo agradecer al Lic. Humberto Fernández, revisor del presente trabajo, quien me brindó su colaboración desde el inicio del mismo, aportando con sus conocimientos y tiempo en el seguimiento del trabajo.

RESUMEN

Los Sistemas Informáticos en la actualidad se han convertido en una herramienta fundamental para la sociedad, fruto del avance tecnológico en estas últimas décadas en la cual se han mostrado, herramientas más complejas capaces de cubrir las necesidades con gran precisión y rapidez, en ese sentido en nuestro medio gran parte de la empresas han optado por estas herramientas para tener beneficios.

Sin embargo existen empresas que aun no han optado por esto, como el casos de “Residencial Opi”, lo cual significa que todos los procesos son relativamente manuales, de tal manera que estos ocasionan retrasos, pérdida de recursos entre otros, en tal sentido el presente trabajo trata de la necesidad de construir un Sistema Informático desarrollado a través del Método de Diseño de Hipermedia Orientado a Objetos (OOHDM), lo cual da solución los problemas identificados en dicha entidad.

El Sistema Informático Gestión Hotelera una herramienta que permite el tratamiento automático de la información, facilitando en gran medida su organización, proceso, transmisión y almacenamiento.

INDICE

CAPITULO I INTRODUCCION

1. ANTECEDENTES	2
2. DESCRIPCIÓN DEL PROBLEMA	2
3. SOLUCION PROPUESTA	3
4. OBJETIVOS	3
4.1 OBJETIVO GENERAL	3
4.2 OBJETIVOS ESPECÍFICOS	3
5. ALCANCES	4
6. METODOLOGIA Y HERRAMIENTAS UTILIZADAS	4
7. RESULTADOS OBTENIDOS	5
8. ORGANIZACIÓN DEL DOCUMENTO	5

CAPITULO II MARCO TEORICO

1. REFERENCIA INSTITUCIONAL	7
2. SISTEMA	7
2.1. CARACTERISTICAS DE LOS SISTEMAS	8
2.2. SISTEMAS INFORMATICOS	8
2.3. ELEMENTOS DE UN SISTEMA INFORMATICO	8
3. SISTEMA DE GESTION HOTELERA	10
4. GESTION ADMINISTRATIVA	11
4.1. GESTION EN HOTELERA	11
4.2. AREAS DE TRABAJO EN UN HOTEL	12
4.2.1 Funciones y realizaciones en un Hotel	13
5. METODO OOHDM PARA EL DESARROLLO DEL SISTEMA	15
5.1. FASES DEL METODO OOHDM	16
6. ELEMENTOS DE IMPLEMENTACION DEL SISTEMA	18
6.1. PATRON DE DISEÑO MODELO VISTA CONTROLADOR	18
6.2. SMARTY	19
6.3. BASE DE DATOS	20
6.4. MYSQL	20
6.4.1 Características principales	21
6.4.2 Ventajas	22
6.5. ARQUITECTURA CLIENTE SERVIDOR	22

6.6. AJAX	23
6.6.1. Tecnologías que engloba AJAX	23
7. APLICACIONES WEB	24
8. PARADIGMA DE PROGRAMACION ORIENTADO A OBJETOS	24
9. PLATAFORMA DE DESARROLLO DEL SISTEMA	25

CAPITULO III

DESARROLLO DEL SISTEMA

1. FASE DE INICIO	
1.1 OBTENCION DE REQUERIMIENTOS	28
1.1.1 Modelo de entrevista y cuestionario	28
1.1.2 Especificación de roles y tareas	31
1.1.3 Especificación de casos de uso	32
2. FASE DE DISEÑO	41
2.1 DISEÑO CONCEPTUAL	41
2.2 DISEÑO NAVEGACIONAL	43
2.2.1 Esquema de clases navegacionales	43
2.2.2 Esquema de contextos navegacionales	45
2.3 DISEÑO DE INTERFAZ ABSTRACTA DEL SISTEMA	46
2.4 IMPLEMENTACION	48
2.4.1 Diseño de la base de datos	49
2.4.2 Definición de clases con PHP 5	50
2.5 DISEÑO DE INTERFACES DEL SISTEMA	56
2.6 PRUEBAS DEL SISTEMA	60

CAPITULO IV

CONCLUSIONES Y RECOMENDACIONES

1. CONCLUSIONES	64
2. RECOMENDACIONES	65

BIBLIOGRAFIA

ANEXOS

CAPITULO I

INTRODUCCION

1. ANTECEDENTES

Los sistemas informáticos de gestión hotelera (SIGH) han evolucionado, incorporando nuevas tecnologías tanto desde el punto de vista funcional (como las redes de computadoras) como de diseño. Es fundamental que los SIGH actuales sigan evolucionando hacia nuevos sistemas que cubran en su totalidad los procesos realizados en una entidad hotelera.

En tal sentido es importante incorporar un Sistema Informático a una entidad hotelera sabiendo que dentro del ámbito hotelero el movimiento de turistas, viajeros (huéspedes potenciales) se constituyen los elementos más importantes para las entidades mencionadas, por otro lado la obtención de utilidades, se basa exclusivamente en la prestación de servicios como son: hospedajes, reservación de habitaciones consumos y/o ventas. Por tanto, un inadecuado control de estos servicios puede traer consecuencias negativas, entre las que se encuentra las pérdidas económicas, mermas y desperdicios.

2. DESCRIPCION DEL PROBLEMA

La actividad de Residencial Opi como cualquier entidad hotelera se enmarca en el rubro de prestación y venta de servicios, sin embargo estas actividades o procesos de la entidad son relativamente manuales, lo cual ocasiona pérdidas referente a tiempo y costo, de esa manera se demuestra ineficiencia a la hora de atender a los clientes, lo cual implica una mala administración en la entidad.

Además la entidad no cuenta con un sistema informático, que coadyuve en los diversos procesos mencionados, por tanto la no disposición de un Sistema Automatizado, hace que todo proceso del control de hospedajes y control de servicios sean completamente manuales, provocando deficiencias en el manejo de información y la atención a los clientes, afectando así en la gestión administrativa.

Lo mencionado anteriormente, el problema se sintetiza en el **“Deficiente control en los servicios que presta la Residencial Opi”**.

En tal sentido es necesario el desarrollo e implementación de un sistema, con el fin de

asistir en la toma de decisiones de manera oportuna y confiable, además ayudara a mejorar la obtención de utilidades.

3. SOLUCION PROPUESTA

El Sistema Informático de Gestión Hotelera para la “Residencial Opi”, a partir de su implementación mejorará la calidad de atención a los clientes, además permitirá la reorganización del personal que trabaja en dicha entidad, de esa manera dará solución a los problemas inicialmente planteados.

4. OBJETIVOS

4.1 OBJETIVO GENERAL

Desarrollar un Sistema Informático de Gestión Hotelera (SIGH) para el control eficiente de los servicios que presta la Residencial Opi, con arquitectura Cliente-Servidor haciendo el uso de la metodología “OOHDM (Método de Diseño Hipermedia Orientado a Objetos).”

4.2 OBJETIVOS ESPECIFICOS

- Analizar los requerimientos del dominio en estudio como etapa inicial.
- Construir el modelo conceptual del sistema para establecer la estructura basada en objetos (además de sus relaciones), a través del modelo de datos Entidad-Relación.
- Diseñar el Modelo de navegación de la aplicación.
- Diseñar las diferentes interfaces abstractas de la aplicación, para definir la forma en el cual deben aparecer los objetos de navegación o contextos navegacionales.
- Implementar el sistema con la utilización de las siguientes herramientas PHP5 para la definición de clases de implementación, para las interfaces (Ajax. Smarty, HTML) y como manejador de base de datos MySQL.
- Realizar las pruebas del sistema con la técnica de Caja Negra.

5. ALCANCES

El sistema de informático de gestión hotelera constará de 7 módulos los cuales serán:

- Administración de Clientes
- Administración de Habitaciones
- Administraciones de Roles de Usuarios
- Administración de Productos
- Caja
- Administración de Opciones de menús
- Modulo de Seguridad

El sistema a desarrollarse basado en web será multiusuario, el cual inicialmente funcionará con un solo cliente, siendo así en el futuro se podrá adicionar clientes en función a los requerimientos por la entidad. Cabe mencionar que el sistema no contempla el manejo contable de la entidad.

6. METODOLOGIA Y HERRAMIENTAS UTILIZADAS

Para el desarrollo del presente proyecto se hará el uso del método científico cualitativo esto de acuerdo al objetivo planteado, para lo cual se tiene como instrumentos de trabajo la observación, entrevista, y cuestionarios, este enfoque cualitativo permitirá clasificar la información relevante y existente a partir del uso de los instrumentos mencionados en el presente trabajo de investigación.

Para el desarrollo del sistema se hará el usos del método de OOHDM (Método de Diseño Hipermedia Orientado a Objetos), del ingles (Object Oriented Hypermedia Design Methodology), el mismo fue creado para el desarrollo de aplicaciones web, que posee las siguientes fases para su desarrollo:

- Obtención de Requerimientos
- Diseño conceptual
- Diseño Navegacional

- Diseño de la interfaz abstracta
- Implementación

7. RESULTADOS OBTENIDOS

- Con la implementación del Sistema Informático de Gestión Hotelera se ha percibido la mejora en lo que es la atención a los clientes.
- Con la implementación del sistema toda la información que maneja la entidad esta centralizado lo cual permite mejor control de los servicios.
- Así mismo la implementación del Sistema Informático de Gestión Hotelera (SIGH) ha permitido la reorganización del personal lo cual significa que cada funcionario de la entidad tiene su función definida, esto con el fin de llevar a cabo un trabajo de manera muy ordenado.

8. ORGANIZACIÓN DEL DOCUMENTO

Para una mejor comprensión o interpretación, el presente documento está organizado por capítulos en tal sentido se presenta de manera resumida el contenido que posee por cada capítulo.

- ✓ **Capítulo I.** Es la parte introductoria del documento, donde se tiene definido la identificación del problema, solución propuesta, objetivos, alcance y la metodología.
- ✓ **Capítulo II.** Contempla las bases teóricas que sustenta el presente trabajo.
- ✓ **Capítulo III.** En este capítulo se describe el proceso del desarrollo del Sistema de Informático.
- ✓ **Capítulo IV.** Este capítulo contempla el desarrollo del trabajo (Sistema Informático) el mismo fue llevado a cabo de acuerdo a la metodología propuesta.
- ✓ **Capítulo V.** Contempla los aspectos más sobresalientes fruto del desarrollo e implementación del sistema.

CAPITULO II

MARCO TEORICO

1. REFERENCIA INSTITUCIONAL

El departamento de Pando ubicado en el extremo norte de Bolivia frontera con los países Perú y Brasil; se caracteriza por ser parte de la Amazonía, posee un potencial turístico incalculable, siendo rubros principales el turismo de aventura, el turismo científico e histórico, los cuales permiten una fuerte afluencia turística, además permite que los ingresos económicos crezcan significativamente, más que todo en los centros turísticos y en el sector hotelero.

La capital del departamento es la ciudad de Cobija, una de las ciudades con más habitantes y la más importante del departamento, el cual se encuentra en plena frontera con el país Brasil, goza de ciertos privilegios como: la zona franca, el cual permite mayor afluencia de visitantes extranjeros como nacionales, en tal sentido la demanda para las entidades hoteleras es muy grande. En particular para la Residencial Opi.

La residencial Opi está ubicado en la calle cívica barrio Mira Flores, en pleno centro de la ciudad de Cobija, la residencial aparece en los planos de la ciudad desde los años 1998. Dos años después ingresa a las actividades hoteleras, con la prestación de servicios como son: hospedaje, lavandería, ventas y otros.

La planta administrativa de la residencial, en conocimiento de que toda la estructura del negocio de la entidad se basa en la buena atención de los clientes, pero todos los procesos y registros se realizan en forma manual los mismos muchas veces ocasionan perdidas en cuanto a sus utilidades, en tal sentido suelen tener una mala gestión administrativa, lo cual hace que los clientes no regresen ni hagan recomendaciones de calidad. Incidiendo en los ingresos de la institución.

2. SISTEMA

Bertalanffy en 1969 definió el sistema como “complejo de componentes interactivos, conceptos característicos de unidades organizadas como interacción, suma, mecanización, centralización, competencia, finalidad,...y su aplicación a fenómenos concretos”.

El termino Sistemas tiene muchas connotaciones "conjunto de elementos interdependientes e interactuantes; grupo de unidades combinadas que forman un todo organizado. El ser humano, por ejemplo es un sistema que consta de varios órganos y miembros; sólo cuando estos funcionan de un modo coordinado el hombre es eficaz. De igual manera, se puede pensar que la organización es un sistema que consta de varias partes interactuantes". En realidad, el sistema es "un todo organizado o complejo; un conjunto o combinación de cosas o partes que forman un todo complejo o unitario".

Actualmente, las empresas han optado por incorporar a su estructura organizacional, Sistemas Informáticos, que operan como soporte a sus actividades.

2.1 CARACTERÍSTICAS DE LOS SISTEMAS

El aspecto más importante del concepto sistema es la idea de un conjunto de elementos interconectados para formar un todo, que presenta propiedades y características propias que no se encuentran en ninguno de los elementos aislados. Es lo que denominamos emergente sistémico: una propiedad o característica que existe en el sistema como un todo y no en sus elementos particulares. Del sistema como un conjunto de unidades recíprocamente relacionadas, se deducen dos conceptos: propósito (u objetivo) y globalismo (o totalidad). Esos dos conceptos reflejan dos características básicas de un sistema.

2.2 SISTEMA INFORMATICO

Según (Fernández 2009) Un sistema informático es el conjunto de elementos de hardware y software orientados al procesamiento Automatizado de la información en una rama concreta de la actividad humana, a los efectos de proveer los resultados informativos en un tiempo y con un costo, tal que no pudieran ser obtenidos por otros medios.

2.3 ELEMENTOS DE UN SISTEMA INFORMÁTICO

- Computadoras
- Medios de programación (Software) S.O, programas de comunicaciones, herramientas, de aplicación.

- Instrucciones destinadas al operador, al usuario y al proveedor de la información al sistema. Estas tienen el objetivo de reglamentar y asegurar la actividad del sistema en su conjunto.
- Informaciones almacenadas en la Base de Datos o que fluyen entre los diferentes elementos estructurales del sistema.
- Funcionarios, especialistas y trabajadores en general, encargados de emitir informaciones o de utilizar la que resulta de la actividad del sistema informático.
- Sensores, captadores de información en máquinas o procesos productivos, comerciales, etc.
- Líneas y equipos de comunicaciones y enlace entre computadoras.
- Dispositivos de almacenamiento de información para ser utilizados por el sistema.
- Documentos o formularios para captar la información de entrada al sistema y para reflejar los resultados. En dependencia del sistema pueden estar presentes o no.
- Equipos auxiliares.

Un sistema informático puede ser tan simple que conste de un solo programa como también más complejo aun que pueda comprender cientos de programas. Puede funcionar con una sola computadora o con sistema de máquinas gigantes.

El sistema informático puede abarcar diferentes funciones, incluir un conjunto de datos, que se utilicen por una función o tarea, proporcionarle información a departamentos y funcionarios.

Está compuesto por diferentes programas, orientado a resolver tareas y funciones, las cuales pueden ser denominadas como subsistemas. A su vez estos pueden subdividirse más de una vez, con lo que se formaría una jerarquía de subsistemas.

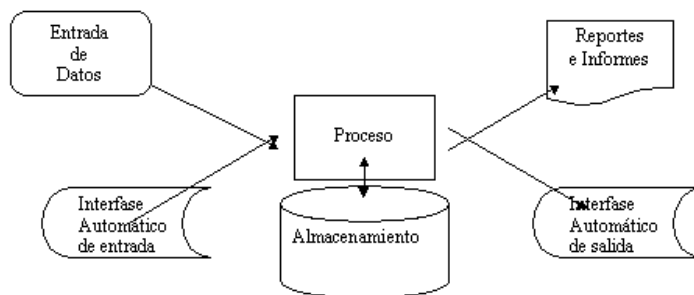


Figura 1: Esquema de un sistema informático
Fuente: Elaboración propia

3. SISTEMAS DE GESTION HOTELERA

En la actualidad la mayoría de las empresas tienen implantados sistemas informáticos que gestionan los distintos procesos que realiza. Estos sistemas cubren una gran parte de los diferentes procesos que tienen las entidades hoteleras.

En los hoteles, como en cualquier organización de negocios, se han implantado sistemas de información de forma generalizada (en su mayoría los hoteles poseen un SI o material informático para su gestión). Es más, en la actualidad es indispensable el uso de un sistema de información para afrontar la alta competitividad del mercado actual y poder ofrecer los niveles de calidad exigidos por el cliente.

En un sistema informático de gestión hotelera (SIGH) se distinguen diferentes procesos, algunos de estos procesos tienen un carácter muy peculiar debido a las particularidades de los servicios que ofrece un hotel (Inkpen, 1994).

Actualmente, los SIGH consisten en un conjunto de bases de datos más el procesamiento asociado. Además, se incluyen una serie de funciones de interconexión con otros sistemas auxiliares como centrales telefónicas o TPV (terminales de punto de venta). Así mismo, es importante que un SIGH aporte las siguientes funciones (Kasavana, 1994):

- Apoya al proceso de toma de decisiones.
- Ayuda a la monitorización y operaciones de control.
- Respuesta a las necesidades dinámicas de la empresa.

4. GESTIÓN ADMINISTRATIVA

Gestión conocida como una herramienta de la administración la cual se encarga de ejecutar todas aquellas órdenes planificadas por la administración, en este caso, de empresas o institución. Más específicamente, la definición de gestión nos indica que se trata de realización de diligencias enfocadas a la obtención de algún beneficio, tomando a las personas que trabajan en la compañía como recursos activos para el logro de los objetivos. Como bien hemos explicado a lo largo de este sitio, la gestión se divide en todas las áreas que comprenden a una empresa, ya que todos los departamentos que la administración tiene bajo su cargo, deben estar involucrados con la gestión empresarial.

Del latín *gestio*, el concepto de gestión hace referencia a la acción y al efecto de gestionar o de administrar. Gestionar es realizar diligencias conducentes al logro de un negocio o de un deseo cualquiera. Administrar, por otra parte, consiste en gobernar, dirigir, ordenar, disponer u organizar.

El término gestión, por lo tanto, implica al conjunto de trámites que se llevan a cabo para resolver un asunto o concretar un proyecto. La gestión es también la dirección o administración de una empresa o de un negocio.

