

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

SEDE VIÑA DEL MAR

JOSÉ MIGUEL CARRERA

SISTEMA DE ADMINISTRACIÓN PARA CAMPING CUCAO

Trabajo de Titulación

para optar al Título de:

TÉCNICO UNIVERSITARIO

EN INFORMÁTICA

Alumnos:

Mauricio Figueroa Nahuelcar

Rodrigo Madrid Valencia

Profesor Guía:

Rodrigo Pinochet Ponce.

2018

RESUMEN

Key words: Sistema, Programa, Servicios, IDE, Base de Datos, Camping Cucao.

El siguiente informe tiene como objetivo proveer a la Microempresa Familiar Camping Cucao, un sistema informático que sea capaz de asistir y mantener el control del camping en temporada de funcionamiento, resolviendo las distintas problemáticas con relación a la mejora de atención hacia los clientes y llevar un registro de los servicios que se prestan dentro del Camping.

Camping Cucao está ubicado en Cucao, Chiloé, Región de los Lagos, como referencia, frente al Parque nacional de Chiloé, es un sector de parcelas divididas en la cual se cuenta con un total de media hectárea para prestar servicios. Allí es donde se encuentran las zonas de camping que son utilizadas por los clientes y donde se encuentra la pequeña oficina donde pueden pedir información y el servicio de caminata.

Con el fin de mejorar los tiempos, la atención a clientes y poder mantener un orden de la información que se maneja a diario el camping, se propone un sistema informático que pueda cumplir con todas estas exigencias que presenta el negocio.

Para la construcción del sistema se utilizará el lenguaje de Programación Visual Basic.NET a través del IDE (Entorno de desarrollo integrado) Visual Studio 2017, siendo esta su última versión disponible hasta el momento. Y para almacenar los datos del sistema se utilizará el sistema de gestión de base de datos Microsoft Access 2016.

Capítulo uno: Se describe la empresa en su totalidad para poder contextualizar el ambiente del negocio y la estructura con la cual funciona actualmente, los problemas que se detectaron y sus respectivas soluciones propuestas.

Capítulo dos: Se muestran principalmente las tablas que formarán parte del sistema, diagrama de Flujos administrativos, entradas y salidas, modelo de datos relacional y estructura de códigos utilizados por el sistema. A su vez son descritas las estructuras funcionales del sistema de manera breve.

Capítulo tres: Por último, en el capítulo tres se dan a conocer el Diagrama de Menú que tendrá el sistema y el Diagrama modular con el cual se trabajará, se muestran las pantallas del sistema dando a conocer cada una de las funciones requeridas.

ÍNDICE

| | |
|---|----|
| RESUMEN | 1 |
| INTRODUCCIÓN | 1 |
| 1. ASPECTOS RELEVANTES DEL DISEÑO LÓGICO..... | 3 |
| 1.1 DESCRIPCIÓN DE LA ORGANIZACIÓN..... | 3 |
| 1.2 DESCRIPCIÓN DE LA SITUACIÓN ACTUAL..... | 4 |
| 1.3 PROBLEMAS DETECTADOS..... | 5 |
| 1.4 DESCRIPCIÓN DEL SISTEMA PROPUESTO..... | 6 |
| 1.4.1 OBJETIVOS DEL SISTEMA PROPUESTO | 6 |
| 1.4.2 BENEFICIOS DEL SISTEMA PROPUESTO..... | 6 |
| 1.4.3 DESCRIPCIÓN GENERAL DE LA SOLUCIÓN PROPUESTA | 7 |
| 1.4.4 DIAGRAMA DE FLUJO ADMINISTRATIVO | 8 |
| 1.4.5 ESTRUCTURA FUNCIONAL DEL SISTEMA..... | 10 |
| 1.4.6 DESCRIPCIÓN DE FÓRMULAS Y CÁLCULOS EMPLEADOS | 10 |
| 1.4.7 INFORMACIÓN QUE SE MANEJARÁ | 12 |
| 1.4.7.1 SALIDAS..... | 12 |
| 1.4.7.2 ENTRADAS..... | 12 |
| 1.4.7.3 ENTIDADES DE INFORMACIÓN | 13 |
| 1.4.7.4 MODELO DE DATOS RELACIONAL..... | 16 |
| 1.4.8 ESTRUCTURA DE CÓDIGOS..... | 17 |
| 1.4.9 CONDICIONANTES DE DISEÑO | 18 |
| 2 MEDIO AMBIENTE COMPUTACIONAL Y DESCRIPCIÓN DE ARCHIVOS..... | 20 |
| 2.1 INTRODUCCIÓN..... | 20 |
| 2.1.1 CONFIGURACIÓN DEL SISTEMA | 20 |
| 2.1.2 SOFTWARE UTILIZADO | 21 |
| 2.2 DESCRIPCIÓN DE ARCHIVOS | 22 |
| 3. DESCRIPCIÓN DE PROGRAMAS | 31 |
| 3.1 DIAGRAMA MODULAR..... | 31 |

| | | |
|-----|---|----|
| 3.2 | DIAGRAMA DE MENÚS | 32 |
| 3.3 | LISTA DE PROGRAMAS | 33 |
| 3.4 | DESCRIPCIÓN DE PROGRAMAS..... | 34 |
| | CONCLUSIONES | 44 |
| | BIBLIOGRAFÍA | 45 |
| | ANEXO 1: CODIGO FUENTE: LOGIN..... | 46 |
| | ANEXO 2: CODIGO FUENTE PANTALLA PRINCIPAL | 47 |
| | ANEXO 3: CODIGO FUENTE AGREGAR CLIENTE | 54 |
| | ANEXO 4: CODIGO FUENTE AGREGAR SOLICITUD | 55 |
| | ANEXO 5: CODIGO FUENTE MODIFICAR USUARIO | 58 |
| | ANEXO 6: CODIGO FUENTE ELIMINAR GUIA..... | 59 |
| | ANEXO 7: CODIGO FUENTE AGREGAR PAGO | 59 |

INTRODUCCIÓN

Hoy en día los Sistemas Informáticos están tomando con mucha fuerza un rol fundamental en el ámbito empresarial, ya que éstos proporcionan eficiencia y apoyo a todos los procesos de negocio de la empresa, teniendo con esto, un mayor control de la información que se maneja a diario, gestionando de manera rápida y segura esta misma.

Otra característica o beneficio que entrega el poseer un Sistema informático dentro de la empresa es el ahorro del tiempo que proporciona, esto porque cada proceso de la empresa se logra realizar de manera automatizada en un periodo de tiempo más corto, lo que conlleva un ahorro de recursos y dinero.

Se quiere implementar un software que apoye y mejore la calidad de una microempresa familiar llamada Camping Cucao la cual es administrada por su dueño Gorki Vila, esta empresa se desenvuelve en el rubro del turismo prestando servicio de alojamiento temporal como camping y posada, desarrollando el turismo específicamente en la localidad de Cucao frente al Parque Nacional de Chiloé, además de esto, se presta el servicio de caminata donde se llevan a los turistas que solicitan este servicio a lugares de atractivo en Cucao.

Para mejorar la gestión de empresa y proporcionar mayor fluidez en su funcionamiento se creará un sistema informático de administración, el cual proporcione y utilice información necesaria para poder realizar los movimientos diarios que se realizan en una temporada de funcionamiento, de esta forma poder entregar información oportuna a cada funcionario y principalmente a su administrador y clientes.

Las funcionalidades que se implementarán serán la entrega de comprobantes con la información de cada servicio solicitado por los clientes, las consultas rápidas pudiendo así guardar documentos en formato PDF o Excel dependiendo de la consulta y poder generar consulta de ingresos de dinero respecto a distintas fechas que se estimen convenientes por el usuario.

Cabe destacar que este sistema estará protegido por Usuario y Contraseña con dos niveles de acceso distintos, uno para el administrador del sistema y del camping a su vez, y otro para las personas que trabajan a diario y solamente necesitan tener acceso a las principales funciones para prestar servicios a los clientes.

CAPÍTULO I:

ASPECTOS RELEVANTES DEL DISEÑO LÓGICO

1. ASPECTOS RELEVANTES DEL DISEÑO LÓGICO

1.1 DESCRIPCIÓN DE LA ORGANIZACIÓN

Camping Cucao es una micro empresa familiar con giro de Otros tipos de hospedaje temporal como camping, albergues, posadas, refugios y similares, junto con prestar servicios de turismo como caminatas guiadas y servicio de información sobre puntos interesantes a visitar para los turistas solo a modo de información turística general.

Su dueño y administrador es Don Gorki Vila Arrué que trabaja en conjunto a 3 personas más, que son las personas que realizan las caminatas guiadas y otras funciones, el camping lleva funcionando cerca de 10 años, está ubicado en Cucao, Chiloé, Región de los Lagos, como referencia, frente al Parque nacional de Chiloé.

Cuenta con un terreno correspondiente a media hectárea en la cual hay 25 sitios que pueden ser utilizados por pasajeros que hayan hecho una reserva vía telefónica preferentemente, pero también se admiten nuevos ingresos, dependiendo de la disponibilidad de los sitios.

Se atiende y reciben personas de las 8:00 AM hasta las 22:00 PM. Los cobros son recibidos durante todo el día, pero al finalizar el turno de recepción de personas, se pasa por cada sitio cobrando a quienes no realizaron el pago, este también incluye derecho a duchas con agua caliente sin costo adicional, en horarios de funcionamiento fijos, desde las 8:00 AM a las 10:30 AM.

Cada día se cuenta con 3 excursiones diarias a Parque Nacional de Chiloé, Muelle de las Almas, Puente Bote y playas, las cuales parten desde las 11:00 AM y se pueden hacer inscripciones el día anterior o el mismo día hasta 10 minutos antes de comenzar, el recorrido. Cada excursión debe contar con un mínimo de 6 pasajeros por recorrido.

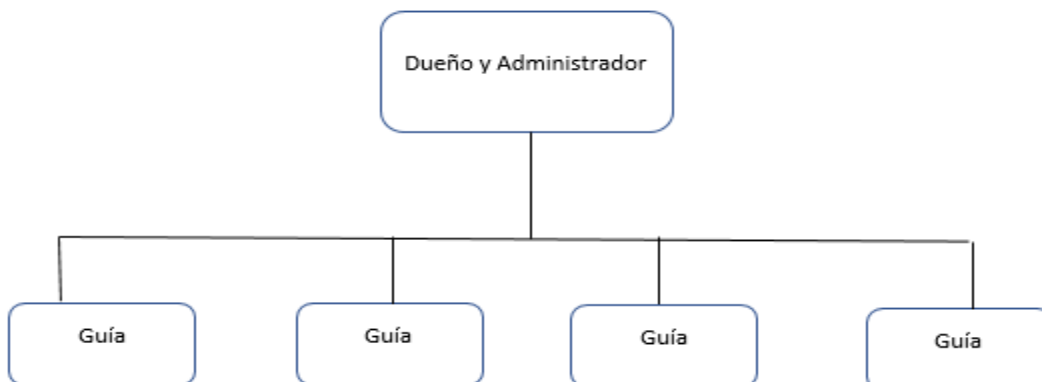


Figura 1.1. Organigrama actual de la organización.

1.2 DESCRIPCIÓN DE LA SITUACIÓN ACTUAL

En la actualidad, este camping no cuenta con ningún sistema computacional. Es decir, que no posee ningún sistema que le permita gestionar las reservas, arriendos de sitios de camping y tampoco tener un orden respecto a los cobros, ingresos y salidas de dinero que se realizan diariamente. Toda reserva y transacción de dinero es registrada en cuaderno escrito a lápiz sin ningún tipo de respaldo o forma de poder hacer consultas rápidas. Cabe destacar que este mismo libro es la base para que al final de temporada se regularicen los libros diarios que se usan para la contabilidad que se le exige como microempresa.

No hay una completa noción de qué sitios están ocupados, por lo que se dan problemas con la cantidad de sitios disponibles, arrendados y las reservas. En ciertas fechas de alta demanda, como lo es la fiesta de la luna que se realiza en las últimas semanas de verano, suelen tener problemas con las reservas, ya que en su mayoría coinciden por un sitio, en la misma fecha por más de una persona.

Bajo las circunstancias antes señaladas, se debe acudir al cuaderno, el cual lleva los registros de cada reserva y a su vez se debe percatar que efectivamente hubo un problema con estas, para solucionar esto solo se habilitan sitios provisorios por una noche o simplemente se conversa con las personas que generalmente no les complica compartir el sitio, ya que el pago es por persona. Además, esto genera el ambiente que el dueño busca, compartir espacios y conocer gente.

Por otro lado, el cobro de la estadía de los pasajeros se realiza con el cuaderno de registros, buscando el nombre que cada cliente que se proporcionó a la hora de hacer el Check-in, sin tener un filtro o sistema de búsqueda, el cobro de todas las personas en caso de estar todos los sitios ocupados tarda alrededor de una hora y en ocasiones un poco más.

Para la implementación de su servicio turístico, se registra de forma manual a los pasajeros que desean realizar la visita. En muchos casos suelen haber confusiones por quienes son pasajeros del camping y quienes vienen de otros campings a solicitar solamente el servicio de caminata, ya que esto también es posible para una persona que venga solo por este servicio.

1.3 PROBLEMAS DETECTADOS

A la empresa se le desarrollará un sistema de administración para el control de toda la información de negocio que se maneja. Actualmente se poseen grandes impedimentos tanto para la información de los clientes turísticos, como para el personal de la empresa que brinda estos servicios por temporada.

Los problemas detectados son:

- Todas las reservas realizadas quedan registradas en un cuaderno junto con el cobro diario del arriendo correspondiente, teniendo problemas en caso de atraso de un pago o de cobros indebidos en caso de registro erróneo.
- Tiempos de consulta o de búsqueda de datos extensos debido a la gran cantidad de información almacenada a mano en un cuaderno junto con los demás datos del arriendo.
- Se pierde el registro de lugares o sitios actualmente ocupados por los clientes, debido a fechas que no son registradas, ya que no se tiene un formato estándar.
- Problemas al modificar en el registro, la zona turística que desea visitar el pasajero, (caminata guiada).
- Problemas con la coordinación de los guías que realizan las caminatas.
- Problemas al identificar el tipo de servicio solicitado por el cliente (estadía y/o caminata guiada).
- Se presentan pérdidas de ganancias al calcular manualmente el dinero recaudado.
- Se observa la necesidad de darle una puntuación de valoración a cada destino de las caminatas para ver si siguen durante la temporada o se eligen nuevos destinos.
- Se le sugiere entregar comprobantes a los pasajeros por cada servicio que tome.
- No existen estadísticas en cuanto al aumento o disminución de turistas en comparación con fechas anteriores.
- Problemas al dar de baja un arriendo, en caso de que se informa la cancelación de éste, existe problemas al buscar la información en el cuaderno, por el nombre de la persona y la fecha en la que se había hecho el pedido en una primera instancia.

