

INTRODUCCIÓN

La presente investigación nace de la necesidad que tienen los administradores del “Hotel Bar Restaurant Fogón Manabita” de la parroquia Tonsupa, quienes quieren mejorar los servicios que ofrecen a los clientes que visitan dicho sector turístico, perteneciente al cantón Atacames, para tener una herramienta que les permita automatizar varios de los procesos que realizan a diario en la empresa, con la finalidad de realizar las tomas de decisiones que conlleven al mejor rendimiento de la misma.

Gracias al diagnóstico realizado mediante una encuesta dirigida a los clientes y trabajadores del mencionado hotel-restaurant, así como una entrevista a su administrador y con las técnicas de observación aplicadas, se pudo comprobar que existen varias deficiencias en la atención de un cliente al momento que éste realiza la petición de un servicio, como lo es la realización de un pedido, o una reservación, ya que no se lo hace de manera precisa, por no contar con una herramienta que le facilite la información de dichos procedimientos, también se pudo detectar varias inconformidades de los clientes al momento de realizar la facturación, es por ello que el grupo de personas que conforman la empresa en todas sus áreas están de acuerdo con la implementación de un sistema que ayude con el trato correcto de la información y que permita a los administradores realizar reportes en tiempo real para la toma de decisiones.

La propuesta presentada en el proyecto tiene como finalidad aportar con un sistema informático basado en software libre que solucione varios de estos problemas, con el análisis de los principales procesos, de tal modo que con el uso de la tecnología necesaria se dé la asistencia y ayude a mejorar los procesos y servicios que ofrece el hotel mejorando la productividad de la misma y haciendo que las tareas diarias sean mucho más fáciles al personal que lo realiza.

De este modo haciendo la implementación de la herramienta propuesta y dando las capacitaciones necesarias para el funcionamiento de la misma, se pueda determinar el uso y la importancia de la tecnología aplicadas en el campo de las empresas privadas, en este caso las hoteleras.

CAPITULO I: FUNDAMENTOS TEÓRICOS

1.1. SISTEMAS DE INFORMACIÓN.

1.1.1 DEFINICIÓN.

Un sistema de información (SI) es un conjunto de elementos orientados al tratamiento y administración de datos e información, organizados y listos para su posterior uso, generados para cubrir una necesidad (objetivo). Dichos elementos formarán parte de alguna de estas categorías:

- Personas.
- Datos.
- Actividades o técnicas de trabajo.
- Recursos materiales en general (típicamente recursos informáticos y de comunicación, aunque no tienen por qué ser de este tipo obligatoriamente).¹

Todos estos elementos del sistema de información interactúan entre sí para procesar los datos (incluyendo procesos manuales y automáticos) dando lugar a información más

¹ http://es.wikipedia.org/wiki/Sistema_de_informaci%C3%B3n

elaborada y distribuyéndola de la manera más adecuada posible en una determinada organización en función de sus objetivos.

1.1.2 SOFTWARE.

1.1.2.1 DEFINICIÓN.

Software se refiere al equipamiento lógico o soporte lógico de una computadora digital, y comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de tareas específicas; en contraposición a los componentes físicos del sistema, llamados hardware.²

Tales componentes lógicos incluyen, entre muchos otros, aplicaciones informáticas como navegadores de internet, procesador de textos, que permite al usuario realizar todas las tareas concernientes a edición de textos; software de sistema, tal como un sistema operativo, que, básicamente, permite al resto de los programas funcionar adecuadamente, facilitando la interacción con los componentes físicos y el resto de las aplicaciones, también provee una interfaz para el usuario.

1.1.2.2 TIPOS DE SOFTWARE.

1.1.2.2.1 SOFTWARE DE SISTEMAS OPERATIVOS.

El software de sistema es el software básico o sistema operativo. Es un conjunto de programas cuyo objeto es facilitar el uso del computador (aísla de la complejidad de cada dispositivo, y presenta al exterior un modelo común de sistema de manejo para todos los dispositivos) y conseguir que se use eficientemente (ejemplo: realizar operaciones mientras se ejecuta un programa). Administra y asigna los recursos del sistema (Hardware) tales como memoria, discos, puertos y dispositivos de comunicaciones, impresoras, pantallas, teclados, etc.

1.1.2.2.2 SOFTWARE DE APLICACIONES.

Por otro lado, al hablar de software de aplicación se refiere a los programas que controlan y optimización la operación de la máquina, establecen una relación básica y

² <http://es.wikipedia.org/wiki/Software>

fundamental entre el usuario y el computador, hacen que el usuario pueda usar en forma cómoda y amigable complejos sistemas de hardware, realizan funciones que para el usuario serían engorrosas o incluso imposibles, y actúan como intermediario entre el usuario y el hardware.

1.1.2.3 SOFTWARE EMPRESARIAL.

Por software empresarial se entiende generalmente cualquier tipo de software de aplicación que está orientado a ayudar a una empresa a mejorar su productividad o a medirla.

Este término engloba una amplia variedad de aplicaciones informáticas que incluyen desde programas de contabilidad y de ofimática, hasta sistemas de planificación de recursos empresariales (ERP), pasando por programas de gestión de clientes (CRM), de recursos humanos, etc.

1.1.3 HERRAMIENTAS DE GESTIÓN.

1.1.3.1 DEFINICIÓN.

Se entiende que las herramientas de gestión son todos los sistemas, aplicaciones, controles, soluciones de cálculo, metodología, etc., que ayudan a la gestión de una empresa en los siguientes aspectos generales:

- Herramientas para el registro de datos en cualquier departamento empresarial
- Herramientas para el control y mejora de los procesos empresariales
- Herramientas para la consolidación de datos y toma de decisiones

Así, entenderemos que si segmentamos la empresa en sus diferentes departamentos genéricos, tendremos herramientas que nos ayudarán a gestionar, organizar, dirigir, planificar, controlar, conocer, etc., cada uno de los departamentos y las relaciones entre ellos y el mundo exterior, ejemplos de ellas las siguientes herramientas:

- Administradores de información personal

- Benchmarking
- Gestores de paquetes
- Minería de datos
- Sistemas de gestión empresarial
- ERP
- CRM
- Software de administración de proyectos
- Subcontratación

1.1.3.2 PLANIFICACIÓN DE RECURSOS EMPRESARIALES (ERP).

Los sistemas de planificación de recursos empresariales (en inglés ERP, Enterprise Resource Planning) son sistemas de gestión de información que integran y automatizan muchas de las prácticas de negocio asociadas con los aspectos operativos o productivos de una empresa.

Los sistemas ERP son sistemas integrales de gestión para la empresa. Se caracterizan por estar compuestos por diferentes partes integradas en una única aplicación. Estas partes son de diferente uso, por ejemplo: producción, ventas, compras, logística, contabilidad (de varios tipos), gestión de proyectos, GIS (sistema de información geográfica), inventarios y control de almacenes, pedidos, nóminas, etc. Sólo podemos definir un ERP como la integración de todas estas partes.

Lo contrario sería como considerar un simple programa de facturación como un ERP por el simple hecho de que una empresa integre únicamente esa parte. Ésta es la diferencia fundamental entre un ERP y otra aplicación de gestión. El ERP integra todo lo necesario para el funcionamiento de los procesos de negocio de la empresa.

1.1.3.3 CUSTOMER RELATIONSHIP MANAGEMENT (CRM).

La gestión de la relación con los clientes es parte de una estrategia de negocio centrada en el cliente. Una parte fundamental de su idea es, precisamente, la de recopilar la mayor cantidad de información posible sobre los clientes, para poder dar valor a la

oferta. La empresa debe trabajar para conocer las necesidades de los mismos y así poder adelantar una oferta y mejorar la calidad en la atención.³

Cuando hablamos de mejorar la oferta nos referimos a poder brindarles soluciones a los clientes que se adecuen perfectamente a sus necesidades, y no como rezan muchos opositores a estas disciplinas generarles nuevas necesidades.

Por lo tanto, el nombre CRM hace referencia a una estrategia de negocio basada principalmente en la satisfacción de los clientes, pero también a los sistemas informáticos que dan soporte a esta, siempre y cuando esto se consiga.

El CRM (del acrónimo inglés "Customer Relationship Management") designa a los programas informáticos que se usan en la empresa para gestionar las relaciones comerciales, fidelización y ventas que mantenemos con los clientes.

1.1.4 ARQUITECTURA DE SOFTWARE.

1.1.4.1 DEFINICIÓN.

La Arquitectura del Software es el diseño de más alto nivel de la estructura de un sistema.

Una Arquitectura de Software, también denominada Arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software para un sistema de información.

La Arquitectura de Software establece los fundamentos para que analistas, diseñadores, programadores, etc. trabajen en una línea común que permita alcanzar los objetivos del sistema de información, cubriendo todas las necesidades.

Una arquitectura de software se selecciona y diseña con base en objetivos y restricciones.

³ http://es.wikipedia.org/wiki/Customer_relationship_management

Los objetivos son aquellos prefijados para el sistema de información, pero no solamente los de tipo funcional, también otros objetivos como la mantenibilidad, adaptabilidad, flexibilidad e interacción con otros sistemas de información. Las restricciones son aquellas limitaciones derivadas de las tecnologías disponibles para implementar sistemas de información.

Unas arquitecturas son más recomendables de implementar con ciertas tecnologías mientras que otras tecnologías no son aptas para determinadas arquitecturas. Por ejemplo, no es viable emplear una arquitectura de software de tres capas para implementar sistemas en tiempo real.

La arquitectura de software define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos. Toda arquitectura debe ser implementable en una arquitectura física, que consiste simplemente en determinar qué computadora tendrá asignada cada tarea.

1.1.4.2 TIPOS DE MODELADOS DE SISTEMAS DE INFORMACIÓN.

1.1.4.2.1 ARQUITECTURA CENTRALIZADA.

En este tipo de arquitectura, los datos y las aplicaciones están en una única máquina. Es uno de los primeros modelos de ordenadores interconectados, llamados centralizados, donde todo el procesamiento de la organización se llevaba a cabo en una sola computadora, normalmente un Mainframe, y los usuarios empleaban sencillos ordenadores personales.

El primer problema de este modelo es que cuando la carga de procesamiento aumentaba se tenía que cambiar el hardware del Mainframe, lo cual es más costoso que añadir más computadores personales clientes o servidores que aumenten las capacidades.

El otro problema que surgió son las modernas interfaces gráficas de usuario, las cuales podían conllevar a un gran aumento de tráfico en los medios de comunicación y por consiguiente podían colapsar.

1.1.4.2.2 ARQUITECTURA CLIENTE-SERVIDOR.

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

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

Una disposición muy común son los sistemas multicapa en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema.

1.1.4.2.3 ARQUITECTURA DE TRES NIVELES.

La programación por capas es un estilo de programación en el que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño; un ejemplo básico de esto consiste en separar la capa de datos de la capa de presentación al usuario.

La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado. Un buen ejemplo de este método de programación sería el modelo de interconexión de sistemas abiertos.

- **Capa de presentación:** es la que ve el usuario (también se la denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario.
- **Capa de negocio:** es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.
- **Capa de datos:** es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Todas estas capas pueden residir en un único ordenador, si bien lo más usual es que haya una multitud de ordenadores en donde reside la capa de presentación (son los clientes de la arquitectura cliente/servidor). Las capas de negocio y de datos pueden residir en el mismo ordenador, y si el crecimiento de las necesidades lo aconseja se pueden separar en dos o más ordenadores. Así, si el tamaño o complejidad de la base de datos aumenta, se puede separar en varios ordenadores los cuales recibirán las peticiones del ordenador en que resida la capa de negocio.

1.2. CICLO DE VIDA DEL SOFTWARE.

1.2.1 NECESIDAD DE UNA METODOLOGÍA.

Cuando surgió la necesidad de adaptar los sistemas informáticos a las exigencias del mercado, el programador realizaba un relevamiento de las solicitudes de quien necesitaba cierto programa o producto software, y con aquellos requerimientos bajo el

brazo comenzaba la dura tarea de codificar. Esta tarea no estaba administrada, supervisada o gestionada de ningún modo, por lo que se iba corrigiendo a medida que surgían los errores, tanto los lógicos provenientes de la codificación, como los de requerimientos solicitados por el cliente o usuario final.

En la década de 1970 los programas fueron creciendo en complejidad, por lo que la antigua técnica de **code & fix** (codificar y corregir) terminó quedando obsoleta.

Esta técnica se basaba en requerimientos ambiguos y sin especificaciones puntuales.

Al no seguir normas para el proyecto, el cliente o usuario sólo impartían especificaciones muy generales del producto final. Se programaba, se corregía, y se volvía a programar sobre la misma marcha del proyecto. El ciclo de vida de este tipo de proyectos finalizaba cuando se satisfacían las especificaciones, no sólo las primeras por las cuales nació la necesidad del programa, sino también todas aquellas que fueron surgiendo sobre la marcha.

Esta técnica tiene las ventajas de no gastar recursos en análisis, planificación, gestión de recursos, documentación, etc., y bien sabemos que es muy cómoda y muchas veces recomendable cuando el proyecto es muy pequeño y es llevado adelante por uno o dos programadores. Por otro lado, cuando el sistema no es pequeño o es más complejo de lo creído (tengamos en cuenta que no hubo análisis) nos trae desventajas en lo que se refiere a costo de recursos, que siempre será mayor del previsto; aumentará el tiempo de desarrollo y la calidad del código será bastante dudosa.

1.2.2 DEFINICIÓN DE METODOLOGÍA.

La metodología para el desarrollo de software es un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con **altas posibilidades de éxito**. Esta sistematización indica cómo se divide un gran proyecto en módulos más pequeños llamados etapas, y las acciones que corresponden en cada una de ellas, esto ayuda a definir entradas y salidas para cada una de las etapas y, sobre todo, normaliza el modo en que se administra el proyecto. Entonces, una metodología para el desarrollo de software son los procesos a seguir sistemáticamente para idear, implementar y mantener

un producto software desde que surge la necesidad del producto hasta que se cumple con el objetivo por el cual fue creado.

Desde un punto de vista general puede considerarse que el ciclo de vida de un software tiene tres etapas claramente diferenciadas, las cuales se detallan a continuación:

- **Planificación:** Se idea un planeamiento detallado que guíe la gestión del proyecto, temporal y económicamente.
- **Implementación:** En esta etapa se acuerda el conjunto de actividades que componen la realización del producto.
- **Puesta en producción:** Cuando el proyecto entra en la etapa de definición, allí donde se lo presenta al cliente o usuario final, cuando se tiene la seguridad que funciona correctamente y responde a los requerimientos solicitados en su momento. Esta etapa es muy importante no sólo por representar la aceptación o no del proyecto por parte del cliente o usuario final sino por las múltiples dificultades que suele presentar en la práctica, alargándose excesivamente y provocando costos no previstos.

A estas tres grandes etapas es conveniente añadir otras dos que, si bien pudieron enunciarse junto a las otras, es conveniente hacer una diferenciación ya que se tiende a menospreciarlas o a no darles la importancia que requieren.

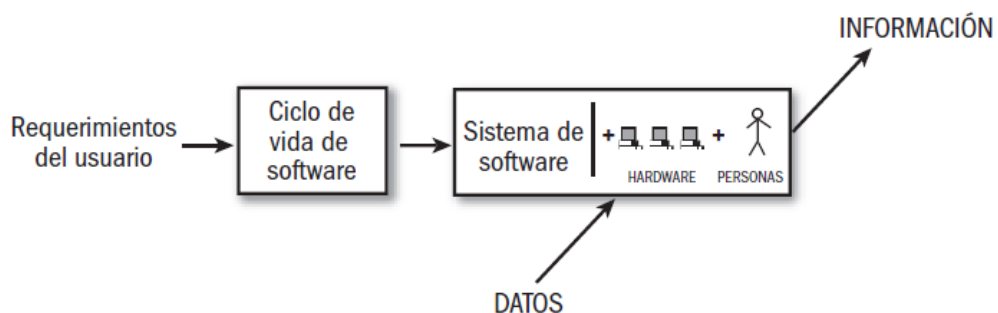


Figura 1. El ciclo de vida de un producto software se desarrolla fuera del ámbito productivo, aunque debemos conocer el entorno (environment) en el que será ejecutado.

- **Inicio:** Éste es el nacimiento de la idea. Aquí se definen los objetivos del proyecto y los recursos necesarios para su ejecución. Hacia dónde se quiere ir, y no cómo se

quiere ir. Las características implícitas o explícitas de cada proyecto hacen necesaria una etapa previa destinada a obtener el objetivo por el cual se escribirán miles o cientos de miles de líneas de código. Un alto porcentaje del éxito del proyecto se definirá en estas etapas que, al igual que la etapa de debugging, muchos líderes de proyecto subestiman.

- **Control en producción:** Control del producto, analizando cómo el proceso difiere o no de los requerimientos originales e iniciando las acciones correctivas si fuesen necesarias. Cuando se dice que hay que corregir el producto, se hace referencia a pequeñas desviaciones de los requerimientos originales que puedan llegar a surgir en el ambiente productivo. Si el programa en estudio no realiza la tarea para lo cual fue creada, esta etapa no es la adecuada para el rediseño. Se incluye también en esta etapa el liderazgo, documentación y capacitación, proporcionando directivas a los recursos humanos, para que hagan su trabajo en forma correcta y efectiva.

1.2.3 OBJETIVOS DE CADA ETAPA.

En cada una de las etapas de un modelo de ciclo de vida, se pueden establecer una serie de objetivos, tareas y actividades que lo caracterizan.

Se hará un repaso y una pequeña descripción de cada una de las etapas del ciclo de vida del software; una vez conocidas las etapas, se tiene que analizar cómo abordarlas en su conjunto. Existen distintos modelos de ciclo de vida, y la elección de un modelo para un determinado tipo de proyecto es realmente importante; el orden de las etapas es uno de estos puntos importantes, Si se elige el modelo de cascada puro en el cual la validación se realiza al final del proyecto, y luego se debe retomar etapas previas, puede resultar no sólo incómodo, sino costoso.

- **Expresión de necesidades:** esta etapa tiene como objetivo el armado de un documento en el cual se reflejan los requerimientos y funcionalidades que ofrecerá al usuario el sistema a implementar (qué, y no cómo, se va a implementar).
- **Especificaciones:** se formaliza los requerimientos; el documento obtenido en la etapa anterior se tomará como punto de partida para esta etapa.

- **Análisis:** se determinan los elementos que intervienen en el sistema a desarrollar, su estructura, relaciones, evolución temporal, funcionalidades, tendremos una descripción clara de qué producto vamos a construir, qué funcionalidades aportará y qué comportamiento tendrá.
- **Diseño:** ya se sabe qué hacer, ahora se tiene que determinar cómo hacer (¿cómo debe ser construido el sistema en cuestión?; se define en detalle entidades y relaciones de las bases de datos, se selecciona el lenguaje que se va a utilizar, el Sistema Gestor de Bases de Datos, etc.).
- **Implementación:** se empieza a codificar algoritmos y estructuras de datos, definidos en las etapas anteriores, en el correspondiente lenguaje de programación o para un determinado sistema gestor de bases de datos. En muchos proyectos se pasa directamente a esta etapa; son proyectos muy arriesgados que adoptan un modelo de ciclo de vida de code & fix (codificar y corregir) donde se eliminan las etapas de especificaciones, análisis y diseño con la consiguiente pérdida de control sobre la gestión del proyecto.
- **Debugging:** el objetivo de esta etapa es garantizar que el programa no contiene errores de diseño o codificación. En esta etapa no se desea saber si el programa realiza lo que solicitó el usuario, esa tarea le corresponde a la etapa de implementación. En ésta se desea encontrar la mayor cantidad de errores. Todos los programas contienen errores: encontrarlos es cuestión de tiempo. Lo ideal es encontrar la mayoría, si no todos, en esta etapa. También se pueden agregar testeos de performance.
- **Validación:** esta etapa tiene como objetivo la verificación de que el sistema desarrollado cumple con los requerimientos expresados inicialmente por el cliente y que han dado lugar al presente proyecto. En muchos proyectos las etapas de validación y debugging se realizan en paralelo por la estrecha relación que llevan. Sin embargo, se tiene que evitar la confusión: se puede realizarlos en paralelo, pero no como una única etapa.

- **Evolución:** en la mayoría de los proyectos se considera esta etapa como **Mantenimiento y evolución**, y se le asigna, no sólo el agregado de nuevas funcionalidades (evolución); sino la corrección de errores que surgen (mantenimiento). En la práctica esta denominación no es del todo errónea, ya que es posible que aun luego de una etapa de debugging y validación exhaustiva, se filtren errores.

Lo que se busca guiándose con una metodología es prolijidad, corrección y control en cada etapa del desarrollo de un programa. Lo que permitirá una forma sistemática para poder obtener un producto correcto y libre de errores.

1.2.4. CLASIFICACIÓN DE LAS METODOLOGÍAS.

Existen dos metodologías que tienen analogía en la práctica con los paradigmas de programación. Metodología estructurada y metodología orientada a objetos.

1.2.4.1.METODOLOGÍA ESTRUCTURADA.

La orientación de esta metodología se dirige hacia los procesos que intervienen en el sistema a desarrollar, es decir, cada función a realizar por el sistema se descompone en pequeños módulos individuales. Es más fácil resolver problemas pequeños, y luego unir cada una de las soluciones, que abordar un problema grande. ⁴

1.2.4.2.METODOLOGÍA ORIENTADA A OBJETOS.

La diferencia básica entre esta metodología de la mencionada anteriormente, es que en la presente no comprende los procesos como funciones sino que arma módulos basados en componentes, es decir, cada componente es independiente del otro. Esto permite que el código sea reutilizable. Es más fácil de mantener porque los cambios están localizados en cada uno de estos componentes.

⁴ http://www.cepeu.edu.py/LIBROS_ELECTRONICOS_3/lpcu097%20-%20001.pdf

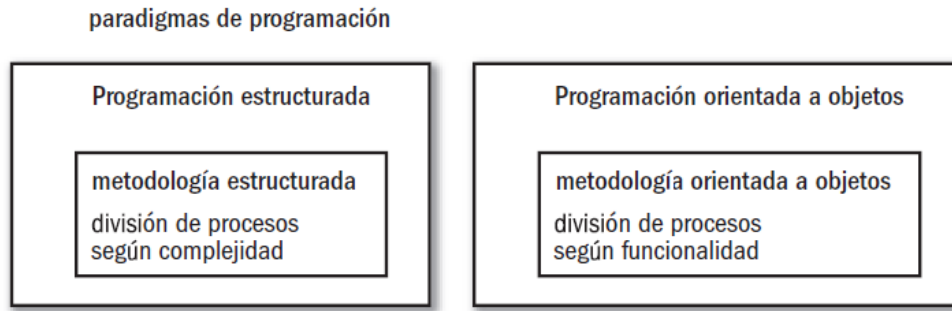


Figura 2. Las distintas metodologías nos ayudarán a abordar el proceso de desarrollo. Un consejo: la mejor metodología, es la que mejor conocemos.

1.2.5. MODELOS DE CICLO DE VIDA.

Las principales diferencias entre distintos modelos de ciclo de vida están divididas en tres grandes visiones:

- El alcance del ciclo de vida, que depende de hasta dónde se desea llegar con el proyecto: sólo saber si es viable el desarrollo de un producto, el desarrollo completo o el desarrollo completo más las actualizaciones y el mantenimiento.
- La calidad y cantidad de las etapas en que se divide el ciclo de vida: según el ciclo de vida que adoptemos, y el proyecto para el cual lo adoptemos.
- La estructura y la sucesión de las etapas, si hay realimentación entre ellas, y si se tiene la libertad de repetir las (iterar).

En los distintos modelos de ciclo de vida se mencionará el riesgo que supone aceptar al elegirlo. Cuando se habla de riesgo, se está refiriendo a la probabilidad que se tiene de volver a retomar una de las etapas anteriores, perdiendo tiempo, dinero y esfuerzo.

1.2.5.1. CICLO DE VIDA LINEAL.

Es el más sencillo de todos los modelos. Consiste en descomponer la actividad global del proyecto en etapas separadas que son realizadas de manera lineal, es decir, cada etapa se realiza una sola vez, a continuación de la etapa anterior y antes de la etapa siguiente. Con un ciclo de vida lineal es muy fácil dividir las tareas, y prever los tiempos (sumando linealmente los de cada etapa).

Las actividades de cada una de las etapas mencionadas deben ser independientes entre sí, es decir, que es condición primordial que no haya retroalimentación entre ellas, aunque sí pueden admitirse ciertos supuestos de realimentación correctiva.

Esto último minimiza, también, las posibilidades de errores durante la codificación y reduce al mínimo la necesidad de requerir información del cliente o del usuario.



Figura 3. La sencillez del ciclo de vida lineal es la razón por la cual es el más elegido en el desarrollo de programas pequeños.

Se destaca como ventaja la sencillez de su gestión y administración tanto económica como temporal, ya que se acomoda perfectamente a proyectos internos de una empresa para programas muy pequeños de ABM (sistemas que realizan Altas, Bajas y Modificaciones sobre un conjunto de datos).

Tiene como desventaja que no es apto para Desarrollos que superen mínimamente requerimientos de retroalimentación entre etapas, es decir, es muy costoso retomar una etapa anterior al detectar alguna falla.

Es válido tomar este ciclo de vida cuando algún sector pequeño de una empresa necesita llevar un registro de datos acumulativos, sin necesidad de realizar procesos sobre ellos más que una consulta simple. Es decir, una aplicación que se dedique exclusivamente a almacenar datos, sea una base de datos o un archivo plano. Debido a que la realización de las etapas es muy simple y el código muy sencillo.

1.2.5.2.CICLO DE VIDA EN CASCADA PURO.

Este modelo de ciclo de vida fue propuesto por Winston Royce en el año 1970. Es un ciclo de vida que admite iteraciones, contrariamente a la creencia de que es un ciclo de vida secuencial como el lineal. Después de cada etapa se realiza una o varias revisiones para comprobar si se puede pasar a la siguiente. Es un modelo rígido, poco flexible, y

Es un ciclo adecuado para los proyectos en los que se dispone de todos los requerimientos al comienzo, para el desarrollo de un producto con funcionalidades conocidas o para proyectos, que aun siendo muy complejos, se entienden perfectamente desde el principio.

Se evidencia que es un modelo puramente teórico, ya que el usuario rara vez mantiene los requerimientos iniciales y existen muchas posibilidades de que se deba retomar alguna etapa anterior. Pero es mejor que no seguir ningún ciclo de vida.

A pesar de todo no es erróneo adaptarlo para alguna aplicación en la que el modelo de ciclo lineal no sea del todo adecuado, y el uso de un modelo de gestión más elaborado no lo justifique.

1.2.5.3.CICLO DE VIDA EN V.

Este ciclo fue diseñado por Alan Davis, y contiene las mismas etapas que el ciclo de vida en cascada puro. A diferencia de aquél, a éste se le agregaron dos subetapas de retroalimentación entre las etapas de análisis y mantenimiento, y entre las de diseño y debugging.

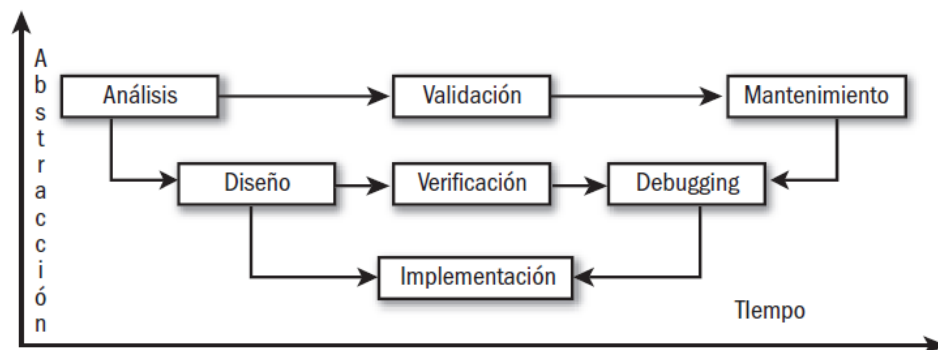


Figura 5. Este modelo ofrece mayor garantía de corrección al terminar el proyecto.

Las ventajas y desventajas de este modelo son las mismas del ciclo anterior, con el agregado de los controles cruzados entre etapas para lograr una mayor corrección.

Se puede utilizar este modelo de ciclo de vida en aplicaciones, que si bien son simples (pequeñas transacciones sobre bases de datos por ejemplo), necesitan una confiabilidad muy alta. Un ejemplo claro en el que no se puede permitir el lujo de cometer errores es

una aplicación de facturación, en la que si bien los procedimientos vistos individualmente son de codificación e interpretación sencilla, la aplicación en su conjunto puede tener matices complicados.

1.2.5.4.CICLO DE VIDA TIPO SASHIMI.

Este ciclo de vida es parecido al ciclo de vida en cascada puro, con la diferencia de que en el ciclo de vida en cascada no se pueden solapar las etapas, y en éste sí. Esto suele, en muchos casos, aumentar su eficiencia ya que la retroalimentación entre etapas se encuentra implícitamente en el modelo.