4.1 GESTION EN HOTELARIA

Administración hotelera tienen origen desde la función básica que posee un administrador de empresas y es importante que se considere que se encuentra inicialmente limitada a la planificación y a la distribución eficientemente realizada de todos los fondos necesarios para cubrir aquellas operaciones que le corresponden a un negocio empresario de este estilo.

La administración hotelera, ha sufrido una evolución drástica de tal manera que, si en una empresa de hotelería se implementa un modelo eficiente para la administración hotelera, es posible que mediante la misma se puede lograr el alcance de todos aquellos objetivos que se impone una empresa al momento de su constitución, especialmente cuando se coordinan los procesos actualizados mediante los cuales los responsables de la empresa hotelera toman las decisiones correspondientes que incluyen todas las actividades que se desarrollan en el hotel en cuestión.

La administración hotelera persigue dos importantes objetivos que poseen una prioridad diferente. Por ejemplos, si hablamos del objetivo primario diremos que se trata de la maximización de todas aquellas acciones que están por el bienestar económico de las personas propietarias de la empresa hotelera. Es importante tener en consideración que este objetivo de la administración hotelera se interpreta como la maximización del precio con las que se costean las acciones comunes, algo que implica una diferencia con respecto a la maximización correspondiente a la ganancia por acción o a la ganancia neta.

El segundo objetivo claro que tiene que ver con la aplicación de la administración hotelera es, definitivamente, el **lograr la satisfacción tanto de los empleados como de la comunidad en la cual la empresa lleva a cabo sus actividades**, por ende, es común asociar esto con los objetivos personales de la empresa, sin embargo, si se habla de la administración hotelera desde un punto de vista financiero, generalmente el mismo resulta tener una relevancia mucho menores cuando a la maximización de los precios correspondientes a las acciones.

4.2 AREAS DE TRABAJO EN UN HOTEL

- Gerencia General
- Administrador alimentos y bebidas
- Recepcionista
- Administración de Pisos
- Anfitriones
- Logística
- Ejecutivo de Ventas

- Asistente comercial
- Jefatura de lencería y lavandería
- Guía turístico
- Organizador de eventos
- Organizador de negocios y recreaciones turísticos

4.2.1 Funciones y realizaciones

- Asistir en la planificación general del establecimiento
- Apoyar en la elaboración de los presupuestos del departamento o área de su responsabilidad.
- Identificar el equipo humano necesario en su departamento o área, en el marco de la planificación estratégica y de las directrices recibidas.
- Dirigir al personal a su cargo, promoviendo su participación en el logro de los objetivos y motivándolos para que tenga una alta capacidad de respuesta a las necesidades de los clientes.
- Organizar, ejecutar y controlar en el ámbito de su competencia la política de calidad definida.
- Evaluar y controlar los ingresos, los costos y rendimientos del ámbito de su competencia para un adecuado seguimiento económico administrativo.
- Organizar los recursos necesarios para prestar un eficiente servicio de recepción, conserjería y comunicaciones.
- Supervisar las reservas, satisfaciendo los deseos y expectativas de los clientes.
- Organizar las entradas y salidas de los clientes, atendiéndolos con inteligencia y cortesía, generando la información que permita la programación de la prestación del servicio.
- Atender a los clientes durante su estancia, satisfaciendo sus expectativas y cumpliendo el estándar de calidad del establecimiento.
- Formalizar la documentación y gestionar la información derivada del alojamiento y los servicios prestados a los clientes en los distintos puntos de atención, de modo que pueda ser útil a la empresa y a otros departamentos.
- Registrar y supervisar los servicios consumidos por los clientes; comprobar la atención adecuada.

- Supervisar el área de telecomunicaciones de modo que los clientes y los departamentos emitan o reciban las comunicaciones con fiabilidad, prontitud y discreción.
- Programar las acciones de mantenimiento y reparación de las instalaciones, mobiliario y equipos de la zona de pisos y áreas públicas, y de los departamentos de lencería y lavandería, supervisando los resultados finales, con el fin de optimizar el servicio.
- Controlar y mantener la dotación de lencería, materiales, utensilios, equipos y mobiliario, necesaria para la prestación del servicio, de forma que se pueda conocer su cantidad, grado de rotación y conservación, obteniendo así el máximo provecho económico de los recursos disponibles.
- Organizar los recursos necesarios para la prestación del servicio de alojamiento en el departamento de pisos y áreas públicas, determinando una estructura eficiente que permita un proceso adecuado y rentable.
- Supervisar los procesos de limpieza y preparación de habitaciones y áreas comunes, asegurando un nivel óptimo de acabado.
- Supervisar los procesos de lavado y arreglo de lencería, ropa de clientes y personal, asegurando un nivel óptimo de acabado.
- Atender las peticiones y los reclamos planteados por los huéspedes, de modo que se les facilite el máximo nivel posible de satisfacción, confort y seguridad.
- Recoger y analizar información que sea útil para la definición del plan de marketing del establecimiento.
- Organizar, preparar y controlar acciones de publicidad o promoción en su departamento o área de responsabilidad.
- Definir, establecer y supervisar los procesos técnicos de aprovisionamiento y almacenamiento de alimentos y bebidas, elaborando la información técnica y dando las instrucciones necesarias para asegurar la calidad de los suministros.
- Definir o establecer los procesos de pre servicio, servicio y post servicio en el área de consumo de alimentos y bebidas, elaborando la información técnica o dando las instrucciones necesarias para asegurar la atención al cliente, conseguir los niveles de calidad necesarios y optimizar el proceso.

- Supervisar los procesos de pre servicio, servicio y post servicio en el área de consumo de alimentos y bebidas, de modo que se eviten y corrijan posibles desviaciones y se satisfagan las expectativas de los clientes.
- Prestar a los clientes el asesoramiento y la atención adecuados para satisfacer sus expectativas y cumplir con los objetivos económicos del establecimiento.
- Dirigir y coordinar al personal dependiente, promoviendo su motivación, participación y aceptación de los objetivos, el desarrollo de su profesionalidad y la mayor capacidad de respuesta a las necesidades de los clientes.

5. METODO OOHDM PARA EL DESARROLLO DE SISTEMAS

El Método de Diseño de Diseño de Hipermedias Orientados a Objetos (OOHDM) del ingles (*Object Oriented Hypermedia Design Methodology*), para diseño de aplicaciones hipermedia y para la Web, lo cual una extensión de HDM con orientación a objetos, que se ha convertido en una de las metodologías más utilizadas en la actualidad. Ha sido usada para diseñar diferentes tipos de aplicaciones hipermedia como galerías interactivas, presentaciones multimedia sobre todo en proyectos de aplicaciones web, diseñado por [D.Rossi, S. D. J. Barbosa].

Este método se inspira en el modelo HDM, pero lo que le distingue claramente del primero es el proceso de concepción orientado a objetos. OOHDM propone el desarrollo de aplicaciones hipermedia mediante un proceso de 5 fases.

- Obtención de requerimientos
- Diseño conceptual
- Diseño Navegacional
- Diseño de la interfaz abstracta
- Implementación

5.1 FASES DEL METODO OOHDM

FASE 1. Obtención de requerimientos

La obtención de requisitos o también denominado elicitación de requisitos, trata de obtener la mayor cantidad de información posible sobre el dominio de la aplicación, esta primera fase del método contempla sub-fases los cuales son:

- **Sub fase 1- Modelo de entrevistas**
- **Sub-fase 2-Identificación de roles (actores) y tareas;**
- **Sub-fase 3-Especificación de casos de uso;**

Sub fase1. La entrevista es una técnica muy utilizada ya que permite tomar conocimiento del problema y por ende las posibles soluciones. Para ello el entrevistador debe identificar muy bien a los entrevistados y estructurar muy bien las preguntas, una buena manera de reforzar la técnica de la entrevista, es combinándola con los cuestionarios y checklist, que son también otra técnica de captura de requerimientos.

Sub-fase 2. Identificación de roles (actores) y tareas Esta sub-fase tiene por objetivo identificar las necesidades reales de los usuarios, esto es posible a través de la interacción entre el desarrollador y el dominio de la aplicación. El desarrollador consigue esa interacción a través del análisis (o revisión) de documentos, y entrevistas con los usuarios a fin de identificar a los actores y sus tareas o necesidades [Henrichs, 2005]. Es importante identificar las diferentes clases de actores ya que ellos no participan en los mismos escenarios, estas clases deben ser confirmadas con los usuarios en la etapa siguiente, o sea durante la especificación de escenarios. Según la notación de OOHDM (Versión 2.0) [Vilain, 2002]:

- ⇒ Un actor es un agente que interactúa con el sistema, es decir un tipo de usuario con un papel definido, así el actor puede representar diversos papeles y un papel puede ser representado por varios actores. Por tanto un actor o un rol es una clase de usuario.
- ⇒ Una tarea representa un objetivo que el usuario desea alcanzar utilizando la aplicación.

Sub fase 3. Un caso de uso es una forma de utilizar la aplicación. Específicamente representa la interacción entre el usuario y el sistema, agrupando las tareas representadas en los escenarios existentes. En el cual se identificara la información relevante en cada uno de ellos, para luego generar un caso de uso coherente.

FASE 2. Diseño conceptual

En esta etapa se diseña el modelo conceptual, donde las clases, relaciones y cardinalidades se definen de acuerdo a reglas que se aplican sobre los UIDs. Cabe destacar que gran parte de ellas provienen de las técnicas de normalización.

FASE 3. Diseño navegacional

En esta etapa de la metodología se pretende desarrollar una topología navegacional que permita a la aplicación ejecutar todas las tareas requeridas por el usuario. La idea principal es unificar una serie de tareas para obtener el diseño navegacional de la aplicación.

En esta fase la metodología esta subdividida en las siguientes etapas: Esquema de clases navegacionales, esquema de contextos navegacionales [Henrichs, 2005].

Sub fase 1. Esquema de clases navegacionales

El esquema de clases navegacionales, refleja una posible vista elegida [Escalona, 2001]. El esquema de clases navegacionales o esquema navegacional define un conjunto de nodos y enlaces que forman parte de una vista navegacional de la aplicación. Una aplicación puede tener uno o más esquemas navegacionales de acuerdo con las vistas existentes de la aplicación. Los nodos y enlaces son originados de las clases y relaciones que componen el esquema conceptual [Henrichs, 2005].

Sub-fase 2. Esquema de contexto navegacional

El esquema de contexto navegacional presenta las diferentes maneras en la que un nodo puede ser accedido, según el esquema de clases navegacionales, el contexto

navegacional es la estructura de la presentación dentro de un determinado contexto [Escalona, 2001].

FASE 4. Diseño de interfaz abstracta

Una vez finalizado el diseño navegacional, será necesario especificar las diferentes interfaces de la aplicación. Esto significa definir, de qué manera aparecerán los objetos navegacionales en la interfaz y cuales objetos activarán la navegación. Para lograr esto se utilizarán ADVs(Vista de Datos Abstracta), modelos abstractos que especifican la organización y el comportamiento de la interfaz, es necesario aclarar que las ADVs representan estados o interfaces y no la implementación propiamente.

FASE 5. Implementación

Una vez terminadas las etapas anteriores, se tendrá el conocimiento del dominio del problema. Así entonces, ya se identificará la información que será mostrada, como estará organizada y cuales funciones permitirá ejecutar la aplicación. Además de ello, se tendrá la idea clara de cómo se quiere que se vean las interfaces.

Bajo este contexto en esta etapa se diseñaran las clases bajo el lenguaje de PHP5 y las interfaces serán desarrollados con la tecnología de eSmarty y Ajax, al mismo tiempo se diseñará la base de datos con la herramienta de MySQL.

6. ELEMENTOS DE IMPLEMENTACION

6.1 PATRON DE DISEÑO MODELO VISTA CONTROLADOR (MVC)

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página; el modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio; y el controlador es el responsable de recibir los eventos de entrada desde la vista.

- **Modelo:** Esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite todo derivar nuevos datos; por ejemplo, no permitiendo comprar un número de unidades negativo, calculando si hoy es el cumpleaños del usuario o los totales, impuestos o importes en un carrito de la compra.
- **Vista:** Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- **Controlador:** Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

6.2 SMARTY

Smarty es un motor de plantillas para PHP. Más específicamente, esta herramienta facilita la manera de separar la aplicación lógica y el contenido en la presentación. La mejor descripción esta en una situación donde la aplicación del programador y la plantilla del diseñador juegan diferentes roles, o en la mayoría de los casos no la misma persona.

Smarty no intenta separar completamente la lógica de la plantilla. No hay problema entre la lógica y su plantilla bajo la condición que esta lógica sea estrictamente para presentación. Un consejo: mantener la aplicación lógica fuera de la plantilla, y la presentación fuera de la aplicación lógica. Esto tiene como finalidad tener un objeto mas manipulable y escalable para un futuro próximo.

La funcionalidad de Smarty es la compilación de la plantilla. De esta manera Smarty lee la plantilla y crea los scripts de PHP. Una vez creados, son ejecutados sobre él. Por consiguiente no existe ningún costo por analizar gramaticalmente cada archivo de template por cada requisición, y cada template puede llevar toda la ventaja del compilador de cache de PHP tal como ZendAccelerator o PHP Accelerator.

Algunas de las características de Smarty:

1. Es extremadamente rápido.
2. Es eficiente ya que puede interpretar el trabajo más sucio.
3. No analiza gramaticalmente desde arriba el template, solo compila una vez.

4. El está atento para solo recompilar los archivos de plantilla que fueron cambiados.
5. Usted puede crear funciones habituales y modificadores de variables customizados, de modo que el lenguaje de la platilla es altamente extensible.
6. Sintaxis de etiquetas delimitadoras para configuración de la plantilla, así lo puede usar {}, {{}}, <!--{}-->, etc.
7. Los constructoresif/elseif/else/endif son pasados por el interpretador de PHP, así la sintaxis de la expresión {if ...} puede ser compleja o simple de la forma que usted quiera.
8. Permite un anidamiento ilimitado de sections, ifs, etc.
9. Es posible incrustar directamente código PHP en los archivos de plantilla, aunque esto puede no ser necesario(no recomendado) dado que la herramienta se puede ajustar.
10. Soporte de caching incrustado
11. Fuentes de Plantilla absoluto
12. Funciones habituales de manipulación de cache
13. Arquitectura de Plugin

6.3 BASE DE DATOS

Según [Navarrete, 1999], “Base de Datos es una colección de información organizada, y relacionada entre ella”. Esta, pueda estar constituido de gran cantidad de información, con posibilidad de acceder en cualquier momento y de manera concurrente; este acceso generalmente se realiza a través de instrucciones denominado consultas.

Una Base de Datos cuenta con las siguientes propiedades:

- a) Representación de algún aspecto del mundo real.
- b) Es un conjunto de datos lógicamente coherente e interrelacionado.
- c) Toda Base de Datos se diseña con un propósito específico.

6.4 MySQL

Es un sistema de gestión de bases de datos relacional, fue creada por la empresa sueca MySQL AB, la cual tiene el copyright del código fuente del servidor SQL, así como también de la marca.

MySQL es un software de código abierto, licenciado bajo la GPL de la GNU, aunque MySQL AB distribuye una versión comercial, en lo único que se diferencia de la versión libre, es en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de otra manera, se vulneraría la licencia GPL.

El lenguaje de programación que utiliza MySQL es Structured Query Language (SQL) que fue desarrollado por IBM en 1981 y desde entonces es utilizado de forma generalizada en las bases de datos relacionales.

MySQL surgió alrededor de la década del 90, Michael Widenis comenzó a usar mSQL para conectar tablas usando sus propias rutinas de bajo nivel (ISAM). Tras unas primeras pruebas, llegó a la conclusión de que mSQL no era lo bastante flexible ni rápido para lo que necesitaba, por lo que tuvo que desarrollar nuevas funciones. Esto resultó en una interfaz SQL a su base de datos, totalmente compatible a mSQL.

6.4.1 Características principales

Inicialmente, MySQL carecía de algunos elementos esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de esto, atrajo a los desarrolladores de páginas web con contenido dinámico, debido a su simplicidad, de tal manera que los elementos faltantes fueron complementados por la vía de las aplicaciones que la utilizan. Poco a poco estos elementos faltantes, están siendo incorporados tanto por desarrolladores internos, como por desarrolladores de software libre. En las últimas versiones se pueden destacar las siguientes características principales:

- El principal objetivo de MySQL es velocidad y robustez.
- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos.
- Cada base de datos cuenta con 3 archivos: Uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla.
- Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo.
- Flexible sistema de contraseñas (passwords) y gestión de usuarios, con un muy buen nivel de seguridad en los datos.
- El servidor soporta mensajes de error en distintas lenguas

6.4.2 Ventajas

- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación.
- Soporta gran variedad de Sistemas Operativos
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Conectividad y seguridad

6.5 ARQUITECTURA CLIENTE SERVIDOR

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

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

Por el contrario, el servidor es la máquina desde la que se suministran servicios y que está a la espera del requerimiento del cliente. Una vez hecho, busca la información solicitada y le envía la respuesta al cliente; incluso puede enviar varios servicios a la vez, lo que es posible porque entre ellos están conectados mediante redes LAN o WAN.

Entre las características fundamentales de esta arquitectura encontramos que tanto el cliente como el servidor pueden realizar tareas en forma conjunta como separada ya que el cliente también tiene sus propias aplicaciones, archivos y bases de datos y que además, pueden estar en la misma plataforma o en plataformas diferentes. Por otra parte, el servidor

puede brindar varios servicios a la vez, tanto al mismo cliente como a clientes múltiples define [Lanzillotta 2005].

6.6 AJAX

AJAX, Asíncronos de Java Script y XML, se conoce como una conjunción de tres tecnologías para el desarrollo de aplicaciones webs interactivas, en tal sentido AJAX engloba a un grupo de tecnologías que trabajan conjuntamente.

Las aplicaciones AJAX usan navegadores web que soportan las tecnologías mencionadas más arriba. Entre estos se incluyen Mozilla, Firefox, Internet Explorer, Opera, Konqueror y Safari, entre otros

6.6.1 Tecnologías que engloba AJAX:

JavaScript: es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en aplicaciones Web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C.

XML: es un metalenguaje extensible de etiquetas desarrollado por el **World Wide Web Consortium (W3C)**. Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML). Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.

HTML: Es un lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web.

CSS: son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML).

Todas las anteriores soportadas por algunos Browsers (Navegadores Web) como Mozilla Firefox, Internet Explorer, entre otros.

7. APLICACIONES WEB

Son aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. InterfazLas interfaces web tienen ciertas limitaciones en las funcionalidades que se ofrecen al usuario. Hay funcionalidades comunes en las aplicaciones de escritorio como dibujar en la pantalla o arrastrar-y-soltar que no están soportadas por las tecnologías web estándar. Los desarrolladores web generalmente utilizan lenguajes interpretados o script en el lado del cliente para añadir más funcionalidades, especialmente para ofrecer una experiencia interactiva que no requiera recargar la página cada vez (lo que suele resultar molesto a los usuarios).