1.4 DESCRIPCIÓN DEL SISTEMA PROPUESTO

1.4.1 OBJETIVOS DEL SISTEMA PROPUESTO

Objetivo general

Posterior a la identificación de los problemas, se ha llegado a la conclusión que la empresa necesita un sistema con el objetivo de solucionar las carencias en el ámbito del control de ingresos y egresos de dinero, control personal, modificación de servicios, organizar información de clientes y trabajadores, y generar informes y comprobantes para los servicios que se entreguen.

Objetivos específicos

- Facilitar el control en los movimientos de dinero, tales como ingresos y egresos.
- Agilizar el proceso para dar de baja alguno de los servicios o modificaciones de éste.
- Reducir tiempos de consultas por parte del usuario.
- Organizar la información de manera correcta y estandarizada
- Implementar una base de datos, a la cual se acceda de manera sencilla por parte del administrador al realizar consultas.
- Tener un manejo de la información personal de cada guía que trabaja en el camping.
- Control de la información personal de cada pasajero que ingresa y se hospeda en el camping.
- Proporcionar información tangible al cliente mediante comprobantes de servicios prestados.

1.4.2 BENEFICIOS DEL SISTEMA PROPUESTO

- Mejoras en los tiempos de recepción de clientes.
- Control de información del personal que trabaja en las temporadas de funcionamiento.
- Información precisa de los clientes y los servicios que a éstos se les ofrece.
- Mantención de la información de los clientes que puedan requerir los servicios.
- Obtener información sobre el cliente o los servicios solicitados por ellos, de manera inmediata desde el sistema.
- Poseer un sistema informático le da a la empresa un aspecto más formal generando una mejor imagen frente a los clientes y a los competidores del rubro, lo que puede llegar a ser un factor más para que los clientes decidan preferir esta empresa en lugar de otras.
- Poder tener información de visitas, determinado por un filtro de fechas a comparar.
- Asignar servicios de Guía turística al personal, según lo solicitado por el cliente
- Entregar comprobantes tangibles a los clientes de cada servicio que se solicite a diario.

1.4.3 DESCRIPCIÓN GENERAL DE LA SOLUCIÓN PROPUESTA

El software que se implementará consiste en un sistema de administración protegido por usuario y contraseña que permite el acceso a los registros de información mediante una interfaz, con ello se podrá estandarizar el texto de todos los registros ingresados y se podrá manipular de mejor manera todas las transacciones, manteniendo la información segura y de uso exclusivo del administrador del sistema.

Dicho lo anterior se mostrará una interfaz al usuario que le permitirá llevar diariamente una contabilidad, mediante un libro diario que podrá consultar, que respaldará sus ingresos diarios de dinero, ya sean los percibidos por zonas de arriendo o caminatas guiadas.

Se tendrá acceso a los servicios actuales prestados a los clientes en caso de que éstos soliciten una consulta, se podrá modificar zonas de arriendo, también de destinos de caminatas en caso de que el dueño lo solicite mediante el administrador.

Se llevará un registro de trabajadores (guías de caminatas) y del servicio solicitado por el cliente (arriendo o caminata guiada), junto a su zona correspondiente sin dar a opción de deuda, ya que cada solicitud genera un pago inmediato, entregando un comprobante instantáneo al cliente.

Por último, el sistema emitirá informes históricos por temporadas de todos los servicios solicitados, junto con las ganancias que generaron y un historial de los clientes que visitaron el camping, con esto se podrá analizar por parte del dueño del camping las virtudes de los servicios prestados o la decadencia en alguno de los servicios.

1.4.4 DIAGRAMA DE FLUJO ADMINISTRATIVO

Solicitud de arriendo o reserva de zona de camping.

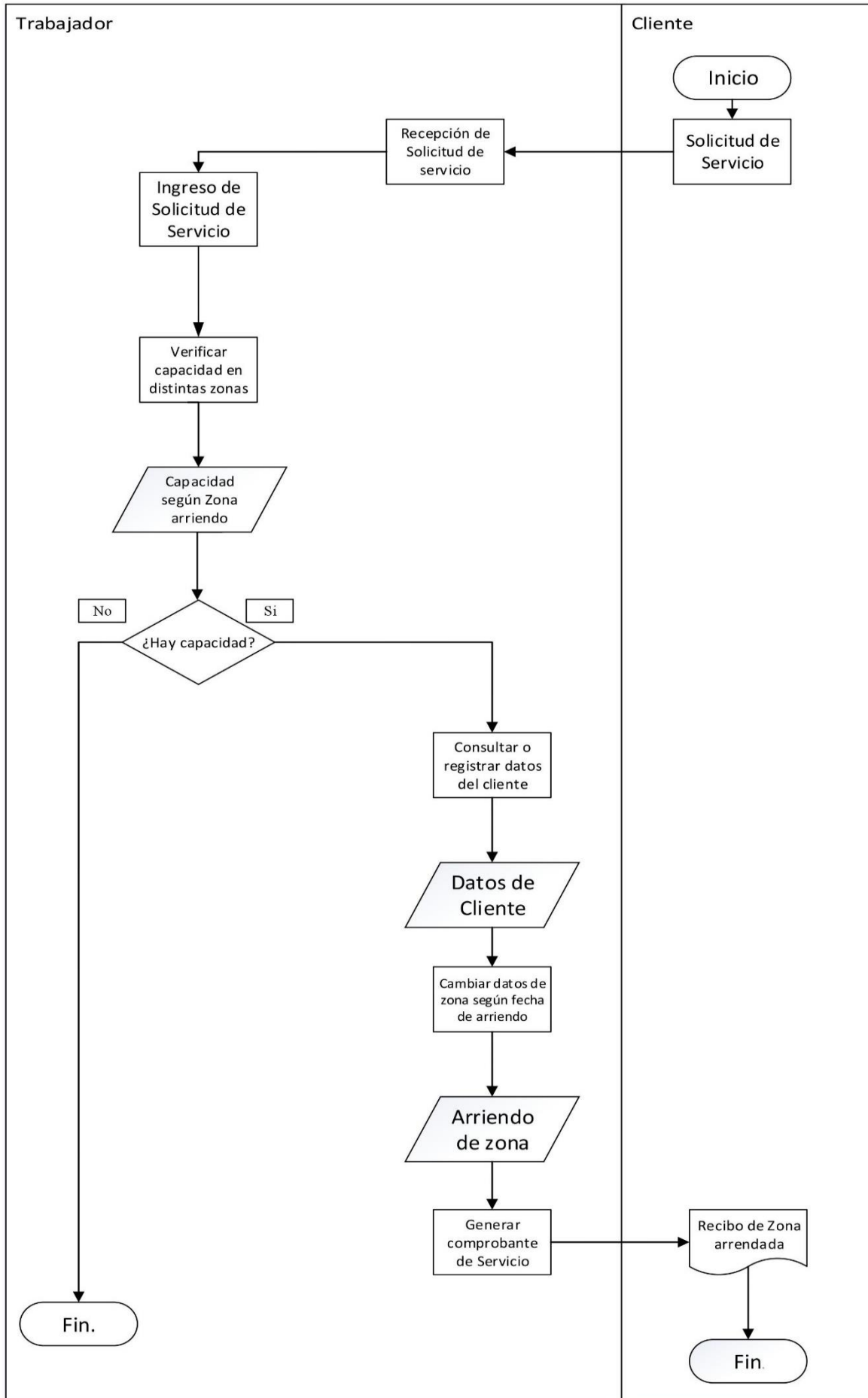


Figura 1.2. Diagrama de Flujo Administrativo

Solicitud del servicio de caminata guiada

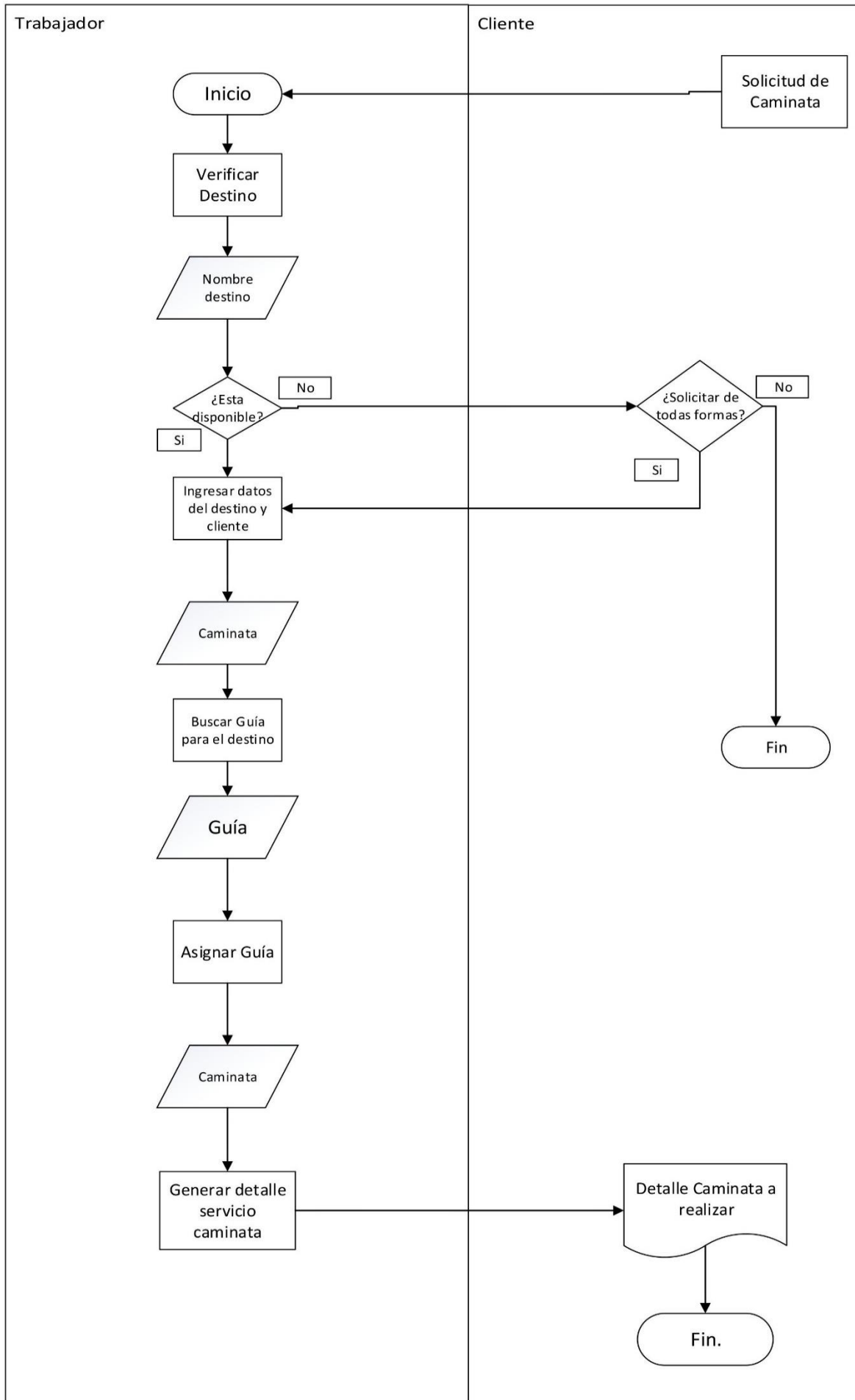


Figura 1.3. Diagrama de Flujo Administrativo

1.4.5 ESTRUCTURA FUNCIONAL DEL SISTEMA

- 1 **Mantenedor de Usuarios:** Permite agregar, eliminar, modificar y consultar todos los usuarios de quienes puedan administrar y hacer uso de los mantenedores.
- 2 **Mantenedor clientes:** Permite agregar, modificar o consultar los clientes registrados en el sistema.
- 3 **Mantenedor de guías turísticas:** Permite agregar, eliminar, modificar o consultar los datos de los guías turísticos del servicio caminata.
- 4 **Mantenedor de Solicitudes:** Permite agregar, eliminar, modificar o consultar las solicitudes realizadas por los clientes registrados.
- 5 **Mantenedor de Servicios:** Permite agregar, eliminar, modificar o consultar los servicios que presta el camping.
- 6 **Mantenedor de pagos:** Permite ingresar y consultar los pagos realizados según los servicios prestados.
- 7 **Consulta Histórica:** Permite realizar una consulta en la tabla histórica donde se pueden buscar los datos según el pago realizado o el identificador del cliente.
- 8 **Informe de Visitas:** Permite generar un informe de las visitas recibidas según las fechas a criterio del usuario o administrador.
- 9 **Informe de Ingresos:** Permite generar un informe de los ingresos generadas por un servicio en particular, de una fecha estimada hasta la fecha actual.

1.4.6 DESCRIPCIÓN DE FÓRMULAS Y CÁLCULOS EMPLEADOS

Las fórmulas empleadas harán referencias a los distintos cálculos monetarios que se van a manejar de acuerdo con el nuevo sistema a implementar, éstos son:

Cálculo 1, Total Ingreso Arriendo: correspondiente al total de ingresos del servicio de arriendo.

$\text{TotalIngresoArriendo} = \text{CantPersonas} * \text{CostoArriendo}$.

- **TotalIngresoArriendo:** Es el costo total del arriendo, calculado por la multiplicación de los siguientes valores:
- **CantidadPersonas:** Hace referencia al total de personas que solicitaron el servicio
- **CostoAriendo:** Hace referencia al costo del servicio previamente definido.

Cálculo 2, Total Ingreso Caminata: correspondiente al total de ingresos del servicio de caminata:

$\text{TotalIngresoCaminata} = \text{CantidadPersonas} * \text{CostoServicio}$.

- TotalIngresoCaminata: Corresponde al total a pagar por la caminata guiada solicitada. Este valor se calcula a través de la multiplicación de:
- CantidadPersonas: Hace referencia a la cantidad de personas que asistirán a la caminata, solicitado por un sólo cliente.
- CostoServicio: Hace referencias al costo del servicio previamente definido.

Cálculo 3, Total Ingreso Mensual: correspondiente a la suma de los ingresos totales de los servicios de caminata y arriendo en el plazo de tiempo de un mes.

$\text{TotalIngresoMensual} = \text{TotalArriendoSumatoria} + \text{TotalCaminataSumatoria}$.

- TotalIngresoMensual: Es el costo total de los servicios solicitados en el camping, calculable a partir de lo siguientes valores:
- TotalArriendoSumatoria: Corresponde a las sumatorias de los TotalArriendo empleados en el mes.
- TotalCaminataSumatoria: Corresponde a las sumatorias de los TotalCaminata empleados en el mes.

Cálculo 4, Total Ingreso por Temporada: correspondiente a la suma total mensual según el periodo de temporada.