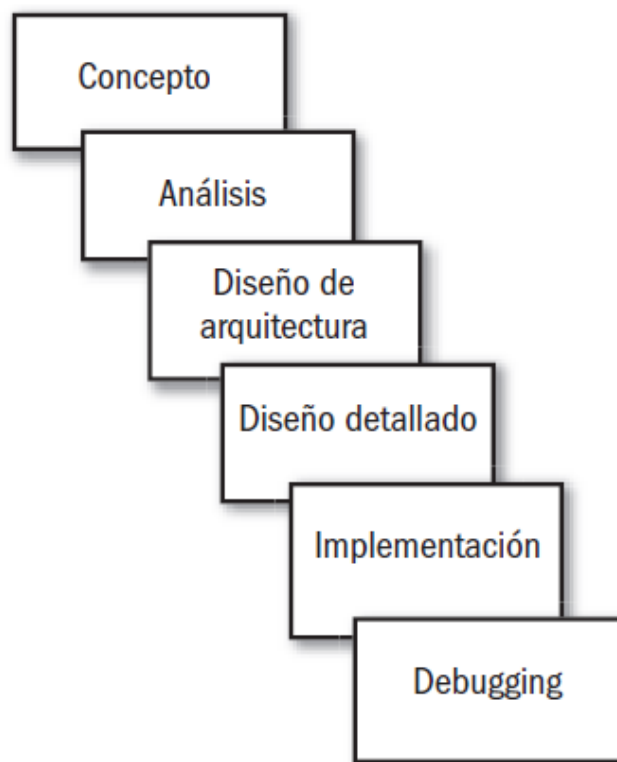


Figura 6. El nombre procede del modelo del estilo japonés de presentar el pescado crudo cortado, en el que los cortes se solapan entre sí.

Se hace notar como ventajas la ganancia de calidad en lo que respecta al producto final, la falta de necesidad de una documentación detallada (el ahorro proviene por el solapado de las etapas). Sus desventajas también se refieren al solapamiento de las etapas: es muy difícil gestionar el comienzo y fin de cada etapa y los problemas de comunicación, si aparecen, generan inconsistencias en el proyecto.

Cuando se desea realizar una aplicación que compartirá los recursos (CPU, memoria o espacio de almacenamiento) con otras aplicaciones en un ambiente productivo, este modelo de ciclo de vida es una opción muy válida. El solapamiento de sus etapas permite en la práctica jugar un poco con el modelo de tres capas ahorrando recursos.

1.2.5.5.CICLO DE VIDA EN CASCADA CON SUBPROYECTOS.

Sigue el modelo de ciclo de vida en cascada. Cada una de las cascadas se dividen en subetapas independientes que se pueden desarrollar en paralelo.

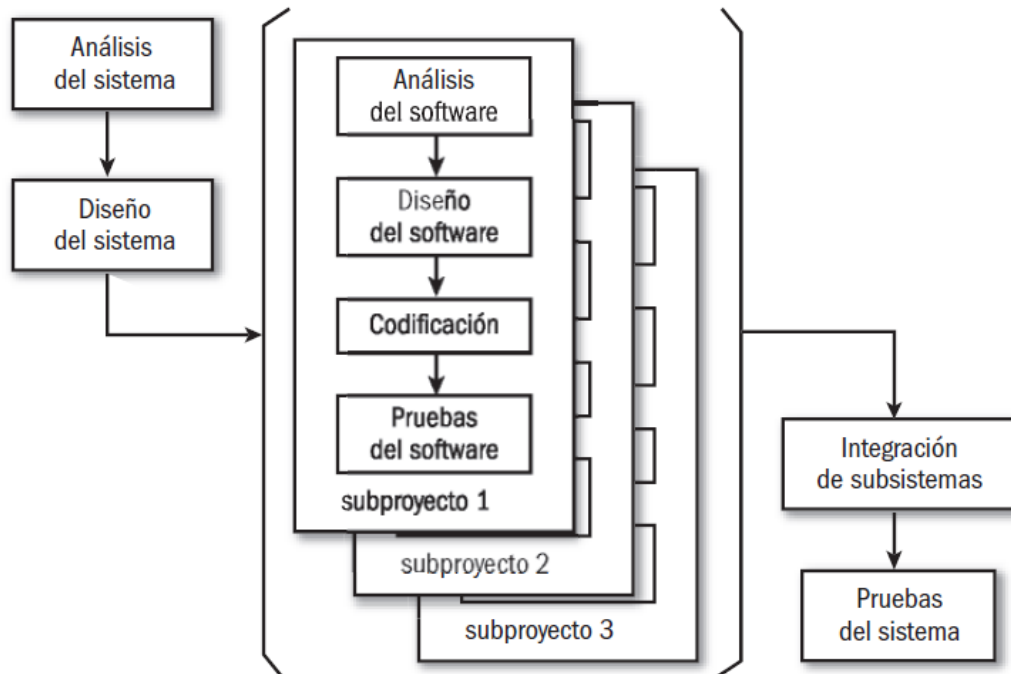


Figura 7. El modelo de ciclo de vida ideal cuando se cuenta con un plantel de programadores numeroso.

La ventaja es que se puede tener más gente trabajando al mismo tiempo, pero la desventaja es que pueden surgir dependencias entre las distintas subetapas que detengan el proyecto temporalmente si no es gestionado de manera correcta.

Se puede utilizar este modelo para administrar cualquier proyecto mencionado en los modelos anteriores. Pero cuidando de administrar muy bien los tiempos.

1.2.5.6.CICLO DE VIDA ITERATIVO.

También derivado del ciclo de vida en cascada puro, este modelo busca reducir el riesgo que surge entre las necesidades del usuario y el producto final por malos entendidos durante la etapa de solicitud de requerimientos.

Es la iteración de varios ciclos de vida en cascada. Al final de cada iteración se le entrega al cliente una versión mejorada o con mayores funcionalidades del producto. El cliente es quien luego de cada iteración, evalúa el producto y lo corrige o propone mejoras.

Estas iteraciones se repetirán hasta obtener un producto que satisfaga al cliente.

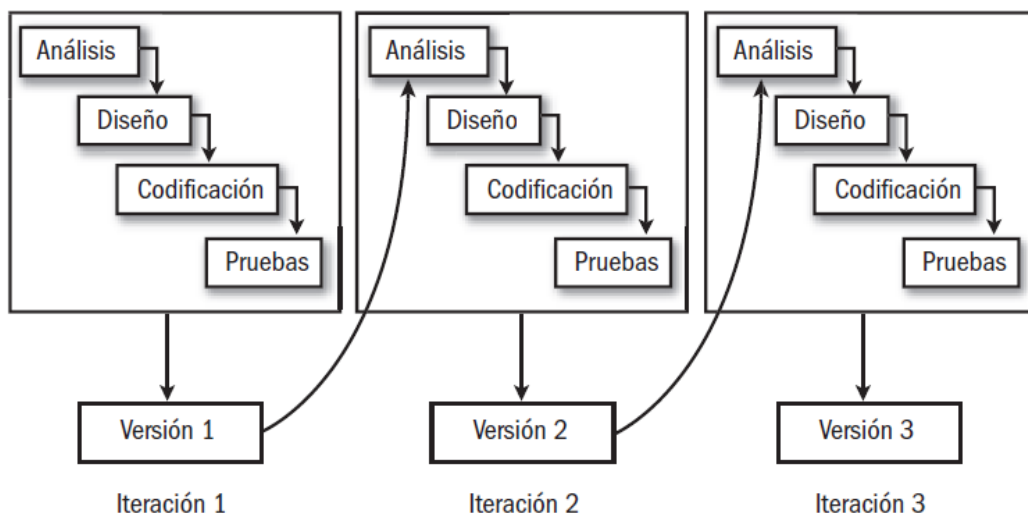


Figura 8. Es un modelo ideal a seguir cuando el usuario necesita entregas rápidas aunque el proyecto no esté terminado.

Se suele utilizar en proyectos en los que los requerimientos no están claros de parte del usuario, por lo que se hace necesaria la creación de distintos prototipos para presentarlos y conseguir la conformidad del cliente.

Se puede adoptar el modelo mencionado en aplicaciones medianas a grandes, en las que el usuario o cliente final no necesita todas las funcionalidades desde el principio del proyecto. Quizás una empresa que debe migrar sus aplicaciones hacia otra arquitectura, y desea hacerlo paulatinamente, es un candidato ideal para este tipo de modelo de ciclo de vida.

1.2.5.7.CICLO DE VIDA POR PROTOTIPOS.

El uso de programas prototipo no es exclusivo del ciclo de vida iterativo. En la práctica los prototipos se utilizan para validar los requerimientos de los usuarios en cualquier ciclo de vida.

Si no se conoce exactamente cómo desarrollar un determinado producto o cuáles son las especificaciones de forma precisa, suele recurrirse a definir especificaciones iniciales para hacer un prototipo, o sea, un producto parcial y provisional. En este modelo, el objetivo es lograr un producto intermedio, antes de realizar el producto final, para conocer mediante el prototipo cómo responderán las funcionalidades previstas para el producto final.

Antes de adoptar este modelo de ciclo debemos evaluar si el esfuerzo por crear un prototipo vale realmente la pena adoptarlo.

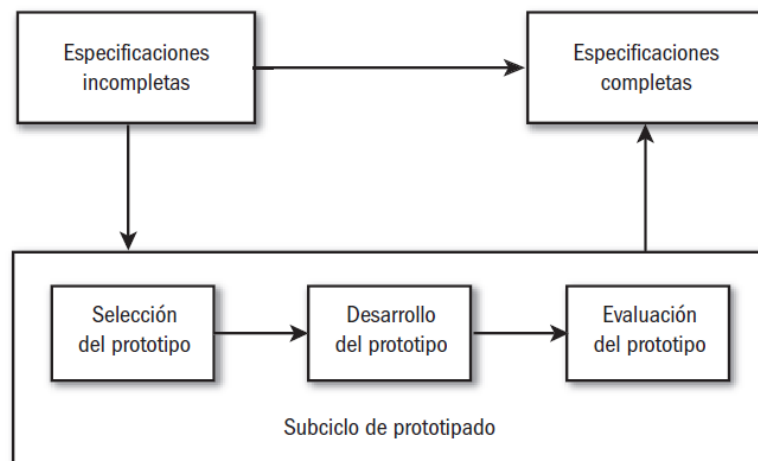


Figura 9. Este modelo nos permite suavizar la transición entre los requerimientos iniciales y finales que surgen en la creación de un proyecto con grandes innovaciones.

Se utiliza mayoritariamente en desarrollos de productos con innovaciones importantes, o en el uso de tecnologías nuevas o poco probadas, en las que la incertidumbre sobre los resultados a obtener, o la ignorancia sobre el comportamiento, impiden iniciar un proyecto secuencial.

La ventaja de este ciclo se basa en que es el único apto para desarrollos en los que no se conoce a priori sus especificaciones o la tecnología a utilizar. Como contrapartida, por

este desconocimiento, tiene la desventaja de ser altamente costoso y difícil para la administración temporal.

1.2.5.8.CICLO DE VIDA EVOLUTIVO.

Este modelo acepta que los requerimientos del usuario pueden cambiar en cualquier momento. La práctica demuestra que obtener todos los requerimientos al comienzo del proyecto es extremadamente difícil, no sólo por la dificultad del usuario de transmitir su idea, sino porque estos requerimientos evolucionan durante el desarrollo y de esta manera, surgen nuevos requerimientos a cumplir. El modelo de ciclo de vida evolutivo afronta este problema mediante una iteración de ciclos **requerimientos–desarrollo–evaluación**.

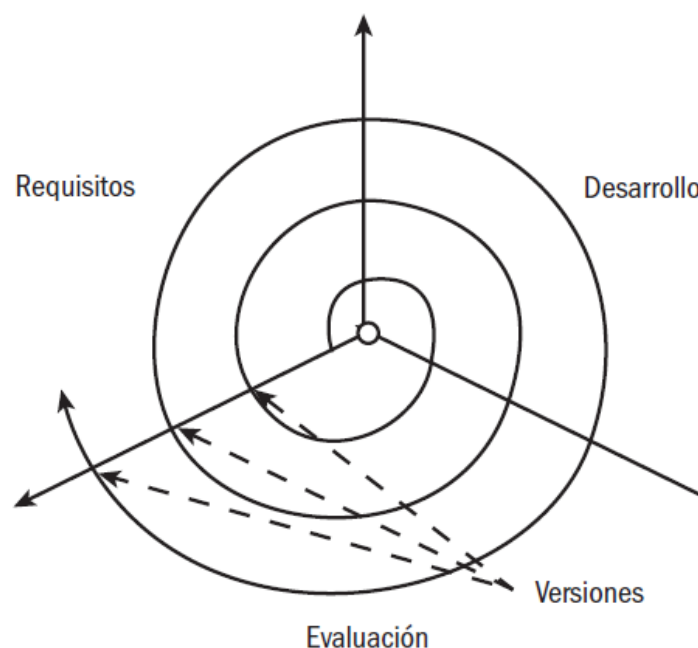


Figura 10. Luego de cada desarrollo obtenemos una nueva versión del producto.

Resulta ser un modelo muy útil cuando desconocemos la mayoría de los requerimientos iniciales, o estos requerimientos no están completos.

Se tiene como ejemplo un sistema centralizado de stock–ventas–facturación, en el cual hay muchas áreas que utilizarán la aplicación. Hay dos complicaciones: la primera, los usuarios no conocen de informática, la segunda, no es uno, sino varios los sectores que pueden pedir modificaciones o hacer nuevas solicitudes.

Además, el pedido de un sector puede influir en los requerimientos del otro. Se hace necesario, entonces, lograr que la aplicación evolucione hasta lograr las satisfacciones de los todos los sectores involucrados.

1.2.5.9.CICLO DE VIDA INCREMENTAL.

Este modelo de ciclo de vida se basa en la filosofía de construir incrementando las funcionalidades del programa.

Se realiza construyendo por módulos que cumplen las diferentes funciones del sistema. Esto permite ir aumentando gradualmente las capacidades del software.

Este ciclo de vida facilita la tarea del desarrollo permitiendo a cada miembro del equipo desarrollar un módulo particular en el caso de que el proyecto sea realizado por un equipo de programadores.

Es una repetición del ciclo de vida en cascada, aplicándose este ciclo en cada funcionalidad del programa a construir. Al final de cada ciclo se entrega una versión al cliente que contiene una nueva funcionalidad. Este ciclo de vida permite realizar una entrega al cliente antes de terminar el proyecto.

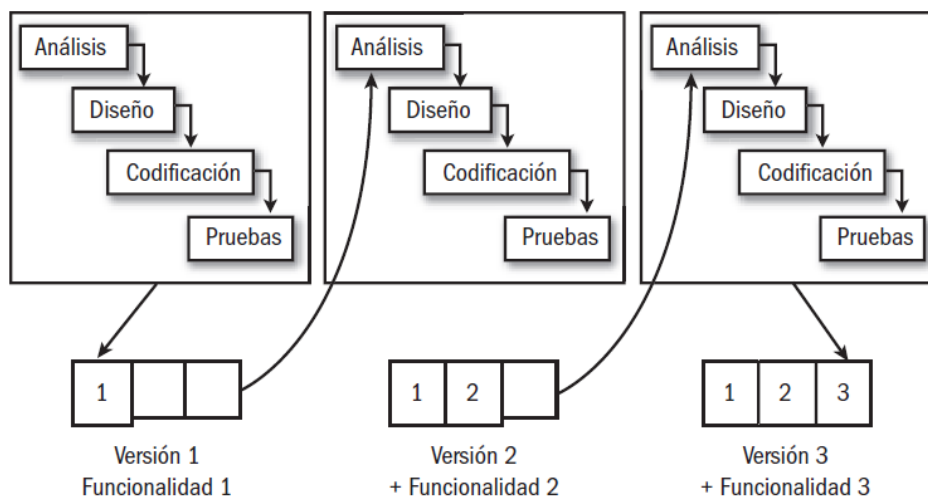


Figura 11. Una forma de reducir los riesgos es ir construyendo partes del sistema adoptando este modelo.

El modelo de ciclo de vida incremental genera algunos beneficios tales como los que se describen a continuación:

- Construir un sistema pequeño siempre es menos riesgoso que construir un sistema grande.
- Como desarrollamos independientemente las funcionalidades, es más fácil relevar los requerimientos del usuario.
- Si se detecta un error grave, sólo desechamos la última iteración.
- No es necesario disponer de los requerimientos de todas las funcionalidades en el comienzo del proyecto y además facilita la labor del desarrollo con la conocida filosofía de **divide & conqueror**.

Este modelo de ciclo de vida no está pensado para cierto tipo de aplicaciones, sino que está orientado a cierto tipo de usuario o cliente. Se puede utilizar este modelo de ciclo de vida para casi cualquier proyecto, pero será verdaderamente útil cuando el usuario necesite entregas rápidas, aunque sean parciales.

1.2.5.10. CICLO DE VIDA EN ESPIRAL.

Este ciclo puede considerarse una variación del modelo con prototipado, fue diseñado por Boehm en el año 1988. Este modelo se basa en una serie de ciclos repetitivos para ir ganando madurez en el producto final. Toma los beneficios de los ciclos de vida incremental y por prototipos, pero se tiene más en cuenta el concepto de riesgo que aparece debido a las incertidumbres e ignorancias de los requerimientos proporcionados al principio del proyecto o que surgirán durante el desarrollo. A medida que el ciclo se cumple (el avance del espiral), se van obteniendo prototipos sucesivos que van ganando la satisfacción del cliente o usuario.

En este modelo hay cuatro actividades que envuelven a las etapas.

- Planificación: Relevamiento de requerimientos iniciales o luego de una iteración.
- Análisis de riesgo: De acuerdo con el relevamiento de requerimientos se decide si se continúa con el desarrollo.
- Implementación: Se desarrolla un prototipo basado en los requerimientos.

- Evaluación: El cliente evalúa el prototipo, si da su conformidad, termina el proyecto. En caso contrario, se incluyen los nuevos requerimientos solicitados por el cliente en la siguiente iteración.

La ventaja más notoria de este modelo de desarrollo de software es que puede comenzarse el proyecto con un alto grado de incertidumbre, se entiende también como ventaja el bajo riesgo de retraso en caso de detección de errores, ya que se puede solucionar en la próxima rama del espiral.

Algunas de las desventajas son: el costo temporal que suma cada vuelta del espiral, la dificultad para evaluar los riesgos y la necesidad de la presencia o la comunicación continua con el cliente o usuario.

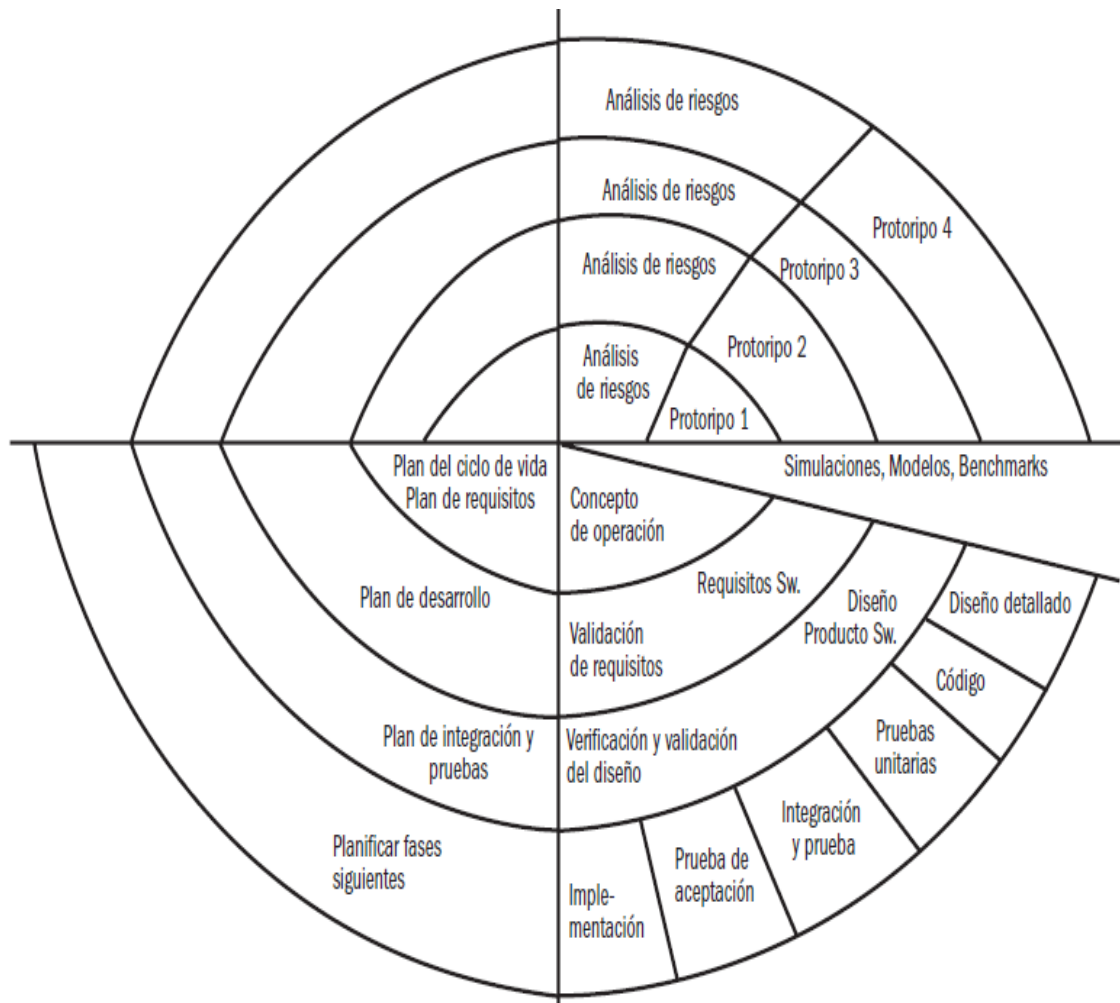


Figura 12. El espiral se repite las veces que sea necesario hasta que el cliente o usuario obtiene la satisfacción de sus necesidades, momento en el cual se retira del espiral.

Se observa que es un modelo adecuado para grandes proyectos internos de una empresa, en donde no es posible contar con todos los requerimientos desde el comienzo y el usuario está en nuestro mismo ambiente laboral.

Se puede citar una aplicación que administre reclamos, pedidos e incidentes, como ejemplo para utilizar este modelo de ciclo de vida, en el que los sectores que utilizarán el sistema son demasiados y con intereses muy diversos como para lograr un relevamiento exhaustivo y completo de los requerimientos.

1.2.5.11. CICLO DE VIDA ORIENTADO A OBJETOS.

Esta técnica fue presentada en la década del 90, tal vez como una de las mejores metodologías a seguir para la creación de productos software.

Puede considerarse como un modelo pleno a seguir, como así también una alternativa dentro de los modelos anteriores.

Al igual que la filosofía del paradigma de la programación orientada a objetos, en esta metodología cada funcionalidad, o requerimiento solicitado por el usuario, es considerado un objeto. Los objetos están representados por un conjunto de propiedades, a los cuales denominamos atributos, por otra parte, al comportamiento que tendrán estos objetos los denominamos métodos.

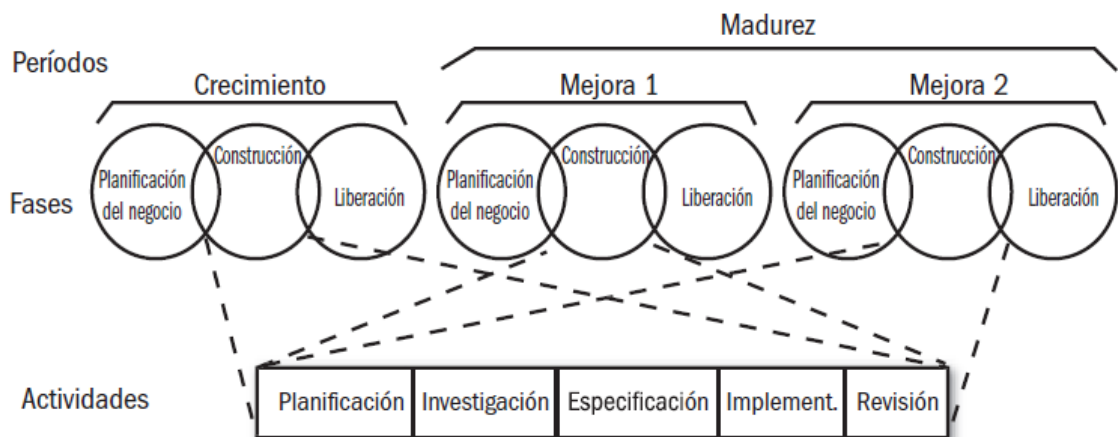


Figura 13. Un modelo muy versátil, tanto para pequeños como para grandes proyectos.

Se puede ver que tanto la filosofía de esta metodología, los términos utilizados en ella y sus fines, coinciden con la idea de obtener un concepto de objeto sobre casos de la vida real.

La característica principal de este modelo es la abstracción de los requerimientos de usuario, por lo que este modelo es mucho más flexible que los restantes, que son rígidos en requerimientos y definición, soportando mejor la incertidumbre que los anteriores, aunque sin garantizar la ausencia de riesgos.

La abstracción es lo que permite analizar y desarrollar las características esenciales de un objeto (requerimiento), despreocupándonos de las menos relevantes.

Favorece la reducción de la complejidad del problema que se desea abordar y permite el perfeccionamiento del producto.

En este modelo se utilizan las llamadas fichas CRC (clase–responsabilidades–colaboración) como herramienta para obtener las abstracciones y mecanismos clave de un sistema analizando los requerimientos del usuario. En la ficha CRC se escribe el nombre de la clase u objeto, sus responsabilidades (los métodos) y sus colaboradores (otras clases u objetos de los cuales necesita). Estas fichas, además, ayudan a confeccionar los denominados casos de uso.

No es correcto suponer que este modelo sólo es útil cuando se escoge para la implementación un lenguaje con orientación a objetos. Se puede utilizar independientemente del lenguaje elegido. Es un modelo a seguir, una técnica, y no obliga a utilizar ningún lenguaje en particular.

1.3. HERRAMIENTAS DE PROGRAMACIÓN.

1.3.1 DEFINICIÓN.

Las herramientas de programación, son aquellas que permiten realizar aplicativos, programas, rutinas, utilitarios y sistemas para que la parte física del computador u ordenador, funcione y pueda producir resultados.⁵

⁵ <http://www.lenguajes-de-programacion.com/herramientas-de-programacion.shtml>

1.3.2 LENGUAJES DE PROGRAMACIÓN.

Un lenguaje de programación es un idioma artificial diseñado para expresar computaciones que pueden ser llevadas a cabo por máquinas, generalmente por las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana.

Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Al proceso por el cual se escribe, se prueba, se depura, se compila y se mantiene el código fuente de un programa informático se le llama programación.

1.3.3 BASES DE DATOS.

1.3.3.1 DEFINICIÓN.

Una base de datos o banco de datos (en ocasiones abreviada BB.DD.) es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. En la actualidad, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital (electrónico), que ofrece un amplio rango de soluciones al problema de almacenar datos.⁶

1.3.3.2 SISTEMAS GESTORES DE BASES DE DATOS.

Los sistemas de gestión de bases de datos (en inglés Database Management System, abreviado DBMS) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan, asegurando su integridad, confidencialidad y seguridad. Por tanto debe permitir:

⁶ http://es.wikipedia.org/wiki/Base_de_datos

- Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD.
- Manipular la base de datos: realizar consultas, actualizarla, generar informes.

1.3.4 HERRAMIENTAS CASE.

1.3.4.1 DEFINICIÓN.

Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

1.3.4.2 TIPOS.

No existe una única clasificación de herramientas CASE y, en ocasiones, es difícil incluirlas en una clase determinada. Podrían clasificarse atendiendo las plataformas que soportan, las fases del ciclo de vida del desarrollo de sistemas que cubren, la arquitectura de las aplicaciones que producen y por su funcionalidad.

Las herramientas CASE en función de las fases del ciclo de vida que abarcan, se pueden agrupar de la forma siguiente:

- Herramientas integradas, I-CASE (Integrated CASE, CASE integrado): abarcan todas las fases del ciclo de vida del desarrollo de sistemas. Son llamadas también CASE workbench.

- Herramientas de alto nivel, U-CASE (Upper CASE - CASE superior o front-end) orientadas a la automatización y soporte de las actividades desarrolladas durante las primeras fases del desarrollo: análisis y diseño.
- Herramientas de bajo nivel, L-CASE (Lower CASE - CASE inferior o back-end) dirigidas a las últimas fases del desarrollo: desarrollo e implantación.
- Juegos de herramientas o toolkits: son el tipo más simple de herramientas CASE. Automatizan una fase dentro del ciclo de vida. Dentro de este grupo se encontrarían las herramientas de reingeniería, orientadas a la fase de mantenimiento.