8. PARADIGMA DE PROGRAMACION ORIENTADO A OBJETOS

La P.O.O. (también conocida como O.O.P., por sus siglas en inglés) es lo que se conoce como un paradigma o modelo de programación. Esto significa que no es un lenguaje específico, o una tecnología, sino una forma de programar, una manera de plantearse la programación. No es la única (o necesariamente mejor o peor que otras), pero se ha constituido en una de las formas de programar más populares en la actualidad.

Lo que caracteriza a la POO es que intenta llevar al mundo del código lo mismo que encontramos en **El Mundo Real** Cuando miramos a nuestro alrededor ¿qué vemos? pues, cosas, **objetos**, pero podemos reconocer estos objetos porque **cada objeto pertenece a una clase**, lo cual permite distinguir, por ejemplo, un perro de un auto (porque son de clases diferentes) y también un TV de otro (porque, aunque sean iguales, cada uno es un objeto distinto). Éste es el modelo que la POO intenta seguir para estructurar un sistema.

En conclusión la POO **no es un lenguaje de programación**, es una forma de enfrentarse a ella. Esto significa que la POO le servirá para desarrollar en muchos de los lenguajes comunes utilizadas en la actualidad.

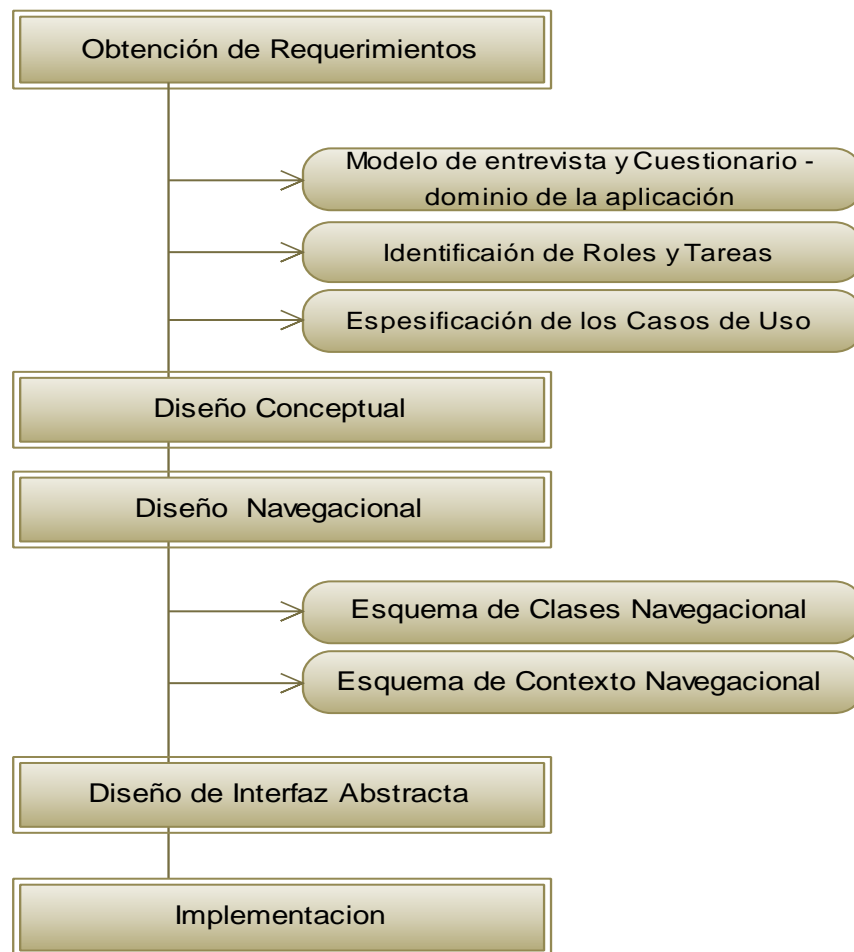
9. PLATAFORMA DE DESARROLLO LINUX

GNU/Linux es un sistema operativo, una plataforma para el desarrollo de aplicaciones aceptada mundialmente por usuarios, instituciones y gobiernos para la cual hay disponibles multitud de lenguajes, bibliotecas y herramientas que, en su mayor parte, siguen la filosofía de código de uso libre. Linux es posiblemente el sistema operativo para el que existe un mayor número de lenguajes, compiladores, bibliotecas y otras herramientas de desarrollo disponibles, lo cual permite a los programadores elegir lo que más se adecúa a los conocimientos y necesidades de cada proyecto en particular.

CAPITULO III DESARROLLO DEL SISTEMA

El presente capítulo describe las fases del desarrollo del Sistema Informático en sus componentes de software, procedimientos y datos, a través de la metodología OOHD, además, de manera que inicialmente se ha procedido a la identificación de los roles y tareas de los usuarios que intervienen en el dominio.

El método OOHD propone cinco actividades para la construcción de una aplicación, en tal sentido a continuación se presentan las fases y sub fases que se han tomado en cuenta para el desarrollo del Sistema de Gestión Hotelera.



*Figura 2: Fases de Método OOHD
Fuente: Elaboración Propia*

1. FASE DE INICIO

1.1 OBTENCION DE REQUERIMIENTOS

La obtención de requisitos, trata de obtener la mayor cantidad de información posible sobre el dominio de la aplicación, para esta primera fase del método se ha considerado tres sub-fases para su efecto, los cuales son:

- **Sub-fase 1-**Modelo de entrevistas y cuestionarios
- **Sub-fase 2-**Identificación de roles y tareas
- **Sub-fase 3-**Especificación de casos de uso

En tal sentido se ha procedido a la ejecución de las entrevistas así mismo los cuestionarios para la identificación de roles y tareas de cada uno de los usuarios quienes forman parte del dominio.

1.1.1 Modelo de Entrevista y Cuestionario – dominio de la aplicación

La entrevista es una técnica muy utilizada en los trabajos de investigación ya que permite tomar conocimiento del problema y por ende las posibles soluciones. Para ello se ha identificado objetivamente a los entrevistados del dominio y se ha estructurado muy bien las preguntas del mismo.

PERFIL DE ENTREVISTADO			
Identificación:			
<i>Cliente</i>			
Descripción:			
<i>Es la persona que esta hospedado en la entidad hotelera por un tiempo determinado</i>			
Calidad de entrevistado:			
Muy accesible	<input checked="" type="checkbox"/>	Poco accesible <input type="checkbox"/>	
Grado de conocimiento del tema:			
Mínima	<input type="checkbox"/>	Media <input type="checkbox"/>	Alta <input checked="" type="checkbox"/>

*Figura 3: Plantillas de entrevista al cliente
Fuente: Elaboración propia*

PERFIL DE ENTREVISTADO			
Identificación:			
<i>Gerente Administrador de la Entidad</i>			
Descripción:			
<i>El administrador general de la entidad hotelera, es la persona que posee la información sobre el funcionamiento de la entidad las reglas y procedimientos sobre el manejo de la información.</i>			
Calidad de entrevistado:			
Muy accesible	<input type="checkbox"/>	Poco accesible <input checked="" type="checkbox"/>	
Grado de conocimiento del tema:			
Mínima	<input type="checkbox"/>	Media <input type="checkbox"/>	Alta <input checked="" type="checkbox"/>

*Figura 4: Plantillas de entrevista al gerente administrador
Fuente: Elaboración propia*

PERFIL DE ENTREVISTADO		
Identificación:		
Recepcionista		
Descripción:		
El recepcionista, representa a un grupo de funcionarios que cumplen la función de decepcionar a los clientes desde su llegada al hotel hasta cuando se marcha.		
Calidad de entrevistado:		
Muy accesible <input type="checkbox"/>	Poco accesible <input checked="" type="checkbox"/>	
Grado de conocimiento del tema:		
Mínima <input type="checkbox"/>	Media <input type="checkbox"/>	Alta <input checked="" type="checkbox"/>

*Figura 5: Plantillas de entrevista al recepcionista
Fuente: Elaboración propia*

PERFIL DE ENTREVISTADO		
Identificación:		
Encargado de Ventas		
Descripción:		
Es el encargado de registrar la venta de productos asimismo de los consumos por parte de los clientes.		
Calidad de entrevistado:		
Muy accesible <input type="checkbox"/>	Poco accesible <input checked="" type="checkbox"/>	
Grado de conocimiento del tema:		
Mínima <input type="checkbox"/>	Media <input type="checkbox"/>	Alta <input checked="" type="checkbox"/>

*Figura 6: Plantillas de entrevista encargado de ventas
Fuente: Elaboración propia*

1.1.2 Especificación de roles y tareas

En esta sub fase se ha realizado la clasificación de los actores, lo cual ha emitido situarse mejor en el dominio de la aplicación, y tener con claridad con que tipo de usuarios se está trabajando, en tal sentido en primera instancia se ha procedido a la clasificación de los tipos de actores que intervienen en el manejo del sistema, lo cual implica identificar a las clases y sub-clases de actores, tal como se muestra en la siguiente figura.

1.1.2.1 Clasificación de Roles

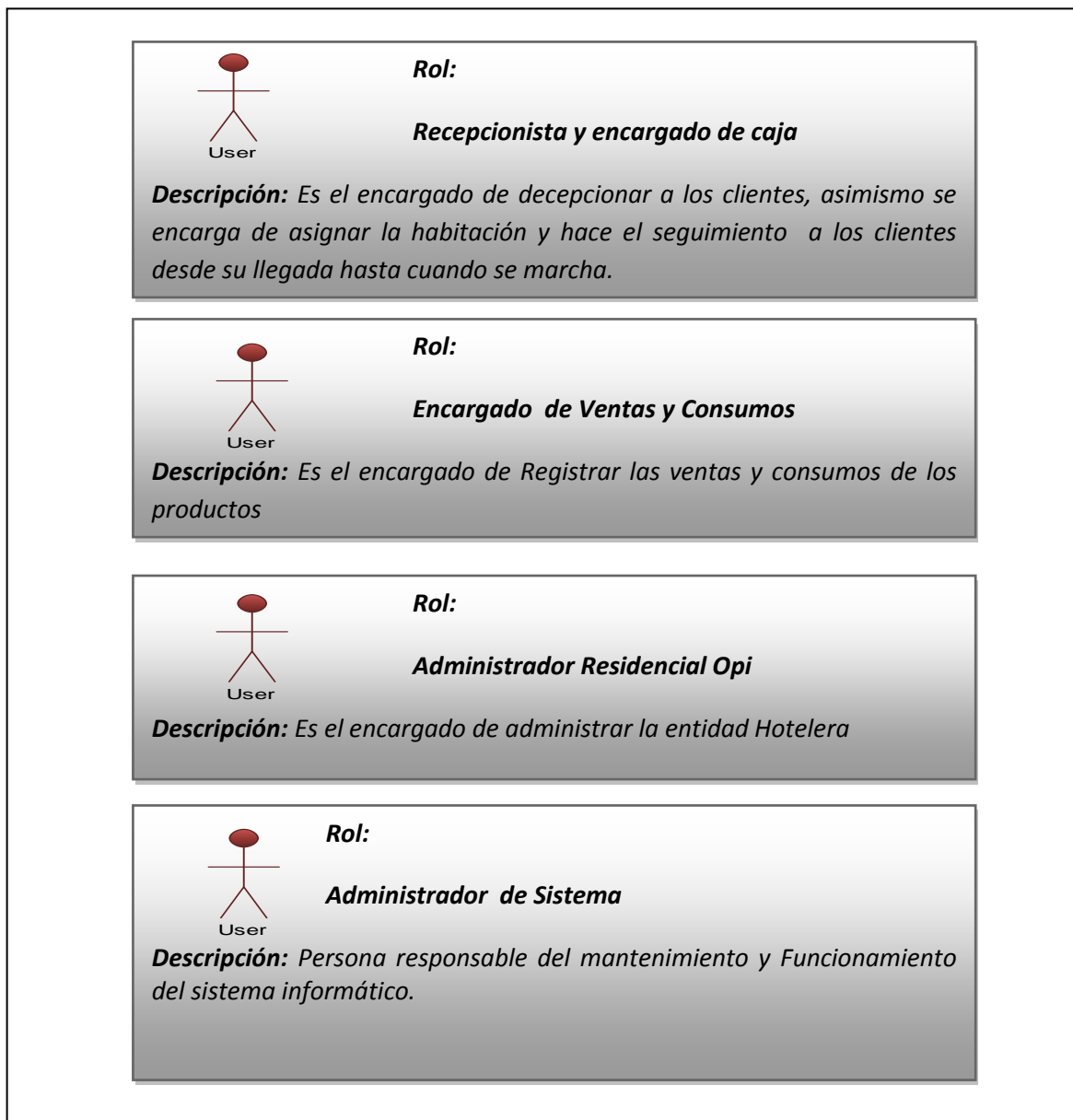


Figura 7: Esquema clasificación de roles
Fuente: Elaboración propia

1.1.2.2 Definición de Roles y Tareas

Definiciones de Tareas	
Rol	Tareas
Rol/Actor Administrador de la Residencial	<i>T1- Alta a persona</i> <i>T2- Baja a persona</i> <i>T3- Modificar datos persona</i> <i>T4- Alta Clientes</i> <i>T5- Baja Clientes</i> <i>T6- Modificar Datos del cliente</i> <i>T7- Alta habitaciones</i> <i>T8- Baja habitaciones</i> <i>T9- Modificar datos de las habitaciones</i> <i>T10- Consultas sobre la estadía de clientes</i> <i>T11- Consultas Sobre la Cuenta del cliente</i> <i>T12- Consulta diaria de los clientes</i> <i>T13- Reporte Global de clientes hospedados</i> <i>T14- Realizar la Copia de seguridad de la base de datos del sistema</i>
Rol/Actor Recepcionista y encargado de caja	<i>T15- Dar alta a personas</i> <i>T16- Dar baja personas</i> <i>T17- Modificar datos personas</i> <i>T18- Dar de alta al clientes</i> <i>T19- Dar de Baja al cliente</i> <i>T20- Modificar datos del cliente</i> <i>T21- Asignar habitación al cliente</i> <i>T22- Realizar el Cobro de dinero en efectivo a los clientes sobre su estadía</i> <i>T23- Revisara estado de la cuenta del cliente.</i> <i>T24- Realizar la copia de seguridad de la base de datos del sistema</i>

	<p><i>T25- Realizar el cierre de habitación (cuando el cliente se marcha).</i></p> <p><i>T26- Extender factura por el tiempo de permanencia del cliente.</i></p>
<p>Rol/Actor</p> <p>Encargado de Ventas y consumos</p>	<p><i>T27- Dar de alta a los productos en el sistema</i></p> <p><i>T28- Registrar la venta de productos</i></p> <p><i>T29- Dar de baja a los productos</i></p> <p><i>T30- Modificar datos del producto</i></p> <p><i>T31- Registra consumos diarios de los clientes</i></p> <p><i>T32- Registrar Productos vendidos</i></p> <p><i>T33- Realizar Copia de seguridad de la base de datos del sistema</i></p>
<p>Rol/Actor</p> <p>Administrador de Sistemas</p>	<p><i>T34- Crear nuevos usuarios</i></p> <p><i>T35- Crear nuevos roles en el sistema</i></p> <p><i>T36- Asignar Roles al usuario</i></p> <p><i>T37- Administrar opciones enlaces</i></p> <p><i>T38- Administrar grupo de enlaces</i></p> <p><i>T39- Realizar copia de seguridad de la base de datos del sistema</i></p>

Tablas 1: Definición de roles y tareas
Fuente: Elaboración propia

1.1.3 Especificación de casos de uso

En esta sub fase se ha identificado los casos de uso del sistema los cuales son presentados a través de plantillas, los cuales muestran la división de las peticiones o acciones que realiza el usuario y las respuestas que da el Sistema. Además permite especificar las precondiciones y post condiciones que deben cumplirse para que el curso normal de acciones se dé.