$\text{TotalIngresoTemporada} = \text{TotalMes}(1) + \text{TotalMes}(2) + \dots + \text{TotalMes}(n)$.

- TotalIngresoTemporada: Corresponde a la sumatoria de todos los servicios solicitados en la temporada, calculable por los totales de los distintos meses, los valores que emplea son:
- TotalMes: Es el costo total de los servicios solicitados en el camping, calculable a partir de las sumas de TotalArriendoSumatoria y TotalCaminataSumatoria.

1.4.7 INFORMACIÓN QUE SE MANEJARÁ

1.4.7.1 SALIDAS

- **Lista de clientes:** Se mostrará por pantalla una lista detallada de todos los clientes que se encuentran en el camping en el momento de emitir la lista.
- **Lista de Guías:** Se mostrará por pantalla una lista detallada de todos los Guías que están trabajando en el camping.
- **Lista de Solicitudes:** Mostrará por pantalla todas las solicitudes generadas, con la opción de poder realizar una búsqueda por filtro.
- **Informe de ganancias:** Emite un informe por pantalla de las ganancias según el tipo de servicio que se prestó en la temporada y permite en detalle una comparación con fechas anteriores.
- **Informe de visitas:** Emite un informe por pantalla del total de visitas turísticas en el lugar en comparación a temporadas pasadas.
- **Informe histórico de clientes:** Emite un informe por pantalla de los datos de todas las personas las cuales estuvieron en el camping.
- **Comprobantes de Solicitudes:** Emitirá un comprobante con opción de generar un archivo PDF con los detalles de la solicitud realizada y del pago efectuado por esta misma.

1.4.7.2 ENTRADAS

- **Datos del usuario:** El administrador del sistema ingresará por teclado los datos del usuario que tendrá acceso al sistema otorgando los permisos que le corresponden.
- **Datos del cliente:** El Administrador o Usuario ingresará por teclado los datos del cliente que requiera de los servicios que ofrece el camping.
- **Datos del guía:** El Administrador o Usuario del sistema ingresará por teclado los datos personales de cada guía que realizará las caminatas que se ofrecen en el camping.
- **Datos del servicio:** El administrador del Sistema ingresará por teclado los servicios que prestará el camping en la temporada ya que pueden variar según temporada o distintas condiciones.

- **Datos de Solicitud:** El administrador ingresará por teclado toda la información para poder generar la solicitud que realice el cliente en un momento determinado para cualquier servicio que esté prestando el camping en ese momento.
- **Datos del pago:** El Usuario del sistema ingresará por teclado los datos del pago que realizará cada pasajero o solicitante de un servicio, ya sea la estadía diaria o el servicio de caminata.

1.4.7.3 ENTIDADES DE INFORMACIÓN

Usuario: Tabla que permite registrar a los usuarios que tienen acceso al sistema y sus distintos perfiles, con los siguientes campos:

- Id_usuario
- Nombre_usuario
- Nombre_real
- Apellidos_usuario
- Contraseña
- Tipo_usuario
- Fono_usuario
- Mail_usuario

Cliente: Tabla que permite registrar a cada cliente que ingresa como pasajero al camping, con los siguientes campos:

- Id_cliente
- Nombre_cliente
- Apellidos_cliente
- Sexo
- Fono_cliente
- Dir_cliente
- Mail_cliente
- Nacionalidad

Guía: Tabla que permite registrar a cada guía que trabaja en el camping con su información personal y experiencia, con los siguientes campos:

- Rut_guia
- Nombre_guia
- Apellidos_guia
- Sexo
- Edad_guia
- Fono_guia
- Anno_exp

Servicio: Tabla que permite almacenar la información los servicios que se prestan.

- Cod_servicio
- Guia_rut_guia
- Estado_zona
- Capacidad_zona
- Capactidad_actual
- Precio_zona
- Nombre_destino
- Descripcion_destino
- Precio_caminata
- Ubicacion_zona
- Tipo_servicio

Pago: Tabla permite registrar los datos de los pagos por cada servicio solicitado.

- Cod_pago
- Solicitud_cod_solicitud
- Cliente_id_cliente
- Monto_pago
- Fecha_cobro

Solicitud: Tabla de intersección entre clientes y servicio, permitiendo registrar las solicitudes por cada cliente de los servicios disponibles.

- Cod_solicitud
- Cliente_id_cliente
- Servicio_cod_servicio
- Guia_rut_guia
- Fecha_solicitud
- Estado

Histórico: Tabla que permitirá almacenar los datos de cada usuario junto con el servicio solicitado en determinada temporada.

- Cod_historico
- Pago_cod_pago
- Cliente_id_cliente
- Nombre_cliente_h
- Apellidos_h
- Sexo_h
- Mail_cliente_h
- Fono_cliente_h
- Monto_pago_h
- Tipo_servicio_h
- Fecha_entrada
- Fecha_salida
- Nacionalidad

1.4.7.4 MODELO DE DATOS RELACIONAL

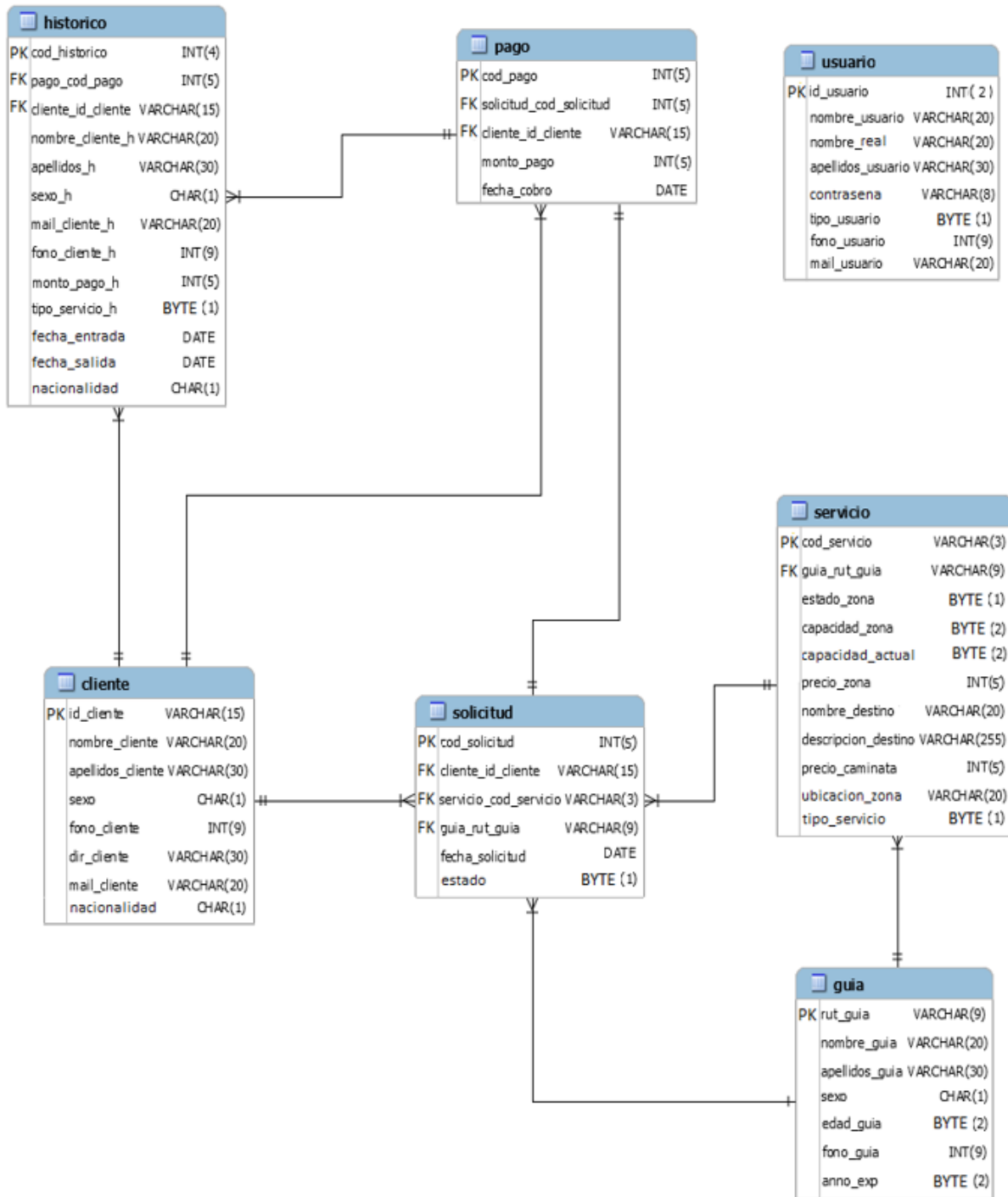


Figura 1.4. Modelo de datos relacional.

1.4.8 ESTRUCTURA DE CÓDIGOS

Id_usuario: Compuesto por nombre asociado al usuario de tipo alfanumérico.

Ejemplo:

Id_usuario hace referencia a: ADMIN, correspondiente nombre de usuario que usará para acceder al sistema

Id_cliente: Compuesto por Rut asociado al cliente o el pasaporte de tipo alfanumérico.

Ejemplo:

Id_cliente en caso de ser el Rut como identificador hace referencia a: 192451485 correspondiente al código natural representante del cliente.

Id_cliente en caso de ser el Pasaporte como identificador hace referencia a: P0300004 correspondiente al código natural representante del cliente.

Cod_servicio: Compuesto por la primera inicial del servicio más un número correlativo de tipo alfabético.

Ejemplo:

Cod_servicio en caso de ser Arriendo la codificación de carácter simple corresponde a: A01.

Cod_servicio en caso de ser Caminata Guiada la codificación de carácter simple corresponde a: C01.

Cod_pago: Compuesto por un número correlativo.

Ejemplo:

Cod_pago corresponde a la siguiente codificación simple: 01, 02 ,03, ..., n.

Cod_solicitud: Compuesto por un número correlativo.

Ejemplo:

Cod_solicitud corresponde a la siguiente codificación simple: 01, 02 ,03, ..., n.

Cod _historico: Compuesto por un número correlativo.

Ejemplo:

Cod_historico corresponde a la siguiente codificación simple: 01, 02 ,03, ..., n.

1.4.9 CONDICIONANTES DE DISEÑO

El sistema será ejecutado en Windows 10, el lenguaje de programación que se utilizará para la creación será Visual Basic ya que nos proporciona gran manejo de botones, pestañas y menú, además de tener una amplia documentación. Junto a esto se utiliza Microsoft Access para el manejo de la base de datos la cual.

Se cuenta con un notebook en el cual funcionará el sistema que cuenta con 8 GB de memoria RAM, 500gb de Disco duro y un procesador Intel Core i3.

El tiempo de respuesta de cada función del sistema, al no necesitar una conexión de internet para funcionar y por las características del hardware con el que se cuenta para el sistema, éste deberá responder prácticamente de manera casi inmediata a cada solicitud que el usuario realice.

CAPÍTULO 2:

MEDIO AMBIENTE COMPUTACIONAL Y DESCRIPCIÓN DE ARCHIVOS

2 MEDIO AMBIENTE COMPUTACIONAL Y DESCRIPCIÓN DE ARCHIVOS

2.1 INTRODUCCIÓN

Como parte fundamental del diseño es la interacción que tendrá el usuario con el sistema, ya que el sistema está dirigido a la empresa de camping, para la administración de los servicios de arriendos de zonas de camping y caminatas guiadas, además de la entrega de datos históricos para el posterior análisis.

2.1.1 CONFIGURACIÓN DEL SISTEMA

El equipamiento hardware consta de un solo computador (notebook), el cual tendrá:

- Software administrador.
- Base de datos del sistema.

Configuración del hardware:

- Procesador : Intel® Core® i3-5005U.
- Tipo de almacenamiento : Disco duro Western Digital a 5400 rpm.
- Capacidad de almacenamiento : 500 GB.
- Tipo de pantalla : Pantalla de 14" LCD notebook.
- Pantalla anexa : Pantalla 28" LED LG.
- Número de pantallas : 2.
- Tipo de impresora : HP Deskjet 4610 series.
- Número de impresoras : 1.
- Puertos : 1 puerto HDMI para conectar pantalla anexa y 4 puertos USB para dispositivos.

2.1.2 SOFTWARE UTILIZADO

Windows 10

Windows 10 es el último sistema operativo desarrollado por Microsoft, diseñado para que sea compatible con cualquier tipo de software, Windows 10 cuenta con una interfaz cómoda que permite dividir la pantalla hasta en 4 aplicaciones distintas que permiten mayor eficiencia en el trabajo y así poder agilizar la interacción en el desarrollo.

VB

Para la construcción del sistema se utilizará el lenguaje de programación VB, se ha escogido este lenguaje porque permitirá usar con facilidad la plataforma de los sistemas Windows, dado que tiene acceso a todas las librerías actuales, además es uno de los lenguajes de uso más extendido, por lo que resulta fácil encontrar información y documentación que podrá ser útil para el desarrollo.

Visual Studio

Visual Studio es un entorno de desarrollo, que permite usar una gran cantidad de herramientas para el desarrollo de aplicaciones de escritorio, Visual Studio da soporte a bastantes lenguajes de programación, como lo son C, C#, C++, VB y otros.

Nos ofrece un gestor de interfaces de usuarios y distintas funcionalidades para poder facilitar la escritura del código y la corrección de éstos mediante la sugerencia de código, control de versiones, comprobaciones de sintaxis y el explorador de proyectos que permitirá encontrar cada componente del proyecto de manera fácil, agilizando la gestión de desarrollo.

ACCESS

Access es un sistema de gestión de bases de datos incluido en el paquete ofimático denominado Microsoft Office. Este permitirá un mejor trabajo en la gestión de base de datos Microsoft Access.

2.2 DESCRIPCIÓN DE ARCHIVOS

A continuación, se señalan las tablas correspondientes al sistema de administración a implementar:

Tabla Usuario

- Nombre : Usuario
- Descripción : Datos de los usuarios que tienen acceso al sistema y sus distintos perfiles
- Clave Primaria : Id_usuario
- Clave Foránea : No posee

Descripción de los registros correspondientes:

| Nombre | Tipo | Longitud | Descripción |
|-------------------|---------|----------|--|
| Id_usuario | INT | 2 | Clave Principal autoincrementable |
| Nombre_usuario | VARCHAR | 20 | Nombre del usuario |
| Nombre_real | VARCHAR | 30 | Nombre real del usuario |
| Apellidos_usuario | VARCHAR | 30 | Apellido del usuario |
| Contraseña | VARCHAR | 8 | Contraseña establecida por el usuario al momento de registrarse en la base de datos, para su posterior acceso al sistema |
| Tipo_usuario | BYTE | 1 | Nivel de permisos del usuario al sistema (1 = ADMINISTRADOR, 2 = TRABAJADOR) |
| Fono_usuario | INT | 9 | Teléfono del usuario |
| Mail_usuario | VARCHAR | 20 | Correo del usuario |

Figura 2.1. Tabla Usuario.