Las herramientas I-CASE se basan generalmente en una metodología. Tienen un repositorio (BD del proyecto) y aportan técnicas para todas las fases del ciclo de vida. Sin embargo, no todas ellas son modernas en el sentido de aprovechar la potencia de las estaciones de trabajo o la utilización de lenguajes de alto nivel o técnicas de construcción de prototipos.

1.3.5 SOFTWARE LIBRE.

1.3.5.1 DEFINICIÓN.

El software libre (en inglés free software, esta denominación también se confunde a veces con gratis por el doble sentido del inglés free en castellano) es la denominación que se le da al software que respeta la libertad de los usuarios sobre su producto adquirido y, por tanto, una vez obtenido puede ser usado, copiado, estudiado, cambiado y redistribuido libremente. Según la Free Software Foundation, el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, modificar el software y distribuirlo modificado.

El software libre suele estar disponible gratuitamente, o al precio de costo de la distribución a través de otros medios; sin embargo no es obligatorio que sea así, por lo tanto no hay que asociar software libre a "software gratuito" (denominado usualmente Freeware), ya que, conservando su carácter de libre, puede ser distribuido comercialmente ("software comercial"). Análogamente, el "software gratis" o "gratuito" incluye en ocasiones el código fuente; no obstante, este tipo de software no es libre en el mismo sentido que el software libre, a menos que se garanticen los derechos de modificación y redistribución de dichas versiones modificadas del programa.

1.3.5.2 VENTAJAS.

Una de sus ventajas es el bajo costo de adquisición, ya que se trata de un software económico ya que permite un ahorro de grandes cantidades en la adquisición de las licencias.

Otra ventaja es la innovación tecnológica, lo cual se debe a que cada usuario puede aportar sus conocimientos y su experiencia y así decidir de manera conjunta hacia donde se debe dirigir la evolución y el desarrollo del software. Este es un gran avance en la tecnología mundial.

Independencia del proveedor, al disponer del código fuente, se garantiza una independencia del proveedor que hace que cada empresa o particular pueda seguir contribuyendo al desarrollo y los servicios del software.

Otra ventaja es lo que se conoce como escrutinio público lo que permite la corrección de errores y la mejora del producto se lleven a cabo de manera rápida y eficaz por cada uno de los usuarios que lleguen a utilizar el producto.

Por último se puede ver que aunque el software se cree y salga al mercado en una sola lengua, el hecho de ser software libre facilita en gran medida su traducción y localización para que usuarios de diferentes partes del mundo puedan aprovechar estos beneficios.

1.3.6 SOFTWARE PRIVATIVO.

El software privativo es aquel software que no es libre ni semilibre. Su uso, redistribución o modificación están prohibidos, requieren que solicite una autorización, o está tan restringido que de hecho no puede hacerlo libremente.

La Fundación para el Software Libre sigue la norma de que únicamente podemos instalar un programa privativo en nuestros ordenadores, cuando lo hacemos temporalmente y con el propósito de escribir una alternativa libre a ese mismo programa.

1.3.7 TIPOS DE LICENCIAS.

Una licencia es aquella autorización formal con carácter contractual que un autor de un software da a un interesado para ejercer "actos de explotación legales". Pueden existir tantas licencias como acuerdos concretos se den entre el autor y el licenciario.

Desde el punto de vista del software libre, existen distintas variantes del concepto o grupos de licencias:

1.3.7.1 LICENCIAS GPL.

Una de las más utilizadas es la *Licencia Pública General de GNU* (GNU GPL). El autor conserva los derechos de autor (copyright), y permite la redistribución y modificación bajo términos diseñados para asegurarse de que todas las versiones modificadas del software permanecen bajo los términos más restrictivos de la propia GNU GPL. Esto hace que sea imposible crear un producto con partes no licenciadas GPL: el conjunto tiene que ser GPL.

Es decir, la licencia GNU GPL posibilita la modificación y redistribución del software, pero únicamente bajo esa misma licencia. Y añade que si se reutiliza en un mismo programa código "A" licenciado bajo licencia GNU GPL y código "B" licenciado bajo otro tipo de licencia libre, el código final "C", independientemente de la cantidad y calidad de cada uno de los códigos "A" y "B", debe estar bajo la licencia GNU GPL.

En la práctica esto hace que las licencias de software libre se dividan en dos grandes grupos, aquellas que pueden ser mezcladas con código licenciado bajo GNU GPL (y que inevitablemente desaparecerán en el proceso, al ser el código resultante licenciado bajo GNU GPL) y las que no lo permiten al incluir mayores u otros requisitos que no contemplan ni admiten la GNU GPL y que por lo tanto no pueden ser enlazadas ni mezcladas con código gobernado por la licencia GNU GPL.

En el sitio web oficial de GNU hay una lista de licencias que cumplen las condiciones impuestas por la GNU GPL y otras que no.

Aproximadamente el 60% del software licenciado como software libre emplea una licencia GPL.

1.3.7.2 LICENCIAS AGPL.

La Licencia Pública General de Affero (en inglés Affero General Public License, también Affero GPL o AGPL) es una licencia copyleft derivada de la Licencia Pública General de GNU diseñada específicamente para asegurar la cooperación con la comunidad en el caso de software que corra en servidores de red.

La Affero GPL es íntegramente una GNU GPL con una cláusula nueva que añade la obligación de distribuir el software si éste se ejecuta para ofrecer servicios a través de una red de ordenadores.

La Free Software Foundation recomienda que el uso de la GNU AGPLv3 sea considerado para cualquier software que usualmente corra sobre una red.

1.3.7.3 LICENCIAS ESTILO BSD.

Llamadas así porque se utilizan en gran cantidad de software distribuido junto a los sistemas operativos BSD. El autor, bajo tales licencias, mantiene la protección de copyright únicamente para la renuncia de garantía y para requerir la adecuada atribución de la autoría en trabajos derivados, pero permite la libre redistribución y modificación, incluso si dichos trabajos tienen propietario. Son muy permisivas, tanto que son fácilmente absorbidas al ser mezcladas con la licencia GNU GPL con quienes son compatibles.

Puede argumentarse que esta licencia asegura “verdadero” software libre, en el sentido que el usuario tiene libertad ilimitada con respecto al software, y que puede decidir incluso redistribuirlo como no libre. Otras opiniones están orientadas a destacar que este tipo de licencia no contribuye al desarrollo de más software libre (normalmente utilizando la siguiente analogía: "una licencia BSD es más libre que una GPL si y sólo si se opina también que un país que permita la esclavitud es más libre que otro que no la permite").

1.3.7.4 LICENCIAS ESTILO MPL Y DERIVADAS.

Esta licencia es de Software Libre y tiene un gran valor porque fue el instrumento que empleó Netscape Communications Corp. para liberar su Netscape Communicator 4.0 y empezar ese proyecto tan importante para el mundo del Software Libre: Mozilla. Se utilizan en gran cantidad de productos de software libre de uso cotidiano en todo tipo de sistemas operativos. La MPL es Software Libre y promueve eficazmente la colaboración evitando el efecto "viral" de la GPL (si usas código licenciado GPL, tu desarrollo final tiene que estar licenciado GPL).

Desde un punto de vista del desarrollador la GPL presenta un inconveniente en este punto, y lamentablemente mucha gente se cierra en banda ante el uso de dicho código. No obstante la MPL no es tan excesivamente permisiva como las licencias tipo BSD. Estas licencias son denominadas de copyleft débil. La NPL (luego la MPL) fue la primera licencia nueva después de muchos años, que se encargaba de algunos puntos que no fueron tenidos en cuenta por las licencias BSD y GNU. En el espectro de las licencias de software libre se la puede considerar adyacente a la licencia estilo BSD, pero perfeccionada.

1.3.7.5 COPYLEFT.

Hay que hacer constar que el titular de los derechos de autor (copyright) de un software bajo licencia copyleft puede también realizar una versión modificada bajo su copyright original, y venderla bajo cualquier licencia que desee, además de distribuir la versión original como software libre. Esta técnica ha sido usada como un modelo de negocio por una serie de empresas que realizan software libre (por ejemplo MySQL); esta práctica *no* restringe ninguno de los derechos otorgados a los usuarios de la versión copyleft. También podría retirar todas las licencias de software libre anteriormente otorgadas, pero esto obligaría a una indemnización a los titulares de las licencias en uso. En algunos países, toda obra derivada está tan protegida como una original, siempre que la obra derivada parta de una autorización contractual con el autor.

En el caso genérico de que el autor retire las licencias "copyleft", no afectaría de ningún modo a los productos derivados anteriores a esa retirada, ya que no tiene efecto

retroactivo. En términos legales, el autor no tiene derecho a retirar el permiso de una licencia en vigencia. Si así sucediera, el conflicto entre las partes se resolvería en un pleito convencional.

1.4. TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN.

1.4.1 DEFINICIÓN.

Se denominan Tecnologías de la Información y las Comunicación al conjunto de tecnologías que permiten la adquisición, producción, almacenamiento, tratamiento, comunicación, registro y presentación de informaciones, en forma de voz, imágenes y datos contenidos en señales de naturaleza acústica, óptica o electromagnética. Las TICs incluyen la electrónica como tecnología base que soporta el desarrollo de las telecomunicaciones, la informática y el audiovisual.

1.4.2 HISTORIA.

Se pueden considerar las Tecnologías de Información y Comunicación (TIC) un concepto dinámico. Por ejemplo, a finales del siglo XIX el teléfono podría ser considerado una nueva tecnología según las definiciones actuales. Esta misma definición podría aplicarse a la televisión cuando apareció y se popularizó en la década de los 50 del siglo pasado. No obstante esto, hoy no se pondrían en una lista de TIC y es muy posible que actualmente los ordenadores ya no puedan ser calificados de nuevas tecnologías.

A pesar de esto, en un concepto amplio, se puede considerar que el teléfono, la televisión y el ordenador forman parte de lo que se llama TIC, tecnologías que favorecen la comunicación y el intercambio de información en el mundo actual.

Después de la invención de la escritura, los primeros pasos hacia una sociedad de la información estuvieron marcados por el telégrafo eléctrico, después el teléfono y la radiotelefonía, la televisión, Internet.

La telefonía móvil y el GPS han asociado la imagen al texto y a la palabra «sin cables», Internet y la televisión son accesibles en el teléfono móvil que es también una máquina de hacer fotos.

El acercamiento de la informática y de las telecomunicaciones, en el último decenio del siglo XX se han beneficiado de la miniaturización de los componentes, permitiendo producir aparatos «multifunciones» a precios accesibles, desde los años 2000.

Los usos de las TIC no paran de crecer y de extenderse, sobre todo en los países ricos, con el riesgo de acentuar localmente la Brecha digital, y social y la diferencia entre generaciones. Desde la agricultura de precisión y la gestión del bosque, a la monitorización global del medio ambiente planetario o de la biodiversidad, a la democracia participativa (TIC al servicio del desarrollo sostenible) pasando por el comercio, la telemedicina, la información, la gestión de múltiples bases de datos, la bolsa, la robótica y los usos militares, sin olvidar la ayuda a los discapacitados (ciegos que usan sintetizadores vocales avanzados), los TIC tienden a tomar un lugar creciente en la vida humana y el funcionamiento de las sociedades.

1.5. GENERALIDADES DEL BAR HOTEL RESTAURANT FOGÓN MANABITA.

1.5.1 HISTORIA.

El Bar Hotel Restaurant Fogón Manabita es una empresa que inició sus actividades como microempresa de servicios alimenticios, está administrada por sus propietarios desde el inicio, hace 10 años aproximadamente, con el tiempo ha pasado a ser uno de los más grandes negocios de hotelería y turismo ubicado en la ciudad de Tonsupa.

Hasta la fecha actual, esta empresa ha pasado por tres ubicaciones distintas donde cada una de ellas tuvo un objetivo en común y aportó en el desarrollo de la misma, de los cuales se hablará a continuación:

Al principio inició como restaurant con una capacidad para 40 personas dentro de la misma parroquia en calidad de arrendatarios, después con las ganancias se logró comprar un terreno a una cuadra antes de llegar al malecón de la playa, y fue aquí donde se estableció además del restaurant una pequeña edificación con un total de siete habitaciones con capacidad para treinta y cinco personas en calidad de hotel.

Luego de este logro, se alcanzó uno mayor que fue el adquirir un local propio, esquinero frente a la playa, que es donde se encuentra actualmente, por supuesto de mayor afluencia de turistas, por ende con este último fomentar mas la economía de la empresa y crecer continuamente, apoyando así la economía de la parroquia, de la provincia y del país.

Es así como en un lapso de tiempo de diez años aproximadamente se crea el Hotel-Restaurant Fogón Manabita como una empresa que compite en el medio en cual se encuentra, una empresa reconocida nacionalmente y que cada día se esfuerza por dar la mejor atención y servicio a sus clientes.

1.5.2 ORGANIZACIÓN.

Esta empresa está administrada por su propietario, y está constituida por diferentes sectores como lo son:

- ✓ Gerencia
 - Staff Contable
 - Cajero
- ✓ Departamento de Ventas
 - Chefs
 - Ayudantes de Cocina
 - Encargado de Salón
 - Saloneros
 - Seguridad
 - Limpieza
- ✓ Departamento de administración
 - Proveedores
 - Sueldos

En esta empresa todas las actividades con supervisadas por el Gerente, quien es también la persona que toma las decisiones en la misma.

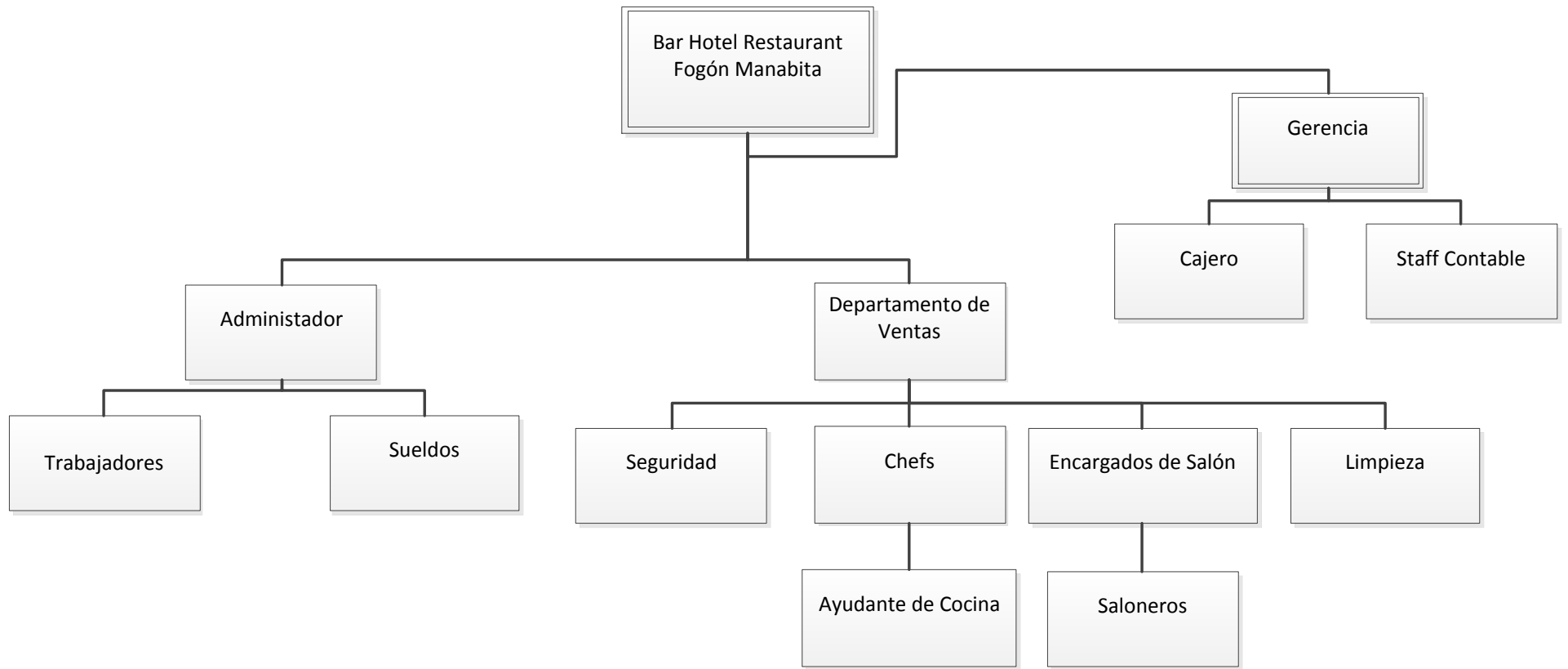


Figura 14. Organigrama del “Bar Hotel Restaurant Fogón Manabita”.

Fuente: Gerencia del Bar Hotel restaurant Fogón Manabiata.

1.5.3 UBICACIÓN.

El Bar Hotel restaurant Fogón Manabita se encuentra ubicado en el malecón de la parroquia Tonsupa, a pocos metros de la playa, dicho lugar es un sitio turístico del cantón Atacames, Esmeraldas – Ecuador. Ver anexo 4.

1.5.4 SERVICIOS.

Los servicios más relevantes en esta empresa son los siguientes: Hospedaje, Alquiler de piscina, Bar, Restaurant, Garaje.

1.5.5 INFRAESTRUCTURA.

El lugar donde funciona la empresa es un edificio de 5 pisos y la planta baja, en las que se distribuyen:

- 10 Habitaciones con capacidad para 8 Personas.
- 15 Habitaciones con capacidad para 6 Personas.
- 15 Habitaciones con capacidad para 4 Personas.
- 12 Habitaciones Matrimoniales.
- 1 Cocina en la planta baja.
- 1 Cocina en el primer piso.
- Restaurant en la planta baja con capacidad para 90 personas.
- Restaurant en el primer piso con capacidad para 50 personas.
- Piscina en la planta baja con capacidad para 40 personas aproximadamente.
- Bar con capacidad para 60 personas aproximadamente.

1.5.6 PARROQUIA TONSUPA.

1.5.5.1 HISTORIA.

Su nombre viene en honor al príncipe TONZUPA, indio de nobles sentimientos y enamorado eterno de su terruño. En este sector se asentaba antiguamente el cementerio indígena, de allí que durante las diferentes construcciones que se han venido edificando

durante las excavaciones de las cimentaciones de los edificios, se han encontrado una infinidad de cerámicas con vistosos colores, figuras finamente decoradas y acabadas, ollas y fragmentos de la cultura atacame, quienes demostraron ser siempre muy amables, amigables y pacíficos, es por ello que cuando naufrago un barco frente a nuestras costas trayendo a los esclavos negros del África estos al desembarcar no encontraron ninguna resistencia por parte de los indios, lo que demuestra que ellos estaban acostumbrados a recibir constantemente a los visitantes de diferentes regiones que hacían de Atacames su centro de comercio e intercambio de productos, ya que ellos se dedicaban a la pesca, agricultura y las artesanías. Aun en la actualidad se pueden apreciar algunas tolas que eran las urnas sagradas donde emprendían el viaje sin retomo.

1.5.5.2 POBLACIÓN.

La parroquia de Tonsupa ha crecido con el paso del tiempo a pasos agigantados determinándolo así el último censo poblacional (INEC 2010⁷) con un número de 10.681 habitantes casi el 28.7% de la población del cantón Atacames, convirtiéndola en la cabecera cantonal con más numero de habitantes.

1.5.5.3 ACTIVIDADES ECONÓMICAS.

En la parroquia Tonsupa, la actividad de las personas es dividida, en temporada de menor afluencia turística, la pesca artesanal y la ganadería representan el ingreso económico para Tonsupa; en tanto que en temporada de afluencia turística el comercio formal e informal es el ingreso representativo.

1.5.5.4 ATRACTIVOS TURÍSTICOS.

El principal atractivo turístico es la playa, donde se puede disfrutar de aguas cristalinas, caminatas, exploración de manglares, paseos en lancha, excelente comida típica de la zona y hermosos atardeceres. Además, se puede disfrutar en los meses de julio y agosto el paso de las ballenas jorobadas muy cerca de las playas.

⁷<http://redatam.inec.gob.ec/cgi-bin/RpWebEngine.exe/PortalAction?&MODE=MAIN&BASE=CPV2010&MAIN=W ebServerMain.inl>

También la parroquia cuenta con un mercado que ofrece artesanías, objetos de coral y accesorios de playa; además se encuentran hoteles, gasolineras, cajeros automáticos, locutorios telefónicos, entre otros servicios.

CAPITULO II: DIAGNÓSTICO.

2.1 ANTECEDENTES.

La investigación presente se la llevó a cabo en el Hotel Restaurant Fogón Manabita ubicado en la parroquia Tonsupa del cantón Atacames de la provincia Esmeraldas, de la República del Ecuador.

Para conocer el problema en estudio, en la presente investigación se tomó tres puntos de vistas distintos pero importantes, uno de ellos es el punto de vista de los trabajadores, otro es el punto de vista del administrador y por último pero no menos importante, el punto de vista del cliente.

Para tener una visión de la empresa en general, el sábado 7 de Agosto del 2010 se realizó una entrevista al gerente propietario Sr. Segundo Quevedo en el hotel que administra, cabe indicar que no se presentaron inconvenientes en el desarrollo de la misma. Ver anexo 3.

Los feriados son fechas que impulsan la visita masiva de turistas al hotel Restaurant Fogón Manabita, por tal motivo se seleccionaron los días 14 y 15 de agosto del 2010 ya que se trasladó para el 13 el feriado del 10 de agosto, además se desarrollaron 2 visitas adicionales los días 21 y 28 del mismo mes, aprovechando las vacaciones de la Sierra.

En estas fechas se realizaron las encuestas a los 142 clientes frecuentes del hotel, determinados por la muestra calculada para la presente investigación. Ver anexo 2.

Por último, el 4 de septiembre del 2010 se realizó la encuesta a los 22 trabajadores de la empresa, para conocer el punto de vista de los mismos, cabe indicar que se seleccionó esta fecha porque la demanda turística es baja y de esa manera los encuestados dispondrían del tiempo necesario para contestar calmadamente a las preguntas realizadas. Ver anexo 1.

De esta forma y sin mayores contratiempos se pudo conocer los puntos de vista de las personas que forman parte del sistema de administración actual del hotel Restaurant Fogón Manabita, obteniendo los datos que se requerían para hacer el análisis de los mismos.

2.2 OBJETIVOS DIAGNÓSTICOS.

Los objetivos de diagnósticos que se plantearon para la presente investigación fueron los siguientes:

- Conocer el funcionamiento y el proceso de administración actual del Hotel.
- Determinar el Grado de satisfacción de los clientes por cada servicio que presta el Hotel.
- Analizar la predisposición de los trabajadores para el uso de un Sistema Informático.
- Determinar la infraestructura informática con la que cuenta el hotel.
- Establecer qué servicios son más apropiados para sistematizar.

2.3 VARIABLES DE DIAGNOSTICO

Las variables determinadas para cada objetivo de diagnóstico son las siguientes:

- **Administrativo.-** Son los procesos que se realizan para la administración correcta del Hotel.

- **Servicios.-** Son los servicios con los que cuenta el hotel y que están dirigidos a la atención de los clientes nacionales y extranjeros.
- **Personal.-** Se refiere a los trabajadores del hotel, que realizan sus actividades ya sea por temporadas o permanentemente.
- **Tecnología.-** Son los dispositivos, hardware y software con los que el hotel cuenta en la actualidad que apoyan a la administración del mismo.
- **Administrativo.-** Son los procesos que se realizan dentro del marco de la administración del Hotel, los cuales pueden ser implementados en un sistema informático.

2.4 INDICADORES:

Para presente investigación se determinaron los siguientes sub aspectos que brindan información para cada variable de diagnóstico:

- Administrativo.
 - ✓ Plan Operativo.
 - ✓ Responsables de Actividades.
 - ✓ Toma de decisiones.
 - ✓ Horarios de Atención.
 - ✓ Procesos de atención.
 - ✓ Organigrama.
- Servicios.
 - ✓ Permanencia del Servicio.
 - ✓ Calidad del Servicio.
 - ✓ Rapidez del Servicio.
 - ✓ Comodidad.
 - ✓ Costo del Servicio.

- Personal.
 - ✓ Nivel Académico.
 - ✓ Capacitación.
 - ✓ Horario de Trabajo.
 - ✓ Experiencia.
 - ✓ Agilidad de Atención.
 - ✓ Número de personas que Atienden.
 - ✓ Disponibilidad de red informática

- Tecnología.
 - ✓ Número de computadoras
 - ✓ Características de las computadoras.
 - ✓ Disponibilidad de red informática

2.5 MATRIZ DIAGNÓSTICA

OBJETIVOS DIAGNÓSTICOS	VARIABLES	INDICADORES	TÉCNICAS	FUENTES DE INFORMACIÓN.
Conocer el funcionamiento y el proceso de administración actual del Hotel.	Administrativo.	<ul style="list-style-type: none"> • Plan Operativo. • Responsables de Actividades. • Toma de decisiones. • Horarios de Atención. • Procesos de atención. • Organigrama. 	Observación. Entrevista.	Administrador del Hotel.
Conocer el Grado de satisfacción de los Clientes por cada servicio que presta el Hotel.	Servicios.	<ul style="list-style-type: none"> • Permanencia del Servicio. • Calidad del Servicio. • Rapidez del Servicio. • Comodidad. • Costo del Servicio. 	Encuesta.	Clientes.
Conocer la predisposición de los trabajadores para el uso de un Sistema Informático.	Personal.	<ul style="list-style-type: none"> • Nivel Académico. • Capacitación. • Horario de Trabajo. • Experiencia. • Agilidad de Atención. • Número de personas que Atienden. 	Entrevista.	Trabajadores del Hotel.
Determinar la infraestructura informática con la que cuenta el hotel.	Tecnología.	<ul style="list-style-type: none"> • Número de computadoras • Características de las computadoras. • Disponibilidad de red informática 	Observación.	Hotel
Determinar qué servicios son más apropiados para sistematizar.	Administrativo.	<ul style="list-style-type: none"> • Tipo de Servicio. • Confiabilidad, Rapidez al ser manual. 	Observación. Entrevista. Encuesta.	Administrador del Hotel.

Tabla 1: Matriz Diagnóstica.

Autor: Dubal Quevedo

2.6 MECÁNICA OPERATIVA.

2.6.1 IDENTIFICACIÓN DE LA POBLACIÓN.

Para la presente investigación se han observado dos poblaciones las cuales se las identifica claramente a continuación:

La primera población está constituida por los Trabajadores del hotel, los mismos que en su total suman la cantidad de 22 personas entre administrativos, cocineros y personal de seguridad.

La segunda población la constituyen los clientes, para los cuales se aplicó un criterio, tomado como referencia el número de clientes que visitaron el hotel en los últimos cuatro años, los cuales dan un valor medio de 1500 personas anuales.

Este número promedio de la población, ha sido calculado por el propietario del hotel quien asegura recibir en sus instalaciones dicha cantidad anualmente.

2.6.2 IDENTIFICACIÓN DE LA MUESTRA.

El tipo de muestra que se aplicó a la población de clientes del hotel fue el muestreo no probabilístico intencional, ya que la selección del individuo a entrevistar se hizo según el criterio: “Se seleccionará a los clientes del hotel considerando su frecuencia de visita anual.”

DETERMINACIÓN DEL TAMAÑO DE LA MUESTRA.

Para la población de los trabajadores del hotel, no se aplicó muestreo debido a que su número es manejable (22 personas) por lo que para estos casos, las normas de investigación recomiendan trabajar con toda la población.

En cuanto a la población de los clientes del hotel, la fórmula de muestreo que mejor se acopla a los requerimientos del problema en estudio es la siguiente:

$$n = \frac{N}{E (N-1)+1}$$

Dónde:

n= Muestra

N=Población.

E=Error.

La población (N) equivale a 1500 personas obtenidas del promedio de los clientes atendidos en los últimos cuatro años, según el cálculo realizado por el propietario del hotel, se considera un error de muestreo (E) del 8%, por lo que al aplicar la fórmula anterior con estos datos se obtiene una muestra(n) aproximada de 142 personas.

2.6.3 INFORMACIÓN PRIMARIA.

Para el presente estudio, se hicieron necesarias dos técnicas de información prioritariamente, la encuesta y la entrevista en el caso más conveniente como se detalla a continuación:

Entrevista al Administrador del Hotel.- Esta entrevista es necesaria para tener un concepto más detallado de cuál es el contexto que maneja la empresa, logrando así conocer detalles del funcionamiento, para entender características específicas que ayuden al investigador a tener una visión clara del objeto en estudio.

La entrevista se la realizó al Sr. Segundo Ramón Quevedo Vergara, propietario y gerente del Hotel Restaurant Fogón Manabita de la parroquia Tonsupa en su lugar de trabajo, expresando de la mejor manera posible aclaraciones al entrevistador, quien en esta ocasión fue la persona encargada de la investigación.

De esta manera con esta entrevista se pudieron conocer aspectos importantes del sistema actual que se lleva de manera manual en la empresa y algunas desventajas del mismo.