Se presentan las especificaciones de los casos de uso del dominio en estudio:

Caso de uso	<i>Registro de Personas</i>	
Roles	<i>Recepcionista y encargado de caja</i>	
Tareas	<i>T15, T16, T7</i>	
Precondiciones	<i>El usuario debe estar registrado en la base de datos</i>	
Descripción		
Peticiones	Respuesta	
<p>1- El usuario ingresa nombre de usuario y contraseña.</p> <p>6- El usuario ingresa los datos de la personas</p> <p>9- Modificar datos de la personas registrados</p> <p>10- Eliminar Persona de la base de datos.</p>	<p>2- El sistema verifica el nombre de usuario y contraseña.</p> <p>3- Si el nombre de usuario y contraseña no son válidos, el sistema informa que los datos son inválidos y vuelve a pedir el nombre de usuario y contraseña.</p> <p>4- Si la contraseña y el nombre de usuario son válidos, el sistema muestra un enlace para el registro personas.</p> <p>5- El sistema muestra el formulario para el registro de personas.</p> <p>7- Si los datos ingresados son inválidos el sistema reporta error de registro</p> <p>8- Si los datos son validos entonces son registrados correctamente en la base de datos del sistema.</p>	
Pos condiciones		

Figura 8: Esquema caso de uso registro de personas en el sistema

Fuente: Elaboración propia

Caso de uso	<i>Registrar clientes en el sistema</i>	
Roles	<i>Recepcionista y encargado de caja</i>	
Tareas	<i>T18, T19, T20, T24</i>	
Precondiciones	<i>el recepcionista debe estar registrado como usuario en el sistema</i>	
Descripción		
Peticiones	Respuesta	
<p><i>1- El usuario ingresa nombre de usuario y contraseña para el acceso al sistema</i></p> <p><i>7- El usuario elige la opción de registrar nuevo cliente</i></p> <p><i>10- El usuario ingresa los datos del cliente</i></p> <p><i>11- Modificar datos del cliente</i></p> <p><i>12- Dar de baja al cliente</i></p> <p><i>13- Realizar copia de Seguridad de la base de datos</i></p> <p><i>14- Revisar estado de la cuenta del cliente</i></p>	<p><i>2- El sistema verifica el nombre de usuario y la contraseña</i></p> <p><i>3- Si el nombre de usuario y contraseña no son validos, el sistema informa que los datos son inválidos y vuelve a pedir usuario y contraseña.</i></p> <p><i>4- Si usuario y contraseña son validos el sistema muestra opción para registrar nuevo cliente.</i></p> <p><i>5- El sistema pide datos personales y documentos que identifiquen al cliente.</i></p> <p><i>6- El sistema verifica si los datos ingresados son correctos.</i></p> <p><i>8- Si los datos son inválidos el sistema pide la verificación de los datos ingresado.</i></p> <p><i>9- Si los datos son correctos, el sistema almacena en la base de datos y muestra mensaje de que los datos han sido registrado.</i></p>	
Pos condiciones		

Figura 9: Esquema caso de uso registro de clientes en el sistema
Fuente: Elaboración propia

Caso de uso	<i>Asignación de habitación</i>	
Roles	<i>Recepcionista y encargado de caja</i>	
Tareas	<i>T21, T23</i>	
Precondiciones	<i>El usuario debe estar registrado en la base de datos</i>	
Descripción		
Peticiones	Respuesta	
<p>1- El usuario ingresa nombre de usuario y contraseña.</p> <p>6- El usuario elige la opción de asignación de habitación.</p> <p>7- El usuario asigna habitación al cliente para su estadía.</p>	<p>2- El sistema verifica el nombre de usuario y contraseña.</p> <p>3- Si el nombre de usuario y contraseña no son válidos, el sistema informa que los datos son inválidos y vuelve a pedir el nombre de usuario y contraseña.</p> <p>4- Si la contraseña y el nombre de usuario son válidos, el sistema muestra una opción para asignar habitación al cliente.</p> <p>5- El sistema muestra las habitaciones disponibles para su asignación al cliente</p>	
Pos condiciones		

*Figura 10: Esquema caso de uso de asignación de habitación
Fuente: Elaboración propia*

Caso de uso	<i>Registro de consumos</i>	
Roles	<i>Encargado de ventas y consumos</i>	
Tareas	<i>T31,</i>	
Precondiciones	<i>Usuario debe estar registrado en el sistema</i>	
Descripción		
Peticiones	Respuesta	
<p><i>1- El usuario ingresa nombre de usuario y contraseña para el acceso al sistema, en caso de que no haya identificado aun</i></p> <p><i>5- El usuario elige la opción para registro de consumos y el tipo de producto</i></p> <p><i>7- El usuario ingresa el producto para a respectiva búsqueda en la base de datos</i></p>	<p><i>2- El sistema verifica el nombre de usuario y contraseña.</i></p> <p><i>3- Si el nombre de usuario y contraseña no son validos, el sistema informa que los datos son inválidos y vuelve a pedir usuario contraseña.</i></p> <p><i>4- Si usuario y contraseña son validos el sistema muestra la opción para el registro de consumos.</i></p> <p><i>6 El sistema muestra la opción de búsqueda del producto que ha solicitado el cliente</i></p> <p><i>8- Si el producto no existe el sistema informa que el producto especificado no es válido y vuelve a pedir el producto a buscar.</i></p> <p><i>9- En caso de que los datos sean correctos el sistema informa que los datos han sido guardados en la base de datos correctamente.</i></p>	
Pos condiciones		

Figura 11: Esquema caso de uso registro de consumos

Fuente: Elaboración propia

Caso de uso	<i>Registro de productos</i>	
Roles	<i>Encargado de ventas</i>	
Tareas	<i>T23</i>	
Precondiciones	<i>Usuario debe estar registrado en el sistema</i>	
Descripción		
Peticiones	Respuesta	
<p>1- El usuario ingresa nombre de usuario y contraseña para el acceso al sistema</p> <p>5- El usuario elige la opción para registro de producto.</p> <p>7- El usuario ingresa datos del producto</p>	<p>2- El sistema verifica el nombre de usuario y contraseña.</p> <p>3- Si el nombre de usuario y contraseña no son validos, el sistema informa que los datos son inválidos y vuelve a pedir usuario contraseña.</p> <p>4- Si usuario y contraseña son validos el sistema muestra opción para registro de productos.</p> <p>6- El sistema muestra un formulario para el ingreso de datos del producto a registrar.</p> <p>8- Si los campos no han sido llenados correctamente el sistema informa que los campos no han sido llenados correctamente.</p> <p>9- En caso de que los datos sean correctos el sistema informa que los datos han sido guardados en la base de datos correctamente.</p>	
Pos condiciones		

Figura 12: Esquema caso de uso registro de productos

Fuente: Elaboración propia

Caso de uso	Venta de productos	
Roles	Encargado de ventas y consumos	
Tareas	T28, T30	
Precondiciones	Usuario debe estar registrado en el sistema	
Descripción		
Peticiones	Respuesta	
<p>1- El usuario ingresa nombre de usuario y contraseña para el acceso al sistema</p> <p>5- El usuario elige la opción para registrar venta de producto</p> <p>7- El usuario ingresa datos del producto</p> <p>10- Registrar producto</p> <p>11- Modificar datos del producto</p>	<p>2- El sistema verifica el nombre de usuario y contraseña.</p> <p>3- Si el nombre de usuario y contraseña no son validos, el sistema informa que los datos son inválidos y vuelve a pedir usuario contraseña.</p> <p>4- Si usuario y contraseña son validos el sistema muestra opción para registro de productos.</p> <p>6- El sistema muestra un formulario para el ingreso de datos del producto a vender.</p> <p>8- Si los campos no han sido llenados correctamente el sistema informa que los campos no han sido llenados correctamente.</p> <p>9- En caso de que los datos sean correctos el sistema informa que los datos han sido guardados en la base de datos correctamente.</p>	
Pos condiciones		

*Figura 13: Esquema caso de uso registro de productos
Fuente: Elaboración propia*

Caso de uso	<i>Pago en caja</i>	
Roles	<i>Recepcionista y encargado de caja</i>	
Tareas	<i>T22, T23</i>	
Precondiciones	<i>Usuario debe estar registrado en el sistema</i>	
Descripción		
Peticiones	Respuesta	
<p><i>1- El cliente se acerca a caja y consulta al encargado de la caja sobre el monto que debe por el tiempo de estadías</i></p> <p><i>4- El recepcionista ingresa nombre de usuario y contraseña para el acceso al sistema.</i></p> <p><i>6- El usuario elige la opción de caja</i></p> <p><i>8- El usuario registra el pago realizado por el cliente y en caso de ser necesario la devolución si tiene sobrante el cliente.</i></p> <p><i>9- El usuario emite el comprobante de pago al cliente</i></p>	<p><i>2- El sistema verifica el nombre de usuario y contraseña.</i></p> <p><i>3- Si el nombre de usuario y contraseña no son validos, el sistema informa que los datos son inválidos y vuelve a pedir usuario y contraseña.</i></p> <p><i>5- Si usuario y contraseña son validos el sistema muestra opción de caja y facturación</i></p> <p><i>7- El sistema reporta el monto total que debe cancelar el cliente por los días de permanencia.</i></p>	
Pos condiciones		

Figura 14: Esquema caso de uso registro de pagos
Fuente: Elaboración propia

2. FASE DE DISEÑO

En esta fase del trabajo se ha procedido con el diseño del sistema en el cual se establecen con claridad cada uno de las fases que contempla la metodología empleada, en tal sentido se cumplen con las tareas definidas para esta etapa.

2.1 DISEÑO CONCEPTUAL

En esta etapa se ha diseñado el modelo orientado a objetos que represente el dominio semántico de la aplicación, usando las técnicas propias de la orientación a objetos. Esta construcción es realizada mediante la definición de clases, subsistemas, relaciones, jerarquías de agregación y especialización, la definición de los tipos de atributos de las clases y cardinalidades [Henrichs, 2005], a continuación se presenta el trabajo realizado.

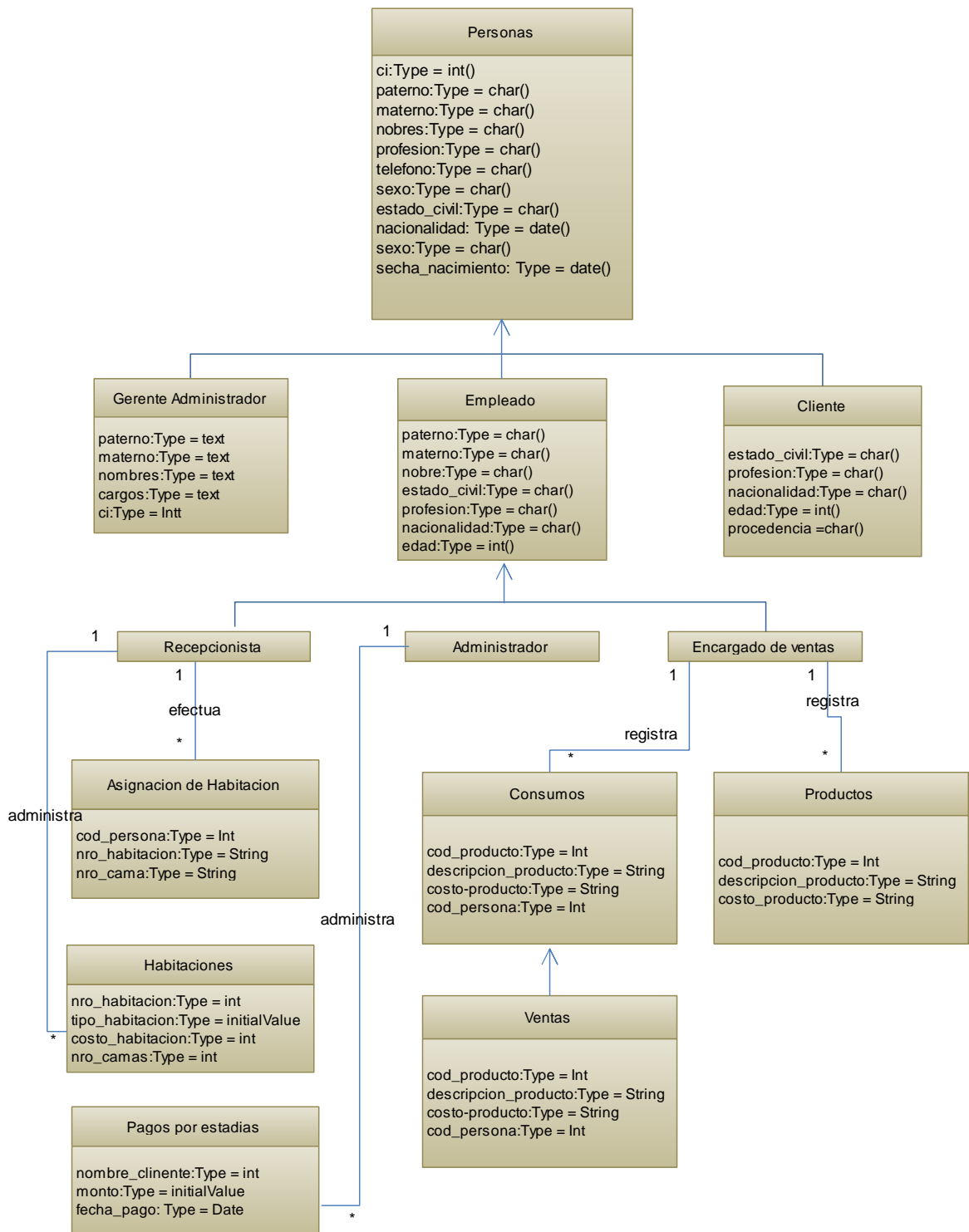


Figura 15: Diseño del Modelo conceptual
 Fuente: Elaboración propia

2.2 DISEÑO NAVEGACIONAL

Todo sistema basado en web son aplicaciones proyectadas para la navegación a través de un espacio de información. La fase anterior (modelo conceptual), propone la construcción de un modelo conceptual en el cual describe las clases de objetos y relaciones del dominio de la aplicación. De este modelo conceptual se pueden originar varios modelos navegacionales es decir la información que será presentada a los usuarios y como se efectuará la navegación entre ellas [Henrichs, 2005].

En esta fase la metodología esta subdividida en dos etapas:

2.2.1 Esquema de clases navegacionales

El esquema de clases navegacionales, refleja una posible vista elegida [Escalona, 2001]. El esquema de clases navegacionales o esquema navegacional define un conjunto de nodos y enlaces que forman parte de una vista navegacional de la aplicación. Una aplicación puede tener uno o más esquemas navegacionales de acuerdo con las vistas existentes de la aplicación. Los nodos y enlaces son originados de las clases y relaciones que componen el esquema conceptual [Henrichs, 2005].

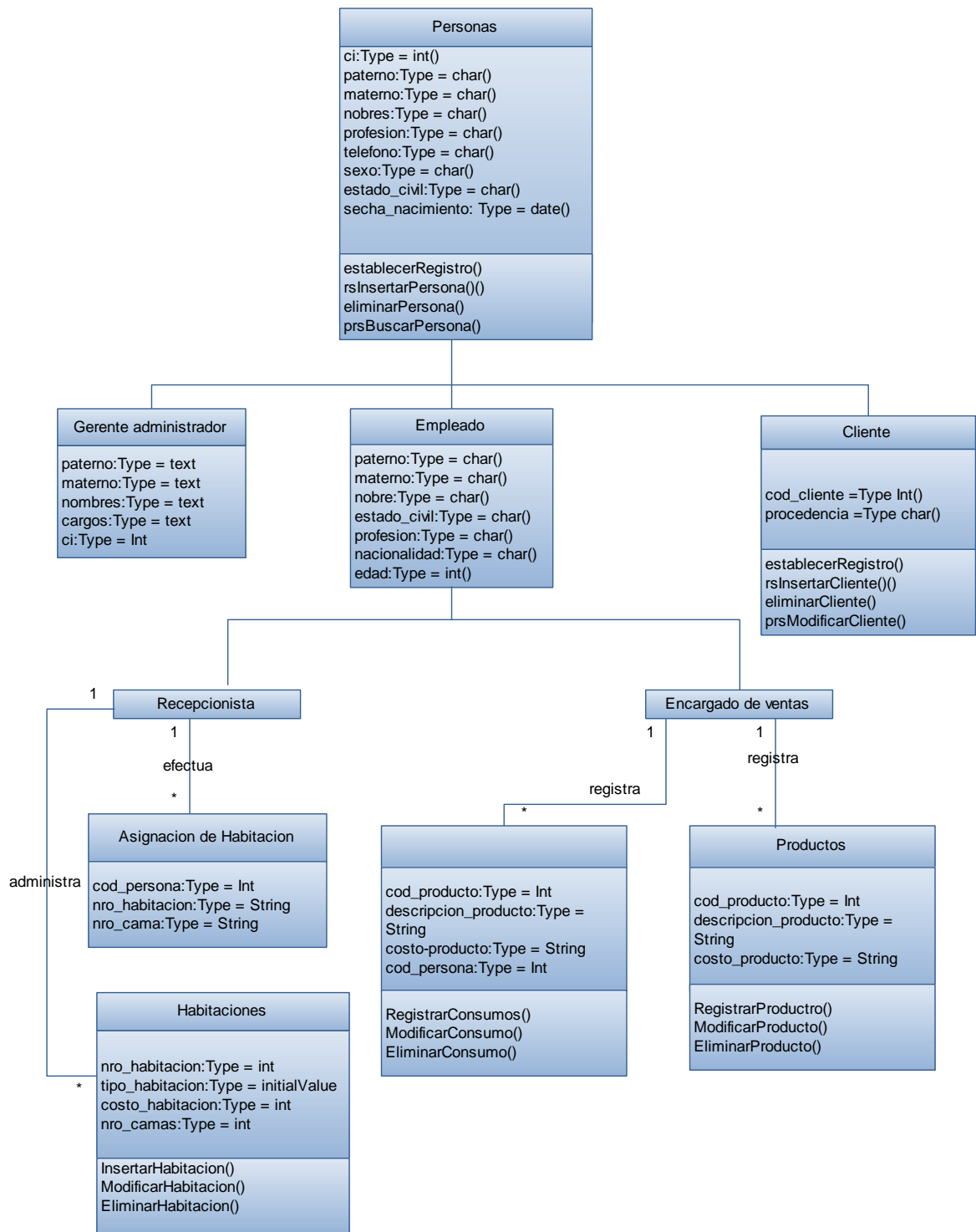


Figura 16: Esquema de clases navegacionales
Fuente: Elaboración propia

2.2.2 Esquema de contextos navegacionales

El esquema de contexto navegacional presenta las diferentes maneras en la que un nodo puede ser accedido, según el esquema de clases navegacionales, el contexto navegacional es la estructura de la presentación dentro de un determinado contexto [Escalona, 2001].

En tal sentido se presenta de manera resumida el esquema del contexto navegacional del sistema.

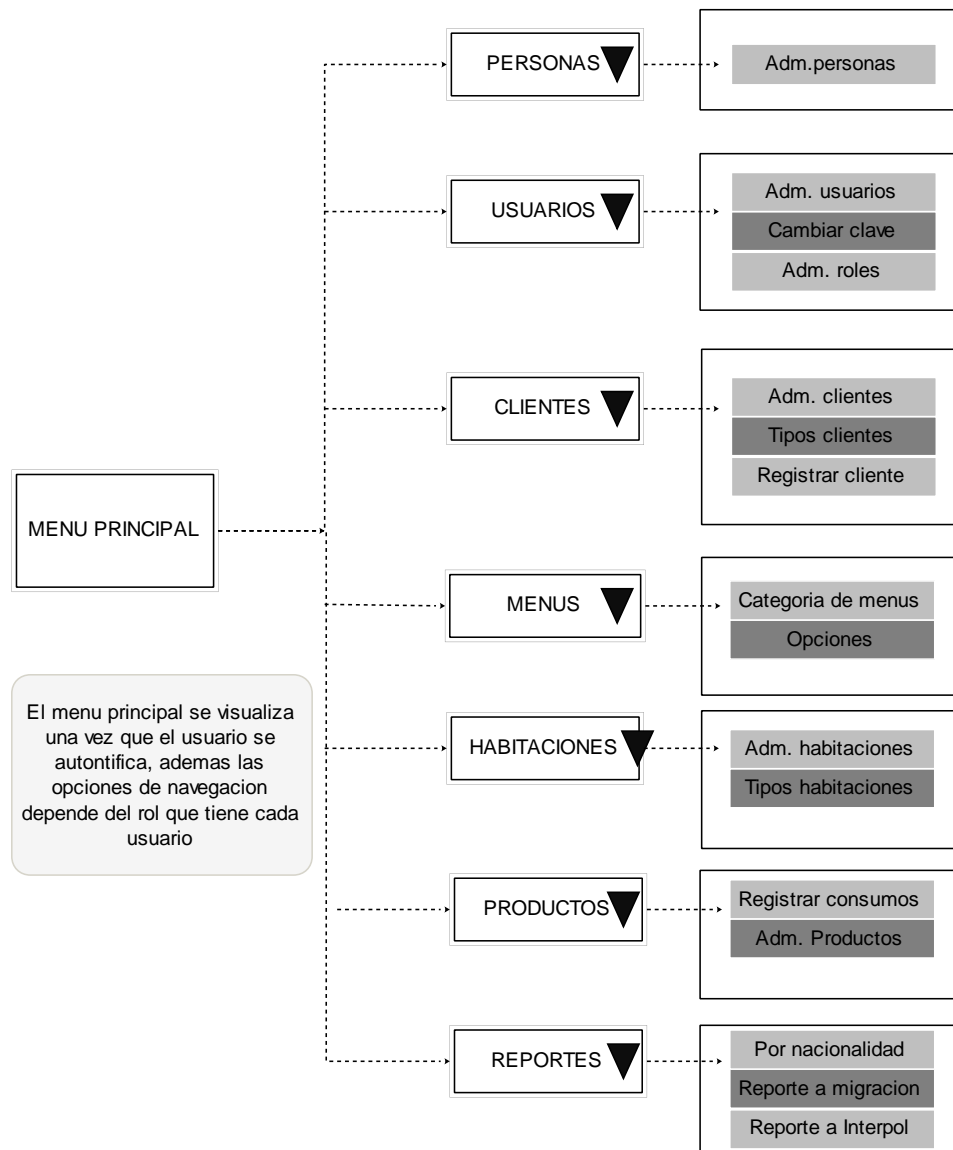


Figura 17: Esquema de contextos navegacionales
Fuente: Elaboración propia

2.3 DISEÑO DE INTERFAZ ABSTRACTA DEL SISTEMA

Una vez definida la estructura navegacional, se ha preparado para que sea perceptible por el usuario y esto es lo se pretende en esta fase. En tal sentido se ha definido los objetos de interfaz que va a percibir el usuario, y en particular el camino en el cuál aparecen los diferentes objetos de navegación, además los objetos de interfaz actúan en la navegación, la forma de sincronización de los objetos del dominio y el interfaz de transformaciones. Al haber una clara separación entre la fase anterior y esta fase, para un mismo modelo de navegación se ha definido modelos de interfaces, permitiendo, así que el interfaz se ajuste mejor a las necesidades del usuario.