Tabla Cliente

- Nombre : Cliente
- Descripción : Datos de cada cliente que ingresa como pasajero al camping.
- Clave Primaria : Id_cliente
- Clave Foránea : No posee

Descripción de los registros correspondientes:

| Nombre | Tipo | Longitud | Descripción |
|------------------|---------|----------|---|
| Id_cliente | VARCHAR | 15 | Identificador del cliente, puede ser; Rut (99999999-K) o número de pasaporte(P12341433) |
| Nombre_cliente | VARCHAR | 20 | Nombre del cliente |
| Apellido_cliente | VARCHAR | 30 | Apellido del cliente |
| Sexo | CHAR | 1 | Sexo del cliente (M/F) |
| Fono_cliente | INT | 9 | Teléfono del cliente |
| Dir_cliente | VARCHAR | 30 | Dirección del cliente |
| Mail_cliente | VARCHAR | 20 | Correo del cliente |
| Nacionalidad | CHAR | 1 | Nacionalidad del cliente, Nacional (N) o Extranjero (E) |

Figura 2.2. Tabla Cliente.

Tabla Guía

- Nombre : Guía
- Descripción : Datos de cada guía que trabaja en el camping con su información personal y experiencia.
- Clave Primaria : Rut_guia
- Clave Foránea : No posee

Descripción de los registros correspondientes:

| Nombre | Tipo | Longitud | Descripción |
|----------------|---------|----------|------------------------------|
| Rut_guia | VARCHAR | 9 | Identificador del guía (RUT) |
| Nombre_guia | VARCHAR | 20 | Nombre del guía |
| Apellidos_guia | VARCHAR | 30 | Apellidos del guía |
| Sexo | CHAR | 1 | Sexo del guía (M/F) |
| Edad_guia | BYTE | 2 | Edad del guía |
| Fono_guia | INT | 9 | Teléfono del guía |
| Anno_exp | BYTE | 2 | Años de experiencia del guía |

Figura 2.3. Tabla Guía.

Tabla Solicitud

- Nombre : Solicitud
- Descripción : Tabla de intersección entre Cliente y Servicio, el cual almacena las claves primarias de estos más su código auto incrementable.
- Clave Primaria : Cod_solicitud + Solicitud_id_cliente + Cod_servicio.
- Clave Foránea :
 - Cliente_id_cliente, correspondiente a la clave primaria de tabla Cliente.
 - Guia_rut_guia, correspondiente a la clave primaria de tabla Guía.
 - Servicio_cod_servicio, correspondiente a la clave primaria de tabla Servicio.
- Descripción de los registros correspondientes:

| Nombre | Tipo | Longitud | Descripción |
|-----------------------|---------|----------|---|
| Cod_solicitud | INT | 5 | Identificador de solicitud a través de un número auto incrementable |
| Cliente_id_cliente | VARCHAR | 15 | Clave primaria de cliente |
| Servicio_cod_servicio | VARCHAR | 3 | Clave primaria de servicio |
| Guia_rut_guia | VARCHAR | 9 | Clave primaria de Guía (RUT) |
| Fecha_solicitud | DATE | - | Corresponde a la fecha actual de la solicitud |
| Estado | BYTE | 1 | Estado correspondiente a la solicitud (X = PAGADO / P = PENDIENTE) |

Figura 2.4. Tabla Solicitud.

Tabla Servicio

- Nombre : Servicio
- Descripción : Datos de los servicios que se prestan.
- Clave Primaria : Cod_servicio
- Clave Foránea : Guia_rut_guia, correspondiente a la clave primaria de tabla Guía.
- Descripción de los registros correspondientes:

| Nombre | Tipo | Longitud | Descripción |
|---------------------|---------|----------|---|
| Cod_servicio | VARCHAR | 3 | Identificador del servicio solicitado Compuesto por la inicial del servicio más un número auto incrementable (A01 = Arriendo, C01= Caminata) |
| Guia_rut_guia | VARCHAR | 9 | Clave foránea correspondiente al guía asociado (RUT) a un destino para una caminata guiada |
| Estado_zona | BYTE | 1 | Corresponde al estado de una zona (0 = NO disponible, 1 = Disponible) |
| Capacidad_zona | BYTE | 2 | Corresponde a la capacidad de la zona definida por el administrador |
| Capacidad_actual | BYTE | 2 | Corresponde a la capacidad actual de la zona según cambie por arriendos |
| Precio_zona | INT | 5 | Valor asignado a cada zona para el servicio de arriendo a cobrar |
| Nombre_destino | VARCHAR | 20 | Nombre del destino para la caminata guiada |
| Descripcion_destino | VARCHAR | 255 | Breve descripción del destino para la caminata guiada |
| Precio_caminata | INT | 5 | Valor asignado a cada destino para el servicio de caminata guiada a cobrar |
| Ubicacion_zona | VARCHAR | 20 | Nombre del lugar de la zona de camping |
| Tipo_servicio | BYTE | 1 | Servicio que ofrece el camping (1 = ARRIENDO ,2 = CAMINATA) |

Figura 2.5. Tabla Servicios

Tabla Pago

- Nombre : Pago
- Descripción : Datos de los pagos por cada servicio solicitado.
- Clave Primaria : Cod_Pago
- Clave Foránea :
 - Solicitud_cod_solicitud, correspondiente a la clave primaria de la tabla Solicitud.
 - Cliente_id_cliente, correspondiente a la clave primaria de la tabla Cliente.
- Descripción de los registros correspondientes:

| Nombre | Tipo | Longitud | Descripción |
|-------------------------|---------|----------|--|
| Cod_pago | INT | 5 | Clave primaria de PAGO auto incrementable |
| Solicitud_cod_solicitud | INT | 5 | Identificador de la solicitud auto incrementable |
| Cliente_id_cliente | VARCHAR | 15 | Identificador del cliente, ya sea con su RUT o pasaporte |
| Monto_pago | INT | 5 | Valor del pago diario a cobrar por el servicio solicitado, no sobrepasa los 10000. |
| Fecha_cobro | DATE | - | Fecha de pago del servicio |

Figura 2.6. Tabla Pago.

Tabla Histórica

- Nombre : Histórico
- Descripción : Datos de cada usuario junto con el servicio solicitado en determinada temporada.
- Clave Primaria : Cod_historico.
- Clave Foránea :
 - Pago_cod_pago, correspondiente a la clave primaria de la tabla Pago.
 - Cliente_id_cliente, correspondiente a la clave primaria de la tabla Cliente.
- Descripción de los registros correspondientes:

| Nombre | Tipo | Longitud | Descripción |
|--------------------|---------|----------|---|
| Cod_historico | INT | 4 | Identificador compuesto por un número correlativo |
| Pago_cod_pago | INT | 5 | Clave primaria de PAGO auto incrementable |
| Cliente_id_cliente | VARCHAR | 15 | Identificador del cliente, puede ser; Rut o número de pasaporte |
| Nombre_cliente_h | VARCHAR | 20 | Nombre del cliente |
| Apellidos_h | VARCHAR | 30 | Apellido del cliente |
| Sexo_h | CHAR | 1 | Sexo del cliente (M/F) |
| Mail_cliente_h | VARCHAR | 20 | Mail del cliente |
| Fono_cliente_h | INT | 9 | Fono contacto del cliente |
| Monto_pago_h | INT | 5 | Valor del pago diario a cobrar por el servicio solicitado |
| Tipo_servicio_h | BYTE | 1 | Corresponde al tipo de servicio solicitado por el cliente |
| Fecha_entrada | DATE | - | Fecha de la solicitud del servicio por parte del cliente |

| | | | |
|--------------|------|---|--|
| Fecha_salida | DATE | - | Fecha en que se realizó el pago de la solicitud, recordar que el pago se realiza diariamente, debido a esto se considera fecha de salida como el ultimo día de pago. |
| Nacionalidad | BYTE | 1 | Corresponde a nacionalidad del cliente, Nacional(N)o Extranjero(E) |

Figura 2.7. Tabla Histórica.

CAPÍTULO 3:

DESCRIPCIÓN DE PROGRAMAS

3. DESCRIPCIÓN DE PROGRAMAS

3.1 DIAGRAMA MODULAR

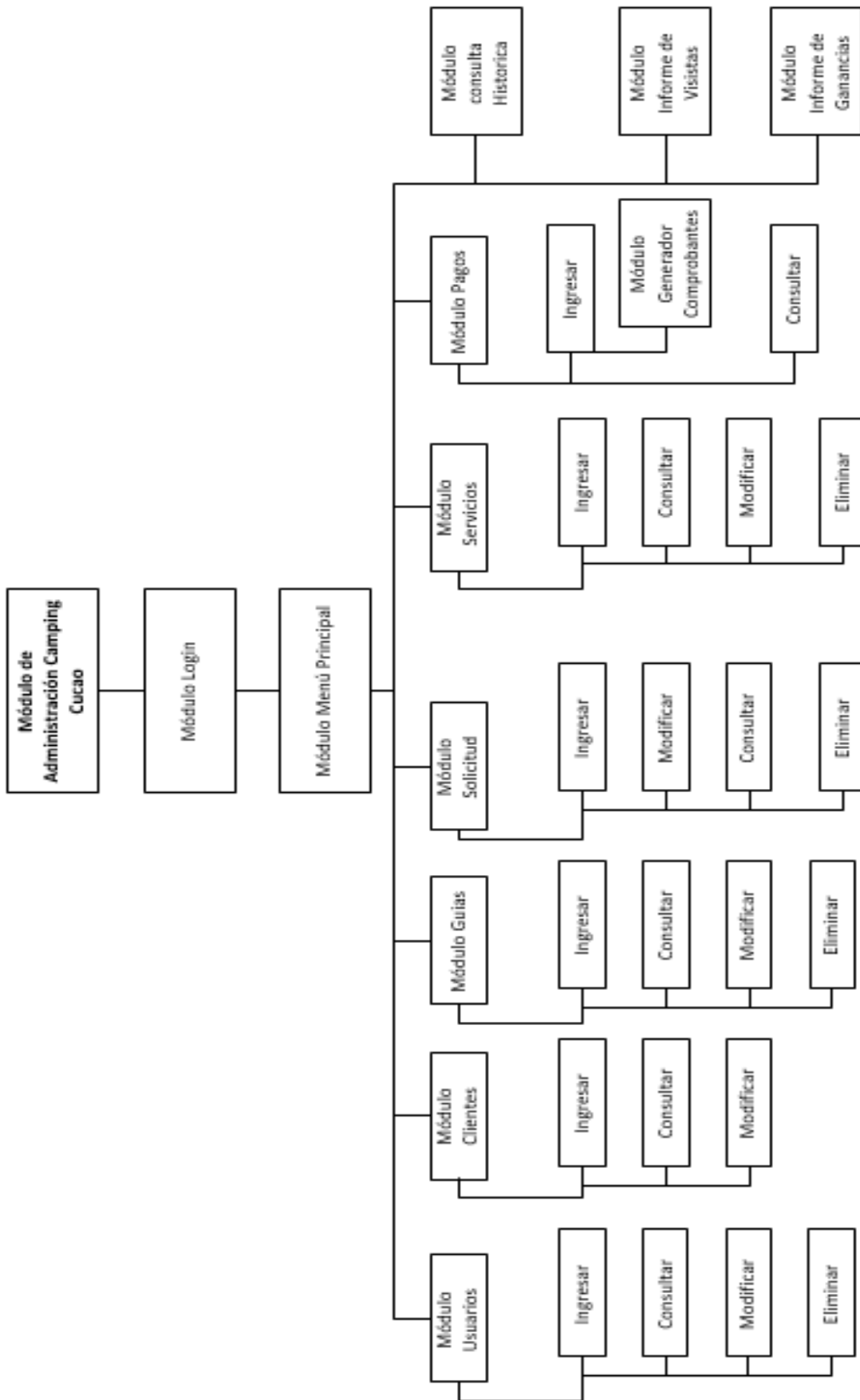


Figura 3.1 Diagrama de modular

3.2 DIAGRAMA DE MENÚS

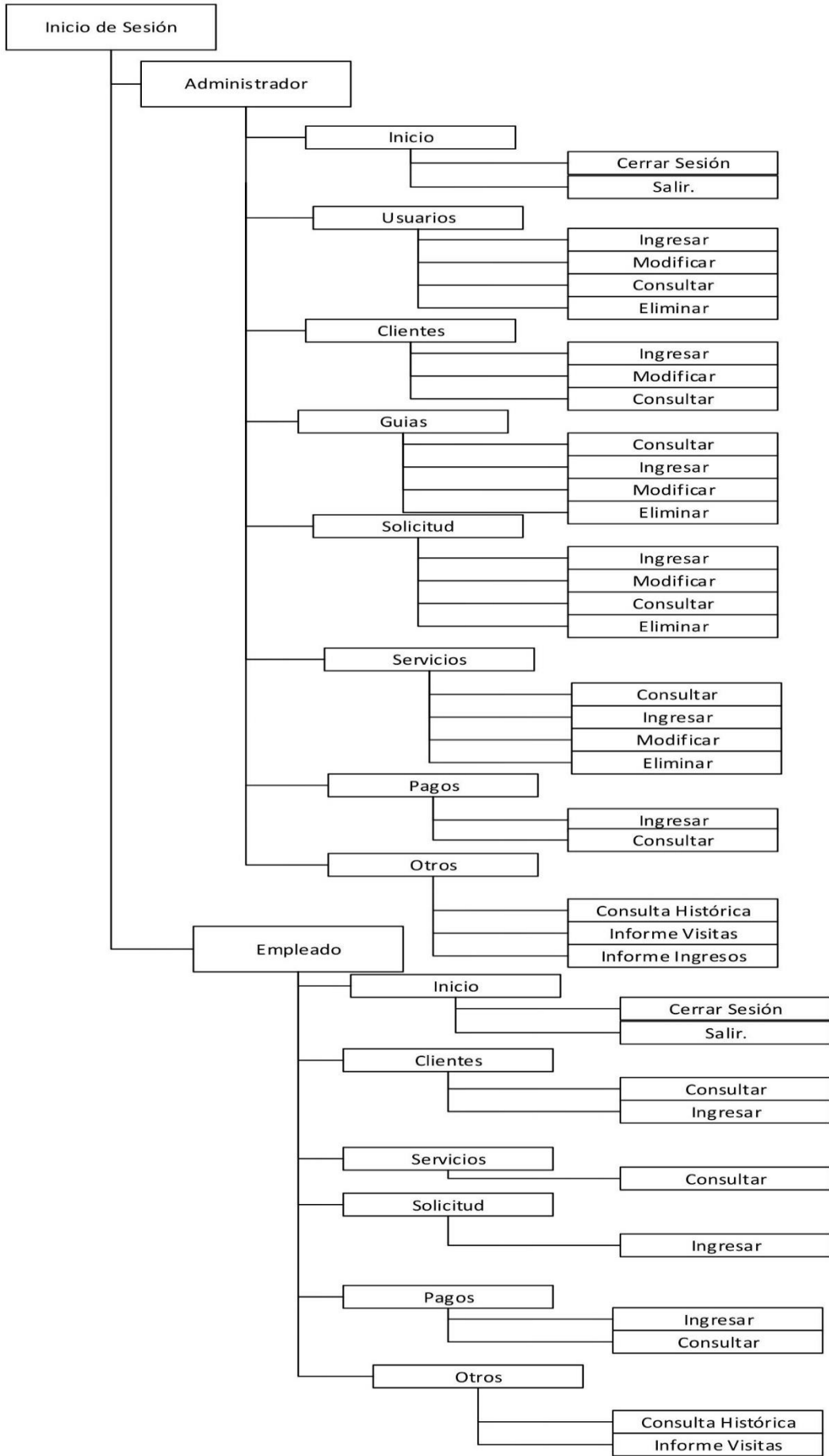


Figura 3.2 Diagrama de menú

3.3 LISTA DE PROGRAMAS

A continuación, se muestra la lista de programas que se implementaron en el sistema, donde se señala con (*) los cuales posteriormente serán descritos junto a su diagrama de flujo.