Encuesta a los Trabajadores del Hotel.- La encuesta destinada a los trabajadores, tuvo como objetivo conocer cuál es la relación que los mismos tienen con el sistema actual de administración del hotel.

Encuesta a los Clientes del Hotel.- La encuesta dirigida a los clientes frecuentes del Hotel, está relacionada y direccionada a conocer qué piensan ellos del sistema de administración actual, para recolectar datos que permitan saber cuál es el grado de satisfacción que tienen los clientes frente a la forma en que se brindan los servicios del hotel en la actualidad.

2.7 TABULACIÓN Y ANÁLISIS DE LA INFORMACIÓN.

2.7.1 TABULACIÓN DE LA ENCUESTA REALIZADA A LOS TRABAJADORES.

PREGUNTA #1:

¿Cuántas horas de servicio presta al hotel aproximadamente por día?

	f	%
De 1-4 Hrs.	0	0
De 4-8 Hrs.	1	4
De 8-12 Hrs.	7	32
Más de 12Hrs.	14	64
TOTAL	22	100

Tabla 2: Horas de servicio diarias de trabajadores.

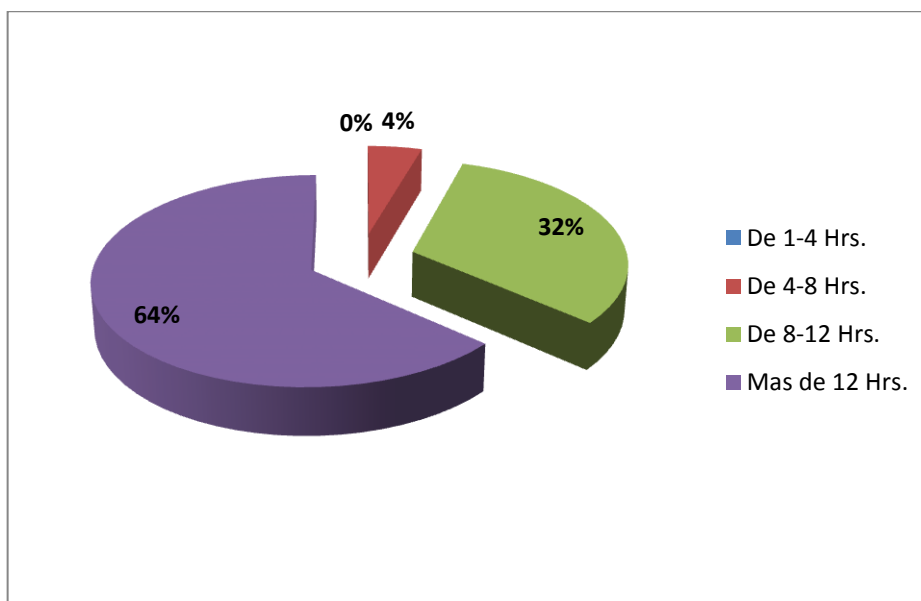


Gráfico 1: Horas de servicio diarias de trabajadores.

Fuente: Encuesta, Agosto del 2010.

Autor: Quevedo Dubal.

ANÁLISIS:

En el hotel restaurant fogón manabita, la mayoría de personas trabajan más de 12 horas, en temporadas playeras ya que las labores inician desde muy temprano y terminan

muchas veces a la madrugada del siguiente día, en menos porcentaje, pero no mínimo, es el de las personas que cumplen entre 8-12 horas de labores al día. Con menor porcentaje se encuentran las personas que trabajan de 4 a 8 horas, y ninguna persona trabaja entre 1 a 4 horas diarias, esto se debe a que no se contratan personas con ese número de horas de labores. Es evidente que las personas tienden a estar muy cansadas al final de una labor, la cual por lo menos en la parte administrativa y contable, es agotador tratándose de llevar las cuentas de forma manual.

PREGUNTA #2:

¿En qué medida maneja usted programas informáticos como Microsoft Office Word y Excel?

	f	%
Nada	8	36
Medianamente	12	55
Mucho	2	9
TOTAL	22	100

Tabla 3: Nivel de conocimientos básicos de computación de trabajadores.

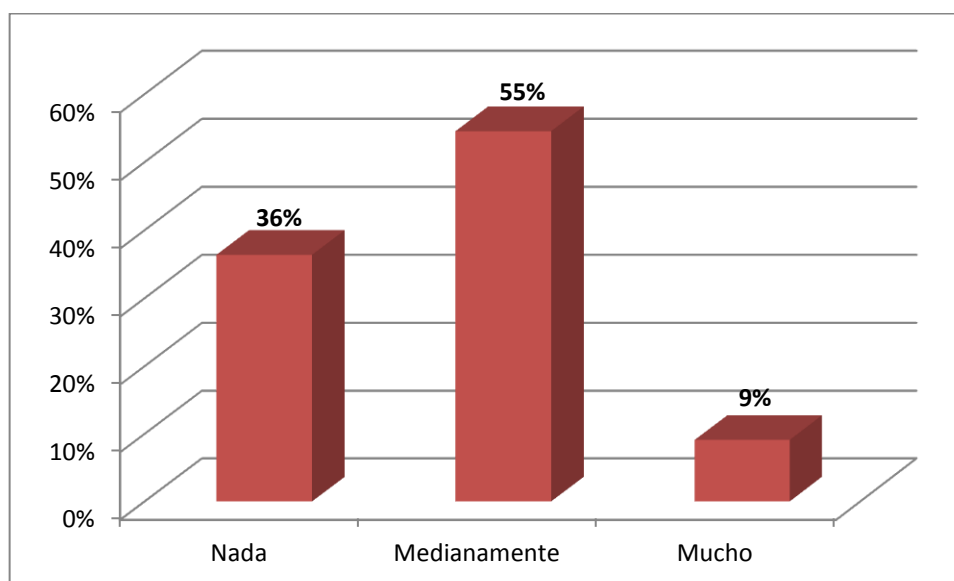


Gráfico 2: Nivel de conocimientos básicos de computación de trabajadores.

Fuente: Encuesta, Agosto del 2010.

Autor: Quevedo Dubal.

ANÁLISIS:

El porcentaje de personas que no tienen conocimientos de computación básica se encuentra en un nivel más bajo respecto a las que saben computación de manera medianamente, pero aun así, el índice es muy alto, y esto se debe a que las personas contratadas para servicios no administrativo, tienen un nivel de educación básico, esto indica que las personas necesitan de capacitaciones para que la empresa cuente con personas más preparadas en el campo informático. El porcentaje de personas que se encuentran en nivel de mucho, corresponde a la parte administrativa del hotel, son personas que sí tienen buenas bases de computación básica.

PREGUNTA #3:

¿De implementarse un Sistema Informático, estaría dispuesto a capacitarse para el uso del mismo?

	f	%
SI	21	95
NO	1	5
TOTAL	22	100

Tabla 4: Disposición para capacitación informática de trabajadores.

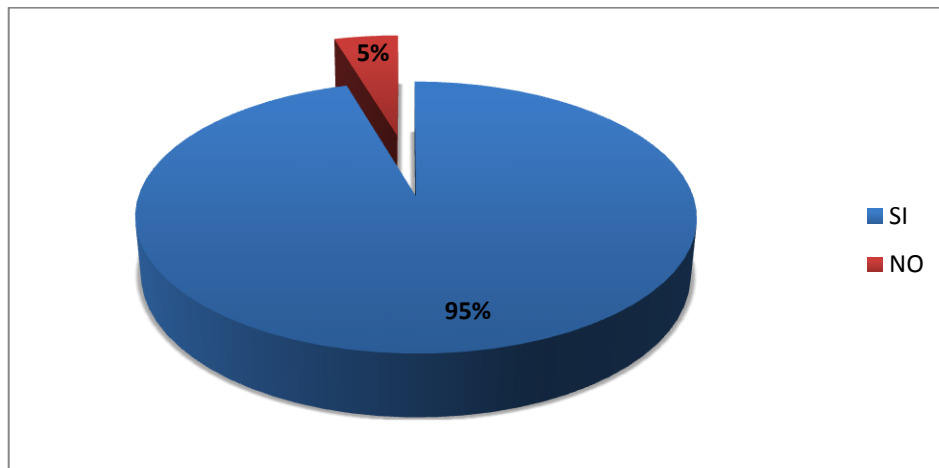


Gráfico 3: Disposición para capacitación informática de trabajadores.

Fuente: Encuesta, Agosto del 2010.

Autor: Quevedo Dubal.

ANÁLISIS:

Como se pudo observar en la pregunta 2, a pesar que hay un porcentaje alto de personas que no conocen de computación, en el grafico actual, es evidente que los trabajadores si quieren adquirir conocimientos sobre estos temas, ya que a pesar de que no la saben, están conscientes que es un ámbito tecnológico que está revolucionando la forma vivir en la actualidad, sin embargo, existe porcentaje mínimo de personas que no quisieran capacitarse.

PREGUNTA #4:

¿Cree que con el uso de un Sistema Informáticos los procesos de Administración (reservaciones, restaurant, bar, facturación) serían más eficientes?

	f	%
SI	20	91
NO	0	0
NO LO SÉ	2	9
TOTAL	22	100

Tabla 5: Eficiencia que se cree tener con un sistema de administración informático.

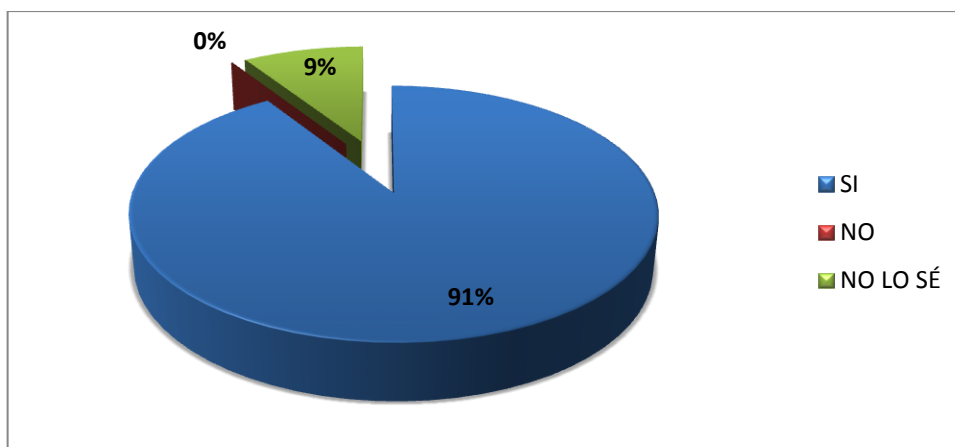


Gráfico 4: Eficiencia que se cree tener con un sistema de administración informático.

Fuente: Encuesta, Agosto del 2010.

Autor: Quevedo Dubal.

ANÁLISIS:

Los resultados ponen en evidencia que el grupo de personas que creen que un sistema de administración informático ayudaría a la empresa con la administración de la misma, es totalmente evidente a las que piensan que no, siendo así que de este último grupo, se tiene un porcentaje de cero, sin embargo existe un mínimo porcentaje de personas que no están seguras o no tienen idea de cuál sería el aporte del mismo a la administración de la empresa.

PREGUNTA #5:

¿Estaría de acuerdo con que sus roles de pago y anticipos se registren en un sistema de aplicación para el mejor control de los mismos?

	f	%
SI	21	95
NO	0	0
NO LO SÉ	1	5
TOTAL	22	100

Tabla 6: Predisposición para formar parte del sistema informático.

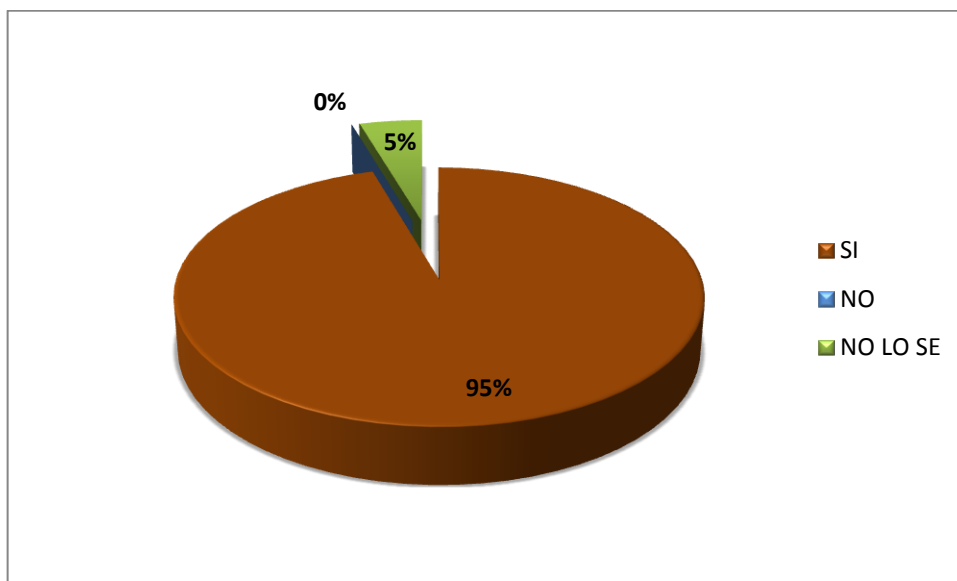


Gráfico 5: Predisposición para formar parte del sistema informático.

Fuente: Encuesta, Agosto del 2010.

Autor: Quevedo Dubal.

ANÁLISIS:

La mayoría de personas están de acuerdo con que es mucho mejor que los registros de los anticipos y de los sueldos sean asistidos mediante un sistema de información, por el cual los mismos expresaron que sería mucho más fiable que todos los movimientos económicos que ellos realicen se registren, para evitar que hayan inconvenientes, discusiones, entre otros, un mínimo porcentaje en el gráfico, representa el no saber o no estar seguro si realmente sería mejor que sea así. Mientras ningún elemento de la población se opone a que sus roles de pagos, y anticipos se registren en un sistema de aplicación.

PREGUNTA #6:

¿Cree que son óptimos los procesos que se realizan dentro del Hotel, para lograr una buena atención a los clientes?

	f	%
SI	6	27
NO	4	18
MAS O MENOS	12	55
TOTAL	22	100

Tabla 7: Atención a clientes.

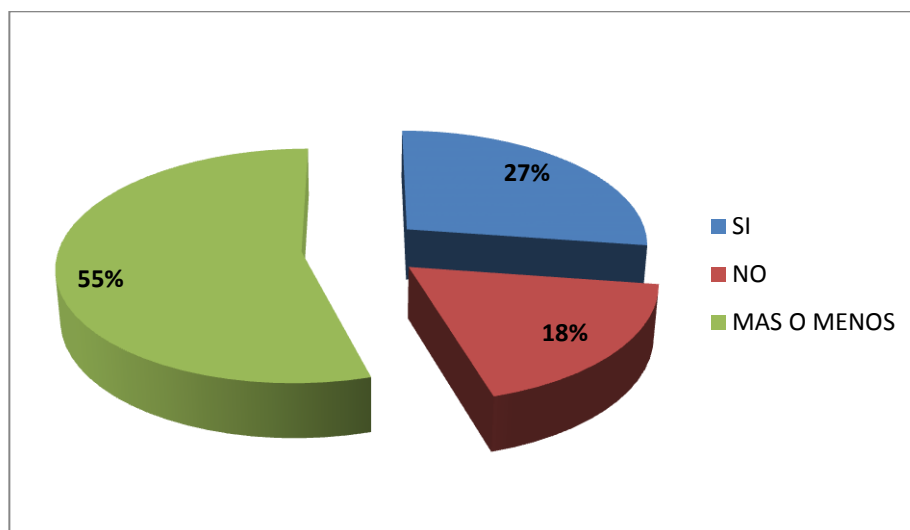


Gráfico 6: Atención a clientes.

Fuente: Encuesta, Agosto del 2010.

Autor: Quevedo Dubal.

Análisis:

Los resultados que se muestran en el gráfico anterior, indican claramente que el sistema actual de administración no es óptimo, hay un gran porcentaje que revela que los procesos son considerados por los trabajadores como regulares por el motivo de que no existe la rapidez, ni el control necesario en algunos procesos del hotel como en la facturación y en las reservaciones, otra parte de la misma, cree que no son óptimos, sin embargo, una parte de esta población, cree que sí lo son, porque consideran que tienen buenos servicios y tratan de que el cliente se sienta de la mejor manera.

PREGUNTA #7:

¿En qué medida el sistema manual de administración actual del hotel satisface todas las necesidades del mismo?

	f	%
EXCELENTE	3	14
MUY BUENO	6	27
BUENO	4	18
REGULAR	9	41
MALO	0	0
TOTAL	22	100

Tabla 8: Efectividad del sistema de administración manual.

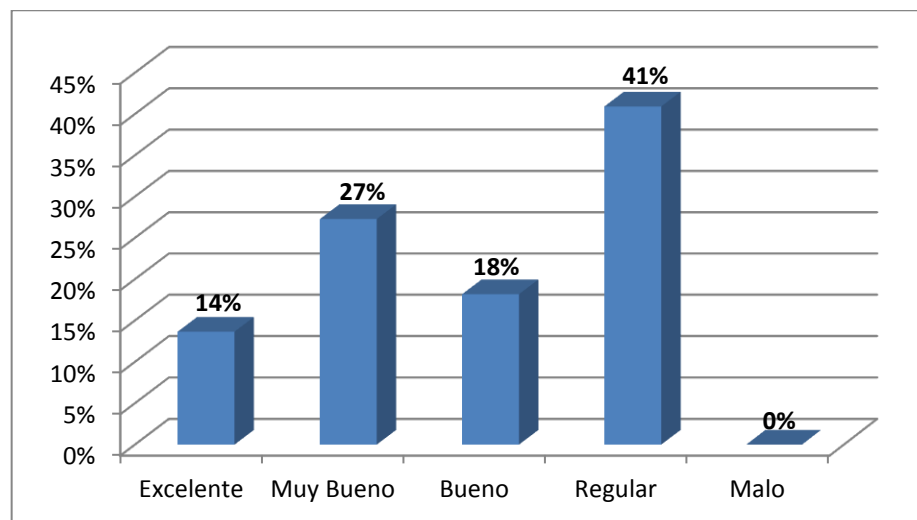


Gráfico 7: Efectividad del sistema de administración manual.

Fuente: Encuesta, Agosto del 2010.

Autor: Quevedo Dubal.

ANÁLISIS:

Los trabajadores creen que el sistema de administración actual no cumple con todas las expectativas de los mismos, porque creen que falta el control en los pagos y en pedidos. Aun así recalcan aspectos positivos del mismo por lo que es considerado regular. El siguiente porcentaje, revela que hay un buen grupo de la población que considera que el sistema es muy bueno sin dejar de lado que el desarrollo del hotel mejoraría si se implementaran herramientas administrativas más apropiadas para realizar ciertos procesos de manera más rápida y eficaz.

2.7.2 TABULACIÓN DE LA ENCUESTA REALIZADA A LOS CLIENTES DEL HOTEL.

PREGUNTA #1:

¿Se siente satisfecho con el proceso de facturación de los servicios que se lleva actualmente en el Hotel Fogón Manabita?

	f	%
SI	92	65
NO	0	0
MAS O MENOS	50	35
TOTAL	142	100

Tabla 9: Satisfacción de clientes por el proceso de facturación actual.

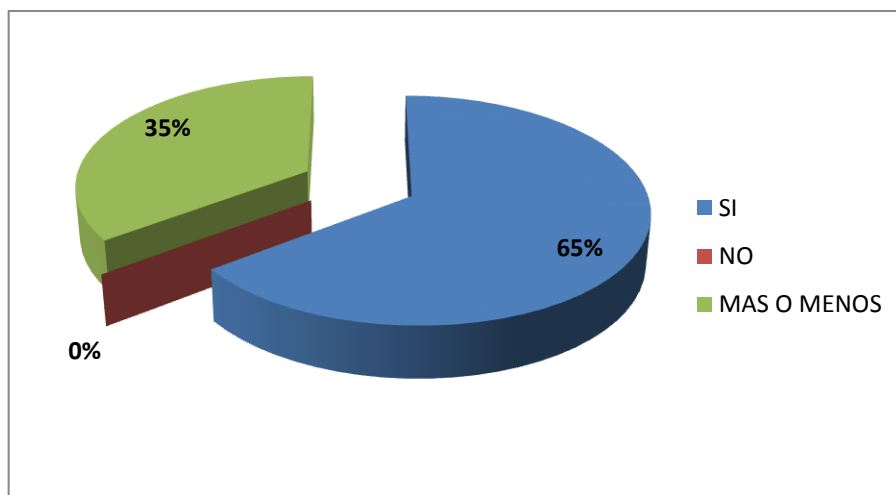


Gráfico 8: Satisfacción de clientes por el proceso de facturación actual.

Fuente: Encuesta, Agosto del 2010.

Autor: Quevedo Dubal.

ANÁLISIS:

Es muy bueno saber que hay un porcentaje nulo de personas que no están satisfechas por el proceso de facturación y que el mayor porcentaje esté sobre los clientes que consideran estar satisfechos del proceso de facturación actual, sin embargo es importante observar que hay un número de la población que no está totalmente satisfechas por varios motivos que se suscitaron al momento o de recibir su factura, este porcentaje cree que el proceso no es el ideal, que sería mucho mejor para ellos y para los cajeros que se realicen los cálculos de los valores de las facturas de manera automática para evitar confusiones.

PREGUNTA #2:

¿Está satisfecho con el proceso de asignación de habitaciones?

	f	%
SI	109	77
NO	33	23
TOTAL	142	100

Tabla 10: Satisfacción de clientes por el proceso de reservación actual.

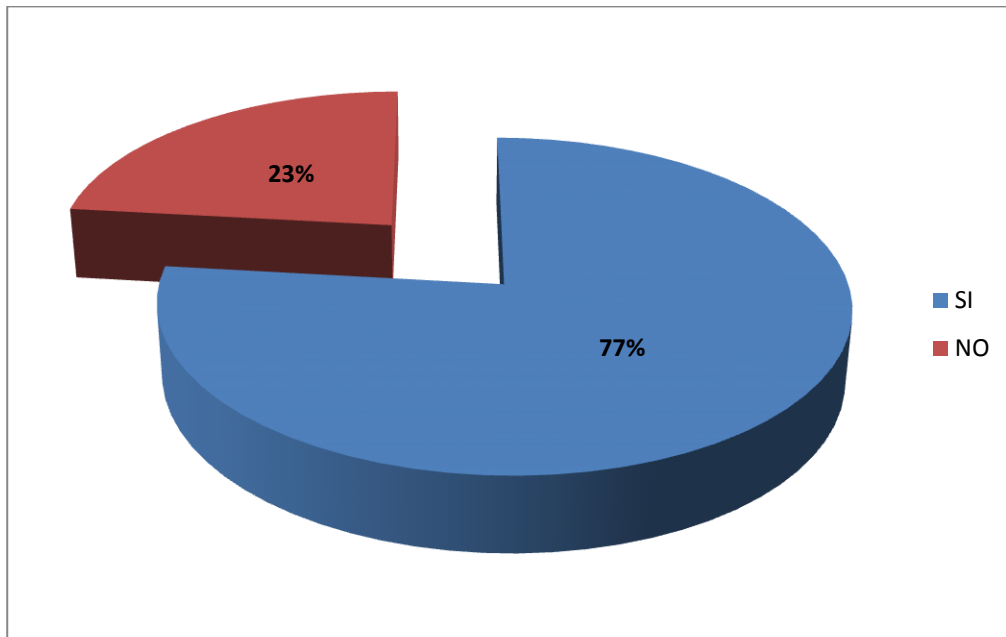


Gráfico 9: Satisfacción de clientes por el proceso de reservación actual.

Fuente: Encuesta, Agosto del 2010.

Autor: Quevedo Dubal.

ANÁLISIS:

Un buen porcentaje está satisfecho con el proceso de asignación de habitaciones, lo cual es favorable para el Hotel, sin embargo, aunque sea pequeño el porcentaje de personas que no está totalmente de acuerdo con el proceso manual, ya que al llegar, existió un poco de incertidumbre por parte de los administradores al confirmar el estado de la habitación, es decir, no se tuvo respuesta automática al momento de la confirmación de la disponibilidad de la misma al instante que llegó el cliente.

PREGUNTA #3:

¿Le gustaría que el Hotel cuente con un mejor control para la devolución de objetos olvidados por los clientes?

	f	%
SI	125	88
NO	17	12
TOTAL	142	100

Tabla 11: Control y devolución de objetos extraviados.

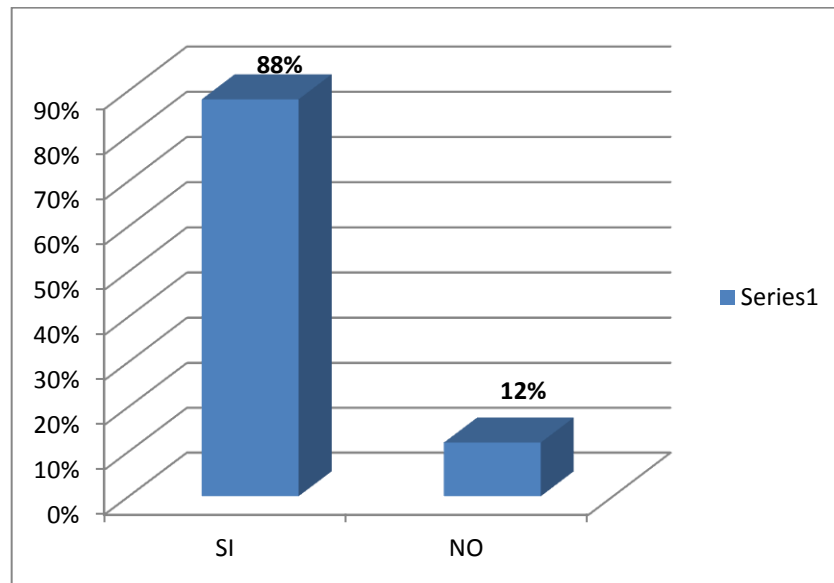


Gráfico 10: Control y devolución de objetos extraviados.

Fuente: Encuesta, Agosto del 2010.

Autor: Quevedo Dubal.

ANÁLISIS:

El máximo porcentaje en el gráfico representa a las personas que creen que se debe tener un sistema con el cual se enlace directamente al cliente con la habitación y de esa manera, si alguno de ellos por olvido deja una de sus pertenencias, se la pueda guardar con una referencia única en la cual se determine de qué cliente es dicho objeto, y guardarlo en bodega hasta una próxima visita, ya que en el caso de ser manual este control, será difícil de implementar, o si se implementa sería difícil de controlar. Hay un mínimo porcentaje de personas que creen que esto no sería muy importante, pero es muy bajo respecto al primer porcentaje.

PREGUNTA #4:

¿Está conforme con la manera que se gestiona el proceso de pedidos de órdenes en el Restaurant (existieron equivocaciones y extravío)?

	f	%
SI	98	69
NO	40	31
TOTAL	142	100

Tabla 12: Clientes con problemas en el proceso de órdenes de pedidos.

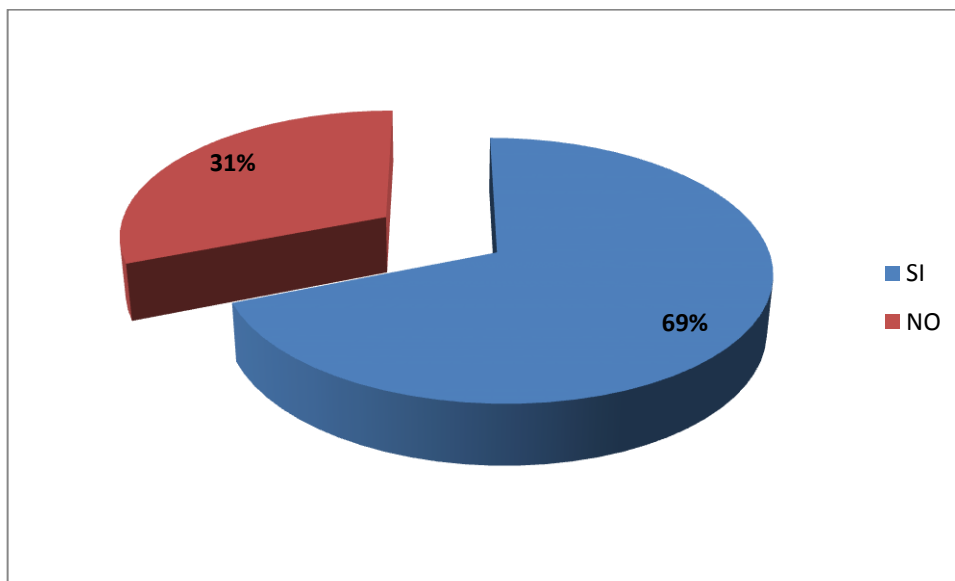


Gráfico 11: Clientes con problemas en el proceso de órdenes de pedidos.