2.4.1 Interfaz Abstracta para el Modulo Personas

La metodología empleada para el presente trabajo justamente es para aplicativos orientado a objetos, lo cual implica que los objetos que son parte del domino tendrán que ser visualizados como objetos más las propiedades de los mismos, en ese contexto se presenta una interfaz abstracta del modulo personas, donde intervienen algunas funcionalidades en base a la definición de este objeto.

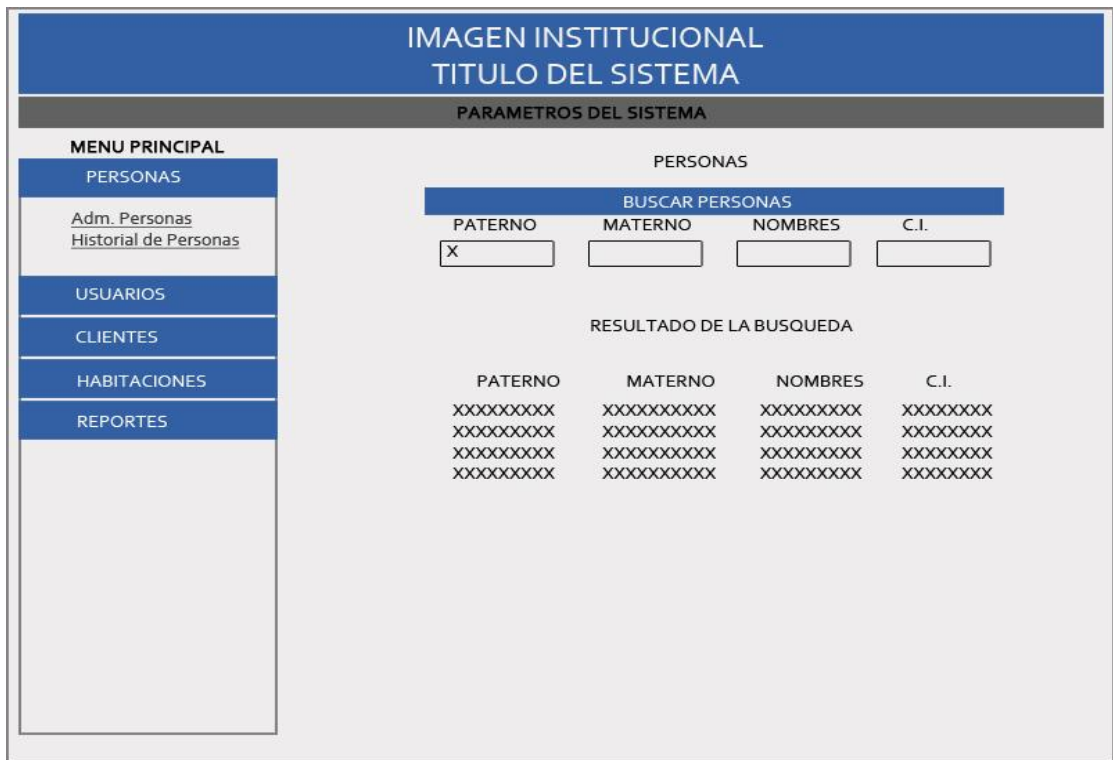


Figura 18: Interfas abstracta del modulo personas
Fuente: Elaboracion propia

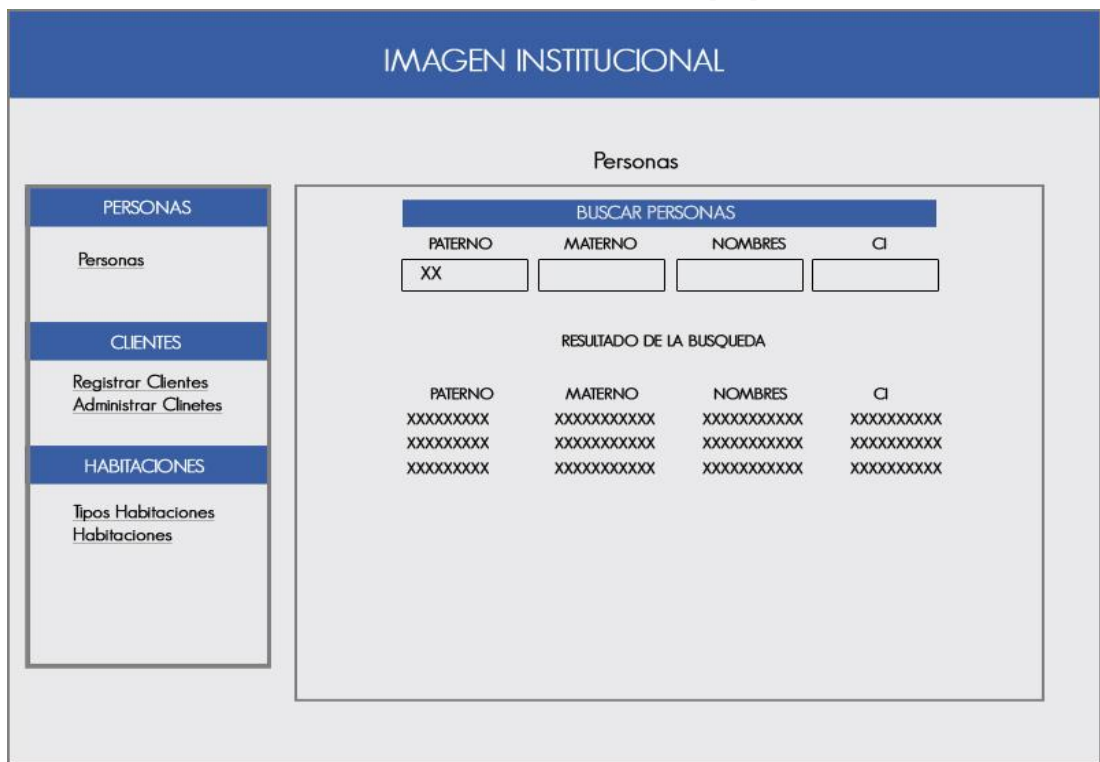
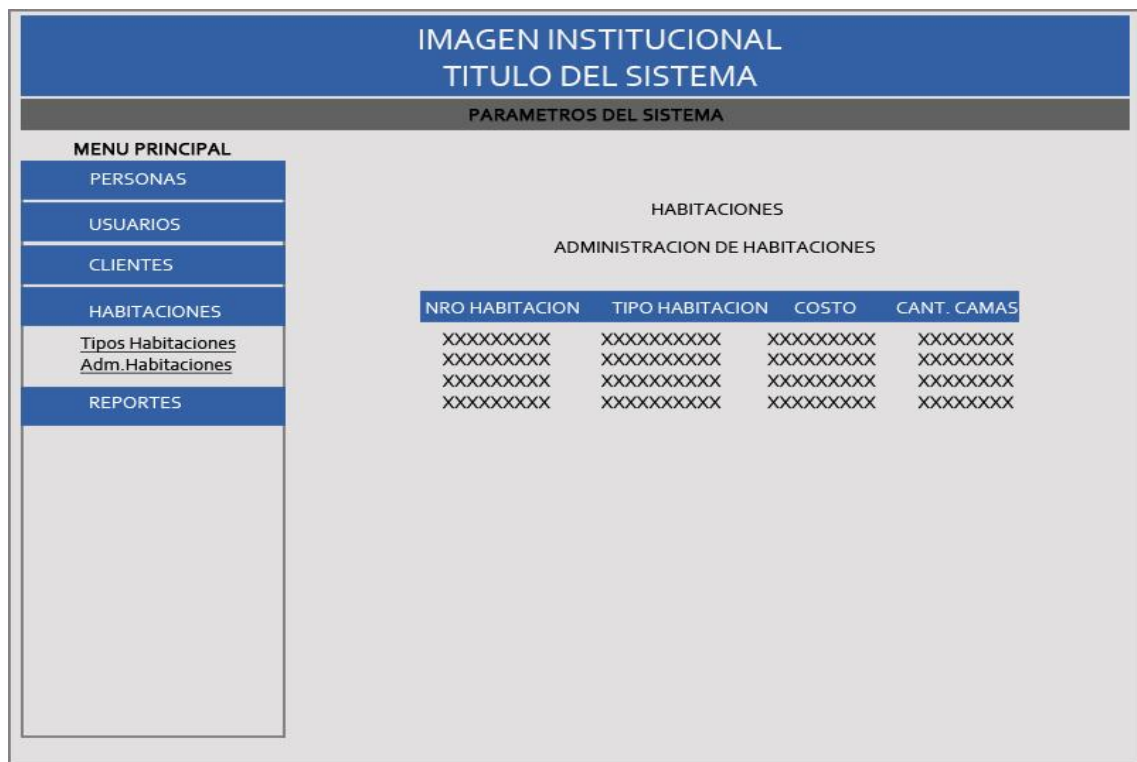


Figura 19: Interfas abstracta del modulo clientes
Fuente: Elaboracion propia



*Figura 20: Interfas abstracta del modulo personas
Fuente: Elaboracion propia*

2.4 IMPLEMENTACION

Una vez terminadas las etapas anteriores, se tendrá el conocimiento del dominio del problema. Así entonces, ya se identificará la información que será mostrada, como estará organizada y cuales funciones permitirá ejecutar la aplicación. Además de ello, se definen la forma en las que deben mostrarse las interfaces.

En esta etapa se han diseñado las clases bajo el lenguaje de PHP5 y las interfaces con la tecnología motor de plantillas Smarty y Ajax, al mismo tiempo se diseñará la base de datos con la herramienta de MySQL.

2.4.1 Diseño de la Base de Datos - Modelo Relacional

En esta etapa se ha diseñado la base de datos con la herramienta MySQL, donde se ha definido como motor de almacenamiento de datos de las tablas el InnoDB. Lo cual permite la integridad de datos a la hora de almacenar la información.

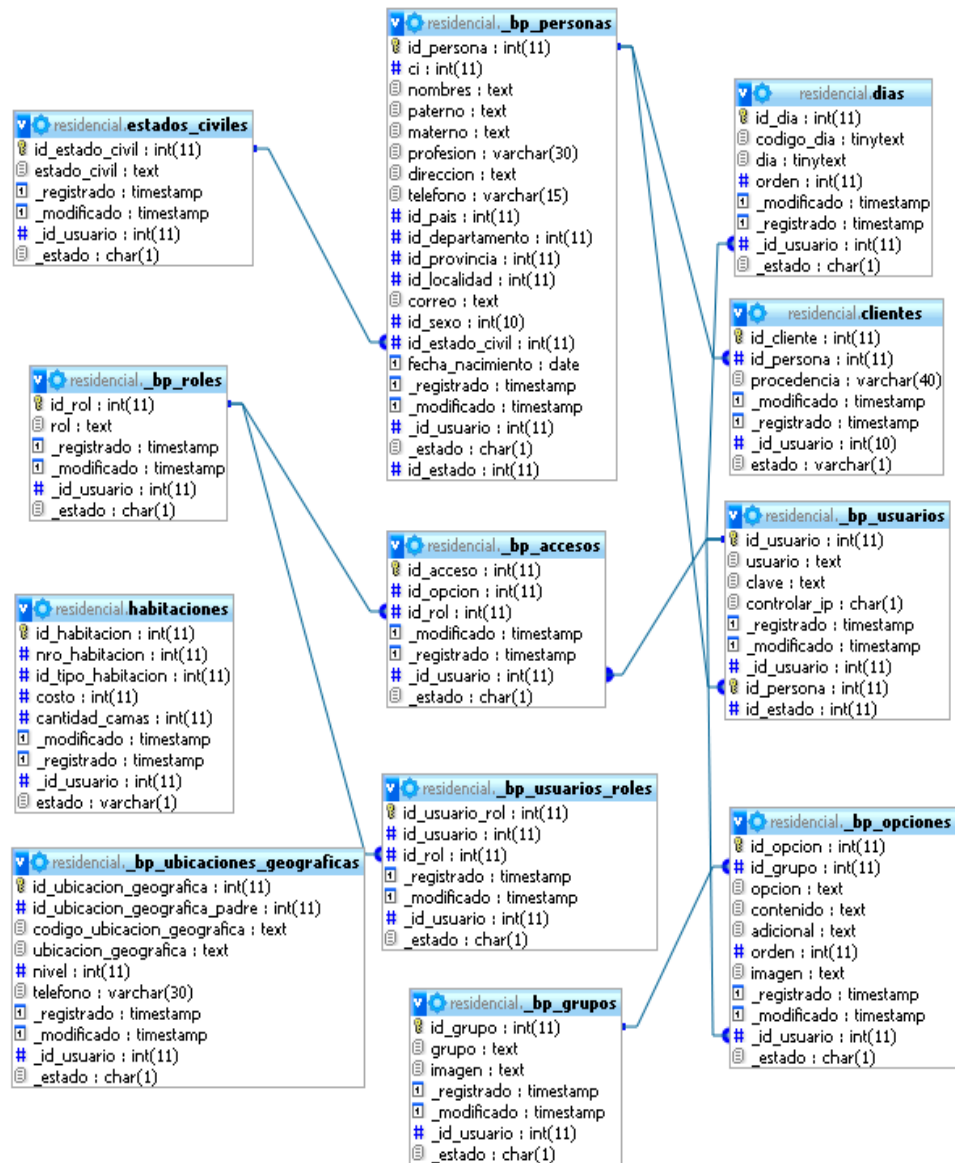


Figura 21: Esquema del modelo Físico de la Base de Datos
Fuente: Elaboracion propia

2.4.2 Definición de clases con ala herramienta PHP

De acuerdo a la metodología empleada para el desarrollo del presente proyecto se ha procedido en primera instancia con la construcción de clases con lenguaje PHP lo cual ha permitido dar una funcionalidad al sistema en su conjunto, en tal sentido se presenta algunas de las clases construidas (VER ANEXO E).

2.4.2.1 Definición de la Clase Conexión

```
<?php
require_once("adodb/adodb.inc.php");
class Conexion{
    private $conServidor = "localhost";
    private $conBasededatos = "residencial";
    private $conUsuario = "root";
    private $conClave = "123456";
    private $db;

    function Conexion(){
        $this->db = NewADOConnection("mysql");
        $this->db->debug = false;
        $this->db->Connect($this->conServidor,$this->conUsuario,$this-
>conClave,$this->conBasededatos);
        $this->db->Execute("SET NAMES 'utf8'");
    }

    function obtenerConexion(){
        return $this->db;
    }

    function obtenerId_conexion(){
        return mysql_connect($this->conServidor, $this->conUsuario, $this->conClave);
    }

    function obtenerVariablesConexion(){
        $registro->conServidor = $this->conServidor;
        $registro->conBasededatos = $this->conBasededatos;
        $registro->conUsuario = $this->conUsuario;
        $registro->conClave = $this->conClave;
        $registro->db = $this->db;
        return $registro;
    }
}
?>
```

2.4.2.2 Definición de la Clase Personas

```
<?
require_once("conexion.class.php");
class Personas extends Conexion{
    public $ci;
    public $nombres;
    public $paterno;
    public $materno;
    public $profesion;
    public $direccion;
    public $telefono;
    public $correo;
    public $id_sexo;
    public $id_estado_civil;
    public $id_pais;
    public $fecha_nacimiento;
    private $_id_usuario;
    private $registrado;

var $registro; //esto es un arreglo
    var $db; //Variable de Conexion con la Base de Datos

    function Personas(){
        $conexion = new Conexion();
        $this->db = $conexion->obtenerConexion();
    }

    function establecerRegistro($ci,
                                $nombres,
                                $paterno,
                                $materno,
                                $profesion,
                                $direccion,
                                $telefono,
                                $correo,
                                $id_sexo,
                                $id_estado_civil,
                                $id_pais,
                                $fecha_nacimiento,
                                $_id_usuario,
                                $registrado,
                                $modificado,
                                $id_estado
    ){
        $this->registro["ci"] = $ci;
        $this->registro["nombres"] = $nombres;
        $this->registro["paterno"] = $paterno;
        $this->registro["materno"] = $materno;
        $this->registro["profesion"] = $profesion;
        $this->registro["direccion"] = $direccion;
        $this->registro["telefono"] = $telefono;
        $this->registro["correo"] = $correo;
```

```

        $this->registro["id_sexo"] = $id_sexo;
        $this->registro["id_estado_civil"] = $id_estado_civil;
        $this->registro["id_pais"] = $id_pais;
        $this->registro["fecha_nacimiento"] = $fecha_nacimiento;
        $this->registro["_id_usuario"] = $_id_usuario;
        $this->registro["_registrado"] = $registrado;
        $this->registro["_modificado"] = $modificado;
        $this->registro["id_estado"] = $id_estado;
    }

    function prsInsertarPersona(){
        $rs = $this->db->AutoExecute("_bp_personas", $this->registro, "INSERT");
        return ($rs);
    }

    function prsModificarPersona($_id_persona) {
        $rs = $this->db->AutoExecute("_bp_personas", $this->registro, "UPDATE"
, "id_persona='".$_id_persona.'"");
        return ($rs);
    }

    function eliminarPersona($_id_persona) {
        $resultado = $this->prsBuscarPersona($_id_persona);
        $this->establecerRegistro(
                                                    $resultado[0]["ci"],
$resultado[0]["nombres"],
$resultado[0]["paterno"],
$resultado[0]["materno"],
$resultado[0]["profesion"],
$resultado[0]["direccion"],
$resultado[0]["telefono"],
$resultado[0]["correo"],
$resultado[0]["id_sexo"],
$resultado[0]["id_estado_civil"],
$resultado[0]["id_pais"],
        $resultado[0]["fecha_nacimiento"],
$_SESSION["sesion_id_usuario"],
$resultado[0]["_registrado"],
                                                    date("Y-m-d H:i:s"),
//_modificado
                                                    2); //id_estado
        return $this->prsModificarPersona($_id_persona);
    }