| | NOMBRE DEL PROGRAMA | OBJETIVO |
|----|----------------------------|--|
| 1 | Inicio de Sesión (*) | Permite al usuario registrado tener acceso al sistema |
| 2 | Menú Principal (*) | Muestra la pantalla de inicio con el menú de todas las funcionalidades del sistema. |
| 3 | Listar Usuario | Muestra un listado de todos los usuarios ya registrados en el sistema |
| 4 | Agregar Usuario | Permite agregar nuevos usuarios al sistema |
| 5 | Modificar Usuario (*) | Permite modificar los datos del usuario |
| 6 | Eliminar Usuario | Permite eliminar usuarios ya registrados |
| 7 | Listar Cliente | Muestra un listado de los clientes registrados en el sistema |
| 8 | Agregar Cliente (*) | Agrega cliente nuevo con sus datos en el sistema |
| 9 | Modificar Cliente | Modifica datos de un cliente ya existente en el sistema |
| 10 | Eliminar Cliente | Elimina un cliente del sistema |
| 11 | Listar Guía | Muestra un listado de los Guías registrados en el sistema |
| 12 | Agregar Guía | Agrega un nuevo Guía al sistema |
| 13 | Modificar Guía | Modifica datos de un Guía ya registrado |
| 14 | Eliminar Guía (*) | Elimina un Guía del sistema |
| 15 | Listar Servicio | Lista los servicios con los que cuenta el Camping para prestar a los clientes |
| 16 | Agregar Servicio | Agrega al sistema un nuevo servicio disponible para ser solicitado |
| 17 | Modificar Servicio | Modifica la información de un servicio |
| 18 | Eliminar Servicio | Eliminar del sistema un servicio registrado |
| 19 | Listar Solicitud | Muestra una lista de todas las solicitudes realizadas por los clientes |
| 20 | Agregar Solicitud (*) | Permite agregar una nueva solicitud de servicio asociada a un cliente. |
| 21 | Modificar Solicitud | Modifica una solicitud realizada. |
| 22 | Eliminar Solicitud | Elimina una solicitud realizada. |
| 23 | Listar Pago | Lista los pagos realizados asociados a una solicitud y un cliente. |
| 24 | Agregar Pago (*) | Permite agregar un pago de una solicitud pendiente por pagar. |
| 25 | Generar Informe | Genera un informe de toda la información rescatada por el sistema en comparación a dos fechas. |

| | | |
|----|----------------------|---|
| 26 | Guardar Estadísticas | Genera un informe de estadísticas relevantes en comparación a dos fechas. |
| 27 | Cerrar Sesión | Permite cerrar la sesión actual del sistema, pudiendo así poder cambiar el usuario. |
| 28 | Reiniciar Capacidad | Reinicia la capacidad de las zonas de camping mientras no esté siendo ocupada |

Figura 3.3 Lista de programas

3.4 DESCRIPCIÓN DE PROGRAMAS

Inicio de sesión

- Nombre: Login
- Objetivo: Otorgar seguridad al sistema restringiendo acceso mediante un usuario y Password.
- Diagrama de bloques:

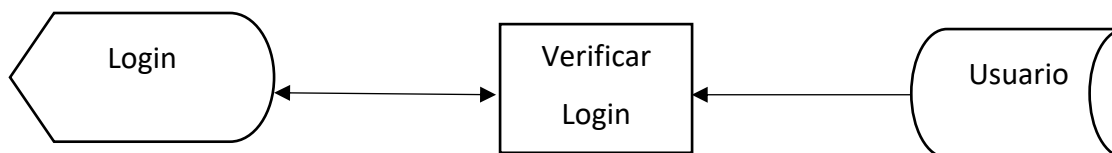
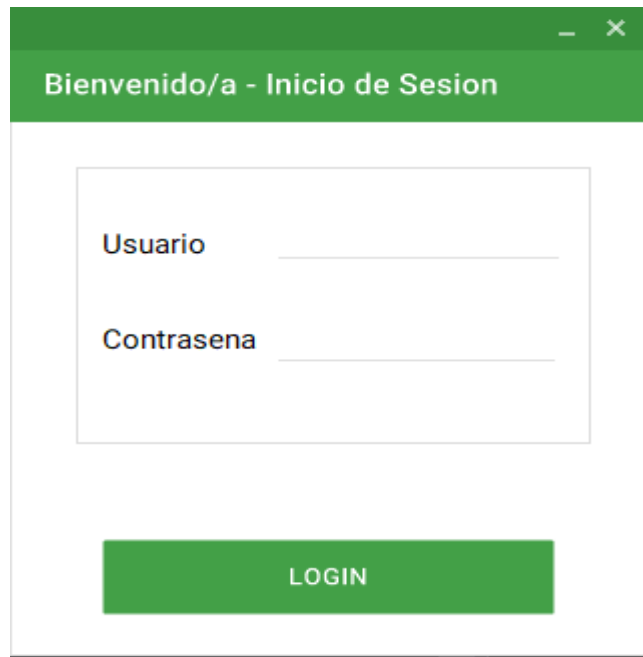


Figura 3.4 Diagrama de bloque Inicio de Sesión

- Reglas del proceso: Se piden los datos del usuario y contraseña de éste, si los datos están correctos se da acceso al menú principal, en caso contrario se despliega una alerta por pantalla señalando que información es correcta.
- Código Fuente: Anexo 1, Página 43

- Diseño de pantalla:



The image shows a login window with a green header bar containing the text "Bienvenido/a - Inicio de Sesion" and standard window control icons (minimize, maximize, close). Below the header is a white rectangular area containing two input fields. The first field is labeled "Usuario" and the second is labeled "Contraseña". Below these fields is a green button with the text "LOGIN" in white capital letters.

Figura 3.5 Pantalla Login.

Menú principal

- Nombre: Inicio del Programa
- Objetivo: Navegar por los distintos formularios mediante una barra de menú en el costado izquierdo
- Diagrama de bloques:

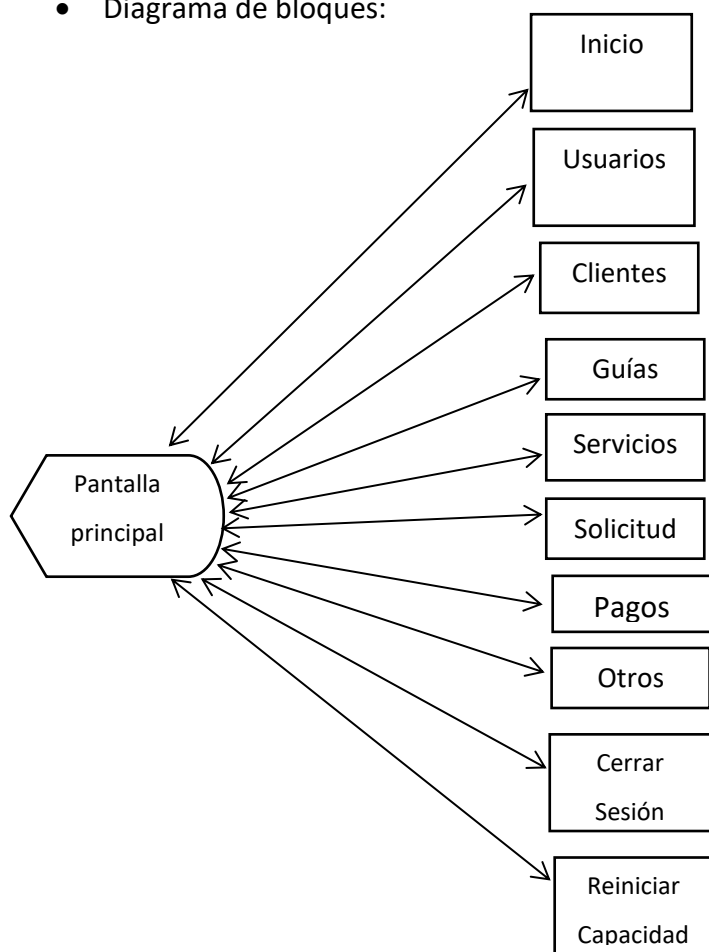


Figura 3.6 Diagrama de bloques Pantalla de Inicio del programa

- Reglas del proceso: Al validar el acceso del usuario mediante el Login se mostrará el menú principal el cual contiene distintos formularios a su disposición e informes para generar.
- Código Fuente: Anexo 2, Página 43

- Diseño de pantalla:



Figura 3.7 Diseño pantalla Principal.

Modificar Usuario

- Nombre: Modificar Usuario
- Objetivo: Modificar los datos de un usuario ya registrado en el sistema
- Diagrama de bloques:

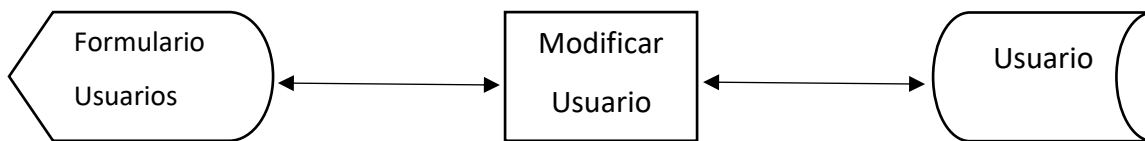


Figura 3.12 Diagrama de bloque modificar Usuario.

- Reglas del proceso: Al presionar el botón usuario, se desplegará el formulario completo del mantenedor de Usuarios. Para poder modificar un usuario se debe seleccionar un Usuario desde la lista que se muestra en la pantalla. Posterior a seleccionarlo automáticamente se llenarán los campos con la información del usuario que se seleccionó, para modificarlo se deben cambiar las cajas de textos o Combo box correspondientes, dependiendo de lo que se desea modificar. Para realizar la modificación simplemente se debe hacer clic en el botón “Modificar”.

Para que se realice la modificación se valida previamente que los datos ingresados sean correctos, posterior a esto se valida que la contraseña ingresada coincida en ambos campos en caso de que hayan sido cambiadas.

En caso de pasar las validaciones se pedirá la contraseña del usuario que está realizando el cambio como método de seguridad. Si esto es correcto se procede a realizar la modificación y se muestra un mensaje emergente con un mensaje de éxito.

- Código Fuente: Anexo 5, Página 50
- Diseño de pantalla:

Panel de Control

| ID | Nombre Usuario | Nombre | Apellidos | Fono | Mail | Nivel |
|----|----------------|----------|-----------------|-----------|---------------|-------|
| 11 | admin | Gorki | Vila Arue | 987645236 | contacto@c... | 1 |
| 16 | Guia1 | Bastian | Perez Figueroa | 988899881 | guia1@camp... | 2 |
| 22 | Guia2 | Mauricio | Figueroa Nahuel | 988899881 | guia2@camp... | 2 |
| 23 | Guia3 | Fabian | Vila Madrid | 669995544 | guia3@camp... | 2 |

Datos usuario

Usuario: Nombre:

Contraseña: Apellidos:

Repita contraseña: Fono:

E-mail:

LIMPIAR CAMPOS

Nombre a buscar: ACTUALIZAR LISTA

AGREGAR MODIFICAR ELIMINAR

Fecha de hoy 09-01-2018

* Nivel 1 corresponde a Administradores
Nivel 2 corresponde a Trabajadores

Figura 3.13 Diseño de pantalla Modificar Usuario.

Agregar Cliente

- Nombre: Agregar Cliente
- Objetivo: Agregar nuevos clientes que ingresen a solicitar un servicio
- Diagrama de bloques:

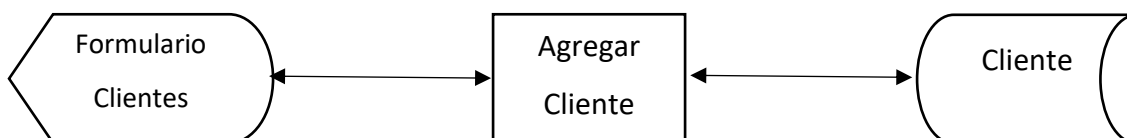


Figura 3.8 Diagrama de bloque Agrega Cliente.

- Reglas del proceso: Al presionar el botón clientes se mostrará el formulario de clientes, luego de presionar el botón agregar se habilitarán las cajas de texto para poder ingresar la información del cliente que se desea ingresar, validando el Rut del cliente, el formato de correo electrónico proporcionado y a su vez que el cliente no exista previamente dentro del sistema mediante su clave en el sistema que es su Rut o Pasaporte. Todos los campos deben estar completos, de lo contrario el programa mostrará un mensaje de error sin poder ingresar el cliente al sistema, finalmente si aprueba las validaciones del sistema se procede a guardar en la base de datos como un nuevo cliente.

- Código Fuente: Anexo3, Página 51
- Diseño de Pantalla

The screenshot shows a software window titled 'FmCliente'. On the right side, there is a 'Datos cliente' form with the following elements:

- Radio buttons for 'RUT' and 'Pasaporte'.
- Input fields for 'Identificacion' and 'Fono'.
- Input fields for 'Nombre' and 'Direccion'.
- Input fields for 'Apellidos' and 'E-mail'.
- Radio buttons for 'Hombre' and 'Mujer', and a dropdown menu.
- A 'LIMPIAR CAMPOS' button.

At the bottom left, there is a search section with 'Seleccione metodo de busqueda' and radio buttons for 'Identificacion' and 'Nombre', followed by an 'Identificacion' input field and an 'ACTUALIZAR LISTA' button. At the bottom right, there are three buttons: 'AGREGAR', 'MODIFICAR', and 'ELIMINAR'.