Fuente: Encuesta, Agosto del 2010.

Autor: Quevedo Dubal.

ANÁLISIS:

Como se observa en el gráfico, el número de personas que han tenido alguna dificultad, en el proceso de pedidos es menor que el que no lo ha tenido, pero aun así es considerable, ya que el sistema de administración actual en distintos caso acarreó como consecuencia la demora o equivocación en el orden de llegada del pedido del cliente.

PREGUNTA #5:

¿Cree que hay un buen control de parte del Hotel para no caer en el error de atender primero a Clientes que llegaron después de otros?

	f	%
SI	105	74
NO	37	26
TOTAL	142	100

Tabla 13: Clientes con problemas en el proceso de despacho de pedidos.

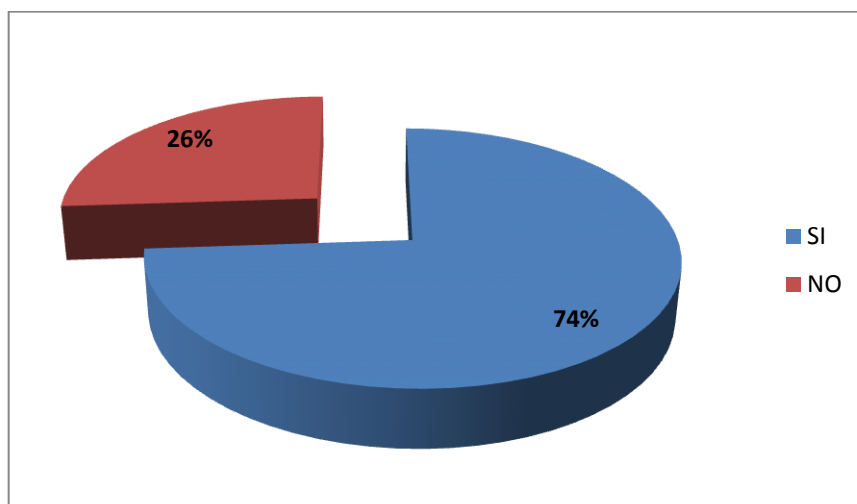


Gráfico 12: Clientes con problemas en el proceso de despacho de pedidos.

Fuente: Encuesta, Agosto del 2010.

Autor: Quevedo Dubal.

ANÁLISIS:

La manera en que se controlan procesos con el sistema de administración actual es aceptable, sin embargo no garantiza en un mayor porcentaje que se cumplan correctamente aspectos como el despachar pedidos, atendiendo a las personas que llegan después de otras, siendo así esto causa en algunos momentos de enojos de los clientes e incluso hasta anulación de pedidos.

PREGUNTA #6:

¿Le gustaría tener la posibilidad de separar una habitación y reservarla para planificar sus vacaciones a tiempo?

	f	%
SI	142	100
NO	0	0
TOTAL	142	100

Tabla 14: Clientes dispuestos a reservar habitaciones antes de temporada.

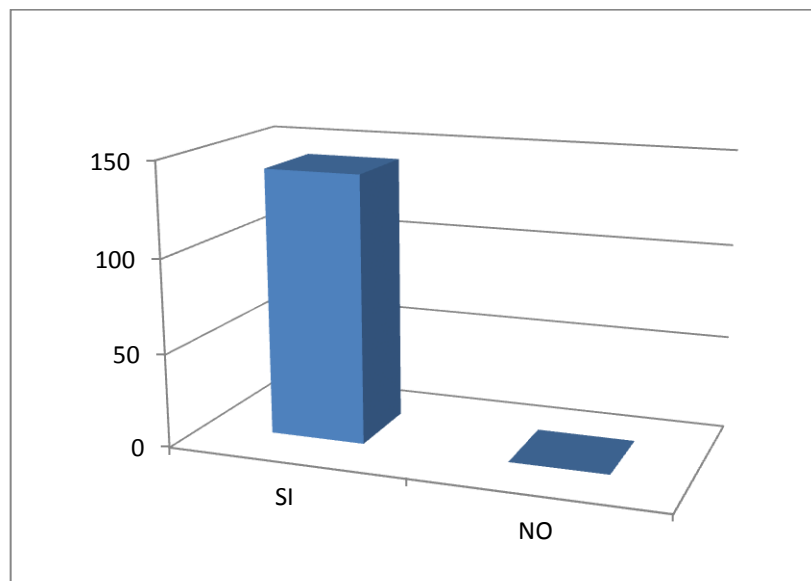


Gráfico 13: Clientes dispuestos a reservar habitaciones antes de temporada.

Fuente: Encuesta, Agosto del 2010.

Autor: Quevedo Dubal.

ANÁLISIS:

De manera absoluta, todos los clientes demostraron estar de acuerdo con que el hotel cuente con un sistema que permita de mejor manera organizar las reservaciones que realizan los clientes, incluso antes de que sea tiempo de temporada, expresando que si existiera un sistema que controle las fechas que se reserva una habitación, habría mejor control, y de esta manera mejor organización, brindando así un mejor servicio y de manera más efectiva a las personas que hacen uso del mismo.

2.7.3 ANÁLISIS DE LA ENTREVISTA.

Entrevista Realizada al Sr. Quevedo Vergara Segundo, gerente propietario del hotel restaurant Fogón Manabita.

Fecha: 7 de Agosto del 2010.

Como se mencionó anteriormente esta entrevista tuvo como finalidad, conocer aspectos importantes del Hotel Restaurant Fogón Manabita de la parroquia Tonsupa, como lo son: organización, servicios y procesos, estructura, infraestructura tecnológica entre otros, de los cuales se hablará a continuación.

La misión del hotel en estudio, es dar buen servicio a los turistas nacionales e Internacionales, y su misión es lograr alcanzar una alta calidad de atención y obtener un reconocimiento a nivel nacional y de esta manera promover el desarrollo turístico del Ecuador.

El hotel consta con seis servicios principales entre los que se encuentran: el servicio de hospedaje con vista al mar, servicio de restaurant, servicio de piscina, parqueadero privado, bar y salón de actos, por estos motivos, cumple con obligaciones legales con el Estado, encontrándose así entre estas instituciones, el municipio de Atacames, el Instituto Ecuatoriano de Seguro Social, Servicio de Rentas Interna, Ministerio de Salud Pública, Cuerpo de Bomberos del Cantón Atacames, Cámara Provincial de Turismo, Ministerio de Turismo y la SAYCE (Sociedad de Autores y Compositores del Ecuador) este último se lo identificó como la institución que otorga permisos de audio en el hotel, bar y salón de actos.

En la actualidad, la empresa consta con dos administradores, uno se encarga únicamente de la parte del hotel y es responsable del mismo, mientras el otro se encarga del restaurant, y ambos, rinden cuenta al gerente propietario, quien es el encargado de realizar labores como la adquisición de materia prima y pago de sueldo a los trabajadores.

El personal necesario para el correcto funcionamiento de la empresa es aproximadamente 22, pero este número no es fijo ya que por lo general se contrata al personal dependiendo de la afluencia de turistas en los feriados, los mismos que se distribuyen de la siguiente manera: entre los cocineros se tienen 2 jefes de cocina y 7 ayudantes, 3 camareros, 5 saloneros, 1 cajero, 1 administrador del hotel, 1 administrador del restaurant y 2 personas en el bar.

El principal problema que se dio a conocer fue el de las confusiones que existen en procesos como la facturación y despacho de pedidos, debido a la falencia que existe en la forma de administración manual, y que es más frecuente por la aglomeración de los clientes.

En cuanto a la infraestructura tecnológica con la que cuenta sistema de administración del hotel consta una caja registradora, y se dio a conocer que hay un computador pero no se lo usa para la administración, sino que esta forma parte del equipo de audio.

2.8 MATRIZ FODA.

FORTALEZAS.

- Cuenta con recursos para adquirir infraestructura tecnológica.
- Es una empresa bien posicionada en el campo hotelero.
- El trabajo en equipo dentro del hotel y en el restaurante, es una cualidad muy importante.
- Posee varios servicios que otros hoteles no cuentan, parqueadero privado, piscina con tobogán, ascensor, habitaciones con vista al mar.
- El personal está acostumbrado a llevar el control de todas las operaciones que se realizan en el hotel.

OPORTUNIDADES.

- La ubicación del hotel hace que sea un lugar clave para la visita de clientes.
- Aumento de la actividad turística de Tonsupa.
- La competencia en muchos casos no cuentan con un sistema de administración.

- Algunos hoteles no cumplen con las normas de facturación porque no tienen un control adecuado.
- Los trabajadores están dispuestos a recibir capacitaciones.
- Tiene clientes fijos y satisfechos.

DEBILIDADES.

- Al ser un hotel-restaurant de gran capacidad de clientela, la atención al cliente, tiende a dejar de ser personalizada.
- El sistema actual presenta inconvenientes para la adecuada administración del hotel.
- Carencia de registro real y oportuno de ingresos y egresos, para el correcto abastecimiento de productos.
- El sistema de administración actual no cuenta con la infraestructura tecnológica.
- Se presentan inconvenientes y retrasos al momento de inventariar las operaciones económicas realizadas por el hotel.
- Un grupo considerable de trabajadores no cuentan con conocimientos básicos de computación.
- En ocasiones, se presentan problemas en el manejo, reserva y asignación de las habitaciones.

AMENAZAS.

- La competencia puede implementar procesos tecnológicos de control que los haga más fuertes.
- Los clientes que no son satisfechos por causa de inconvenientes, no regresen más.
- La empresa no cuenta con información histórica que permita la toma de decisiones.
- Que la competencia capacite a su personal en el manejo de herramientas informáticas.

2.9 MATRIZ DE LA ESTRATEGIAS FA, FO, DO, DA.

	AMENAZAS	OPORTUNIDADES
F O R T A L I Z A S	<p>Predisposición económica y del personal de hotel para el uso de un sistema informático de administración hotelera, por lo que existe la factibilidad de implementarlo antes que la competencia.</p> <p>Al desarrollar un sistema informático, se daría un buen servicio al cliente y se evitarían los inconvenientes y los clientes insatisfechos.</p>	<p>Se facilita el uso adecuado del sistema informático de administración ya que el personal está acostumbrado a llevar el control de todas las operaciones que se realizan en el hotel.</p> <p>El sistema informático corregirá los inconvenientes que se pueden presentar en una facturación manual por datos incorrectos y permite el ahorro de tiempo.</p>
D E B I L I D A D E S	<p>Que la administración decida invertir a medias o definitivamente no invertir en la infraestructura tecnológica que se requiere para el funcionamiento adecuado del sistema informático.</p> <p>Que el personal no esté lo suficientemente capacitado en el manejo del sistema informático, lo que generaría que el sistema funcione inadecuadamente.</p>	<p>Con el sistema informático se minimizarían al máximo los problemas de manejo, reserva y asignación de habitaciones ya que se llevaría un control automatizado de los mismos.</p> <p>El sistema permitirá el registro eficiente de todas las actividades económicas que realice el hotel, lo que permitirá a la administración tomar las decisiones adecuadas basadas en datos oportunos y reales.</p>

Tabla 15. Matriz de análisis FA, FO, DO, DA.

Autor: Dubal Quevedo

2.10 DIAGNÓSTICO DEL PROBLEMA.

Gracias a los estudios que se realizaron sobre el sistema de administración actual que se lleva dentro del hotel restaurant fogón manabita, se han determinado los siguientes problemas:

En el proceso de facturación existen falencias, ya que al ser realizado manualmente se cometen errores de cálculo de los valores que en ella se detallan.

El proceso de asignación de habitaciones, es un poco rudimentaria y poco precisa ya que no se cuenta con una herramienta que permita determinar de forma inmediata cuales son las habitaciones que están disponibles con una certeza de ser así, si no que se recurre a preguntar entre ellos cual es la que se encuentra en capacidad para ser rentada.

Los roles de pagos de los trabajadores se realizan de forma manual, así como también los anticipos y todas las cuentas que tienen que ver con ellos, como descuentos por no cumplir con el horario de trabajo entre otros.

Con el sistema de administración actual se torna impreciso, tedioso, y retardado tratar de tener al final de la labor diaria o en un determinado tiempo, cuál ha sido el total en ventas que se ha alcanzado, siendo así esta una problemática que incide directamente con la toma de decisiones que vallan a favor del crecimiento de la empresa.

Los pedidos que realizan los clientes, son atendidos en un orden determinado, pero al existir la aglomeración de estos se pierde el orden de llegada ya que en muchos casos, se repiten los números de orden ocasionando que se atiendan primero a clientes que llegan después de otros.

CAPITULO III: PROPUESTA.

3.1. ANTECEDENTES DE LA PROPUESTA.

El “Bar Hotel Restaurant Fogón Manabita” es una empresa que inició sus actividades como microempresa de servicios alimenticios, está administrada por sus propietarios desde el inicio, hace 8 años aproximadamente, con el tiempo ha pasado a ser uno de los más grandes negocios de hotelería y turismo en la parroquia Tonsupa del cantón Atacames de la provincia Esmeraldas.

En esta empresa los controles de administración son poco eficientes ya que se la realiza de forma manual, tanto con la parte organizacional interna, como la de control de reservaciones y del servicio de alimentación.

Al ser dirigida por sus propietarios, no hay un control exclusivo de los ingresos y egresos reales que esta entidad proporciona (sin dejar de lado el proceso de facturación).

Frente a esta situación, analizando el proceso de funcionamiento del establecimiento, se ha observado muchas faltas de control de las actividades que se realizan en el mismo, además se debe optimizar la atención al cliente ya que los procesos de reservación y de restaurant, se los realiza de forma manual, lo cual origina ciertos inconvenientes.

Este proyecto realizado se circunscribió en el estudio de la infraestructura y procesos de ventas en el restaurant, facturación general, gestión de reservaciones y control de recursos humanos, y así plasmarlos en el sistema informático desarrollado, para la administración del hotel.

3.2. ANÁLISIS DEL SISTEMA.

3.2.1. ANÁLISIS DEL REQUERIMIENTO.

Se requiere un sistema que permita asignar de manera precisa turnos a los pedidos que se realizan en el restaurant para que estos sean atendidos por los meseros en manera secuencial respetando el orden de llegada de cada cliente para que el primero que llegue, sea el primero en ser atendido.

Luego de ellos el sistema debe permitir realizar la facturación de los pedidos que se realizan en el restaurant y a la vez, implemente el registro de las ventas que se realizan, permitiendo posteriormente la obtención de información para la administración de la empresa.

Además se requiere un sistema que permita realizar las reservaciones de habitaciones, implementando el registro de Check In, el proceso de Check Out, llevando el registro de los anticipos que el cliente realiza, el mismo debe gestionar correctamente la disponibilidad de las habitaciones para que permita realizar consultas de disponibilidad de manera ágil y efectiva. A la vez, el sistema debe de realizar la facturación de los valores de reservaciones, llevando un historial de todas las reservaciones realizadas, el registro de los huéspedes para posteriormente obtener reportes respecto a los procesos dichos con anterioridad.

El sistema debe permitir también el registro adecuado de las asistencias de los trabajadores, en el momento que inician y finalizan sus labores, que registre los anticipos, préstamos a la empresa y las multas a dichos trabajadores para que al final del periodo laboral, el sistema genere el Rol de Pagos a los trabajadores con los valores acreditados y debitados, llevando de la misma forma que los otros procesos un historial que posteriormente permita realizar reportes de cada transacción por la fecha que fue realizado.

Finalmente pero no menos importante se requiere de un sistema que sea amigable con el usuario, ágil, fácil de usar, seguro y efectivo.

3.2.2. ESTUDIO DE LOS REQUERIMIENTOS BÁSICOS.

Para asignar de manera precisa turnos el sistema implementará los procesos necesarios para que el control y asignación de turnos sea lo más rápido posible, permitiendo asignar al mismo, los datos del mesero, y la mesa a la que se refiere.

Para que se realice la facturación de los pedidos, el sistema utilizará un registro donde se encuentren los productos y la carta que se vende en el restaurant para poder registrar y calcular el valor del pedido que luego será facturado, para ello se debe implementar el registro de los clientes mediante su cedula de identidad para que sean identificados posteriormente tanto en el restaurant como en el hotel.

Se debe también crear registros de las habitaciones con las características principales para que estas sean asignadas apropiadamente al cliente que las requiera, luego de ello, se podrá crear las reservaciones con los datos de los clientes, luego realizar el Check In, el Check Out, realizar los abonos y anular una reservación en caso de ser necesario.

También se debe desarrollar las opciones necesarias para el registro de los datos e información del trabajador, para poder posteriormente el registro de las asistencias, de los anticipos que realizan con la información referente a dicho proceso, esto permitirá que el sistema genere los roles de cada trabajador descontando o acumulando de manera automática los valores correspondientes.

Luego de los procesos de registros anteriormente se debe trabajar en los reportes que permitan obtener la información que el usuario requiera en el momento que lo desee.

Los procesos específicos requeridos, los mismos que se sistematizarán son los siguientes:

- Gestión de Turnos.
- Facturación de Restaurant.

- Reservaciones de Habitaciones, Check In, Check Out.
- Facturación de Reservaciones.
- Control de Asistencia.
- Control y registro de anticipos, registros de pagos de sueldos y realización de Roles a los trabajadores.
- Reportes de ventas.
- Reportes de Reservaciones.
- Reporte de Habitaciones.
- Registro de Clientes del Restaurant.
- Registro de Huéspedes.
- Registro de agenda de recurso humano.
- Agenda de Recordatorios.

3.2.3. ACTIVIDADES DE LA DETERMINACIÓN DE REQUERIMIENTOS.

Las actividades que se proponen en el sistema se muestran en la siguiente tabla:

Proceso	Actual	Propuesto
Gestión de Turnos.	Manual.	Proceso Sistematizado de Asignación de Turnos.
Facturación de Restaurant.	Manual.	Sistema que facture y registre las ventas de restaurant.
Reservaciones de Habitaciones, Check In, Check Out.	Manual, Libro de Registros de Reservaciones.	Sistema que gestione la disponibilidad de las habitaciones y permita registro de las reservaciones, realizar el Check in y Check Out de los clientes.
Control de Asistencia.	Manual, libro de registros.	Sistema que registre y controle las asistencias de los trabajadores.

Control y registro de anticipos, registros de pagos de sueldos y realización de Roles a los trabajadores.	Manual, Libro de registros de trabajadores.	Sistema que registre y controle los anticipos e historial de pagos de sueldos de trabajadores.
Reportes de ventas.	Manual.	Reportes de Ventas.
Reportes de Reservasiones.	Manual.	Reportes de Reservasiones.
Reporte de Habitaciones.	Manual.	Reporte de Habitaciones.
Registro de Clientes del Restaurant.	No existe.	Base de datos del sistema.
Registro de Huéspedes.	Manual, Libro de Huéspedes	Opción del sistema que registre los huéspedes en la base de datos con los datos requeridos.
Registro de agenda de recurso humano.	Manual.	El sistema implementará una opción que gestione los datos referentes al personal de la empresa.
Agenda de Recordatorios.	Manual.	Agenda de Notas Rápidas.

Tabla 16. Tabla de Determinación de Requerimientos.

Autor: Dubal Quevedo

3.3. DISEÑO.

Para el diseño se utilizará los diagramas de modelado de sistemas entre ellos:

- Diagrama de Flujo de Datos.
- Modelo Entidad Relación.
- Modelo Relacional.
- Diccionario de Datos.
- Diagrama de Clases.

3.3.1. DIAGRAMAS DE FLUJO DE DATOS DEL SISTEMA.

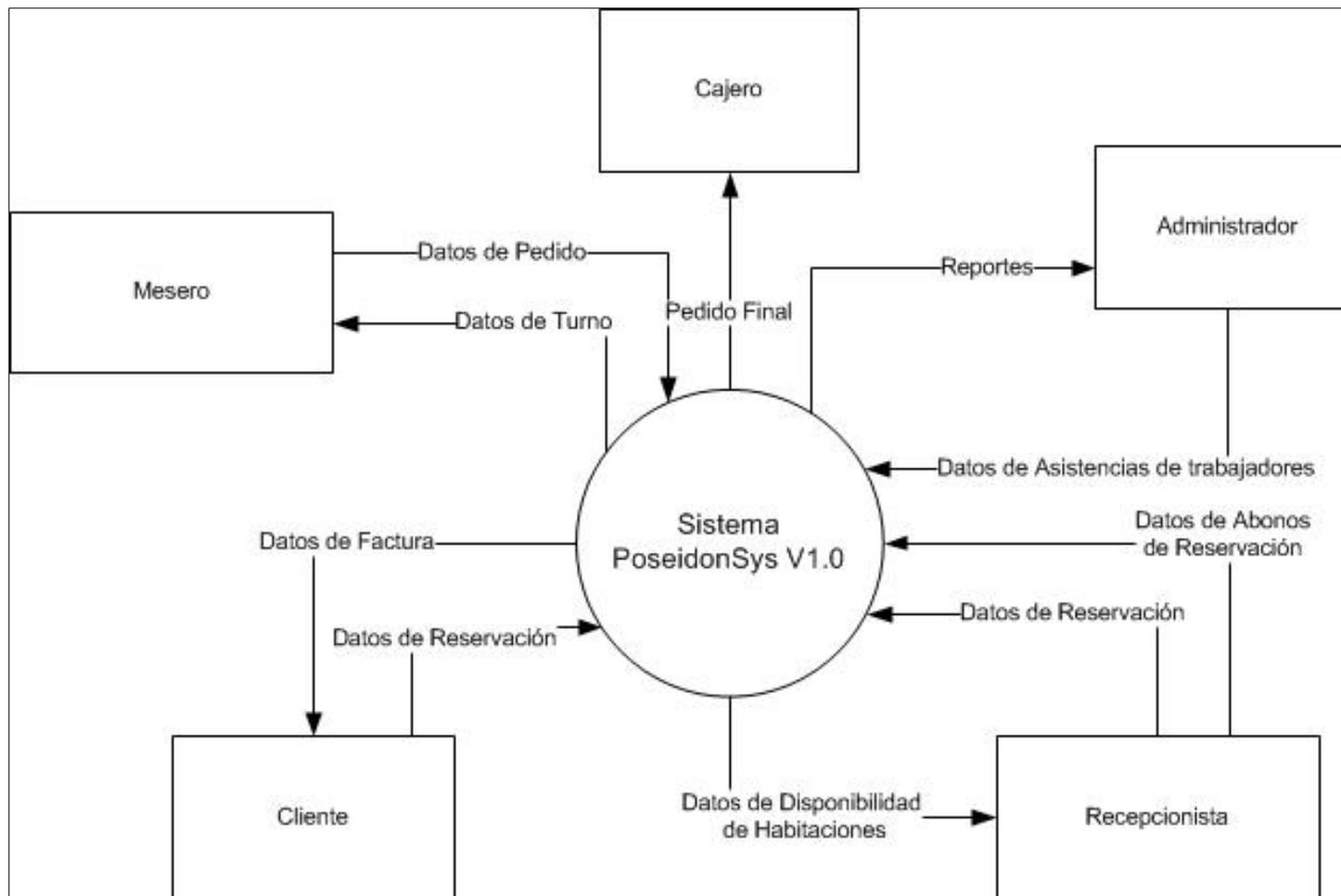


Diagrama de Flujo de Datos. Nivel 0.

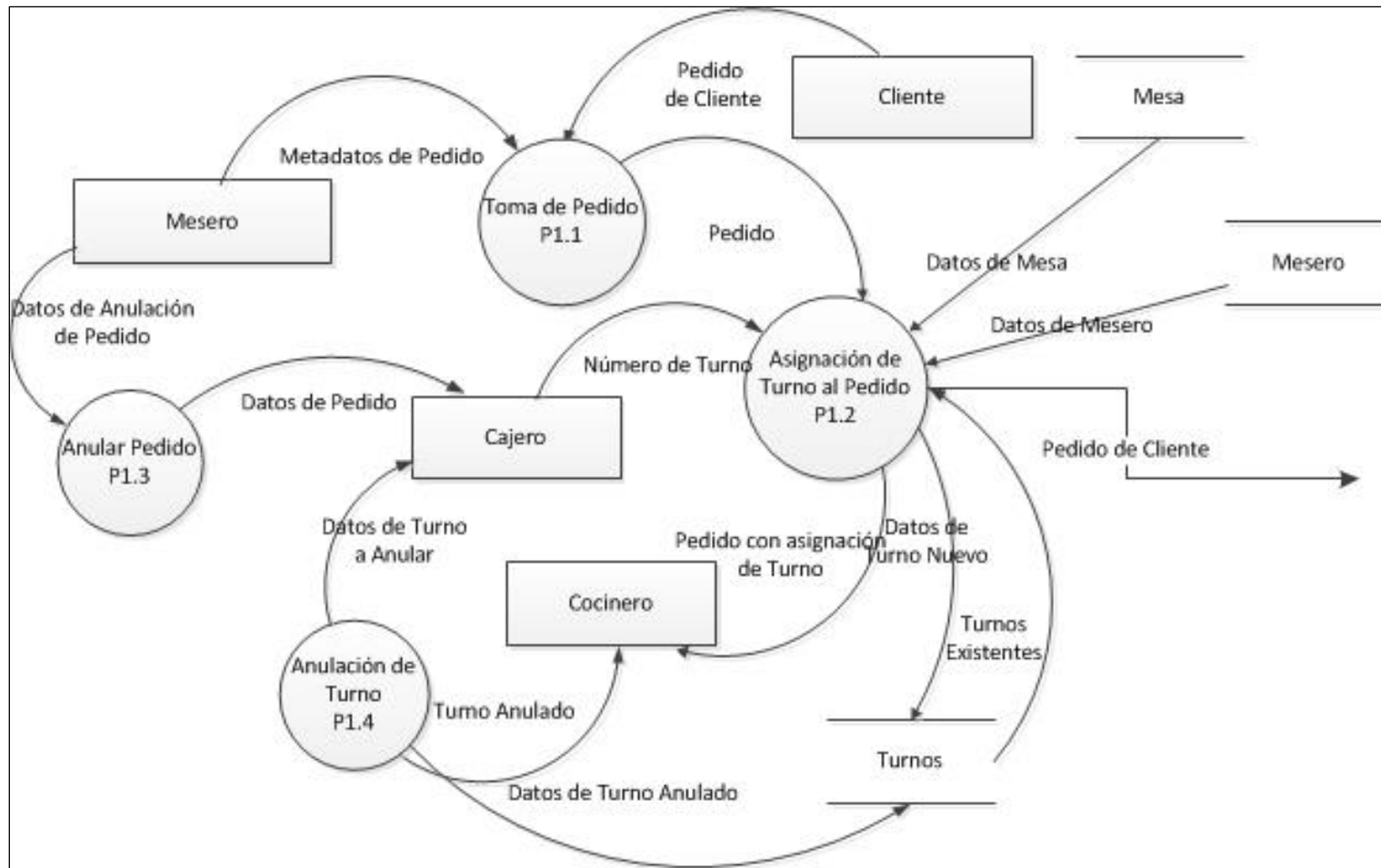


Diagrama de Flujo de Datos de Pedidos de Restaurant. Nivel 2.