```

```

    }

    function prsBuscarPersona($__id_persona) {
        $sql = $this->db->Prepare("select p.*, pas.pais, dep.departamento,
pro.provincia,
loc.localidad, s.sexo,s.cod_sexo, c.estado_civil
        from _bp_personas p
        JOIN paises pas USING(id_pais)
        JOIN departamentosdep USING(id_pais, id_departamento)
        JOIN provincias pro USING(id_pais, id_departamento, id_provincia),
        localidades loc, sexos s, estados_civiles c
        WHERE
        p.id_localidad=loc.id_localidad AND
        p.id_sexo=s.id_sexo AND
        p.id_estado_civil=c.id_estado_civil AND
        p.id_persona=?");
        $rs = $this->db->GetAll($sql, array($__id_persona));
        return ($rs);
    }

    function prsBuscarPersonaQueSiTenganItems($__id_persona) {
        $sql = $db->Prepare("SELECT p.ci, p.nombres, p.paterno, p.materno, c.cargo,
i.id_item
        FROM _bp_personas p, _bp_items i, _bp_cargos c
        WHERE p.id_persona= ?
        AND p.id_persona = i.id_persona
        AND i.id_cargo = c.id_cargo
        AND i._estado<> 'X'
        GROUP BY p.id_persona
        ");
        $rs = $db->GetAll($sql, array($__id_persona));
        return ($rs);
    }
}
?>

```

2.2.4.3 Definición de la clase habitaciones

```
<?
require_once("conexion.class.php");
class Habitaciones extends Conexion{
    public $nro_habitacion;
    public $id_tipo_habitacion;
    public $costo;
    public $cantidad_camas;
    private $registrado;
    private $_id_usuario;
    var $registro; //esto es un arreglo
    var $db; //Variable de Conexion con la Base de Datos
function Habitaciones(){
    $conexion = new Conexion();
    $this->db = $conexion->obtenerConexion();
}
function establecerRegistro($nro_habitacion,
                            $id_tipo_habitacion,
                            $costo,
                            $cantidad_camas,
                            $modificado,
                            $registrado,
                            $_id_usuario
){
    $this->registro["nro_habitacion"] = $nro_habitacion;
    $this->registro["id_tipo_habitacion"] = $id_tipo_habitacion;
    $this->registro["costo"] = $costo;
    $this->registro["cantidad_camas"] = $cantidad_camas;
    $this->registro["_modificado"] = $modificado;
    $this->registro["_registrado"] = $registrado;
}
function habitInsertarHabitaciones(){
    $rs = $this->db->AutoExecute("habitaciones", $this->registro,
    "INSERT");
    return ($rs);
}
function habitListarHabitaciones($arreglo)
{
    $sql = $this->db->Prepare("SELECT h.*, th.tipo_habitacion
    FROM habitaciones h, tipos_habitacionsth
    WHERE h.id_tipo_habitacion = th.id_tipo_habitacion
    AND h.estado<> 'X'
    ORDER BY h.nro_habitacion ASC LIMIT ?OFFSET ?
    ");
    $rs = $this->db->GetAll($sql, $arreglo);
    return ($rs);
}
function rlsListarRoles() {
    $sql = $this->db->Prepare("SELECT r.*, u.*
    FROM _bp_rols r, _bp_usuarios u
    WHERE r._id_usuario = u.id_usuario
    AND r._estado<> 'X'

```

```

        ");
        $rs = $this->db->GetAll($sql);
        return ($rs);
    }
    function buscarHabitacion($id_habitacion){
        $sql = $this->db->Prepare("SELECT *
        FROM habitaciones
        WHERE id_habitacion= ?
        AND estado<> 'X'");
        $rs = $this->db->GetAll($sql, array($id_habitacion));
        return ($rs);
    }
    function modificarHabitacion($id_habitacion) {
        $rs = $this->db->AutoExecute("habitaciones", $this->registro,
        "UPDATE" ,"id_habitacion='".$id_habitacion."'");
        return ($rs);
    }
    function eliminarHabitacion($id_habitacion) {
        $resultado = $this->buscarHabitacion($id_habitacion);
        $this->establecerRegistro(
        $resultado[0]["nro_habitacion"],
        $resultado[0]["id_tipo_habitacion"],
        $resultado[0]["costo"],
        $resultado[0]["cantidad_camras"],
        date("Y-m-d H:i:s"), //_modificado
        $resultado[0]["_registrado"],
        $_SESSION["sesion_id_usuario"],
        'X'
        ); //
        return $this->modificarHabitacion($id_habitacion);
    }
}
?>

```

2.5 DISEÑO DE INTERFACES

Tomando en cuenta que las interfaces son las pantallas que se muestran ante el usuario para que puedan ser operados desde allí, en tal sentido Una vez diseñado la base de datos y todo lo referente a clases se ha procedido al diseño de las interfaces con las herramientas que han sido citado con anterioridad para este acápite del trabajo.

2.5.1 Diseño del interfaz de autenticación de usuario

En esta etapa se ha construido el interfaz de para la autenticación de usuario utilizando las tecnologías mencionadas con anterioridad para esta etapa como resultado de la utilización de estas herramientas se presenta una figura del interfaz acabado.



*Figura 22: Interfaz de autenticación de usuario
Fuente: Elaboración propia*



*Figura 23: Interfaz Principal del Sistema
Fuente: Elaboración propia*

2.5.2 Implementación del Modulo Personas

*Figura 24: Diseño Modulo Personas
Fuente: Elaboración propia*

2.5.3 Implementación del Modulo Habitaciones

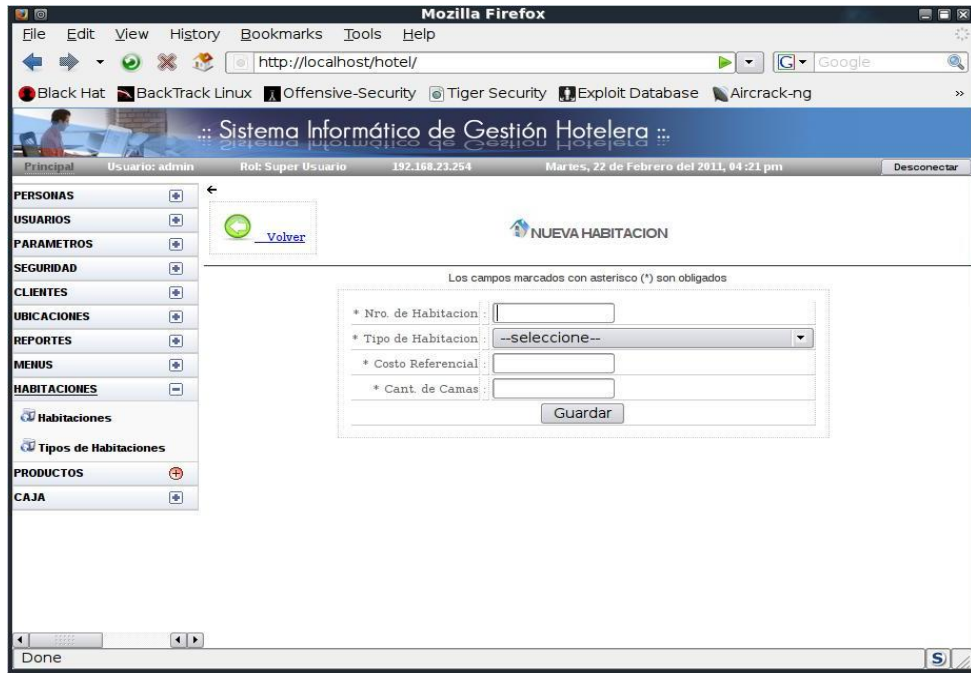


Figura 25: Registro de habitaciones
Fuente: elaboración propia

2.5.4 Implementación del Modulo Asignación de Habitación

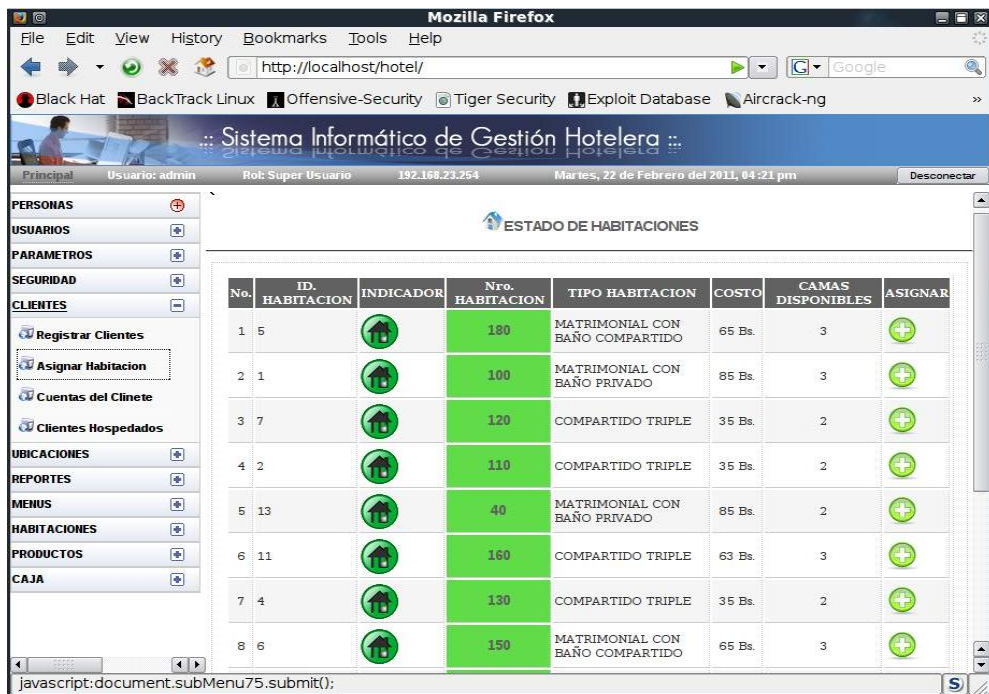


Figura 26: Asignación de Habitación
Fuente: elaboración propia

2.5.5 Implementación del Modulo Clientes

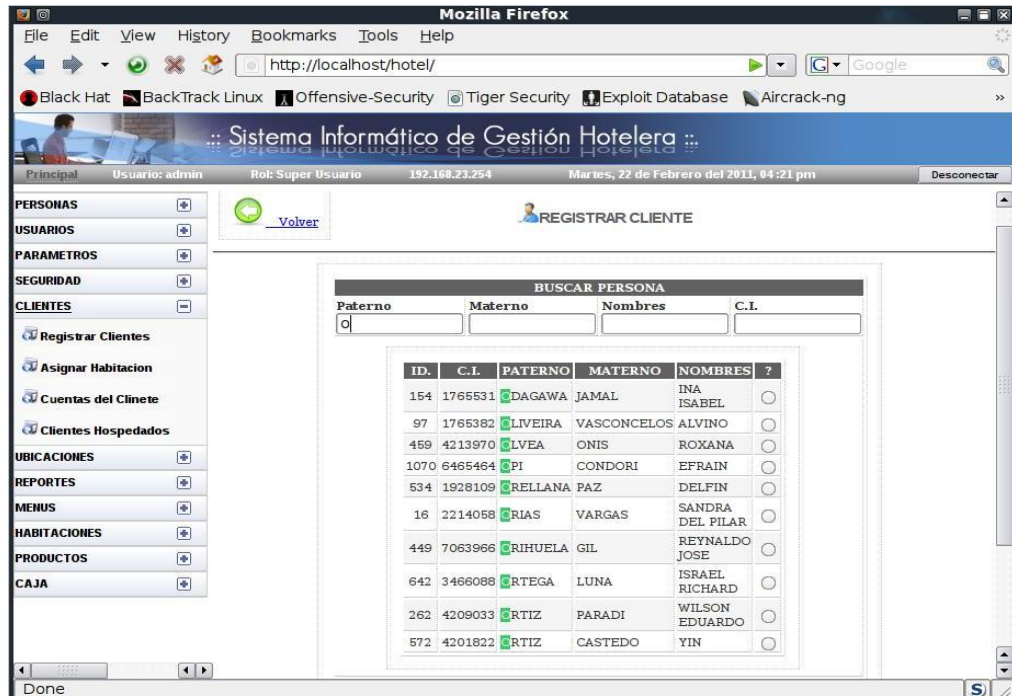


Figura 27: Administración de clientes

Fuente: elaboración propia

2.5.6 Implementación del Modulo Caja

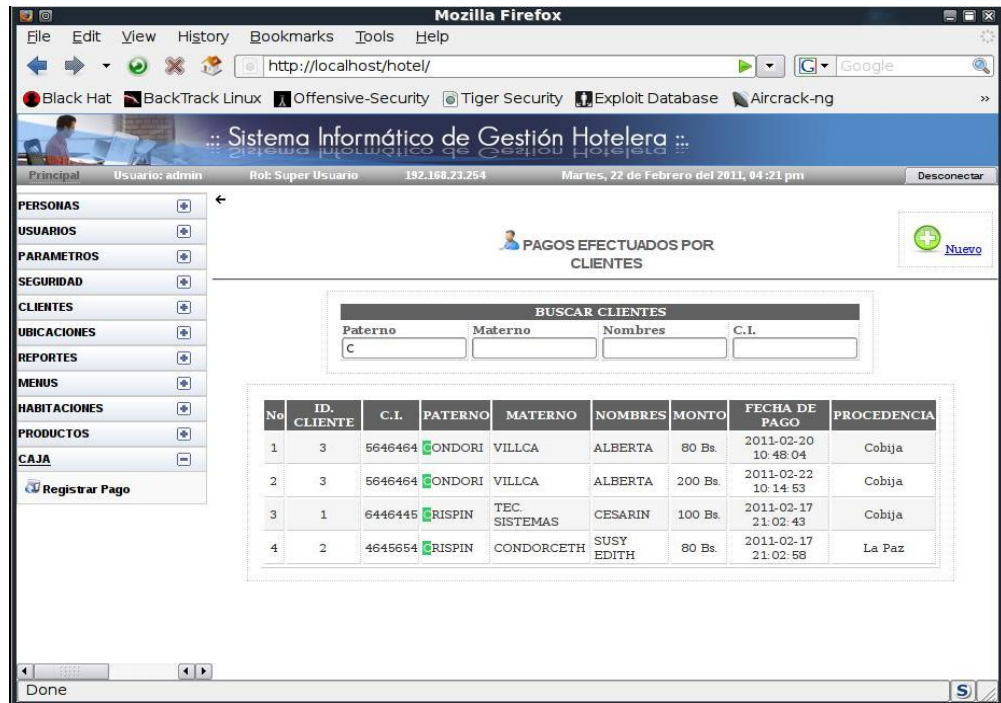


Figura 28: Administración Caja

Fuente: elaboración propia

2.6 PRUEBAS DEL SISTEMA

Por la magnitud del sistema se determina utilizar los casos de prueba utilizando la técnica de la *Caja Negra*, para verificar las condiciones sobre las cuales trabaja el sistema. Para este efecto se seleccionó como caso crítico la historia de usuario Registro de personas.

CASO DE PRUEBA NRO. 1: Registro de Personas

PROPOSITO:

Verificar que los valores introducidos deben ser de tipo numérico y alfabeto de acuerdo a los datos requeridos en el formulario

PRE REQUISITOS

Estar registrado en el sistema como usuario

DATOS DE PRUEBA

DOC. IDENTIFICACION: {**5724FFF120**, []=-//..}

PATERNO: {**VEGA**, 20.4, -34, []=-, 111111}

MATERNO: {**MIRANDA**, 15.3, -40, [], 33333323333232}

NOMBRES: {**DANIEL**, 15.3, -40, [], 333333323323333232}

PROFESION: {**ESTUDIANTE**, 15.3, -40, [], 3323333232}

ESATDO CIVIL: {**SOLTERO**, 15.3, -40, [], 33333333333232}

NACIONALIDAD: {**BOLIVIANO**, 15.3, -40, [], 33333333333232}

Nota: Los valores marcados en negrita son validos

PASOS

Preparatorio, Menú PERSONAS >> Personas >> Nuevo Registro >>

RESULTADOS ESPERADOS

Todos los valores que no están en negrita deben ser rechazados asimismo los valores vacios. El documento de identificación es de tipo alfanumérica debido a que existen varios tipos de documentos por cada País.

EVALUACIÓN DE LA PRUEBA

Superado con éxito

CASO DE PRUEBA 2: Registro de Habitaciones

PROPOSITO:

Verificar el funcionamiento del formulario ingreso de nuevo habitación.

PRE REQUISITOS:

La habitación no está registrada

El tipo de habitación está registrado.

Ingresar como usuario Almacenero.

DATOS DE PRUEBA

Nro. De Habitación &: {**10,20,30** aaaaaabc, ?[]=- , _}

Tipo de habitación: {**Simple, Matrimonial"**, -34, aaaaaa, []=-, 1111111}

Costo de la Habitación: {**35, 85,120**, -40, gg, '[],332333232}

Nota: Los valores marcados en negrita son validos

PASOS

Preparatorio, Menu HABITACIONES >> habitaciones >> Nuevo Registro >>

RESULTADOS ESPERADOS

Mostrar el mensaje de alerta y no permitir el registro de valores inválidos.