Figura 3.9 Diseño de pantalla Agrega Cliente.

Eliminar Guía

- Nombre: Eliminar Guía
- Objetivo: Eliminar del sistema un Guía que ya se encuentra registrado en el camping
- Diagrama de Bloques:

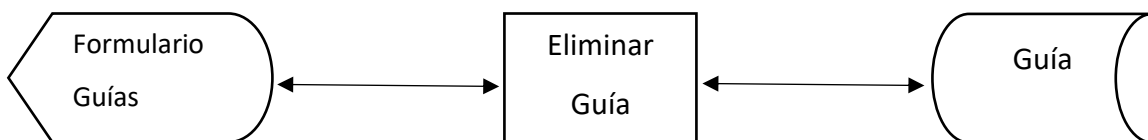


Figura 3.14 Diagrama de bloque Eliminar Guía.

- Reglas del proceso: Se procede a seleccionar un guía desde la lista que se muestra en el formulario, pudiendo a su vez realizar una búsqueda por filtro mediante la identificación del guía en este caso el Rut, luego de haber seleccionado el guía se llenarán los campos con la información, se presiona el botón "Eliminar", posterior a esto se pregunta si realmente desea eliminar el Guía, dependiendo de la respuesta que se reciba mediante las opciones Aceptar o Cancelar, se eliminara el Guía o de lo contrario se cancela la operación.
- Código Fuente: Anexo 6, Página 53

- Diseño de Pantalla:

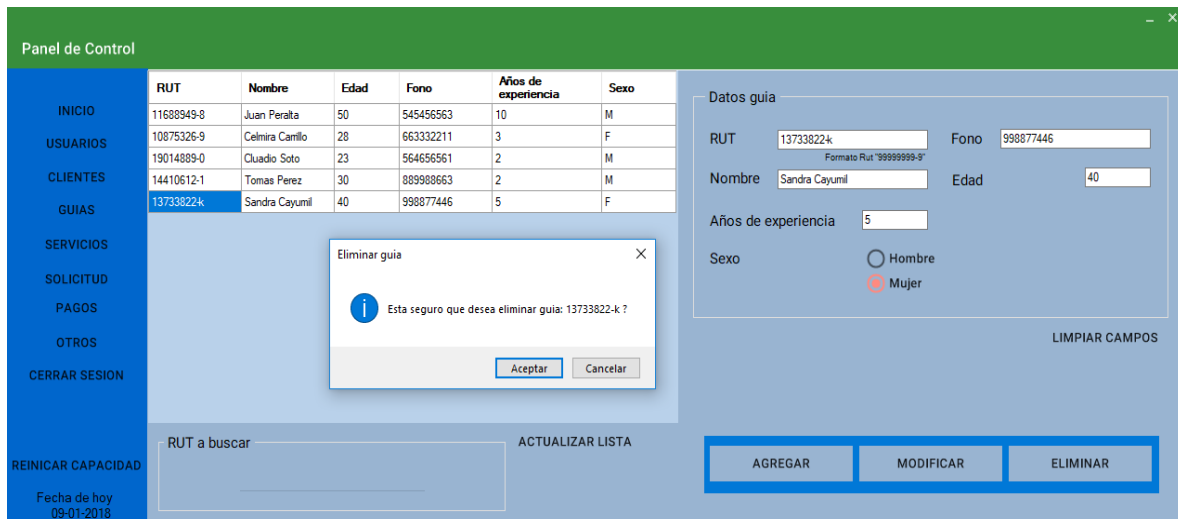


Figura 3.15 Diseño de pantalla Eliminar Guía.

Agregar Solicitud

- Nombre: Agregar Solicitud
- Objetivo: Poder generar una solicitud asociada a un Cliente y un servicio especifico
- Diagrama de bloques:

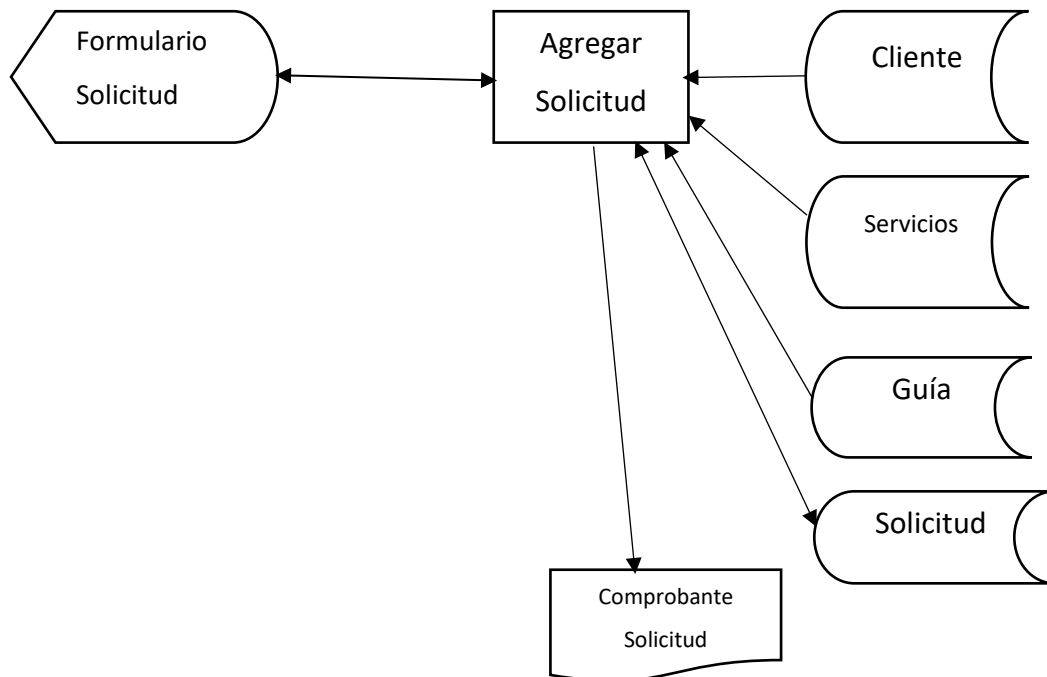


Figura 3.10 Diagrama de Bloques Agregar solicitud.

- Reglas del proceso: Al ingresar al formulario de Solicitud, se despliega el DataGridView de todas las solicitudes existentes, para agregar una nueva debe presionar el botón “Seleccione un cliente” al cual se le asociará la solicitud que se ingresará, posterior a esto se llena en el costado la información del Cliente que se seleccionó y se ingresa la fecha de la solicitud y el tipo de servicio que solicitará, ya sea caminata o arriendo, en cualquiera de los dos casos habrá una lista con el nombre de los destinos disponibles para realizar la caminata o el nombre de la zona de camping que se podrá arrendar.

Al seleccionar Caminata se le asociará automáticamente un Guía que realiza la Caminata, en caso de seleccionar un arriendo de zona de camping se asociará un Rut único que corresponderá al encargado de los arriendos. Luego de todo el proceso se presiona el botón “Agregar” y mostrará un mensaje emergente donde mostrará la fecha, el servicio y el cliente al cual fue asociada la solicitud ingresada, en caso de ingresar dos veces la misma solicitud con los mismos datos, se muestra un mensaje de error mostrando que la solicitud ya existe. Por último, se muestra la solicitud en la lista de solicitudes ingresadas y se genera un documento en formato PDF con la información completa del servicio solicitado.

- Código Fuente: Anexo 4, Página 54
- Diseño de Pantalla:

| Código solicitud | ID cliente | Guia asignado | Fecha solicitud |
|------------------|------------|---------------|-----------------|
| 171 | 9417095-8 | 000000000 | 09-01-2018 |
| 172 | 5626904-5 | 10875326-9 | 09-01-2018 |
| 173 | 16635480-3 | 10875326-9 | 09-01-2018 |
| 174 | 19192252-2 | 13733822k | 09-01-2018 |
| 175 | 13733822k | 19014889-0 | 09-01-2018 |

Panel de Control

Asignacion servicio

CLICK PARA SELECCIONAR CLIENTE

Fecha: martes, 9 de enero de 2011

Servicio: Arriendo Caminata

Destino Zona: zona uno

Datos del cliente

ID: Seleccione Cliente

Nombre: Seleccione Cliente

Apellidos: Seleccione Cliente

Guia

AGREGAR MODIFICAR ELIMINAR

Fecha de hoy: 09-01-2018

*En caso de arriendo el guia estara representado con RUT 0

Figura 3.11 Diseño de pantalla Agrega Solicitud.

Agregar Pago

- Nombre: Agregar Pago
- Objetivo: Poder agregar un pago asociado a una solicitud con estado pendiente, ya que no ha sido pagada.
- Diagrama de Bloques:

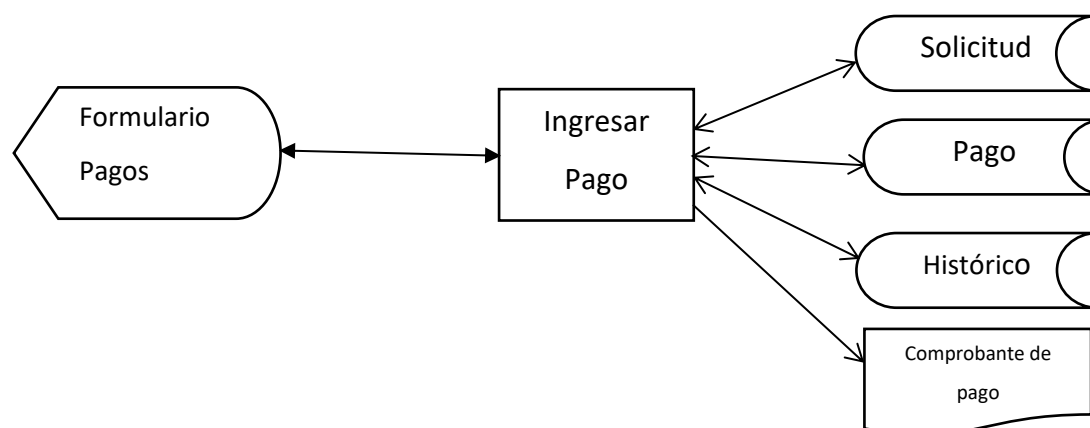


Figura 3.16 Diagrama de bloque Agregar pago

- Reglas del proceso: Se procede a ingresar al formulario de Pagos, luego para ingresar un pago debe hacer clic en el botón “Clic para seleccionar solicitud”, luego de esto se abrirá un listado emergente en el cual se puede realizar una búsqueda por filtro, de todas las solicitudes en estado “pendiente” a las cuales se le puede realizar el pago. Luego de seleccionarla se llena la información en campos correspondientes con la ID del Cliente, el código del servicio solicitado y el Guía asociado.

Cabe destacar que en caso de que aparezca “No corresponde” quiere decir que el Servicio es un arriendo ya que en estricto rigor no tiene un Guía asociado. A su vez se muestra el monto a pagar y la fecha que son campos no editables. De esta forma solo quedará realizar el pago mediante el botón “Agregar” y mostrará un mensaje que confirma la operación desplegando “Pago ingresado correctamente. Comprobante generado con éxito”.

- Código Fuente: Anexo 7, Página 54

- Diseño de pantalla:

Panel de Control

| Codigo pago | Codigo solicitud | ID cliente | Monto | Fecha cobro |
|-------------|------------------|------------|-------|-------------|
| 44 | 171 | 9417095-8 | 4500 | 01-09-2018 |
| 45 | 173 | 16635480-3 | 4500 | 01-09-2018 |
| 46 | 175 | 137338224 | 1500 | 01-09-2018 |
| 47 | 174 | 19192252-2 | 1500 | 01-09-2018 |
| 48 | 172 | 5626904-5 | 4500 | 01-09-2018 |

INICIO
 USUARIOS
 CLIENTES
 GUIAS
 SERVICIOS
 SOLICITUD
 PAGOS
 OTROS
 CERRAR SESION
 REINICAR CAPACIDAD
 Fecha de hoy 09-01-2018

Ingresar valores
 Primero debe seleccionar una solicitud pendiente
 Aceptar

Datos solicitud
 CLICK PARA SELECCIONAR SOLICITUD
 ID Seleccione SOLICITUD
 ID servicio Seleccione SOLICITUD
 Guia Seleccione SOLICITUD

Pago
 Fecha martes, 9 de enero de
 Monto

Seleccione metodo de busqueda (Seleccione)
 Busqueda ACTUALIZAR LISTA
 AGREGAR

Figura 3.17 Diseño de pantalla Agregar Pago.

CONCLUSIONES

Para poder realizar este trabajo se necesitaron aproximadamente nueve meses desde que comenzó el análisis hasta finalizar con la construcción total del software, cabe destacar que para poder llegar a la construcción del software fue necesario interiorizarse en el lenguaje de desarrollo en esta caso Visual Basic.NET, además de investigar como poder confeccionar documentos PDF y EXCEL desde los formularios del sistema, además, de investigar cada error que se producía mediante la implementación de distintos módulos, esto fue investigado durante la construcción con documentación de foros y archivos de Microsoft orientada al desarrollo.

La influencia de la formación profesional adquirida para enfrentar el desarrollo de un software desde cero es primordial, ya que el desarrollo puede realizarse mediante cualquier lenguaje de programación, mientras cumpla los requisitos para el desarrollo. Pero cabe destacar que en la formación profesional es imprescindible el adquirir los conocimientos necesarios en la lógica de programación y en el análisis de requerimientos, ya que un mal análisis llevará a un software con errores y quizás implementaciones innecesarias.

El sistema entrará en funcionamiento para el verano de 2018 en marcha blanca, mientras funciona el sistema antiguo que es manual mediante un cuaderno, para poder con esto encontrar fallas y poder arreglarlo para su posterior corrección en caso de que existan.

En un futuro este proyecto puede ampliarse a muchas más funciones si es necesario ya que el Camping tiene una proyección muy grande, en un futuro se espera poder construir cabañas, contratar más personal y poder prestar más servicios, esto por el crecimiento año tras año del turismo en la zona.

Para finalizar podemos decir que este proyecto nos hizo interiorizarnos de buena forma en el trabajo de problemáticas reales, necesidades y problemas que fueron detectados en la etapa de análisis, pudiendo así experimentar la fusión de conocimientos con habilidades humanas, una mezcla esencial que marca la diferencia de quienes no hayan realizado esto anteriormente.