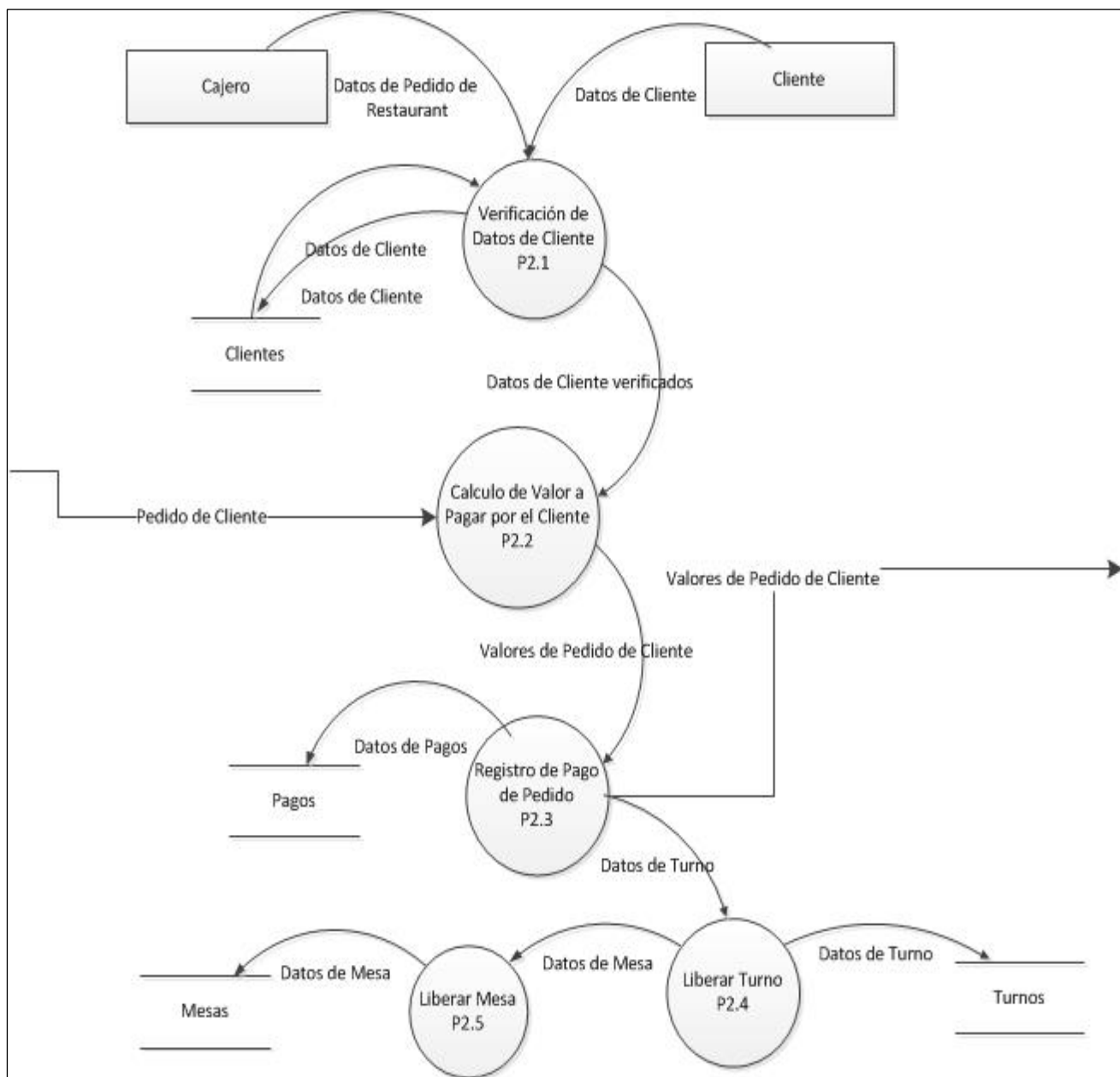


Diagrama de Flujo de Datos del Cálculo de valores del Pedido. Nivel 2.

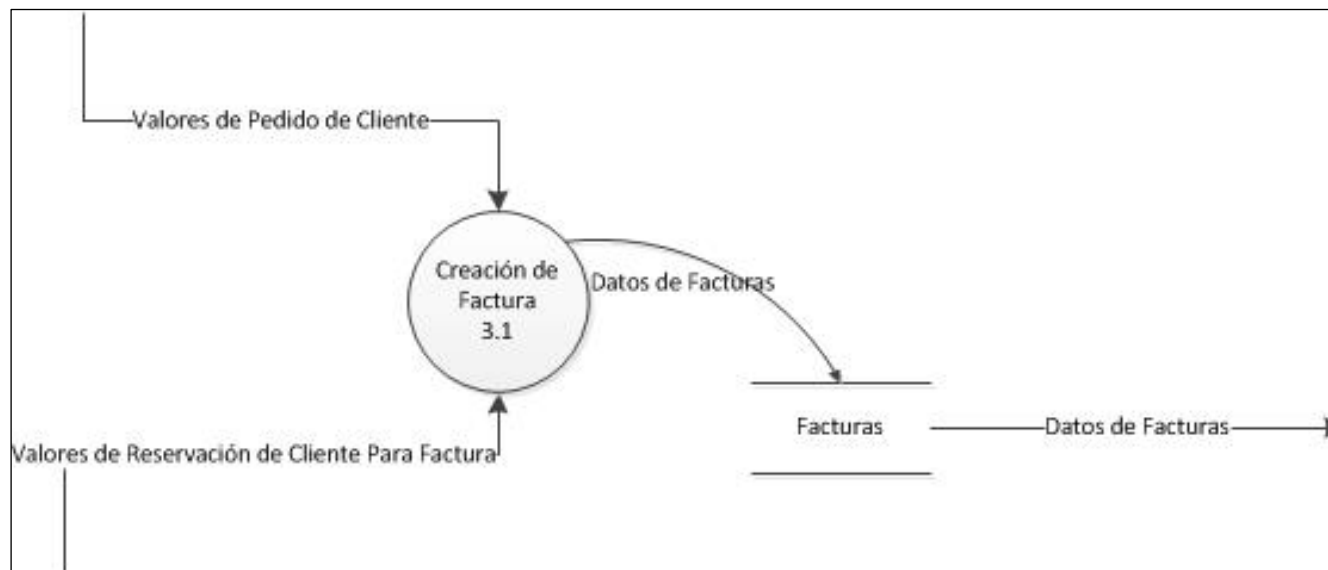


Diagrama de Flujo de Datos de Creación de Factura del Restaurant y del Hotel. Nivel 2.

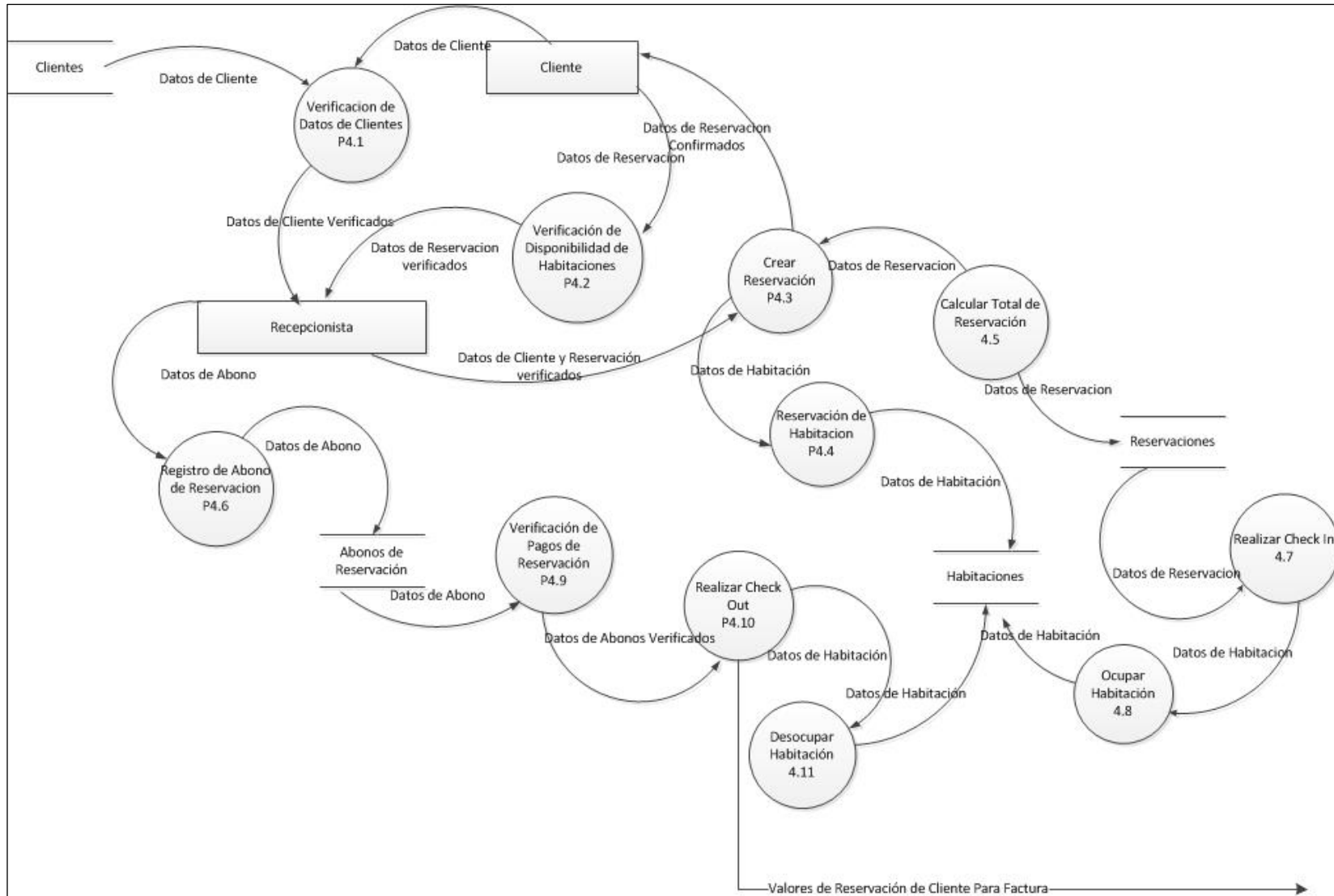


Diagrama de Flujo de Datos de Gestión de Reservas. Nivel 2.

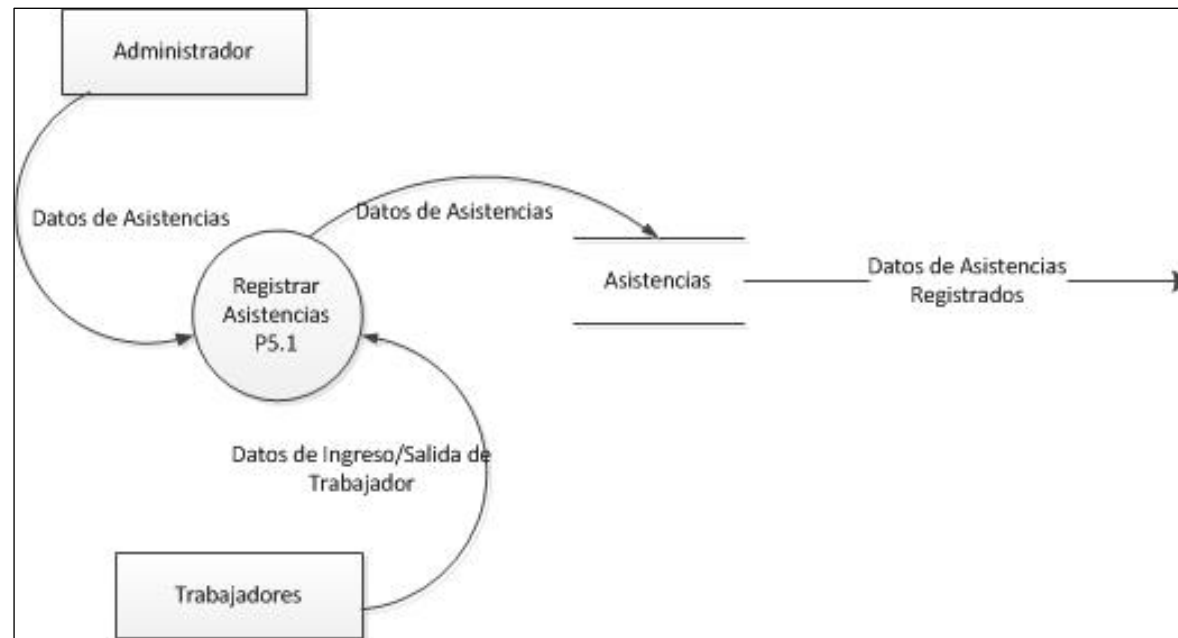


Diagrama de Flujo de Datos de Registro de Asistencias de Trabajadores. Nivel 2.

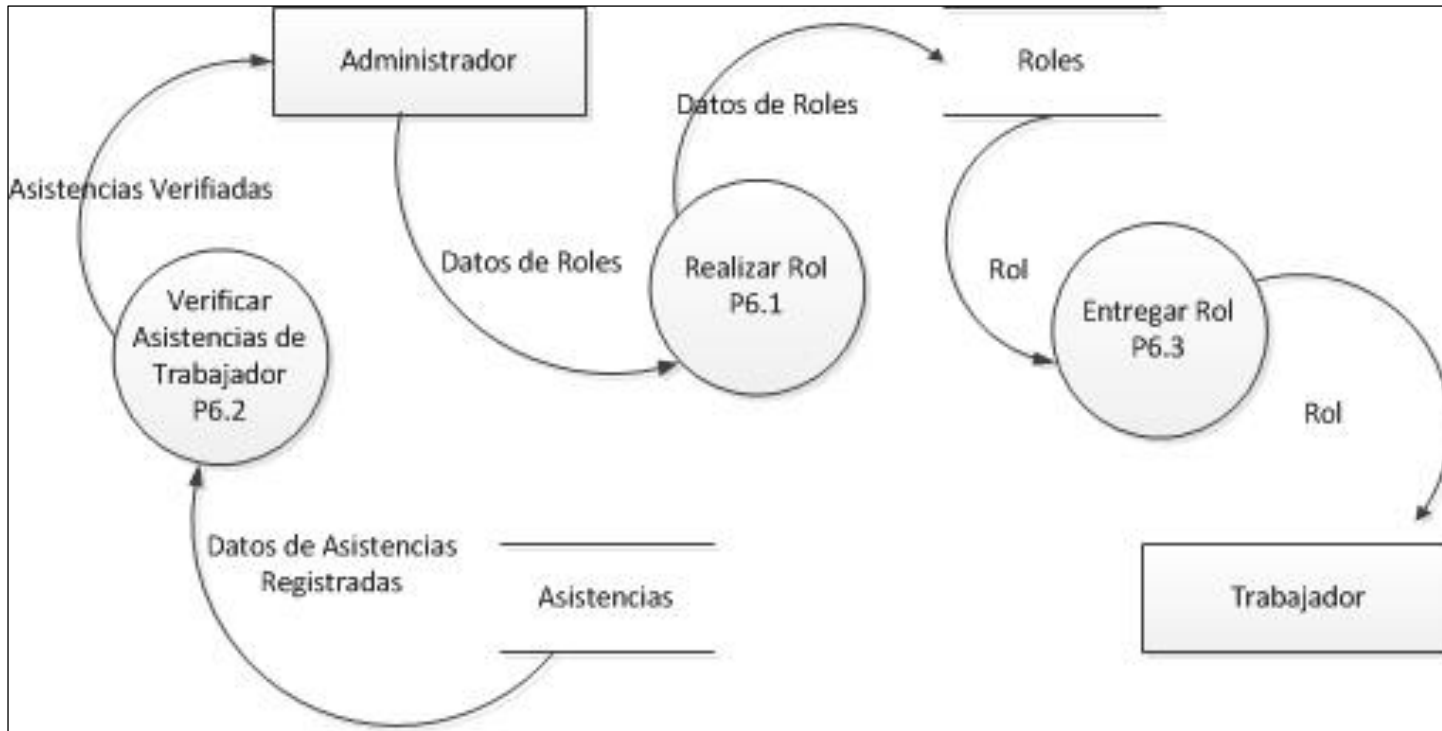


Diagrama de Flujo de Datos Módulo de Realización de Roles de Trabajadores. Nivel 2.

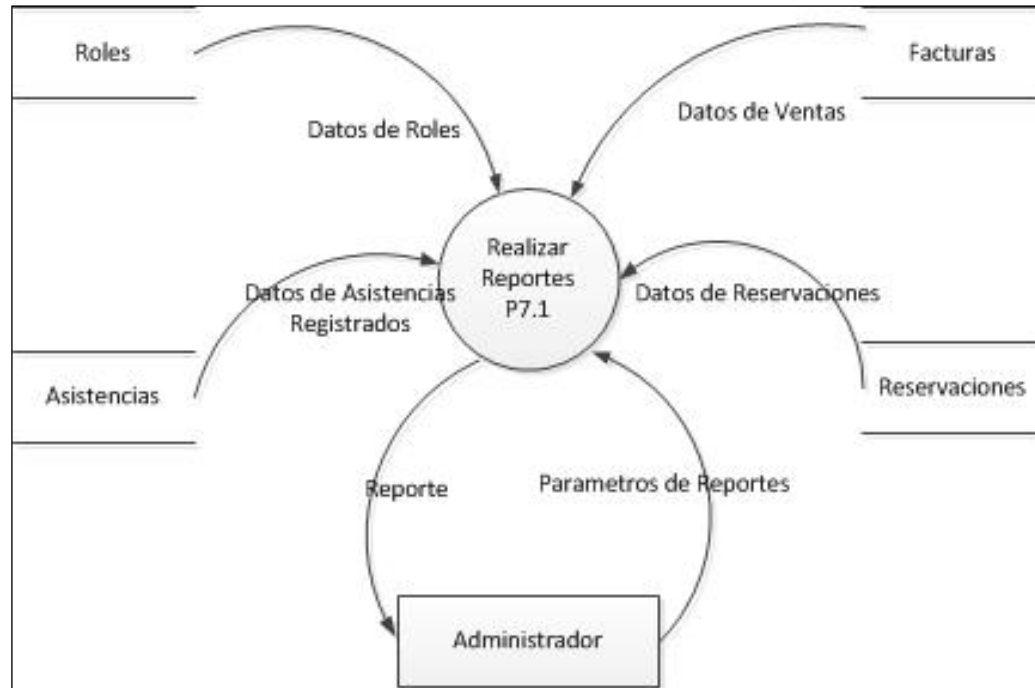


Diagrama de Flujo de Datos de creación de Reportes. Nivel 2

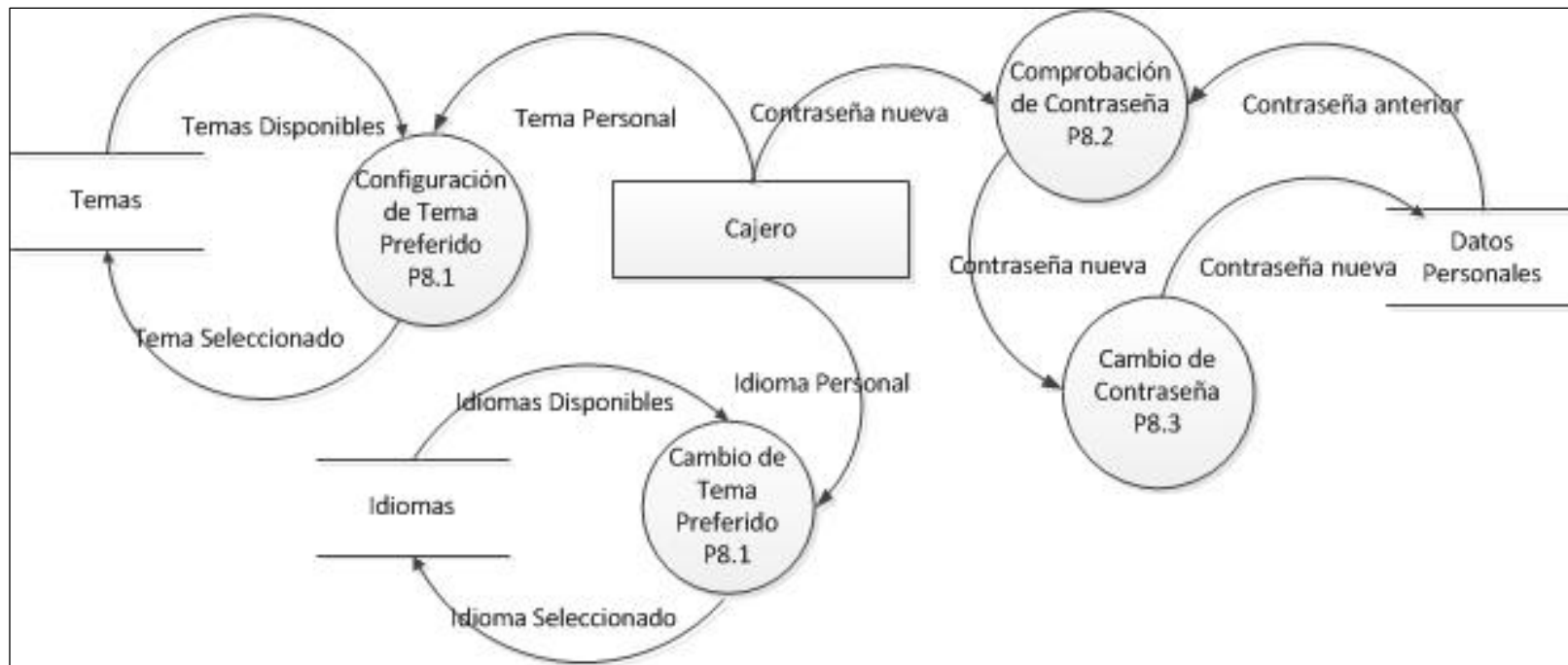


Diagrama de Flujo de Datos de Configuraciones Personales. Nivel 2

3.3.2. MODELO ENTIDAD - RELACIÓN

3.3.3. MODELO RELACIONAL

3.3.4. DICCIONARIO DE DATOS

Esquema *public*:

Tabla: public.bancos

Detalle: Tabla que almacena los datos de los bancos, los mismos que son usados para indicar la pertenencia de las tarjetas de créditos y transacciones que implican referencias a los bancos.

Name	Type	Primary	Not Null	Comment
ban_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Clave Única Identificador de Bancos
ban_nombre	varchar(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Nombre del Banco
ban_detalle	varchar(500)	<input type="checkbox"/>	<input type="checkbox"/>	Detalle de Banco
ban_activo	boolean	<input type="checkbox"/>	<input type="checkbox"/>	Bandera que indica si está o no activo un banco
ban_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de Creación del Banco
ban_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de última modificación del Registro

Tabla: public.clientes

Detalle: Permite almacenar los datos de los clientes del hotel y del restaurant, permite guardar detalle como el email y la nacionalidad lo que permite dar una atención más personalizada al cliente.

Name	Type	Primar	Not Null	Comment
di_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Clave Primaria, Identificador único del Cliente
di_nombre	varchar(40)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Nombres del Cliente
di_apellidos	varchar(60)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Apellidos del Cliente
di_ci	varchar(13)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Cédula/Ruc de Identificación del cliente
di_direccion	varchar(200)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Dirección del Cliente
di_telefono1	varchar(10)	<input type="checkbox"/>	<input type="checkbox"/>	Campo que almacena el número de teléfono del cliente
di_telefono2	varchar(15)	<input type="checkbox"/>	<input type="checkbox"/>	Campo que almacena el número de celular del cliente
di_email	varchar(50)	<input type="checkbox"/>	<input type="checkbox"/>	Dirección de Correo Electrónico del Cliente
di_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de Creación del Cliente
di_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de última modificación del Cliente
di_fr_id_nac	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número que guarda la relación de la nacionalidad del Cliente
di_sexo	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Bandera que indica si el sexo es masculino(true) o femenino(false)
di_activo	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Bandera que indica si el cliente esta activo(true) o bloqueado(false)
di_obser	varchar(500)	<input type="checkbox"/>	<input type="checkbox"/>	Campo que guarda los detalles u observaciones del cliente

Tabla: public.deta_abono_cheque

Detalle: Permite almacenar datos referentes al registro de un abono a una reservación realizado con cheque.

Name	Type	Primary	Not Null	Comment
det_abo_dep	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Identificador único de registro de abono de reservación con cheque
det_abo_titular	varchar(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Nombre del titular del cheque
det_abo_num_cta	bigint	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número de cuenta del cheque
det_abo_valor	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Monto girado en el cheque
fr_id_res_abono	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Código que guarda la relación con la tabla de abonos de reservación
det_abo_num_cheque	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número de cheque girado
det_abo_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de registro de los detalles del pago con cheque
det_abo_obser	varchar(250)	<input type="checkbox"/>	<input type="checkbox"/>	Observación de la transacción
det_abo_fecha	date	<input type="checkbox"/>	<input type="checkbox"/>	Fecha que el cliente determina para el cobro de cheque en caso de ser postfechado
fr_id_banco	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número que guarda la relación con el identificador del banco

Tabla: public.deta_abo_deposito

Detalle: Permite almacenar datos referentes al registro de un abono que el cliente realiza a una reservación ya sea mediante una transferencia o un depósito bancario.

Name	Type	Primary	Not Null	Comment
det_abo_dep	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Identificador único de detalles de abono de reservación con depósito o transferencia
det_abo_valor	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Valor que el cliente deposita o transfiere
fr_id_res_abono	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Código que guarda la relación con la tabla de abonos de reservación
det_abo_obser	varchar(250)	<input type="checkbox"/>	<input type="checkbox"/>	Observación de la transacción
det_abo_num_trans	bigint	<input type="checkbox"/>	<input type="checkbox"/>	Número de depósito o transferencia
det_abo_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha que se registra la transferencia
det_abo_fech_dep	date	<input type="checkbox"/>	<input type="checkbox"/>	Fecha que el cliente realiza la transacción
fr_id_cuenta	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Identificador que guarda relación con la cuenta bancaria del Hotel a la que se realiza la transacción
es_deposito	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Bandera que determina si la transacción fué depósito(true) o transferencia(false)

Tabla: public.deta_abo_efectivo

Detalle: Permite registrar datos de abonos que el cliente realiza a una reservación con dinero en efectivo.

Name	Type	Primary	Not Null	Comment
det_abo_efe	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Identificador único de detalles de abono de reservación en efectivo
fr_id_res_abono	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Código que guarda la relación con la tabla de abonos de reservación
det_abo_valor	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Valor del abono en efectivo
det_abo_obser	varchar(250)	<input type="checkbox"/>	<input type="checkbox"/>	Observación de la transacción
det_abo_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de registro de transacción de abono
det_abo_fecha	date	<input type="checkbox"/>	<input type="checkbox"/>	Campo que indica cuando el cliente realiza el abono en efectivo

Tabla: public.deta_abo_tar

Detalle: Permite registrar datos de abonos que el cliente realiza a una reservación o de consumo en el restaurant con tarjeta de crédito o débito.

Name	Type	Primary	Not Null	Comment
det_abo_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Identificador único de detalles de abono de reservación con tarjetas de debito o crédito
det_abo_num_tar	bigint	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número de la tarjeta del cliente
fr_id_banco	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Identificador que guarda la relación del banco a la que pertenece la tarjeta
fr_id_tar	integer	<input type="checkbox"/>	<input type="checkbox"/>	Número que indica la compañía o empresa de la tarjeta
det_abo_mes_dif	integer	<input type="checkbox"/>	<input type="checkbox"/>	Número de meses a los que se difiere el valor del abono. 0 Si no se difiere.
restaurant	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Indica si el cobro se realizó por conceptos de Hotel(false) o Restaurant(true)
fr_id_res_abono	integer	<input type="checkbox"/>	<input type="checkbox"/>	Código que guarda la relación con la tabla de abonos de reservación
det_abo_titular	varchar(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Nombre del titular de la tarjeta
tarjeta_credito	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Bandera que indica si la tarjeta es de debito(false) o credito(true)
fr_id_fac	integer	<input type="checkbox"/>	<input type="checkbox"/>	Indica el número de la factura con la cual se cobra, en caso de ser de Restaurant.
det_abo_obser	varchar(250)	<input type="checkbox"/>	<input type="checkbox"/>	Observaciones adicionales de la transacción
det_abo_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de registro del pago
det_abo_rec	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Valor recargado por el pago con tarjeta
det_abo_subto	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Monto Subtotal del consumo
det_abo_total	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Valor Total del abono con tarjeta

Tabla: public.factura

Detalle: Tabla que almacena las facturas emitidas en el restaurant por las ventas realizadas y en el hotel por las reservaciones finalizadas con éxito de los clientes.

Name	Type	Primary	Not Null	Comment
fac_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Identificador único de la factura.
fac_fr_ci	varchar(13)	<input type="checkbox"/>	<input type="checkbox"/>	Cedula/Ruc que guarda relación con los datos del cliente.
fac_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de Registro/Emisión de la factura.
fac_subto	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Valor del Subtotal de la Factura
fac_desc	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Valor del Descuento aplicado a la Factura.
fac_iva	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Valor del IVA
fac_total	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Total por el que se emite la factura.
fac_es_rest	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Bandera que indica si la factura se emitió por consumo en el restaurant(true) o en hotel(false)
fac_fr_id_usu	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número que guarda la relación con el usuario que realiza la emisión de la factura.
fac_estado	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Indica el estado de la factura, 1 (activa) o 2 (anulada)
fac_num	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Numero de la Factura Emitida
fac_tipo_pago	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Tipo de Pago: 1.- Efectivo 2.- Tarjeta y Otro Tipo de pago

Tabla: public.fac_det_hot

Detalle: Almacena los detalles de las facturas emitidas en el hotel, con los datos de cada habitación de la reservación.

Name	Type	Primary	Not Null	Comment
id_fac_det_hot	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Identificador único de detalle de factura del Hotel
fac_det_hot_cant	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Numero de días de Reservacion
fr_res_hab_detalle	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número que guarda relación con el detalle de la habitación, dentro de la reservación
fac_det_hot_total	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Monto de importe de la habitación
fr_fac_id	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Numero que guarda la relacion con el identificador de la factura de hotel

Tabla: public.fact_detalle

Detalle: Almacena los detalles de las facturas emitidas en el restaurant, con los valores de cada ítem (carta).

Name	Type	Primary	Not Null	Comment
fac_det_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Identificador único de detalles de factura del Restaurant
fac_det_cantidad	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Numero de Items pedido por el cliente
fac_det_id_carta	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número que guarda la relación para determinar la carta pedida por el cliente
fac_det_pu	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Valor unitario de la carta facturada
fac_det_total	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Total registrado por la venta del ítem seleccionado
fac_fr_id_fac	integer	<input type="checkbox"/>	<input type="checkbox"/>	Número que guarda la relación con el identificador de la factura
fac_det_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha que indica el registro en la base de datos

Tabla: public.fac_detalle

Detalle: Almacena los datos de las plantillas de impresión de facturas, que son las medidas que se utilizarán para imprimir una factura ya sea del hotel o de restaurant.