EVALUACIÓN DE LA PRUEBA

Superado con éxito

CASO DE PRUEBA 3: Búsqueda de personas

PROPOSITO:

Verificar el funcionamiento del formulario búsqueda de personas

PRE REQUISITOS

La persona deberá estar registrado está registrado

Ingresar como usuario almacenero

DATOS DE PRUEBA

Caja de texto paterno: { **Vega**, '[]=-65646545 }

Caja de texto materno: { **Calle**, '[]=- 12121212 }

Caja de texto nombres: { **Daniel**, '[]=- 02131330 }

Caja de texto C.I.: { **75241720o**, '[]=- }

Nota: Los valores marcados en negrita son validos

PASOS

Preparatorio, Menú Personas >> Buscar Personas>> posicionar el cursor en cualquiera de las cajas de texto, PATERNO, MATERNO, NOMBRES, CI

RESULTADOS ESPERADOS

Mostrar el mensaje de alerta y no permitir el registro de valores inválidos.

EVALUACIÓN DE LA PRUEBA

Superado con éxito.

CAPITULO IV

CONCLUSIONES Y RECOMENDACIONES

1. CONCLUSIONES

Como conclusión del presente trabajo la cual es “Desarrollo de Sistema Informático de Gestión Hotelera para la Residencial Opi”, pretende dar solución a los problemas planteados con anterioridad, se tiene como resultado de ello elementos importantes que contempla el trabajo desarrollado.

- Se ha concluido de manera satisfactoria con la fase de obtención de requerimientos en la que se ha obtenido la mayor cantidad de información a través de las entrevistas realizadas.
- Se ha concluido con el diseño conceptual del sistema de tal manera que los objetos que intervienen en el trabajo desarrollado son percibidos de manera clara.
- La fase de diseño navegacional ha sido concluido de manera satisfactoria donde cada nodo u objetos son definidas de manera específica.
- Se ha concluido con la fase de implementación donde se ha utilizado las siguientes herramientas para esta etapa: PHP5, Smarty, Ajax,HTML,MySql
- Con la implementación del Sistema de Gestión Hotelera se han automatizado diversos procesos tales como registro de clientes, asignación de habitaciones al cliente, registro de ventas-consumos, informes de clientes hospedados entre otros. En tal sentido se logró reducir de manera considerable los gastos que anteriormente se realizaban para efectuar estas tareas.
- Las búsquedas son ligeramente rápidas al momento de realizar una consulta sobre la existencia de clientes hospedados en la Residencial.

2. RECOMENDACIONES

Una vez culminado con el desarrollo del proyecto se ha visto la necesidad de considerar algunos aspectos tales como:

- Integrar al sistema desarrollado con los módulos **Modulo de contabilidad**
- Asimismo integrar un **modulo para reserva de habitaciones on-line**, los cuales darán mayor funcionalidad al sistema implementado.

BIBLIOGRAFIA

BIBLIOGRAFIA

[Roberto Hernández Sampieri, Metodología de la Investigación, Cuarta Edición, 2006]

[Roger S. Pressman, Ingeniería del Software, Sexta Edición, 2006]

[Francisco Charte Ojeda, Sistema Operativo Linux, primera edición, 2003]

[Bertalanffy, Ludwin, *Teoría General de Sistemas*. Primera edición, 1969]

[Sistema de Gestión, “Gestión Hotelera”, URL

<http://www.datahousecompany.com.ar/>]

Visitado el 28 de Junio del 2010

[Turismo, “Pando - Turismo”, URL

http://www.bolivia.com/Turismo/ciudades/pando/que_visitar.htm]

Visitado el 6 de Noviembre del 2008

[Patrones Modelo Vista Controlador, Modelo Vista Controlador,

URL <http://www.proactiva-calidad.com/java/patrones/mvc.html>]

Visitado el 13 de Noviembre del 2008

[Base de Datos, MySQL,

URL <http://www.sobl.org/traducciones/practical-postgres/node19.html>]

Visitado el 13 de Noviembre del 2008

ANEXO A
CUADRO DE INVOLUCRADOS DE LA RESIDENCIAL OPI

CUADRO DE INVOLUCRADOS DE LA RESIDENCIAL OPI

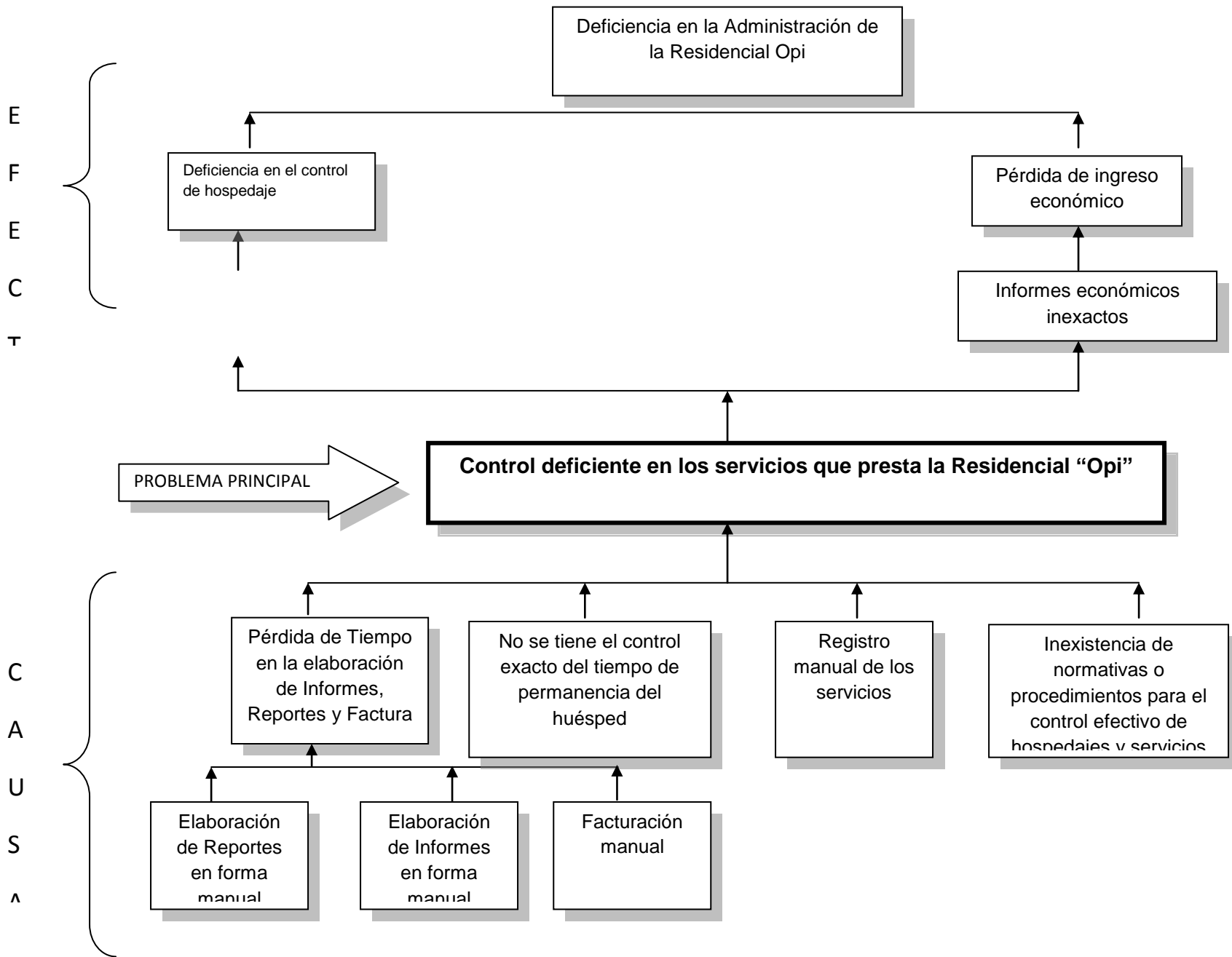
Grupos	<i>Intereses</i>	Problemas Percibidos	Recursos y mandatos	Interés en un Proyecto	Conflictos potenciales
<p>CLIENTES (huésped, Turistas, Viajeros,)</p>	<p>-Atención de calidad y óptimos servicios.</p> <p>-Excelente atención personal</p>	<p>-No se cuenta con la información oportuna de las habitaciones disponibles.</p> <p>-Demora en totalizar el monto a pagar por todos los servicios prestados en la entidad.</p> <p>-Facturación manual</p>		<p>-Los procedimientos en el control de servicios deben ser mejorados.</p>	<p>-----o-----</p>
	<p>- Brindar una atención</p>	<p>-Informes estadísticos y</p>	<p>-Predisposición para la</p>	<p>-Mejorar el control de los</p>	

<p>ADMINISTRACION (Gerente General, Gerente de Créditos, Gerente Contable)</p>	<p>con calidad y óptimos servicios a los clientes.</p> <p>-Cumplimiento de los procedimientos de recepción, servicios y partida del huésped.</p> <p>-Rapidez y confiabilidad en los procesos de recepción, servicios y partida.</p> <p>-Mejorar la administración económica.</p> <p>-Mejorar la prestación de servicios.</p>	<p>económicos poco fiables.</p> <p>-No se cuenta con información oportuna para la toma de decisiones.</p> <p>-Deficiencia en el control de servicio de hospedaje.</p> <p>-Deficiencia en el control de servicios y consumos.</p>	<p>compra de los requerimientos de equipos.</p> <p>-Apoyo con recursos humanos dispuestos a ser capacitados sobre el manejo de Sistema.</p>	<p>servicios en la entidad.</p>	<p>-----o-----</p>
--	--	--	---	---------------------------------	--------------------

<p>EMPLEADOS (Recepcionista, Camarero, botón, Personal de limpieza).</p>	<p>-Mejorar los procesos y sistematizarlos.</p> <p>-Contar con un Sistema de Información Hotelera para el control de servicios como ser: hospedajes, ventas, consumos, y otros.</p> <p>-Contar con tutoriales Interactivos en cuanto al manejo del Sistema de Información.</p> <p>-Contar con un manual de funciones.</p>	<p>-Los reportes realizados son manuales.</p> <p>-Facturación manual</p> <p>-No se tiene un control adecuado sobre el movimiento de clientes y servicios prestados en la entidad.</p> <p>-No existe un Sistema de Información para el control de servicios.</p> <p>-No existe un control exacto de la entrada y salida del huésped</p>	<p>- Material de escritorio.</p> <p>-Información necesaria y apoyo del personal de Residencial Opi.</p>	<p>-Capacitación al personal</p>	<p>-----o-----</p>
--	---	--	---	----------------------------------	--------------------

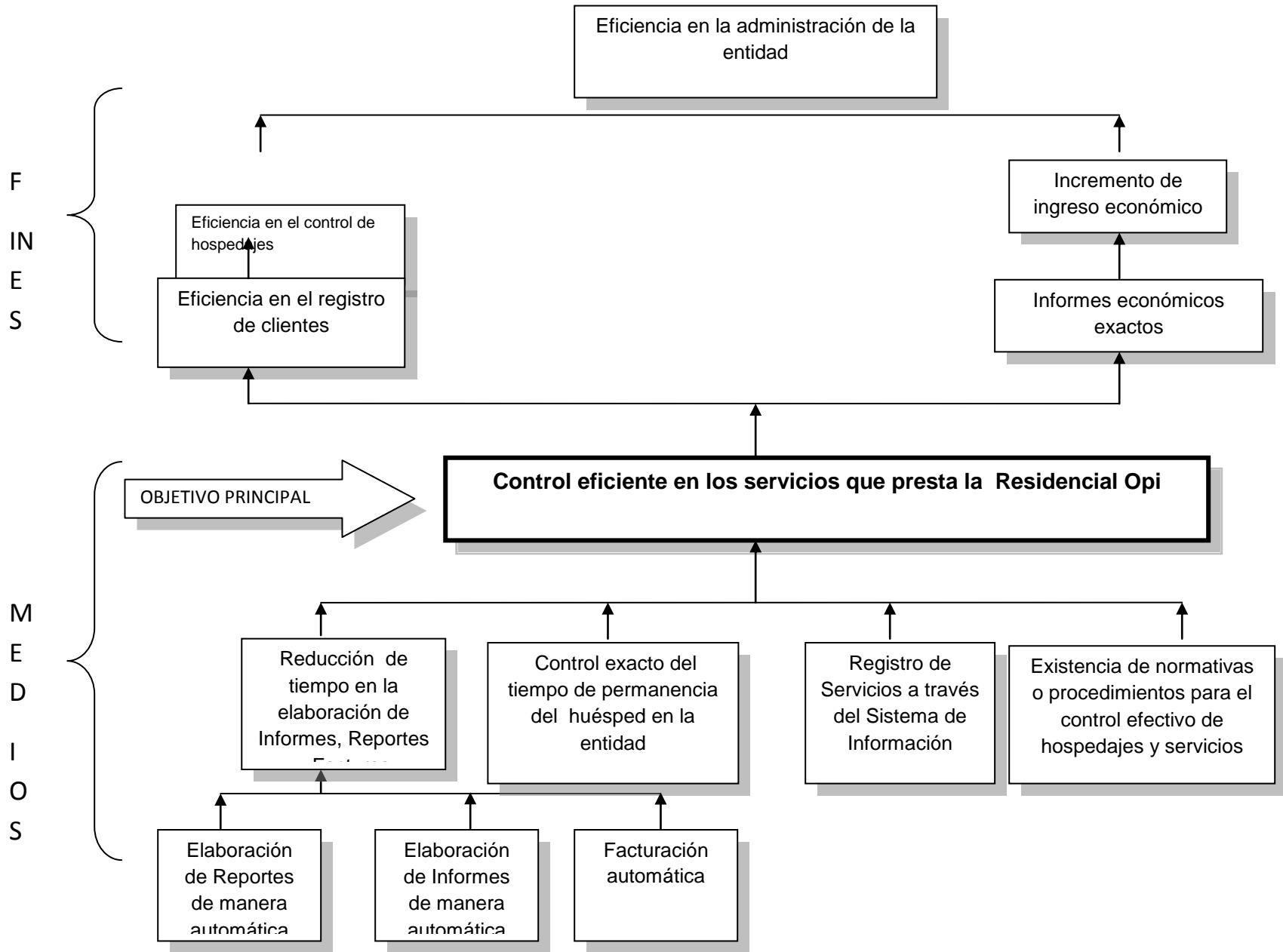
		durante su estadía.			
--	--	---------------------	--	--	--

ANEXO B
ÁRBOL DE PROBLEMAS



ANEXO C
ÁRBOL DE OBJETIVOS

ÁRBOL DE OBJETIVOS



ANEXO D

**MARCO LÓGICO DEL SISTEMA PARA EL CONTROL DE
HOSPEDAJES Y SERVICIOS EN RESIDENCIAL OPI**

MARCO LÓGICO DEL SISTEMA PARA EL CONTROL DE HOSPEDAJES Y SERVICIOS EN RESIDENCIAL OPI

Resumen narrativo de objetivos	Indicadores objetivamente verificables	Medios de verificación	Supuestos importantes
<p style="text-align: center;">FIN</p> <p>Eficiencia en la administración de la entidad</p>	<p>1.- Se ha mejorado la administración de los servicios de la Residencial Opi a más del 50%. Desde la implementación del sistema.</p>	<ul style="list-style-type: none"> - Informes por parte del administrador del sistema - Informes estadísticos de ingreso de clientes. - Informes estadísticos de ingreso económico. 	<ul style="list-style-type: none"> - El propietario realizará todas las inversiones necesarias.
<p style="text-align: center;">PROPÓSITO</p> <p>Control eficiente de hospedajes y servicios de la Residencial Opi a partir de la implantación de un sistema de información con</p>	<p>1.- Se ha reducido a más del 50% el tiempo invertido en cada proceso de control de los servicios desde la</p>	<ul style="list-style-type: none"> - Informes del Administrador de la Residencial Opi 	<ul style="list-style-type: none"> - El personal involucrado esta capacitado en el uso de TIC's.

<p>arquitectura Cliente-Servidor desarrollado bajo la metodología “<i>Extension de Aplicación Web</i>” en ingles ”<i>Web Application Extension for UML – Process Conallen</i>”</p>	<p>implantación del sistema.</p> <p>2.- Entrega oportuna de informes y reportes para la toma de decisiones de los involucrados.</p>		<p>-El sistema de información es utilizado.</p> <p>- Predisposición al cambio por parte de los involucrados beneficiarios.</p>
<p style="text-align: center;">COMPONENTES</p> <p>1.- Sistema de control de Hospedaje y servicios para Residencial Opi funcionando.</p> <p>2.- Documentación Técnica del Sistema impresa.</p> <p>3.- Manual de usuario Impreso</p>	<p>1.- Sistema desarrollado en un semestre.</p> <p>3.- Desarrollo de la documentación técnica en 1 mes.</p> <p>2.- Desarrollo del manual de usuario</p>	<p>- Informes del Administrador de la Residencial Opi.</p> <p>- Reportes emitidos por el Sistema.</p> <p>- CD de instalación del Sistema.</p> <p>- Certificación del sistema probado e instalado en Residencial Opi.</p>	<p>- Disponibilidad de software, hardware y otros recursos.</p> <p>- Disponibilidad de recursos humanos capacitados.</p>

<p>4.- Tutorial Interactivo para el manejo del Sistema</p>	<p>completo en un mes.</p> <p>2.- Desarrollo del tutorial interactivo en un mes.</p>	<p>- Manual de usuario completo.</p> <p>- Tutorial interactivo completo</p>	
<p>ACTIVIDADES</p> <p>1.1.- Manejo de proyecto</p> <p>1.2.- Captura de requerimientos</p> <p>1.3.- Análisis</p> <p>1.4.- Diseño</p> <p>1.5.- Implementación</p> <p>1.6.- Prueba</p> <p>1.7.- Desarrollo</p> <p>1.8.- Configuración y manejo de cambios</p> <p>2.1.- Elaboración de la</p>	<p>- Cronograma de actividades</p> <p>1.1.- 1 personas/ 2meses</p> <p>1.2.- 1 personas/ 1mes</p> <p>1.3.- 1 personas/ 3meses</p> <p>1.4.- 1 personas/ 2meses</p> <p>1.5.- 1 personas/ 1mes</p> <p>1.6.- 1 personas/ 2meses</p> <p>1.7.- 1 persona/ 1mes</p>	<p>- Informes del Tutor.</p> <p>-Informe del Revisor.</p> <p>- Documentación del desarrollo del Sistema.</p>	<p>- Existe el apoyo suficiente de parte de la Residencial Opi para brindar toda la información necesaria.</p> <p>- Cumplimiento del cronograma de actividades.</p> <p>- Cumplimiento del envío de informes.</p> <p>- Personal dispuesto a pasar</p>

documentación técnica. 3.1.- Análisis del Sistema. 3.2.- Elaboración del manual de usuario. 3.3.- Elaboración del tutorial interactivo	- Tres equipo de computación con accesorios y software. - Material de escritorio.		cursos de actualización.
---	--	--	--------------------------