BIBLIOGRAFÍA

Microsoft Developer Network. Visual Basic Lenguaje Reference. [en línea]
<[https://msdn.microsoft.com/en-us/library/sh9ywfdk\(v=vs.140\).aspx](https://msdn.microsoft.com/en-us/library/sh9ywfdk(v=vs.140).aspx)>

Net-Information. VB.NET Crystal Reports. [en línea]
<http://vb.netinformations.com/crystalreport/vb.net_crystal_report_step_by_step>

Plantilla de diagrama de flujos para WORD [en línea]
<<https://www.lucidchart.com/pages/es/plantilla-de-diagrama-de-flujo-para-word>>

Canal Visual Basic.NET. Manuales.NET: Manual VB.NET. [en línea]
<<http://www.canalvisualbasic.net/manual-net/vb-net/>>

ANEXO 1: CODIGO FUENTE: LOGIN

```

Imports MaterialSkin
Public Class Login
    Public using As New Usuario 'usuario variable publico
    Dim BsnUsuario As New BsnUsuario()
    Dim Usuario As New Usuario
    Dim BsnValidarUsuario As New BsnValidarUsuario()

    Private Sub Login_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Dim SkinManager As MaterialSkinManager = MaterialSkinManager.Instance
        SkinManager.AddFormToManage(Me)
        SkinManager.Theme = MaterialSkinManager.Themes.LIGHT
        SkinManager.ColorScheme = New ColorScheme(Primary.Green600, Primary.Green700,
Primary.Green200, Accent.Red100, TextShade.WHITE)
    End Sub

    Private Sub btnAceptar_Click_1(sender As Object, e As EventArgs) Handles
btnAceptar.Click
        'solo para mandarlo al formulario eliminar usuario
        using.NombreUsuario = txtUsuario.Text
        using.Contrasena = txtPass.Text

        'CREA LA CLASE PARA USARLA DE FORMA PRIVADA EN ESTE FORMULARIO
        Dim Validarusuario As New ValidarUsuario()
        'SIRVE PARA LUEGO VALIDAR SI EXISTE
        Validarusuario.NombreUsuario = txtUsuario.Text
        Validarusuario.Contrasena = txtPass.Text
        'EXTRA EL LVL DEL USUARIO
        BsnValidarUsuario.GetLevel(Validarusuario)
        'MsgBox(Validarusuario.TipoUsuario)
        using.Tipo_usuario = Validarusuario.TipoUsuario

        If BsnValidarUsuario.ValidaUsuario(Validarusuario) Then
            Usuario.NombreUsuario = Validarusuario.NombreUsuario
            Usuario.Contrasena = Validarusuario.Contrasena
            BsnUsuario.IDusuarioPassYname(Usuario)
            using.Id_Usuario = Usuario.Id_Usuario
            Me.Hide()
            Principal.Show()
        Else
            MsgBox("Usuario o contrasena incorrecta", vbCritical + vbSystemModal +
vbOKOnly, "Login")
        End If
    End Sub
End Class

```

ANEXO 2: CODIGO FUENTE PANTALLA PRINCIPAL

```

Imports CampingCucao.Login
Imports MaterialSkin
Public Class Principal
    Dim Usuario As String = Login.txtUsuario.Text
    Private Sub CerrarSesionToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles CerrarSesionToolStripMenuItem.Click
        Me.Dispose()
        Login.Show()
    End Sub
    Private Sub SalirToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
SalirToolStripMenuItem.Click
        Application.Exit()

    End Sub

    Private Sub Principal_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Dim fecha_manana As Date
        fecha_manana = "12/12/2017"
        lblFechaHoy.Text = DateTime.Today.Date
        'lblFechaHoy.Text = DateTime.Today.Date
        If Login.usering.Tipo_usuario = 2 Then
            'btnUsuarios.Enabled = False
            btnGuias.Enabled = False
            btnReiniciarCapacidad.Visible = False
            btnOtros.Enabled = False
        ElseIf Login.usering.Tipo_usuario = 1 Then
            btnUsuarios.Enabled = True
            btnGuias.Enabled = True
            btnReiniciarCapacidad.Visible = True
            btnOtros.Enabled = True
        End If

        If Usuario = "1" Then
            EliminarUsuarioToolStripMenuItem.Enabled = False

        End If

        Dim SkinManager As MaterialSkinManager = MaterialSkinManager.Instance
        SkinManager.AddFormToManage(Me)
        SkinManager.Theme = MaterialSkinManager.Themes.LIGHT
        SkinManager.ColorScheme = New ColorScheme(Primary.Green700, Primary.Green700,
Primary.Green700, Accent.Red100, TextShade.WHITE)
    End Sub

```

```

Private Sub btnUsuarios_Click(sender As Object, e As EventArgs) Handles
btnUsuarios.Click
    ConsultarClienteSolicitud.Close()
    Listar_Solicitud_Modificar.Close()
    ConsultarPagoSolicitud.Close()
    Consultar_Guia_Servicio.Close()
    ConsultarArriendo.Close()
    ConsultarCaminata.Close()
    tabAdmin.Visible = True
    'FORMS CLIENTES
    FmCliente.Close()
    'FORMS GUIAS
    FmGuia.Close()
    'FORMS SERVICIO
    FmServicio.Close()
    'FORMS SOLICITUD
    FmSolicitud.Close()
    'FORMS PAGO
    FmPago.Close()
    'HISTORICO
    FmInformes.Close()
    'MOSTRAR FORM
    FmUsuario.TopLevel = False
    FmUsuario.FormBorderStyle = System.Windows.Forms.FormBorderStyle.None
    FmUsuario.Parent = tabMantenedoresAdmin.TabPages(0)
    FmUsuario.Show()
    If Login.usering.Tipo_usuario = 2 Then
        FmUsuario.btnEliminar.Enabled = False
        FmUsuario.btnModificar.Enabled = Enabled
        FmUsuario.btnAceptar.Enabled = False
    End If
End Sub

```

```

Private Sub btnClientes_Click(sender As Object, e As EventArgs) Handles
btnClientes.Click
    ConsultarClienteSolicitud.Close()
    Listar_Solicitud_Modificar.Close()
    ConsultarPagoSolicitud.Close()
    Consultar_Guia_Servicio.Close()
    ConsultarArriendo.Close()
    ConsultarCaminata.Close()
    tabAdmin.Visible = True
    'FORMS USURIO
    FmUsuario.Close()
    'FORMS CLIENTES
    'FmCliente.Close()
    'FORM SERVICIO

```

```

FmServicio.Close()
'FORMS GUIAS
FmGuia.Close()
'FORMS SOLICITUD
FmSolicitud.Close()
'FORMS PAGO
FmPago.Close()
'HISTORICO
FmInformes.Close()
'MOSTRAR FORM
FmCliente.TopLevel = False
FmCliente.FormBorderStyle = System.Windows.Forms.FormBorderStyle.None
FmCliente.Parent = tabMantenedoresAdmin.TabPages(0)
FmCliente.Show()
If Login.usering.Tipo_usuario = 2 Then
    FmCliente.btnEliminar.Enabled = False
    'FmUsuario.btnModificar.Enabled = False
End If
End Sub

Private Sub btnGuias_Click(sender As Object, e As EventArgs) Handles btnGuias.Click
    ConsultarClienteSolicitud.Close()
    Listar_Solicitud_Modificar.Close()
    ConsultarPagoSolicitud.Close()
    Consultar_Guia_Servicio.Close()
    ConsultarArriendo.Close()
    ConsultarCaminata.Close()
    tabAdmin.Visible = True
    'FORMS USSUARIO
    FmUsuario.Close()
    'FORMS CLIENTES
    FmCliente.Close()
    'FORMS GUIAS
    'FmGuia.Close()
    'FORM SERVICIO
    FmServicio.Close()
    'FORMS SOLICITUD
    FmSolicitud.Close()
    'FORMS PAGO
    FmPago.Close()
    'HISTORICO
    FmInformes.Close()
    'MOSTRAR FORM
    FmGuia.TopLevel = False
    FmGuia.FormBorderStyle = System.Windows.Forms.FormBorderStyle.None
    FmGuia.Parent = tabMantenedoresAdmin.TabPages(0)
    FmGuia.Show()

```

```

If Login.usering.Tipo_usuario = 2 Then
    FmGuia.btnEliminar.Enabled = False
    'FmUsuario.btnModificar.Enabled = False
End If
End Sub

```

```

Private Sub btnServicios_Click(sender As Object, e As EventArgs) Handles
btnServicios.Click
    ConsultarClienteSolicitud.Close()
    Listar_Solicitud_Modificar.Close()
    ConsultarPagoSolicitud.Close()
    Consultar_Guia_Servicio.Close()
    ConsultarArriendo.Close()
    ConsultarCaminata.Close()
    tabAdmin.Visible = True
    'FORMS USURIO
    FmUsuario.Close()
    'FORMS CLIENTES
    FmCliente.Close()
    'FORMS GUIAS
    FmGuia.Close()
    'FORM SERVICIO
    'FmServicio.Close()
    'FORMS SOLICITUD
    FmSolicitud.Close()
    'FORMS PAGO
    FmPago.Close()
    'HISTORICO
    FmInformes.Close()
    'MOSTRAR FORM
    FmServicio.TopLevel = False
    FmServicio.FormBorderStyle = System.Windows.Forms.FormBorderStyle.None
    FmServicio.Parent = tabMantenedoresAdmin.TabPages(0)
    FmServicio.Show()
    If Login.usering.Tipo_usuario = 2 Then
        FmServicio.btnEliminar.Enabled = False
        FmServicio.btnModificar.Enabled = False
    End If
End Sub

```

```

Private Sub btnSolicitud_Click(sender As Object, e As EventArgs) Handles
btnSolicitud.Click
    ConsultarClienteSolicitud.Close()
    Listar_Solicitud_Modificar.Close()
    ConsultarPagoSolicitud.Close()
    Consultar_Guia_Servicio.Close()
    ConsultarArriendo.Close()

```

```

ConsultarCaminata.Close()
tabAdmin.Visible = True
'FORMS USURIO
FmUsuario.Close()
'FORMS CLIENTES
FmCliente.Close()
'FORMS GUIAS
FmGuia.Close()
'FORM SERVICIO
FmServicio.Close()
'FORMS SOLICITUD
'FmSolicitud.Close()
'FORMS PAGO
FmPago.Close()
'HISTORICO
FmInformes.Close()
'MOSTRAR FORM
FmSolicitud.TopLevel = False
FmSolicitud.FormBorderStyle = System.Windows.Forms.FormBorderStyle.None
FmSolicitud.Parent = tabMantenedoresAdmin.TabPages(0)
FmSolicitud.Show()
If Login.usering.Tipo_usuario = 1 Then
    'FmServicio.btnEliminar.Enabled = False
    'FmServicio.btnModificar.Enabled = False
End If
End Sub

Private Sub btnPagos_Click(sender As Object, e As EventArgs) Handles btnPagos.Click
    ConsultarClienteSolicitud.Close()
    Listar_Solicitud_Modificar.Close()
    ConsultarPagoSolicitud.Close()
    Consultar_Guia_Servicio.Close()
    ConsultarArriendo.Close()
    ConsultarCaminata.Close()
    tabAdmin.Visible = True
    'FORMS USURIO
    FmUsuario.Close()
    'FORMS CLIENTES
    FmCliente.Close()
    'FORMS GUIAS
    FmGuia.Close()
    'FORM SERVICIO
    FmServicio.Close()
    'FORMS SOLICITUD
    FmSolicitud.Close()
    'FORMS PAGO
    'FmPago.Close()

```

```

'HISTORICO
FmInformes.Close()
'MOSTRAR FORM
FmPago.TopLevel = False
FmPago.FormBorderStyle = System.Windows.Forms.FormBorderStyle.None
FmPago.Parent = tabMantenedoresAdmin.TabPages(0)
FmPago.Show()
If Login.using.Tipo_usuario = 2 Then
    FmServicio.btnEliminar.Enabled = False
    FmServicio.btnModificar.Enabled = False
End If
End Sub

Private Sub btnOtros_Click(sender As Object, e As EventArgs) Handles btnOtros.Click
    ConsultarClienteSolicitud.Close()
    Listar_Solicitud_Modificar.Close()
    ConsultarPagoSolicitud.Close()
    Consultar_Guia_Servicio.Close()
    ConsultarArriendo.Close()
    ConsultarCaminata.Close()
    tabAdmin.Visible = True
    'FORMS USURIO
    FmUsuario.Close()
    'FORMS CLIENTES
    FmCliente.Close()
    'FORMS GUIAS
    FmGuia.Close()
    'FORM SERVICIO
    FmServicio.Close()
    'FORMS SOLICITUD
    FmSolicitud.Close()
    'FORMS PAGO
    FmPago.Close()
    'HISTORICO
    'FmInformes.Close()
    'MOSTRAR FORM
    If Login.using.Tipo_usuario = 1 Then
        FmInformes.TopLevel = False
        FmInformes.FormBorderStyle = System.Windows.Forms.FormBorderStyle.None
        FmInformes.Parent = tabMantenedoresAdmin.TabPages(0)
        FmInformes.Show()

        'FmServicio.btnEliminar.Enabled = False
        'FmServicio.btnModificar.Enabled = False
    End If
End Sub

```

```

Private Sub btnReiniciarCapacidad_Click(sender As Object, e As EventArgs) Handles
btnReiniciarCapacidad.Click
    Dim BsnServicio As New BsnServicio()
    Dim BsnPago As New BsnPago
    Dim Pago As New Pago
    Dim count As Integer = 0
    If MsgBox("Esta seguro que desea reiniciar las capacidades de las zonas de
arriendos?", vbExclamation + vbSystemModal + vbOKCancel, "Reiniciar capacidad") =
DialogResult.OK Then
        BsnPago.GetUltimoPago(Pago)
        'Convert.ToDateTime(lblFechaHoy.Text).Date

        If
            Pago.Fecha_cobro.ToString("MM-dd-yyyy") <
Convert.ToDateTime(lblFechaHoy.Text).Date.ToString("MM-dd-yyyy") Then
            BsnServicio.ReiniciarCapacidad()
            MsgBox("Se han reiniciado las capacidades satisfactoriamente",
vbInformation)
        Else
            MsgBox("No se puede realizar accion, aun no es un nuevo dia.", vbExclamation
+ vbSystemModal + vbOKOnly, "Reiniciar capacidad")
        End If

    End If
End Sub

Private Sub btnCerrarSesion_Click(sender As Object, e As EventArgs) Handles
btnCerrarSesion.Click
    Me.Dispose()
    Login.txtPass.Text = ""
    Login.txtUsuario.Text = ""
    Login.Show()
End Sub

Private Sub Principal_FormClosed(sender As Object, e As FormClosedEventArgs) Handles
MyBase.FormClosed
    Login.Dispose()
    Me.Dispose()
End Sub
End Class