Name	Type	Primary	Not Null	Comment
plan_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Identificador único de la plantilla de facturas
plan_detalle	varchar	<input type="checkbox"/>	<input type="checkbox"/>	Nombre de la plantilla de facturación
xmargen	integer	<input type="checkbox"/>	<input type="checkbox"/>	Distancia hasta la coordenada X de inicio del margen.
ymargen	integer	<input type="checkbox"/>	<input type="checkbox"/>	Distancia hasta la coordenada Y de inicio del margen.
xfec	integer	<input type="checkbox"/>	<input type="checkbox"/>	Distancia hasta la coordenada X de inicio de impresión de la fecha.
yfec	integer	<input type="checkbox"/>	<input type="checkbox"/>	Distancia hasta la coordenada Y de inicio de impresión de la fecha.
xdet1	integer	<input type="checkbox"/>	<input type="checkbox"/>	Distancia hasta la coordenada X de inicio de impresión del nombre de cliente.
ydet1	integer	<input type="checkbox"/>	<input type="checkbox"/>	Distancia hasta la coordenada Y de inicio de impresión del nombre de cliente.
xdet2	integer	<input type="checkbox"/>	<input type="checkbox"/>	Distancia hasta la coordenada X de inicio de impresión del RUC del cliente.
ydet2	integer	<input type="checkbox"/>	<input type="checkbox"/>	Distancia hasta la coordenada Y de inicio de impresión del RUC del cliente.
xdet3	integer	<input type="checkbox"/>	<input type="checkbox"/>	Distancia hasta la coordenada X de inicio de impresión de la dirección del Cliente.
ydet3	integer	<input type="checkbox"/>	<input type="checkbox"/>	Distancia hasta la coordenada Y de inicio de impresión de la dirección del Cliente.
xdet4	integer	<input type="checkbox"/>	<input type="checkbox"/>	Distancia hasta la coordenada X de inicio de impresión del teléfono del cliente.
ydet4	integer	<input type="checkbox"/>	<input type="checkbox"/>	Distancia hasta la coordenada Y de inicio de impresión del teléfono del cliente.
xdet5	integer	<input type="checkbox"/>	<input type="checkbox"/>	Distancia hasta la coordenada X de inicio de impresión del número de items.
ydet5	integer	<input type="checkbox"/>	<input type="checkbox"/>	Distancia hasta la coordenada Y de inicio de impresión del número de items.
xdet6	integer	<input type="checkbox"/>	<input type="checkbox"/>	Distancia hasta la coordenada X de inicio de impresión del detalle de items.
ydet6	integer	<input type="checkbox"/>	<input type="checkbox"/>	Distancia hasta la coordenada Y de inicio de impresión del detalle de items.
xdet7	integer	<input type="checkbox"/>	<input type="checkbox"/>	Distancia hasta la coordenada X de inicio de impresión del precio unitario.
ydet7	integer	<input type="checkbox"/>	<input type="checkbox"/>	Distancia hasta la coordenada Y de inicio de impresión del precio unitario.
xdet8	integer	<input type="checkbox"/>	<input type="checkbox"/>	Distancia hasta la coordenada X de inicio de impresión del precio de venta.
ydet8	integer	<input type="checkbox"/>	<input type="checkbox"/>	Distancia hasta la coordenada Y de inicio de impresión del precio de venta.
xdet9	integer	<input type="checkbox"/>	<input type="checkbox"/>	Distancia hasta la coordenada X de inicio de impresión del subtotal.
ydet9	integer	<input type="checkbox"/>	<input type="checkbox"/>	Distancia hasta la coordenada Y de inicio de impresión del subtotal.
xdet10	integer	<input type="checkbox"/>	<input type="checkbox"/>	Distancia hasta la coordenada X de inicio de impresión del valor del Descuento.
ydet10	integer	<input type="checkbox"/>	<input type="checkbox"/>	Distancia hasta la coordenada Y de inicio de impresión del valor del Descuento.
xdet11	integer	<input type="checkbox"/>	<input type="checkbox"/>	Distancia hasta la coordenada X de inicio de impresión del valor del IVA.
ydet11	integer	<input type="checkbox"/>	<input type="checkbox"/>	Distancia hasta la coordenada Y de inicio de impresión del valor del IVA.
xdet12	integer	<input type="checkbox"/>	<input type="checkbox"/>	Distancia hasta la coordenada X de inicio de impresión del Total de la Factura.
ydet12	integer	<input type="checkbox"/>	<input type="checkbox"/>	Distancia hasta la coordenada Y de inicio de impresión del Total de la Factura.
maxnumdet	integer	<input type="checkbox"/>	<input type="checkbox"/>	Campo que determina el número de detalles impresos por cada factura
espaciado	integer	<input type="checkbox"/>	<input type="checkbox"/>	Valor que determina el espaciado entre los detalles impresos en la factura
activa	boolean	<input type="checkbox"/>	<input type="checkbox"/>	Determina si la plantilla está o no activa
maxnumcar	integer	<input type="checkbox"/>	<input type="checkbox"/>	Determina el número máximo de caracteres impresos en el detalle de factura
maxnumcarn	integer	<input type="checkbox"/>	<input type="checkbox"/>	Determina el número máximo de caracteres impresos para el nombre del cliente en la factura
maxnumcard	integer	<input type="checkbox"/>	<input type="checkbox"/>	Determina el número máximo de caracteres impresos para la dirección del cliente en la factura
fac_fv	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha que caduca el facturero, fecha de validez
plan_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de creación de la plantilla de facturación
plan_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de última modificación de la plantilla
fac_num	integer	<input type="checkbox"/>	<input type="checkbox"/>	Número de Factura a imprimir

Tabla: public.nacionalidades

Detalle: Almacena las nacionalidades que serán utilizadas para determinar la nacionalidad de los clientes y de los trabajadores.

Name	Type	Primary	Not Null	Comment
nac_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Clave primaria, identifica una Nacionalidad.
nac_nombre	varchar(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Nombre de la nacionalidad, campo mostrado en la interfaz de usuario.
nac_pais	varchar(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	País de referencia de la Nacionalidad
nac_activa	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Indica si se muestra(true) o no(false) la nacionalidad
nac_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de Registro de la Nacionalidad
nac_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de última modificación de la nacionalidad

Tabla: public.res_plantilla

Detalle: Permite almacenar datos de los comprobantes de reservación denominados como plantillas de reservación, permiten definir la forma de imprimir comprobantes de reservación.

Name	Type	Primary	Not Null	Comment
plan_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Identificador único de la plantilla de impresion de comprobante de reservación.
plan_detalle	varchar(200)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Permite colocar un detalle o nombre a la plantilla.
max_num_res	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Determina el número máximo de detalles(Habitaciones Reservadas) para la impresión correcta de comprobante.
plan_nom_arch	varchar(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Nombre de Archivo de Plantilla extensión .jasper
plan_det	boolean	<input type="checkbox"/>	<input type="checkbox"/>	Determina la plantilla como predeterminada para la impresión
plan_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de registro/creación de la plantilla
plan_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de registro/creación de la plantilla

Tabla: public.tar_ban_acep

Detalle: Permite almacenar datos de las tarjetas que son aceptadas al momento de registrar un abono o pago mediante tarjetas de crédito o débito, almacenando el porcentaje de interés que se aplica al cobro en caso de ser así determinado.

Name	Type	Primary	Not Null	Comment
tar_ban_ac_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Clave primaria que identifica la tarjeta bancaria aceptada
fr_id_tar	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Numero que relaciona la empresa de la tarjeta
fr_id_banco	integer	<input type="checkbox"/>	<input type="checkbox"/>	Numero que identifica el banco asociado a la tarjeta
tar_ban_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de registro de tarjeta aceptada
tar_ban_recargo_debito	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Valor porcentual del interes generado con tarjeta de debito
tar_ban_recargo_credito	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Valor porcentual del interes generado con tarjeta de credito
tar_ban_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de modificacion del registro

Tabla: public.tar_cod_inicio

Detalle: Debido a que es obligatorio que el código de una tarjeta inicie con un número dependiendo de la empresa a la que pertenece dicha tarjeta, esta tabla guarda los dígitos con los que debe iniciar el número de cada tarjeta de crédito, permitiendo al sistema saber si existe inconsistencia de este tipo al ingresar los datos de la tarjeta.

Name	Type	Primary	Not Null	Comment
tar_ini	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Identificador único del codigo de verificacion de tarjetas
fr_id_tar	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Numero que guarda la relacion con la tarjeta
cod_inicia	smallint	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Código verificador/inicio de los números de la tarjeta indicada.
ta_cod_ini_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de registro del codigo de inicio
ta_cod_ini_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de última modificación

Tabla: public.tarjetas

Detalle: Guarda los datos de las empresas a las que pertenecen las tarjetas de crédito o de débito.

Name	Type	Primary	Not Null	Comment
tar_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Código clave que identifica las empresas de tarjetas
tar_nombre	varchar(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Nombre de la tarjeta
tar_compania	varchar(200)	<input type="checkbox"/>	<input type="checkbox"/>	Compañía de la tarjeta
tar_activa	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Bandera que determina si está(true) o no(false) activa la tarjeta.
tar_detalle	varchar(500)	<input type="checkbox"/>	<input type="checkbox"/>	Detalle de la tarjeta
tar_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de registro de la tarjeta
tar_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de modificación de la tarjeta

Tabla: public.tipo_abono

Detalle: Tabla que almacena los tipos de abonos que se pueden realizar en el hotel, para determinar cuáles tipos de pagos están o no disponibles.

Name	Type	Primary	Not Null	Comment
tip_abo_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Clave primaria que identifica el tipo de abono/pago de reservaciones o consumo de restaurant
tip_abo_nombre	varchar(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Nombre que identifica el tipo de abono/pago
tip_abo_detalle	varchar(200)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Detalle del tipo de abono/pago
tip_abo_activo	boolean	<input type="checkbox"/>	<input type="checkbox"/>	Determina si el tipo de pago está o no activo al momento de realizar un cobro al cliente.
tip_abo_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de registro del tipo de pago
tip_abo_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de modificación de tipo de pago

Tabla: public.tipo_tarjeta

Detalle: Determina los tipos de tarjetas existentes.

Name	Type	Primary	Not Null	Comment
tip_tar_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Identificador del tipo de tarjetas
tip_tar_nombre	varchar(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Nombre del tipo de tarjeta existente

Tabla: public.tra_roles

Detalle: Almacena los detalles de los roles que se generan a los trabajadores en un año y mes determinado, con los datos de ingresos y egresos de los mismos.

Name	Type	Primary	Not Null	Comment
rol_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Clave primaria que identifica el registro del rol generado del trabajador.
fr_tra_id	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número que guarda la relación con el código del trabajador.
rol_anio	integer	<input type="checkbox"/>	<input type="checkbox"/>	Año por el cual el rol es generado
rol_mes	integer	<input type="checkbox"/>	<input type="checkbox"/>	Mes de referencia por el cual el rol se generó
rol_tot_ing	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Monto de Total de Ingresos que generó el trabajador
rol_tot_egr	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Valor de Total de Egresos que generó el trabajador
rol_tot_rol	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Monto de Total del Rol que se pagó al trabajador
rol_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de registro de rol en la Base de Datos
rol_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de modificación de rol en la Base de Datos
rol_val_sueldo	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Sueldo del trabajador cuando se generó el rol
rol_num_horas_ext	integer	<input type="checkbox"/>	<input type="checkbox"/>	Número de horas extras que realizó el trabajador
rol_val_hor_extras	double precision	<input type="checkbox"/>	<input type="checkbox"/>	Valor comisionado por el numero de horas extras realizadas
rol_val_bono	double precision	<input type="checkbox"/>	<input type="checkbox"/>	Valor total de bonos acreditados al trabajador
rol_val_antipico	double precision	<input type="checkbox"/>	<input type="checkbox"/>	Valor total debitado por anticipos realizados por el trabajador
rol_val_pres_emp	double precision	<input type="checkbox"/>	<input type="checkbox"/>	Valor debitado por prestamos realizados a la empresa
rol_val_pres_jess	double precision	<input type="checkbox"/>	<input type="checkbox"/>	Valor debitado por prestamos del IESS
rol_val_jess	double precision	<input type="checkbox"/>	<input type="checkbox"/>	Valor debitado por valor de seguro social
rol_val_multa	double precision	<input type="checkbox"/>	<input type="checkbox"/>	Valor debitado por multas realizadas al trabajador

Esquema *hotel*:

Tabla: hotel.cliente_reservacion

Detalle: Es la tabla que permite registrar a todos los clientes que ingresarán al hotel mediante una reservación realizada, o en el registro de Check In por el recepcionista.

Name	Type	Primary	Not Null	Comment
di_res_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Identificador único del registro de clientes que ingresan en una reservación
fr_di_id	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número que relaciona el cliente por el identificador el cliente
fr_res_id	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número que relaciona el identificador de la reservación
di_res_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha y hora que se registra al cliente en la reservación

Tabla: hotel.hab_detalle

Detalle: Permite relacionar los componentes de una habitación con la habitación que lo contiene para determinar con qué servicios cuenta la habitación y tener un control de los objetos con los que cuenta cada habitación.

Name	Type	Primar	Not Null	Comment
hab_det_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Clave identificadora de los componentes que conforman la habitación
fr_id_hab	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Numero que relaciona el código de la habitacion
fr_id_hab_det_tip	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Numero que relaciona el componente de la habitación
hab_det_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de Registro del componente en la habitacion

Tabla: hotel.hab_detalle_tipo

Detalle: Contiene un inventario de todos los objetos que pertenecen a la empresa y que están designados a cada habitación.

Name	Type	Primar	Not Null	Comment
hab_det_tip_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Clave de Identificación única del activo o componente de habitación
hab_det_tip_nombre	varchar(400)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Nombre del activo o componente
hab_det_tip_act	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Indica si el componente está(true) o no(false) activo.
hab_det_tip_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de ingreso del componente
hab_det_tip_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de modificación del componente
hab_det_tip_cod	varchar(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Código de Identificación única del activo o componente de habitación

Tabla: hotel.hab_estado

Detalle: Objeto de la base de datos que permite almacenar los estados actuales posibles de las habitaciones.

Name	Type	Primar	Not Null	Comment
hab_est_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Código identificador de estado de la habitación
hab_est_nombre	varchar(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Nombre que identifica el estado de la habitación
hab_est_detalle	varchar(200)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Detalle del estado de la habitación
hab_est_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de registro del tipo de estado de la habitación
hab_est_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de modificación del estado de la habitación
hab_est_activo	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Determina si un estado de habitación está o no activo

Tabla: hotel.hab_tipo

Detalle: Permite guardar los tipos de las habitaciones con las que cuenta el hotel para clasificarlas en grupos según corresponda para una mejor organización de las mismas.

Name	Type	Primar	Not Null	Comment
hab_tip_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Identifica el tipo de habitación con las que cuenta el Hotel
hab_tip_nombre	varchar(50)	<input type="checkbox"/>	<input type="checkbox"/>	Nombre del tipo de habitación
hab_tip_detalle	varchar(250)	<input type="checkbox"/>	<input type="checkbox"/>	Detalle del tipo de habitación
hab_tip_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de registro del tipo de habitación
hab_tip_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de modificación del tipo de habitación
hab_tip_activo	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Determina si el tipo de habitación está activo(true) o no (false)

Tabla: hotel.habitaciones

Detalle: Almacena el catálogo de todas las habitaciones existentes en el hotel con las características necesarias que las identifican, las mismas que serán reservadas por los clientes.

Name	Type	Primary	Not Null	Comment
hab_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Número que identifica internamente y de manera única una habitación del Hotel
hab_cod	varchar(5)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Código o número con la que se hace identifica una habitación
hab_descripcion	varchar(500)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Detalles u observaciones de la habitación o de su estado actual.
hab_fr_id_estado	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número que guarda relación con el estado actual de la habitación.
hab_num_bloq	integer	<input type="checkbox"/>	<input type="checkbox"/>	Determina el número de bloque donde se encuentra situada la habitación físicamente.
hab_num_piso	integer	<input type="checkbox"/>	<input type="checkbox"/>	Determina el número de piso del bloque donde se encuentra situada la habitación físicamente.
hab_fumar	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Determina si es permitido fumar dentro de la habitación
hab_capacidad	integer	<input type="checkbox"/>	<input type="checkbox"/>	Determina la capacidad maxima de la habitación sean niños o adultos
hab_uu	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Determina la fecha de ultimo uso de la habitación
hab_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Determina la fecha de modificación de la habitación
hab_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Determina la fecha de registro de la habitación
hab_fr_id_tipo	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número que guarda la relación con el tipo de la habitación
hab_ext	varchar(10)	<input type="checkbox"/>	<input type="checkbox"/>	Indica el número de extensión de la habitación en caso de haberla.
hab_pn	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Valor que incica la tarifa diaria de un niño en la habitación
hab_pa	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Valor que incica la tarifa diaria de un adulto en la habitación
hab_ph	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Valor que incica la tarifa diaria de la habitación sin importar numero o tipo de ocupantes

Tabla: hotel.res_estado

Detalle: Tabla que contiene los posibles estados de una reservación desde que se crea hasta que la misma finalice dentro del sistema.

Name	Type	Primary	Not Null	Comment
res_est_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Código que identifica de manera única el estado de la reservación
res_est_nombre	varchar(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Nombre de estado de reservación
res_est_detalle	varchar(1000)	<input type="checkbox"/>	<input type="checkbox"/>	Detalle del estado de reservación
res_est_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de registro del estado de la reservación
res_est_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de última modificación del estado de la reservación

Tabla: hotel.res_estado

Detalle: Debido a que en una reservación se pueden incluir varias habitaciones con detalles diferentes, esta tabla permite almacenar los detalles de reservación de cada habitación, incluso con fechas de reservación distintas entre sí.

Name	Type	Primary	Not Null	Comment
res_hab_det_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Número que identifica de forma única los datos de reservación de una habitación que conforma la reservación del cliente
fr_id_res	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número que guarda la relación hacia la reservación mediante el código de reservación
fr_id_hab	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número que relaciona el detalle de reservación con la habitación implicada
res_hab_num_nin	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Determina en número de niños que ingresarán a la habitación
res_hab_num_adu	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Determina en número de adultos que ingresarán a la habitación
res_hab_fecha1	timestamp(2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Determina la fecha y hora que los huéspedes ingresan a la habitación
res_hab_fecha2	timestamp(2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Determina la fecha y hora que los huéspedes deben abandonar la habitación
res_hab_total	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Valor de importe de la reservación de la habitación
res_hab_fi	timestamp(2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Fecha de registro del detalle de la reservación en la base de datos
res_hab_um	timestamp(2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Fecha de modificación del detalle de la reservación en la base de datos
res_hab_tip_res	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Tipo de tarifa aplicada en la reservacion de la habitacion. Por persona o por valor de la habitacion.
res_hab_check_in	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Determina si se realizó o no el Check In de la reservación.
res_hab_fecha_in	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha y Hora que los clientes ingresan (Check In) a la habitación
res_hab_check_out	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Determina si se realizó o no el Check Out de la reservación.
res_hab_fecha_out	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha y Hora que los clientes salen (Check Out) de la habitación
res_hab_activa	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Determina si el detalle está o no activo. No está activa cuando se anula o finaliza.

Tabla: hotel.reser_abonos

Detalle: Esta tabla sirve para almacenar los valores que el cliente abona por una reservación y el saldo que se genera, sin importar el tipo del pago, permite determinar rápidamente cuales han sido los abonos de la reservación, para luego ser referenciada desde las tablas que almacenan más detalles de los tipos de abonos de reservaciones desde el esquema *public*.

Name	Type	Primary	Not Null	Comment
res_abo_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Clave Única de Registro de un Abono de Reservación
fr_res_id	integer	<input type="checkbox"/>	<input type="checkbox"/>	Número que guarda la relación entre el abono y la reservación
fr_id_usu	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Identificador del usuario que registra el abono
res_abo_valor	double precision	<input type="checkbox"/>	<input type="checkbox"/>	Monto del abono
res_abo_saldo	double precision	<input type="checkbox"/>	<input type="checkbox"/>	Saldo pendiente de pago de la reseración
res_abo_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha que se registra el pago en la base de datos
res_abo_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha que se modifica el pago en la base de datos
fr_id_tipo_pago	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número que guarda la relación con el tipo de pago del abono

Tabla: hotel.reservacion

Detalle: Tabla que guarda los detalles generales de la reservación, con los totales de las personas que ingresarán, valor total de la reservación, datos del cliente, datos de usuario, fecha de reservaciones y su estado.

Name	Type	Primary	Not Null	Comment
res_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Código que identifica de manera única la reservación. Número de reservación.
res_ci_ci	varchar(13)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Cédula/RUC del cliente que realiza la reservación.
res_id_usu	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Código que relaciona al usuario que realiza la reservación
res_num_nin	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Total de niños que ingresarán a la(s) habitacion(es)
res_num_adu	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Total de adultos que ingresarán a la(s) habitacion(es)
res_fecha1	timestamp(2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Fecha mínima de ingreso a las habitaciones.
res_fecha2	timestamp(2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Fecha máxima de salida de las habitaciones.
res_dia_inst	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Numero de días de la reservación
res_subto	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Subtotal de la reservación
res_descpto	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Valor del descuento aplicado a la reservación
res_iva	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Valor del IVA de reservación.
res_total	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Total de la reservación.
res_abono	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Valor total abonado por el cliente
res_saldo	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Valor residual adeudado por el cliente
res_id_est	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número que relaciona la reservación con el estado de la misma
res_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de registro de la reservación en la base de datos
res_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de última modificación de la reservación
res_obs	varchar(250)	<input type="checkbox"/>	<input type="checkbox"/>	Observaciones de la Reservación
res_fac	boolean	<input type="checkbox"/>	<input type="checkbox"/>	Bandera que indica si la reservación se ha o no facturado.

Esquema *restaurant*:

Tabla: restaurant.carta

Detalle: Guarda los datos de los productos que se venden en el restaurant con los datos de configuraciones que permiten visualizar y organizar las cartas en la interfaz de usuario para optimizar el uso de la misma.

Name	Type	Primary	Not Null	Comment
car_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Número que identifica de manera única un producto vendido en el restaurant.
car_fr_id_grupo	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número que identifica la pertenencia de una carta dentro de un grupo
car_rank	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Permite asignar un orden de visualización por reelevancia o preferencia de la carta dentro de su grupo
car_nombre	varchar(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Guarda el nombre completo de la carta o producto
car_label	varchar(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Muestra un nombre corto de la carta o producto
car_detalle	varchar(40)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Guarda el detalle de la carta.
car_pvp	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Almacena el valor de la carta incluido el IVA
car_usa_imagen	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Determina si se muestra o no la imagen de la carta
car_usa_label	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Detrmina si se muestra o no el nombre corto de la carta
car_activo	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Permite determinar si la carta está activa o no
car_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Guarda la fecha de registro de la carta en la base de datos
car_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Guarda la fecha de última modificación de la carta
car_favorito	boolean	<input type="checkbox"/>	<input type="checkbox"/>	Determina si la carta es considerada como favorita o no

Tabla: restaurant.grupos

Detalle: Almacena registros que permiten clasificar la carta o productos de consumo del restaurant, lo que permite organizarlas en grupos para que el usuario pueda crear un pedido rápidamente.

Name	Type	Primary	Not Null	Comment
gru_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Número que identifica de manera única un grupo de carta del restaurant.
gru_nombre	varchar(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Guarda el nombre completo del grupo de carta
gru_label	varchar(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Muestra un nombre corto del grupo
gru_detalle	varchar(200)	<input type="checkbox"/>	<input type="checkbox"/>	Guarda el detalle del grupo
gru_usa_imagen	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Determina si se muestra o no la imagen del grupo para mejor visualización
gru_usa_label	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Detrmina si se muestra o no el nombre corto del grupo
gru_activo	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Permite determinar si el grupo de carta está activo o no
gru_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Guarda la fecha de registro del grupo en la base de datos
gru_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Guarda la fecha de modificación del grupo en la base de datos
gru_rank	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Permite asignar un orden de visualización por reelevancia o preferencia del grupo
gru_es_bebida	boolean	<input type="checkbox"/>	<input type="checkbox"/>	Determina si el grupo de carta se considera como de un grupo de bebidas

Tabla: restaurant.mesas

Detalle: Almacena un registro por cada mesa del restaurant con las características de la misma. Permiten posteriormente la asignación de un turno al sitio donde se sirve el cliente que sirve como referencia para identificar el pedido al momento de la facturación.

Name	Type	Primary	Not Null	Comment
mes_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Número que identifica de manera única una mesa en el sistema
mes_cod	varchar(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Determina el código de la mesa, como se hace referencia hacia ella en el restaurant
mes_detalle	varchar(200)	<input type="checkbox"/>	<input type="checkbox"/>	Detalle de la mesa
mes_cap	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número de personas que caben en la mesa
mes_ocupada	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Determina el estado actual de la mesa, si está o no ocupada
mes_fi	timestamp(2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Determina la fecha que se registra en el sistema
mes_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Determina la fecha de modificación de la mesa
mes_activa	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Permite saber si la mesa está o no activa

Tabla: restaurant.turnos

Detalle: Guarda todos los turnos que los cajeros crean para asignar un orden de llegada y atención a los clientes que realizan un pedido en el restaurant, almacenando también los datos de referencias de los mismos para ser identificados al momento de la facturación.

Name	Type	Primary	Not Null	Comment
tur_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Número que identifica de manera única un turno creado
tur_numero	serial	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Determina el número de turno asignado a un pedido u orden de atención
tur_mesero	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Guarda la relación para determinar el mesero que realiza el pedido
tur_mesa	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Guarda la relación para determinar la mesa a la que se asigna el pedido
tur_id_estado	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Guarda la relación que determina el estado del turno
tur_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Guarda la fecha que se crea/asigna el turno
tur_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Guarda la fecha que se modifica el turno
tur_usuario	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Permite determinar cual es el cajero que realiza la apertura del turno.
tur_observacion	varchar(1000)	<input type="checkbox"/>	<input type="checkbox"/>	Asigna una observación al turno.

Tabla: restaurant.turnos_estados

Detalle: Almacena los posibles estados de un turno desde el momento de su creación hasta el momento que es atendido.

Name	Type	Primary	Not Null	Comment
est_tur_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Determina de manera única un estado del turno
est_tur_nombre	varchar(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Guarda el nombre del estado del turno
est_tur_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Almacena la fecha de creación del estado del turno
est_tur_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Determina la fecha de modificación del estado del turno
est_tur_detalle	varchar(200)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Permite agregar un detalle del estado del turno

Esquema sistema:

Tabla: sistema.actividades

Detalle: Guarda los cargos que pueden ser asignados a los trabajadores en el hotel. Esta tabla es relacionada para asignar una actividad específica a un trabajador.

Name	Type	Primary	Not Null	Comment
act_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Permite determinar de manera única un cargo o actividad que se realiza dentro de la empresa
act_nombre	varchar(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Determina el nombre del cargo en la empresa
act_detalle	varchar(500)	<input type="checkbox"/>	<input type="checkbox"/>	Permite almacenar un detalle de la actividad que puede realizar un trabajador
act_activa	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Determina si el cargo está o no activo
act_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Guarda la fecha de registro del cargo en la base de datos
act_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Guarda la fecha de modificación del cargo
act_fr_id_lugar	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Identificador del lugar o área de trabajo dentro de la empresa

Tabla: sistema.cuentas_bancarias

Detalle: Permite almacenar las cuentas bancarias del hotel, en la cual se realizan los depósitos y las transferencias que los usuarios realizan por adquirir servicios de reservación.

Name	Type	Primary	Not Null	Comment
cue_ban_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Determina de forma única una cuenta bancaria de la empresa
fr_id_banco	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Guarda la relación con el banco a la que pertenece la cuenta
fr_id_tipo	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Tipo de cuenta 1 Si es de ahorro 2 si es corriente
cue_ban_activa	boolean	<input type="checkbox"/>	<input type="checkbox"/>	Determina si la cuenta está o no activa
cue_ban_obser	varchar(500)	<input type="checkbox"/>	<input type="checkbox"/>	Permite agregar un detalle a la cuenta bancaria
cue_ban_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Almacena la fecha que se registra la cuenta en la base de datos
cue_ban_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Determina la fecha de modificación de la cuenta bancaria
cue_ban_label	varchar(100)	<input type="checkbox"/>	<input type="checkbox"/>	Guarda un detalle con el numero, tipo, y nombre de banco de la cuenta bancaria
cue_ban_numero	varchar(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número de la cuenta bancaria

Tabla: sistema.estado_civil

Detalle: Alacena los posibles estados civiles que se asignarán a un trabajador.