ANEXO E
OPERACIONALIZACION DE VARIABLES

OPERACIONALIZACION DE VARIABLES

Variable	Concepto	Dimensión	Indicador	Instrumento
Personal administrativo y adiestramiento en el manejo de sistemas	Todos los funcionarios involucrados en el trabajo de la institución	Procesos (Hospedajes)	Personal de la entidad encargado en el control de servicios	Entrevista y observación
Organización	Estructura organizacional de la entidad	Normativas	Estatutos cámara Hotelera	Entrevista
			Procedimientos para la recepción de huéspedes	Observación y entrevista
			Manual de procedimiento y Seguridad del Sistema.	Entrevista
Clientes	Todos los potenciales huéspedes	personas	Servicios que requieren los clientes	Entrevista y Cuestionario
Sistemas informáticos	Sistema que automatiza los diferentes procesos de la entidad	Tecnología Informática	Sistema Informático	Entrevista
			Manejador de bases de Datos	Cuestionario
			Hojas de Calculo	Cuestionario
Recursos económicos y tecnológicos	Recursos económicos con la que cuenta la entidad.	Económica	Costos Mínimos en Equipos Calidad de Diseño de los programadores.	Observación y entrevista
	Documentación que guía al usuario	Tecnológica	Normativas para el uso del sistema	Observación y entrevista

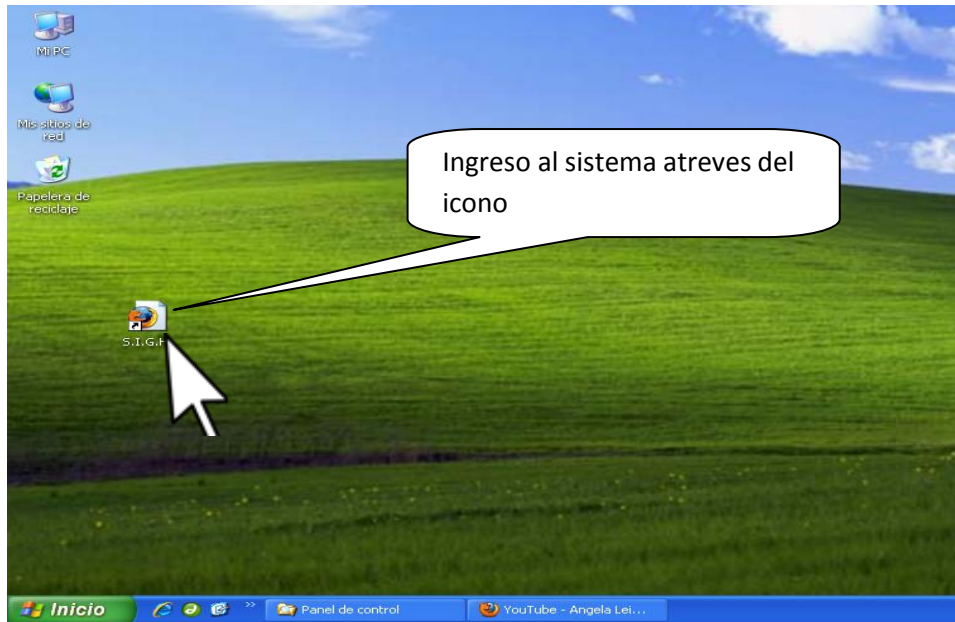
ANEXO F
MANUAL DE USUARIO

MANUAL DE USUARIO
SISTEMA INFORMÁTICO DE GESTIÓN HOTELERA

MANUAL DE USUARIO

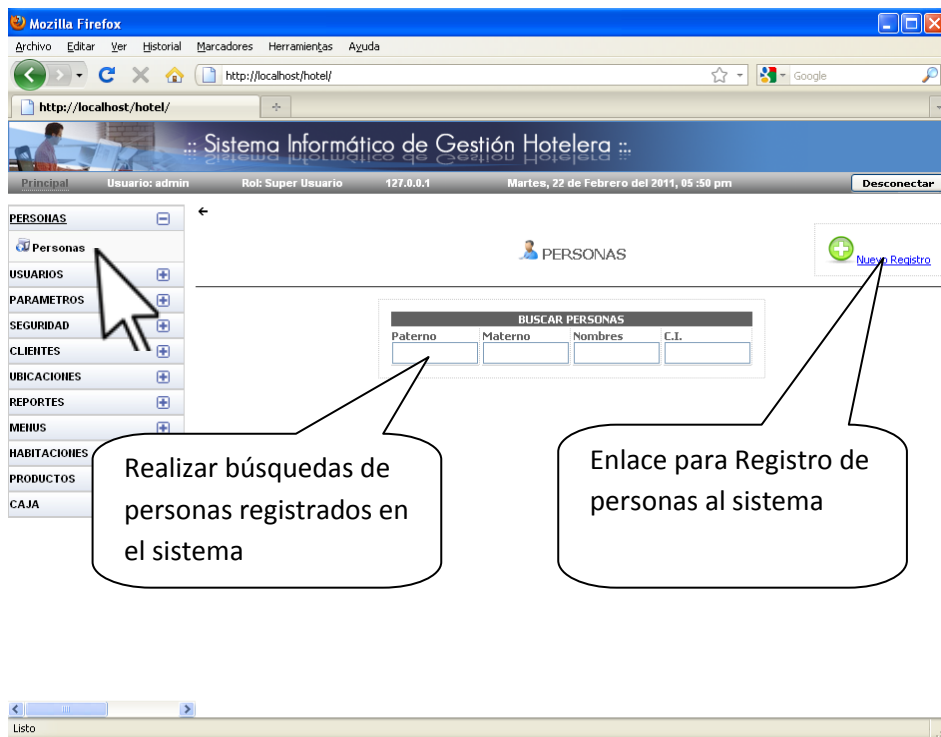
1.- ACCESO AL SISTEMA

El ingreso al sistema se lo realiza atreves del icono con el nombre S.I.G.H. en el escritorio de la computadora (**doble click para el acceso**).



2.- REGISTRO DE PERSONAS AL SISTEMA

Una vez ingresado a la pantalla principal desde allí deberá seleccionar la opción **PERSONAS >>Personas>>Nuevo**



Ingresar los datos del cliente en el siguiente formulario y presione guardar

The screenshot shows a web browser window with the URL `http://localhost/hotel/`. The page title is "Sistema Informático de Gestión Hotelera". The user is logged in as "admin" with the role "Super Usuario". The main content area is titled "NUEVA PERSONA" and contains a form with the following fields:

- * Doc. Identificación:
- * Nombres:
- * Apellido paterno:
- * Apellido materno:
- * Profesion:
- Telefono:
- Correo:
- * Sexo: Masculino Femenino
- * Estado civil:
- * Pais:
- * Fecha Nacimiento:

A "Guardar" button is located at the bottom of the form. A mouse cursor is pointing at the button. A "Volver" button is also visible in the top left of the form area.

3.- REGISTRO DE CLIENTES

Para registrar clientes debe ir a la opción **CLIENTES>>Registrar Clientes>> Nuevo**

The screenshot shows the "CLIENTES" menu in the hotel management system. The menu items are:

- PERSONAS
- USUARIOS
- PARAMETROS
- SEGURIDAD
- CLIENTES
 - Registrar Clientes
 - Asignar Habitación
 - Cuentas del Cliente
 - Clientes Hospedados
- UBICACIONES
- REPORTE
- MENUS
- HABITACIONES
- PRODUCTOS
- CAJA

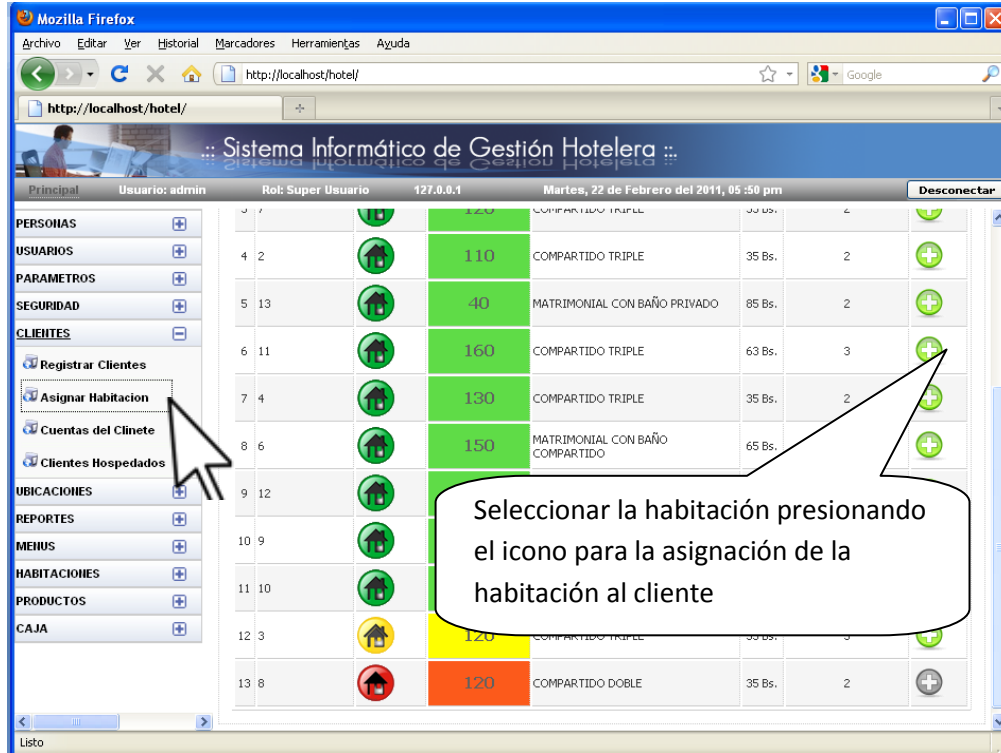
Annotations include:

- A callout box pointing to the "Registrar Clientes" menu item with the text: "Buscar clientes registrados en el sistema".
- A callout box pointing to the "Nuevo" button (a green plus sign) with the text: "Enlace para el formulario de registro de clientes".

The "BUSCAR CLIENTES" search form is also visible, with fields for "Paterno", "Materno", "Nombres", and "C.I.". The status bar at the bottom shows the JavaScript code: `javascript:document.subMenu72.submit();`

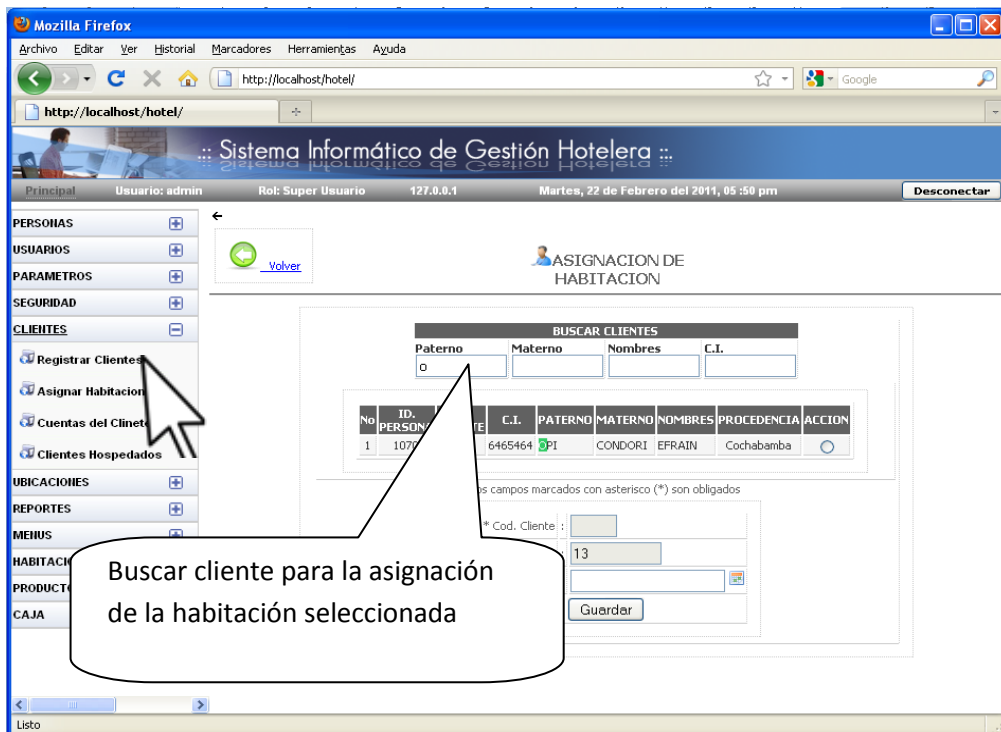
4.- ASIGNACION DE HABITACION

Para la asignación de habitación al cliente realizar desde la opción **CLIENTES>>Asignar Habitación**



Seleccionar la habitación presionando el icono para la asignación de la habitación al cliente

Una vez seleccionado el icono deberá buscar el cliente en el siguiente formulario.



Buscar cliente para la asignación de la habitación seleccionada

5.- REGISTRAR CONSUMOS

Para el registro de consumos debe realizar desde menú **PRODUCTOS>>Registrar consumos**

Buscar producto a través de nombre o cod. De producto

No	CODIGO	PRODUCTO	GRUPO	CANT	COSTO UNIT.	?
1	CC	EPILLO DENTAL	BEBIDAS	24	2 Bs.	✓
2	PH	PAPEL HIGIENICO	LIMPIEZA	48	1.5 Bs.	✓
3	PET	PET GOLA	BEBIDAS	24	10 Bs.	✓
4	CC	GOLA	BEBIDAS	20	15	✓

Seleccionar presionando el icono

Una vez seleccionado el producto el sistema muestra la siguiente interfaz en la que se debe buscar el nombre del cliente de quien va a consumir el producto.

REGISTRO DE CONSUMOS

BUSCAR CLIENTES			
Paterno	Materno	Nombres	C.I.
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Los campos marcados con asterisco (*) son obligados

* Cod. Cliente :

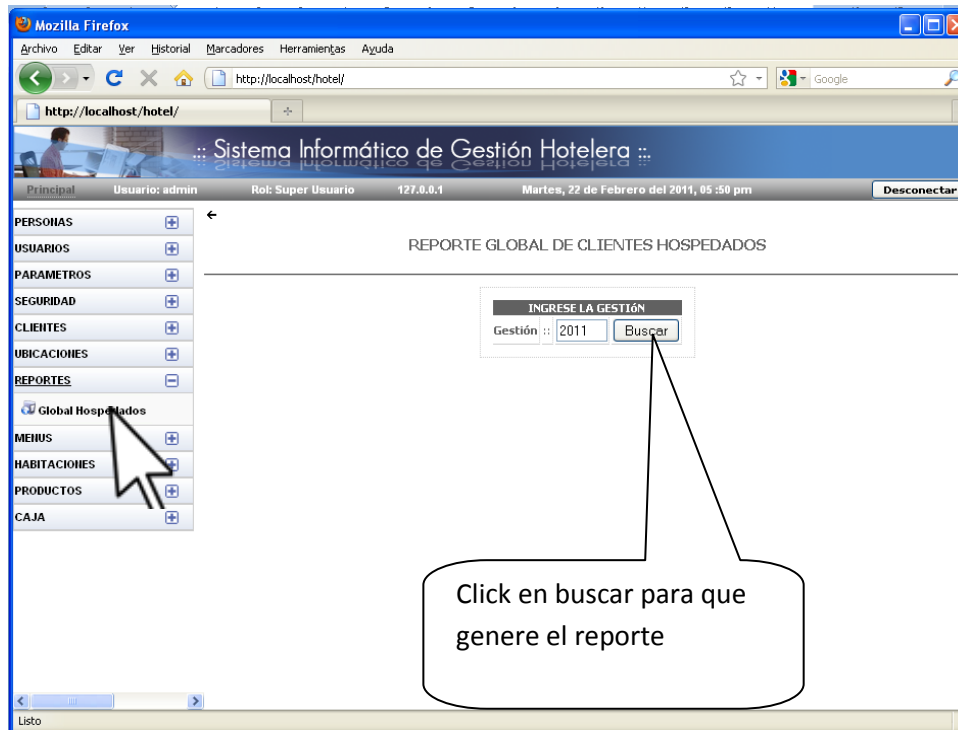
* Cod. Producto : 29

* Fecha :

Guardar

6.- SACAR REPORTE GLOBAL DE HOSPEDADOS

Para realizar esa tarea seleccione opción de menú **REPORTES>>Reporte global** Como se muestra en la siguiente captura de pantalla.



El resultado será la generación de un reporte en una ventana nueva que como la que se muestra en la figura siguiente.

The screenshot shows a new browser window displaying the generated report. The page header includes the logo of 'Residencial Opi' and the text 'SISTEAM INFORMATICO DE GESTION EN HOTELERIA "Residencial OPI" COBLJA - P A N D O'. The report title is 'REPORTE GLOBAL DE CLIENTES HOSPEDADOS'. The table below contains the following data:

FECHA	NOMBRES	No. HABIT.	TIPO HABIT.	NACIONALIDAD	PROCEDENCIA	PROFESION	ESTADO CIVIL	EDAD	C.I.
2011-02-22 11:36:13	ZENTENO CALLISAYA DAVID	150	SS	BOLIVIA	Cobija	ING INFORMATICO	Soltero(s)	2011-02-22	4854447
2011-02-21 11:02:52	FACO RAMOS GIOVANNA	150	SS	BOLIVIA	nberalta	UNIVERSITARIA	Soltero(s)	1986-07-31	7611592
2011-01-30 22:29:10	OPI CONDORI EFRAIN	130	CT	BOLIVIA	Cochabamba	ING INFORMATICO	Soltero(s)	2011-01-14	6465464
2011-01-30 11:19:27	CONDORI VILLCA ALBERTA	110	CT	BOLIVIA	Cobija	CONTADORA	Soltero(s)	1987-12-30	5646464
2011-01-12 13:56:43	CRISPIN TEC SISTEMAS CESARIN	100	CT	BOLIVIA	Cobija		Soltero(s)	1983-01-05	6446445
2010-02-03 05:52:29	VACA LOPEZ YARCENIA	100	CT	BOLIVIA	Tinidad		Soltero(s)	1989-07-12	9270220
2009-12-16 04:18:26	VACA CUELLAR FABIOLA	110	CT	BOLIVIA	Riberalta		Soltero(s)	1987-10-05	7599223
2009-12-16 04:18:26	VACA CUELLAR FABIOLA	110	CT	BOLIVIA	Riberalta		Soltero(s)	1987-10-05	7599223