```

ANEXO 3: CODIGO FUENTE AGREGAR CLIENTE

```

Private Sub btnAceptar_Click(sender As Object, e As EventArgs) Handles btnAceptar.Click
    Dim empty =
    Me.Controls.OfType(Of TextBox)().Where(Function(txt) txt.Text.Length = 0)
    If empty.Any Or cmbSexo.Text = "" Then
        MsgBox("Favor llene todos los campos", vbExclamation, "Agregar cliente")
    Else
        If rbRut.Checked Then
            Cliente.Id_cliente = txtID_cliente.Text
            BsnCliente.VerificaCliente(Cliente)
            If Cliente.Id_cliente = "" Then
                Cliente.Id_cliente = txtID_cliente.Text
                If ValidarRut(Cliente.Id_cliente) = "Rut Correcto" Then
                    Cliente.Nombre_cliente = txtNombre_cliente.Text
                    Cliente.Apellido_cliente = txtApellido_cliente.Text
                    Cliente.Fono_cliente = CStr(txt_Fono.Text)
                    Cliente.Dir_cliente = txt_direccion.Text
                    If ValidaEMail(txt_Mail.Text) = True Then
                        Cliente.Mail_cliente = txt_Mail.Text
                        If cmbSexo.Text = "M" Then
                            Cliente.Sexo_cliente = "M"
                        End If
                        If cmbSexo.Text = "F" Then
                            Cliente.Sexo_cliente = "F"
                        End If
                        'aca recién ingresa
                        BsnCliente.IngresarCliente(Cliente)
                        MsgBox("Cliente ingresado con éxito!", vbInformation)
                        dgv_cliente.DataSource = BsnCliente.GetCliente
                    Else MsgBox("Formato Correo incorrecto!", vbExclamation) 'if
validaEmail
                End If
            Else MsgBox("Forma rut Incorrecto", vbExclamation) ' if ValirRut
            End If
        Else MsgBox("Cliente ya existe!", vbExclamation) 'if verificacliente
        'txt_pasaporte.Text = "x"
        End If

    ElseIf rbPasaporte.Checked Then
        Cliente.Id_cliente = txt_pasaporte.Text
        BsnCliente.VerificaCliente(Cliente)
        If Cliente.Id_cliente = "" Then
            Cliente.Id_cliente = txt_pasaporte.Text
            Cliente.Nombre_cliente = txtNombre_cliente.Text
            Cliente.Apellido_cliente = txtApellido_cliente.Text
            Cliente.Mail_cliente = txt_Mail.Text
        End If
    End If
End Sub

```

```

Cliente.Fono_cliente = CStr(txt_Fono.Text)
Cliente.Dir_cliente = txt_direccion.Text
Cliente.Nacionalidad = "E"

If cmbSexo.Text = "M" Then
    Cliente.Sexo_cliente = "M"
End If

If cmbSexo.Text = "F" Then
    Cliente.Sexo_cliente = "F"
End If

'aca recién ingresa
BsnCliente.IngresarCliente(Cliente)
MsgBox("Cliente ingresado con éxito!", vbInformation)
'txtID_cliente.Text = "x"

dgv_cliente.DataSource = BsnCliente.GetCliente
Else MsgBox("Cliente ya existe!", vbExclamation) 'if verificacliente
End If
End If
End If
End Sub

```

ANEXO 4: CODIGO FUENTE AGREGAR SOLICITUD

```

Private Sub btnAceptar_Click(sender As Object, e As EventArgs) Handles btnAceptar.Click
    Dim empty =
    Me.Controls.OfType(Of TextBox)().Where(Function(txt) txt.Text.Length = 0)
    If empty.Any Then
        MsgBox("Favor llene todos los campos", vbExclamation, "Agregar solicitud")
    Else
        If lblIDCliente.Text <> "Seleccione Solicitud" Then
            If dtpFechaSolicitud.Value < DateTime.Today.Date Then
                dtpFechaSolicitud.Value = DateTime.Today.Date
                MsgBox("Error fecha ingresada. No puede ser menor a la fecha actual",
vbExclamation)
            Else
                solicitud.Fecha_solicitud = dtpFechaSolicitud.Value.Date
                'dtpFechaSolicitud.Value.Date
                solicitud.Cod_servicio = servicio.Cod_servicio
                'VERIFICAR SI CLIENTE ESTA SOLICITANDO EL MISMO SERVICIO 2 VECES EN LA
MISMA FECHA
                'ASIGNA CLIENTE SELECCIONADO CON EL MISMO DIA
                If rbArriendo.Checked And servicio.Tipo_servicio = "A" Then
                    solicitud.Solicitud_rut_guia = "00000000"
                    BsnSolicitud.GetIDClienteArriendo(solicitud, lblIDCliente.Text,
dtpFechaSolicitud.Value.Date)
                    If lblIDCliente.Text = solicitud.Solicitud_id_cliente Then

```

```

        MsgBox("Imposible asignar servicio. Cliente ya cuenta con el
servicio ARRIENDO para la fecha: " + dtpFechaSolicitud.Value.Date, vbCritical +
vbSystemModal + vbOKOnly, "Ingreso solicitud")
    Else
        solicitud.Solicitud_id_cliente = lblIDCliente.Text
        solicitud.Fecha_solicitud = dtpFechaSolicitud.Value.Date
        'AGREGAR A LA BD

        BsnSolicitud.InsertarSolicitud(solicitud)
        MsgBox("Solicitud ingresada satisfactoriamente para la fecha:
" + dtpFechaSolicitud.Value.Date + " Comprobante generado", vbInformation + vbSystemModal
+ vbOKOnly, "Ingreso solicitud")
        ConsultarClienteSolicitud.Close()
        BsnServicio.GetServiceSolicitud(solicitud) '
solicitud.tipo_servicio

        BsnGuia.GetNombreguia(solicitud) ' solicitud.nombre_guia
        BsnServicio.Getzona(solicitud) 'solicitud.nombre_destino
        GenerarComprobante()

        If solicitud.Tipo_servicio = "A" Then
            BsnServicio.Getzona(solicitud)
            'da nombre de la zona para arriendo
        ElseIf solicitud.Tipo_servicio = "C" Then
            BsnServicio.GetDestino(solicitud)
            'da nombre del destino para la caminata
        End If
        BsnGuia.GetNombreguia(solicitud)
    End If
Else
    If rbCaminata.Checked And servicio.Tipo_servicio = "C" Then
        BsnSolicitud.GetIDClienteCaminata(solicitud,
lblIDCliente.Text, dtpFechaSolicitud.Value.Date)
        If lblIDCliente.Text = solicitud.Solicitud_id_cliente Then
            MsgBox("Imposible asignar servicio. Cliente ya cuenta con
el servicio CAMINATA para la fecha: " + dtpFechaSolicitud.Value.Date, vbCritical +
vbSystemModal + vbOKOnly, "Ingreso solicitud")
        Else
            solicitud.Solicitud_id_cliente = lblIDCliente.Text
            solicitud.Fecha_solicitud = dtpFechaSolicitud.Value.Date
            solicitud.Solicitud_rut_guia = dgvGuia.Item(0, 0).Value()
            'AGREGAR A LA BD
            solicitud.Nombre_destino = cbDestino.Text
            BsnSolicitud.InsertarSolicitud(solicitud)
            MsgBox("Solicitud CAMINATA ingresada satisfactoriamente
para la fecha: " + dtpFechaSolicitud.Value.Date + " Comprobante generado", vbInformation +
vbSystemModal + vbOKOnly, "Ingreso solicitud")
            ConsultarClienteSolicitud.Close()

```

```
        BsnServicio.GetServiceSolicitud(solicitud) '
solicitud.tipo_servicio
        If solicitud.Tipo_servicio = "A" Then
            BsnServicio.Getzona(solicitud)
            'da nombre de la zona para arriendo
        ElseIf solicitud.Tipo_servicio = "C" Then
            BsnServicio.GetDestino(solicitud)
            'da nombre del destino para la caminata
        End If

        BsnGuia.GetNombreguia(solicitud) ' solicitud.nombre_guia
        'solicitud.nombre_destino
        GenerarComprobante()
    End If
End If
End If
Else
    MsgBox("Favor seleccione un cliente", vbExclamation, "Ingreso solicitud")
End If
End If
End Sub
```

ANEXO 5: CODIGO FUENTE MODIFICAR USUARIO

```

Private Sub btnModificar_Click(sender As Object, e As EventArgs) Handles
btnModificar.Click
    Dim empty =
    Me.Controls.OfType(Of TextBox)().Where(Function(txt) txt.Text.Length = 0)
    If empty.Any Or cmb_tipoUsuario.Text = "" Then
        MsgBox("Favor llene todos los campos", vbExclamation, "Modificar usuario")
    Else
        If Login.usering.Tipo_usuario = cmb_tipoUsuario.Text Then
            If MsgBox("Esta seguro que desea modificar usuario: " +
Usuario.NombreUsuario + " ?", vbOKCancel, "Modificar usuario") = DialogResult.OK Then
                'nombre usuario
                Usuario.NombreUsuario = txt_Usuario.Text
                'mail
                If Login.usering.NombreUsuario = Usuario.NombreUsuario Then
                    IngresarNoDuplicado()
                Else
                    'Usuario.NombreUsuario = txt_Usuario.Text
                    BsnUsuario.VerificaUsuario(Usuario)
                    'verifica que el nombre usuario no exista
                    If Usuario.NombreUsuario = "" Then
                        IngresarNoDuplicado()
                    Else
                        MsgBox("Nombre Usuario ya esta registrado", vbExclamation,
"Modificar usuario")
                    End If
                End If
                'contraseñas
            End If
        ElseIf Login.usering.Tipo_usuario <> cmb_tipoUsuario.Text And
Login.usering.NombreUsuario <> txt_nombre.Text Then
            Usuario.NombreUsuario = txt_Usuario.Text
            BsnUsuario.VerificaUsuario(Usuario)
            If Usuario.NombreUsuario = "" Then
                IngresarNoDuplicado()
            Else
                MsgBox("Nombre Usuario ya esta registrado", vbExclamation, "Modificar
usuario2")
            End If
        Else
            MsgBox("Imposible cambiar su nivel de acceso", vbExclamation, "Modificar
usuario")
        End If
    End If
End Sub

```

ANEXO 6: CODIGO FUENTE ELIMINAR GUIA

```

Private Sub btnEliminar_Click(sender As Object, e As EventArgs) Handles btnEliminar.Click
    If Guia.Rut_guia <> "" Then
        If MsgBox("Esta seguro que desea eliminar guia: " + Guia.Rut_guia + " ?",
vbInformation + vbOKCancel, "Eliminar guia") = DialogResult.OK Then
            'ELIMINA
            BsnGuia.EliminarGuia(Guia)
            MsgBox("Consulta elminida satisfactoriamente", vbInformation)
            'CARGA NUEVAMENTE DATAGRID
            dgv_guia.DataSource = BsnGuia.GetGuia()
            txt_buscar.Text = ""
        End If
    Else
        MsgBox("Debe seleccionar un registro a modificar", vbExclamation)
    End If
End Sub

```

ANEXO 7: CODIGO FUENTE AGREGAR PAGO

```

Private Sub btnAceptar_Click(sender As Object, e As EventArgs) Handles btnAceptar.Click
    If lblGuia.Text = "Seleccione SOLICITUD" Or lblIDCliente.Text = "Seleccione
SOLICITUD" Or lblServicio.Text = "Seleccione SOLICITUD" Then
        MsgBox("Favor complete todos los valores antes de continuar", vbExclamation +
vbSystemModal + vbOKOnly, "Ingresar valores")
    Else
        Pago.Solicitud_cod_solicitud = lblCodSolicitud.Text
        Pago.Cliente_id_cliente = lblIDCliente.Text
        Pago.Fecha_cobro = dtpFecha.Value.Date
        Pago.Monto_pago = Int(txtMonto.Text)

        Servicio.Cod_servicio = lblServicio.Text
        Cliente.Id_cliente = lblIDCliente.Text
        Solicitud.Cod_solicitud = lblCodSolicitud.Text
        'SUMA 1 A CAPACIDAD ACTUAL
        BsnServicio.ActualizarCapacidad(Servicio)
        'EXTRAE CAPACIDAD TOTAL
        BsnServicio.GetCapActual(Servicio)
        'EXTRACE CAPACIDAD ACTUAL
        BsnServicio.GetCapTotal(Servicio)
        'SI CAPACIDAD TOTAL = CAPACIDAD ACTUAL, ACTUALIZA ESTADO DE ZONA A NO
DISPONIBLE
        If Servicio.Capacidad_zona <= Servicio.Capacidad_Actual Then
            BsnServicio.CapacidadCero(Servicio)
        End If
        BsnPago.IngresarPago(Pago)
        BsnPago.GetCodigoPago(Pago)
        ' SE ASIGNAN LOS DATOS DE PAGO
    End If
End Sub

```

```

Historico.Pago_cod_pago = Pago.Cod_pago
Historico.Monto_pago_h = Pago.Monto_pago
Historico.Fecha_salida = Pago.Fecha_cobro
' SE EXTRAEN DATOS DEL CLIENTE PARA INGRESAR EN TABLA HISTORICA
BsnCliente.GetDatosCliente(Cliente)
Historico.Cliente_id_cliente = Cliente.Id_cliente
Historico.Nombre_cliente_h = Cliente.Nombre_cliente
Historico.Mail_cliente_h = Cliente.Mail_cliente
Historico.Fono_cliente_h = Cliente.Fono_cliente
Historico.Sexo_h = Cliente.Sexo_cliente
Historico.Apellidos_cliente_h = Cliente.Apellido_cliente
Historico.Nacionalidad_H = Cliente.Nacionalidad
' SE EXTRAEN DATOS DE SERVICIO PARA INGRESAR EN TABLA HISTORICA
BsnServicio.GetDatosServicio(Servicio)
Historico.Tipo_servicio_h = Servicio.Tipo_servicio
' SE EXTRAEN DATOS DE SOLICITUD
'BsnSolicitud.GetDatosSolicitud(Solicitud)
Historico.Fecha_entrada = Convert.ToDateTime(lblFechaSolicitud.Text).Date
' SE INGRESA LOS DATOS A HISTORICO
BsnHistorico.InsertarHistorico(Historico)
'MODIFICA ESTADO SOLICITUD A PAGADO (X)
BsnSolicitud.ModificarEstadoSolicitud(Solicitud)

MsgBox("Pago ingresado correctamente. Comprobante generado con exito",
vbInformation + vbSystemModal + vbOKOnly, "Ingresar pago")
GenerarComprobante()
lblGuia.Text = "Seleccione SOLICITUD"
lblIDCliente.Text = "Seleccione SOLICITUD"
lblServicio.Text = "Seleccione SOLICITUD"
txtMonto.Text = ""
ConsultarPagoSolicitud.Close()
btnAbrirSolicitud.Enabled = True
dtpFecha.Enabled = False
dtpFecha.Value = DateTime.Today.Date
dgvPago.DataSource = BsnPago.GetPago
End If
End Sub

```