Name	Type	Primary	Not Null	Comment
est_civ_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Determina de manera única un estado civil
est_civ_nombre	varchar(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Nombre del estado civil
est_civ_activo	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Determina si el estado está o no disponible
est_civil_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de ingreso del tipo de estado civil en la base
est_civil_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de ingreso del tipo de estado civil en la base

Tabla: sistema.historia_trabajo

Detalle: Guarda el historial de trabajo con las fechas de ingreso y salida de cada persona que labora en la empresa.

Name	Type	Primary	Not Null	Comment
hist_trab	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Identifica de manera única un registro de la historia de los cargos que ha realizado de un trabajador en la empresa
tra_id	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Guarda la relación con el trabajador
tra_sueldo	double precision	<input type="checkbox"/>	<input type="checkbox"/>	Indica el sueldo asignado por el cargo que ocupó.
actividad	varchar(150)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Guarda el nombre del cargo que ocupó el trabajador
tra_act_fi	date	<input type="checkbox"/>	<input type="checkbox"/>	Guarda la fecha que el trabajador inicia las actividades en la empresa
tra_act_fs	date	<input type="checkbox"/>	<input type="checkbox"/>	Guarda la fecha que el trabajador finaliza las actividades en la empresa
tra_es_afi	boolean	<input type="checkbox"/>	<input type="checkbox"/>	Determina si se afilió o no al trabajador al IESS

Tabla: sistema.idiomas

Detalle: Es una tabla del sistema que registra los idiomas en los que puede estar traducido el sistema, la misma que permite también definir un idioma determinado a cada usuario.

Name	Type	Primary	Not Null	Comment
id_idioma	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Identifica de manera única un idioma soportado por el sistema
detalle	varchar(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Nombre del idioma, mostrado al usuario
idioma	char(2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Iniciales que determinan el idioma
pais	char(2)	<input type="checkbox"/>	<input type="checkbox"/>	Iniciales que determinan el país relacionado con el idioma
activo	boolean	<input type="checkbox"/>	<input type="checkbox"/>	Determina si el Idioma está o no activo en el sistema
idio_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Guarda la fecha de registro del idioma en la base de datos.
idio_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Guarda la fecha de modificación del idioma en la base de datos.

Tabla: sistema.interfaz

Detalle: Guarda los detalles básicos de todas las interfaces del módulo de restaurant y de hotel, relacionándolas a un módulo, la misma que permitirá posteriormente el control de los accesos de usuarios a las opciones sistema.

Name	Type	Primary	Not Null	Comment
int_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Determina de manera única una interfaz del sistema
int_detalle	varchar(30)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Nombre de la interfaz
int_activa	boolean	<input type="checkbox"/>	<input type="checkbox"/>	Determina si la interfaz está o no activa
int_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de registro de la interfaz en la base de datos
int_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de modificación del registro en la base de datos
fr_modulo	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Identificador del módulo al que pertenece la interfaz

Tabla: sistema.modulos

Detalle: Almacena los módulos del sistema e identifican la pertenencia de una interfaz.

Name	Type	Primary	Not Null	Comment
mod_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Identifica de manera única un módulo del sistema
mod_detalle	varchar(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Nombre del módulo del sistema
mod_activo	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Determina si el módulo está o no activo
mod_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Determina la fecha de registro del módulo en la base de datos.
mod_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de modificación de módulo.

Tabla: sistema.notas

Detalle: Es la tabla determinada específicamente para guardar los recordatorios, notas, citas, etc... que el usuario registra en la agenda del sistema de eventos importantes.

Name	Type	Primary	Not Null	Comment
not_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Identifica de manera única una nota registrada en la Agenda del sistema
not_fecha	date	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Determina la fecha a la que se relaciona la nota creada.
not_detalle	varchar(5000)	<input type="checkbox"/>	<input type="checkbox"/>	Contenido de la(s) nota(s) realizadas en la fecha indicada
not_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha que se registra la nota en la base de datos.

Tabla: sistema.opciones

Detalle: Almacena las opciones de las interfaces principales de los módulos de hotel y restaurant, a las cuales se restringe o concede acceso a los usuarios según las acciones que se le permitan realizar en el sistema.

Name	Type	Primary	Not Null	Comment
opc_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Identifica de manera única las opciones a las que se tiene acceso en las interfaces.
opc_detalle	varchar(200)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Detalle de la opción
opc_activa	boolean	<input type="checkbox"/>	<input type="checkbox"/>	Determina si la opción está o no activa
opc_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Guarda la fecha de modificación de la opcion del sistema
opc_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Guarda la fecha de ingreso de la opcion al sistema
fr_interfaz	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Establece la pertenencia de la opción a una interfaz por el identificador de la interfaz

Tabla: sistema.sesiones

Detalle: Guarda las sesiones creadas por los administradores junto a un resumen de número de turnos y reservaciones realizados en dicho intervalo de tiempo. Permite saber también si hay o no acceso al sistema cuando exista una sesión abierta.

Name	Type	Primary	Not Null	Comment
ses_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Identifica de manera única una sesión creada por el administrador
ses_f1	timestamp(2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Fecha de creación de la sesión
ses_f2	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de cierre o finalización de la sesión
ses_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de modificación de la sesión.
ses_deta	varchar(200)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Detalle de la sesión
ses_tur_abi	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número de turnos que se abrieron en la sesión.
ses_tur_can	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número de turnos que se anularon en la sesión.
ses_tur_exi	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número de turnos que se atendieron exitosamente en la sesión.
ses_res_abi	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número de reservaciones que se crearon en la sesión.
ses_res_can	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número de reservaciones que se anularon en la sesión.
ses_res_exi	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número de reservaciones que se atendieron con éxito en la sesión.
ses_abierta	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Determina si la sesión está abierta(true) o cerrada(false)

Tabla: sistema.skins

Detalle: Guarda los temas o skins del sistema para permitir cambiar la apariencia al sistema según sea el gusto del usuario.

Name	Type	Primary	Not Null	Comment
id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Identificador único de un tema/skin disponible en el sistema
descripcion	varchar(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Nombre del tema que se muestra al usuario
varchar	varchar(100)	<input type="checkbox"/>	<input type="checkbox"/>	Valor interno de configuración del tema
fecha_ingreso	timestamp(2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Fecha de registro del tema en la base de datos
fecha_modif	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de modificación del tema
grupo	integer	<input type="checkbox"/>	<input type="checkbox"/>	Valor que permite ordenar los temas
activo	boolean	<input type="checkbox"/>	<input type="checkbox"/>	Determina si el tema puede o no ser usado por un usuario

Tabla: sistema.tra_anticipos

Detalle: Registra los anticipos que un trabajador realiza a la empresa con un detalle y el estado del mismo, permitiendo tener un historial de anticipos del mismo.

Name	Type	Primary	Not Null	Comment
ant_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Identifica de manera única el registro de un anticipo de un trabajador.
fr_tra_id	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número que guarda la relación con el trabajador.
ant_fecha1	timestamp(2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Fecha que se realiza el anticipo.
ant_detalle	varchar(500)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Detalle del anticipo
ant_valor	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Valor del anticipo
ant_activo	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Determina si el anticipo está pendiente de debitar(true) o si está inactivo(false)
ant_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha que se registra el anticipo en la base de datos.
ant_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de modificación del registro de anticipo.

Tabla: sistema.tra_asistencia

Detalle: Guarda todos los registros de las asistencias que realiza un trabajador en una sesión, permitiendo tener un historial de asistencias de trabajador con el tiempo que estuvo en la empresa y con una observación en caso de ser necesario.

Name	Type	Primary	Not Null	Comment
tra_asi_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Identifica de manera única una asistencia de un trabajador
fr_tra_id	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Identificador que relaciona la asistencia con el trabajador
fr_ses_id	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número que guarda la relación con la sesión en que se registra la asistencia del trabajador
tra_asi_fa	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de ingreso/legada del trabajador a la empresa
tra_asi_fs	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de salida del trabajador de la empresa
tra_asi_obser	varchar(200)	<input type="checkbox"/>	<input type="checkbox"/>	Observación de registro de asistencia
tra_asi_cul	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Bandera que determina si se registró el ingreso y la salida del trabajador.
tra_asi_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha y hora que se registra la asistencia en la base de datos.
tra_asi_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha y hora que se modifica la asistencia en la base de datos.
tra_asi_tiempo	time	<input type="checkbox"/>	<input type="checkbox"/>	Tiempo que permanece el trabajador en la empresa.

Tabla: sistema.tra_cursos

Detalle: Permite almacenar la información referente a los cursos a los que asistió un trabajador, con la información de dicha capacitación.

Name	Type	Primary	Not Null	Comment
tra_cur_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Identificador único de un curso realizado por un usuario.
tra_cur_id_trab	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número que relaciona el curso al trabajador que lo realiza.
tra_cur_insti	varchar(200)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Institución en la cual se realiza el curso.
tra_cur_detalle	varchar(300)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Detalle relacionado con el curso realizado.
tra_cur_fecha1	date	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Fecha que el trabajador inicia el curso.
tra_cur_fecha2	date	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Fecha que el trabajador culmina el curso.
tra_cur_num_horas	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número de horas de duración del curso.
tra_cur_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha que se registra el curso realizado en la base de datos.
tra_cur_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha que se modifica el curso realizado en la base de datos.

Tabla: sistema.tra_exp_profesional

Detalle: Es una tabla que registra la trayectoria profesional de un trabajador con los datos de dichas experiencias realizadas en su vida laboral.

Name	Type	Primary	Not Null	Comment
tra_exp_pro_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Código que identifica de manera única un registro de la experiencia de un trabajador.
tra_exp_id_trab	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número que relaciona un registro con el trabajador correspondiente
tra_exp_pro_inst	varchar(200)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Institución en la que desempeñó el trabajador
tra_exp_pro_cargo	varchar(200)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Cargo ocupado por el trabajador.
tra_exp_pro_fecha1	date	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Fecha de inicio de trabajo del empleado.
tra_exp_pro_fecha2	date	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Fecha de salida de la empresa del empleado.
tra_exp_pro_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de registro de la experiencia del trabajador.
tra_exp_pro_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de modificación de la experiencia del trabajador.

Tabla: sistema.tra_inf_academica

Detalle: Guarda la información académica de un trabajador para conocer el nivel de estudios que posee dicho empleado.

Name	Type	Primary	Not Null	Comment
tra_inf_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Código que identifica de manera única un registro de estudio de un trabajador.
tra_inf_id_trab	integer	<input type="checkbox"/>	<input type="checkbox"/>	Número que relaciona un registro con el trabajador correspondiente
tra_inf_institucion	varchar(200)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Institución donde realiza los estudios.
tra_inf_instuccion	varchar(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Tipo de instrucción que realiza el trabajador.
tra_inf_fecha1	date	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de inicio de instrucción del trabajador.
tra_inf_fecha2	date	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de finalización de instrucción del trabajador.
tra_inf_observacion	varchar(200)	<input type="checkbox"/>	<input type="checkbox"/>	Detalle de la instrucción del trabajador.
tra_inf_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha que se registra la información en la base.
tra_inf_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha que se modifica la información en la base.

Tabla: sistema.tra_instruccion

Detalle: Guarda los tipos de niveles de estudios que sirven para ingresar la información académica de un trabajador.

Name	Type	Primary	Not Null	Comment
instru_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Número que identifica de forma única el tipo de instrucción
instru_nombre	varchar(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Nombre de la instrucción.

Tabla: sistema.tra_ref_personal

Detalle: Almacena los datos de referencias personales de los trabajadores.

Name	Type	Primary	Not Null	Comment
tra_ref_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Número que identifica de manera única la referencia personal del trabajador
tra_ref_id_trab	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número que relaciona un registro con el trabajador correspondiente
tra_ref_ci	varchar(13)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Cédula de la persona que da la referencia personal del trabajador
tra_ref_nombre	varchar(200)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Nombre de la persona que da la referencia personal
tra_ref_telefonos	varchar(200)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Numero(s) de telefono de la persona que sirve de referencia al trabajador.
tra_ref_detalle	varchar(300)	<input type="checkbox"/>	<input type="checkbox"/>	Detalle de la referencia personal del trabajador
tra_ref_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de ingreso de la referencia personal del trabajador en la base de datos.
tra_ref_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de modificación de la referencia personal del trabajador en la base de datos.

Tabla: sistema.trabajadores

Detalle: Guarda los datos personales de los trabajadores como nombres, estado civil, fecha de nacimiento y datos de contacto.

Name	Type	Primary	Not Null	Comment
tra_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Identificador único del trabajador dentro del sistema.
tra_ci	varchar(13)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Cédula del trabajador
tra_nombre	varchar(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Nombres del trabajador
tra_apellido	varchar(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Apellidos del trabajador
tra_fn	date	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de nacimiento del trabajador.
tra_id_est_civ	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número que guarda la relación con el estado civil, determina el estado civil del trabajador.
tra_sexo	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Bandera que determina si el sexo del trabajador es masculino(true) o femenino(false)
tra_id_nacio	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número que determina la nacionalidad del trabajador
tra_celular	varchar(20)	<input type="checkbox"/>	<input type="checkbox"/>	Número de celular del trabajador
tra_tel1	varchar(20)	<input type="checkbox"/>	<input type="checkbox"/>	Número de teléfono convencional principal.
tra_dir1	varchar(300)	<input type="checkbox"/>	<input type="checkbox"/>	Dirección principal del trabajador.
tra_tel2	varchar(20)	<input type="checkbox"/>	<input type="checkbox"/>	Número de teléfono convencional opcional.
tra_dir2	varchar(300)	<input type="checkbox"/>	<input type="checkbox"/>	Dirección opcional del trabajador.
tra_obser	varchar(500)	<input type="checkbox"/>	<input type="checkbox"/>	Observación adicional del trabajador.
tra_activo	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Determina si el trabajador está activo actualmente en la empresa.
tra_fi	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha que se registra un trabajador en la base de datos.
tra_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha que se modifica un trabajador en la base de datos.
tra_alias	varchar(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Alias o Nombre Corto de identificación del trabajador.
tra_sueldo	double precision	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Sueldo que recibe el trabajador actualmente.
fr_id_actividad	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número que determina el último cargo que realizó el trabajador.
tra_presente	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Determina si el trabajador se encuentra presente en la empresa en la actualidad.
tra_act_fi	date	<input type="checkbox"/>	<input type="checkbox"/>	Fecha que el trabajador inició su última actividad en la empresa.
tra_act_fs	date	<input type="checkbox"/>	<input type="checkbox"/>	Fecha que el trabajador finaliza su última actividad en la empresa.
tra_es_afi	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Determina si el trabajador está afiliado(true) al seguro social.
con_historial	boolean	<input type="checkbox"/>	<input type="checkbox"/>	Bandera de uso interno que determina si se crea un historial del cargo del trabajador.

Tabla: sistema.usu_opciones

Detalle: Es una tabla que guarda la relación entre usuarios y opciones, para conceder permisos a las opciones de las interfaces a cada usuario, para que estas se muestren o no disponibles al usuario cuando ingrese al módulo de restaurant o del hotel.

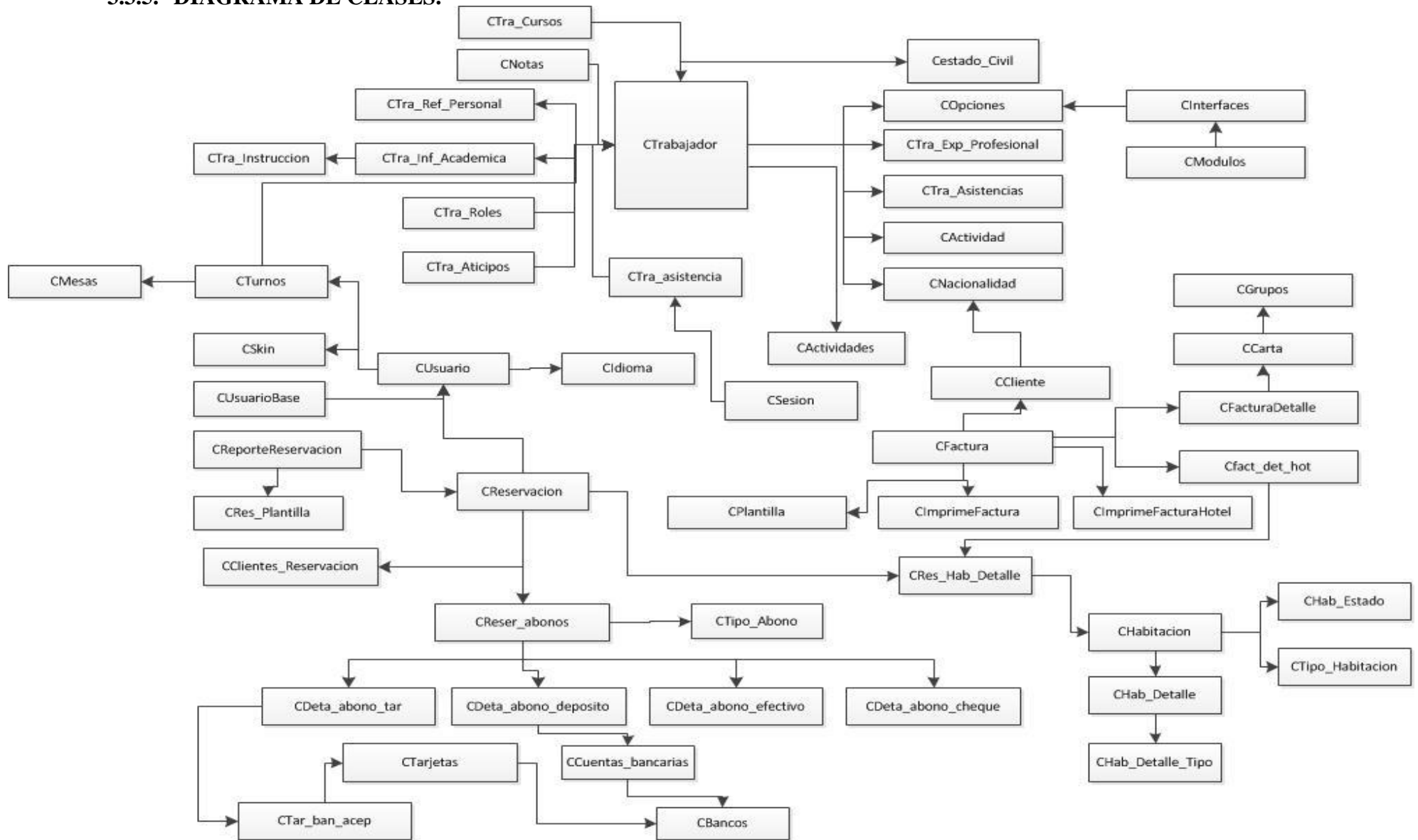
Name	Type	Primary	Not Null	Comment
id_usu_opciones	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Código que identifica de manera única un acceso de un usuario a una opción del sistema.
fr_usuario_id	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número que guarda la relación hacia el usuario.
fr_opcion_id	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número que guarda la relación hacia la opción del sistema.
fi_usu_opcion	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de asignación de acceso del usuario a la opción.
um_usu_opcion	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de modificación de acceso del usuario a la opción.

Tabla: sistema.usuarios

Detalle: Guarda la información relacionada con los usuarios del sistema, y las configuraciones personales definidas por el mismo.

Name	Type	Primary	Not Null	Comment
usu_id	serial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Número que identifica de manera única a un usuario del sistema.
usu_nombres	varchar(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Nombres del Usuario
usu_apellidos	varchar(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Apellidos del Usuario
usu_observacion	varchar(200)	<input type="checkbox"/>	<input type="checkbox"/>	Observación adicional del Usuario
usu_fec_reg	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de creación del Usuario
usu_fec_um	timestamp(2)	<input type="checkbox"/>	<input type="checkbox"/>	Fecha de modificación del Usuario
usu_skin	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número que guarda la relación con el Skin, para determinar el tema preferido de usuario.
usu_login	varchar(20)	<input type="checkbox"/>	<input type="checkbox"/>	Nombre de acceso de usuario para la autenticación de base de datos y de sistema
usu_activo	boolean	<input type="checkbox"/>	<input type="checkbox"/>	Determina si el usuario está activo o no.
usu_fr_idioma	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Número que guarda la relación con los idiomas, para determinar el idioma preferido de usuario.
usu_sexo	boolean	<input type="checkbox"/>	<input type="checkbox"/>	Determina el sexo del usuario. True si es masculino o false si es femenino.
usu_mod_tac	boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Determina si el usuario usa el sistema en un terminal táctil(true) o no(false).

3.3.5. DIAGRAMA DE CLASES.



3.4. DESARROLLO DEL SISTEMA.

3.4.1.DETERMINACIÓN DE PLATAFORMAS.

El sistema puede ser implementado sobre varias plataformas, ya sea Windows, Linux, Mac, sin embargo tanto para los terminales como el servidor (que será el mismo terminal) se usará la plataforma Windows.

3.4.2.DETERMINACIÓN DEL LENGUAJE DE PROGRAMACIÓN.

Java es un lenguaje de programación y es la primera plataforma informática creada por Sun Microsystems en 1995. Es la tecnología subyacente que permite el uso de la programación orientada a objetos, como herramientas, juegos y aplicaciones de negocios. Java se ejecuta en más de 850 millones de ordenadores personales de todo el mundo y en miles de millones de dispositivos, como dispositivos móviles y aparatos de televisión. Por el motivo de ser rápido, seguro y fiable, para el desarrollo del sistema actual, se determina Java como el lenguaje de programación, ya que además de ser multiplataforma y muy potente es software libre lo que disminuye los costos de implementación.

3.4.3.DETERMINACIÓN DEL SISTEMA GESTOR DE BASE DE DATOS.

PostgreSQL es un poderoso sistema de bases de datos objeto-relacional. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación de fiabilidad, integridad de datos y la corrección. Se ejecuta en todos los principales sistemas operativos, incluyendo Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows. Por esos motivos de rendimiento y fiabilidad, además de la característica de ser software libre, se determina PostgreSQL como el Sistema Gestor de Base de Datos para el desarrollo del sistema. Sus principales características son las siguientes:

Límite	Valor
Tamaño máximo de la base de datos	Ilimitado
Tamaño máximo de tabla	32 TB
Tamaño Fila máxima	1,6 TB

Tamaño máximo de campo	1 GB
Número máximo de filas por tabla	Ilimitado
Número máximo de columnas por tabla	250 - 1600 en función de los tipos de columna
Los índices máximos por la tabla	Ilimitado

Tabla 17: Características de PostgreSQL.

Fuente: www.postgresql.org/about/

3.4.4. DETERMINACIÓN DEL TIPO DE ESTRUCTURA DEL SISTEMA.

El tipo de estructura a utilizar será la Estructura Cliente-Servidor, ya que en esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

3.4.4.1. REQUERIMIENTOS DE CLIENTE.

Para el correcto funcionamiento del sistema se requieren los siguientes componentes en el terminal, o cliente:

- Disco Duro: 120 GB.
- Procesador: Intel Pentium 4, o superior.
- Memoria Ram: 1 DDR2 1 GB. Se recomienda 2 GB.
- Main Board: Que soporte el procesador y las memorias.
- Monitor: Cualquier monitor.
- Batería o UPS de 550 Vatios (un computador).
- Impresora Matricial.

Requerimientos opcionales para un mejor rendimiento del sistema:

- Monitor Touch 15” o de 17”: Requeridos para terminales táctiles.
- Tarjeta aceleradora gráfica 125 MB.
- Disco duro externo de 500 GB para guardar respaldos de la base de datos.
- Lector quemador de CD DVD Rom, para guardar los respaldos de la base de datos en CDs o DVDs.

- Impresora y Lector de Códigos de Barra para el registro de los componentes de las habitaciones.

3.4.4.2. REQUERIMIENTOS DEL SERVIDOR.

Para el servidor se requiere como mínimo las siguientes características:

- Procesador Perntium IV 3.0 GHZ doble núcleo.
- Disco duro SCII 120 GB.
- Memoria Ram 1 GB.
- Tarjeta de red 100/100

3.4.5. IMPLEMENTACIÓN DEL SISTEMA.

3.4.5.1. IMPLANTACIÓN.

La implementación del sistema se realizó, por motivos de la infraestructura tecnológica actual con la que cuenta el hotel, solo en un ordenador el mismo que servirá de servidor y de terminal como caja del restaurant, reservación hotelera y administraciones del mismo. Ver anexo 6 y 7.

3.4.5.2. CAPACITACIÓN.

Después de la configuración e instalación del sistema, se realizó la capacitación a las personas, que interactúan con el sistema; entre ellas, los trabajadores que laboran como meseros, camareros, recepcionistas, chef y administradores. Primero se realizó una capacitación teórica de los procesos y opciones implementadas en el sistema (ver anexo 8) y posteriormente se realizó el entrenamiento del uso del programa informático Poseidón Sys V1.0 (ver anexo 9).

3.4.5.3. PRUEBAS Y USO DEL SISTEMA.

Durante el desarrollo del sistema, se realizaron pruebas por procesos, es decir pruebas de las secciones que se iban programando en cada módulo, sin embargo una vez

implementado en su totalidad, se procedió a hacer una prueba de todos sus módulos, con los procesos críticos, a lo cual respondió el sistema sin ninguna anomalía (ver anexo 10). Es así, que por este suceso, el propietario certifica el correcto funcionamiento de cada uno de los módulos que el sistema implementa (ver anexo 11).

3.4.5.4. PLATAFORMA DE USUARIO.

La plataforma utilizada en la implementación del sistema hotelero será Windows XP pero podría ser aplicada a las versiones como 2000, XP, Vista, 7 o superior.

Se elige esta plataforma, ya que no es muy exigente con los recursos de hardware y además los trabajadores tienen conocimientos básicos sobre ella.

3.4.5.5. REQUERIMIENTO DE SOFTWARE.

Los requerimientos necesarios para que funcione el software aplicado en la plataforma mencionada son los siguientes:

Para que el sistema pueda ser utilizado en un terminal, sólo se necesita tener instalado el **Java Runtime Environment o JRE**, el mismo que incluye entre otros el compilador para Java.

En el servidor es necesario tener instalado y configurado el Sistema Gestor de Base de Datos seleccionado, PostgreSQL 9.1 o posterior.

CAPITULO IV: ANÁLISIS DE IMPACTO.

Los niveles de impacto se califican y valoran teniendo en consideración la siguiente tabla:

Valor	Impacto
-3	Impacto Alto negativo
-2	Impacto Medio negativo
-1	Impacto Bajo negativo
0	No hay impacto
1	Impacto Bajo positivo
2	Impacto Medio positivo
3	Impacto Alto positivo

Tabla 18: Clasificación de los niveles de impacto.

4.1. IMPACTO ADMINISTRATIVO

MATRIZ DE IMPACTO

Nivel de Impacto \ Indicador	-3	-2	-1	0	1	2	3
Organización Interna.							X
Atención a Clientes.						X	
Apoyo a toma de decisiones.							X
Registro eficientes de datos.							X
TOTAL						2	9

$$\Sigma = 11$$

$$\text{Nivel de Impacto Administrativo} = \frac{\Sigma}{\text{N. I.}}$$

$$\text{N. I.} = 4$$

$$\text{Nivel de Impacto Administrativo} = 2.75 \approx 3 \text{ Alto positivo}$$

ANÁLISIS:

Gracias a las herramientas implementadas en el sistema hotelero, la organización interna tiene un impacto alto positivo, debido a que los procesos de administración y control han sido sistematizados ayudando a planificar, organizar, gestionar y controlar los factores disponibles como reservaciones, facturación, ventas, y recursos humanos.

La atención a clientes toma un valor medio positivo debido a que el sistema permite una atención más personalizada, ayudando a gestionar procesos organizados de turnos para atención a clientes, pero depende también del factor humano, es decir de la calidad de atención que los trabajadores otorgan a los clientes que ingresan a la empresa en búsqueda de los servicios prestados.

El apoyo a la toma de decisiones asume el calificativo de alto positivo, debido a que el sistema desarrollado e implementado, permite generar reportes de varios tipos en los distintos ámbitos dando en tiempo real, un soporte al administrador para las respectivas tomas de decisiones referentes a temas de control de personal, ventas y reservaciones realizadas. Gracias al aporte que brinda el sistema desarrollado, el personal administrativo puede tomar decisiones que le permitan llevar a la empresa al desarrollo y progreso de la misma.

El registro eficiente de los datos tiene un valor alto positivo, debido a que el sistema guarda los datos de manera organizada, normalizada y segura para su posterior visualización, análisis y proceso para obtener información válida y veraz en el momento que el usuario desee. No hay límites de ingreso de datos en la herramienta creada, siempre y cuando la estructura informática física lo soporte.

4.2. IMPACTO TECNOLÓGICO

MATRIZ DE IMPACTO

Nivel de Impacto	-3	-2	-1	0	1	2	3
Indicador							
Uso de Tecnologías de la Información y la comunicación.							X
Agilizar Procesos.							X
Automatización de registros.							X
Utilización de Software Libre.							X
TOTAL							12

$$\Sigma = 12$$

$$\text{Nivel de Impacto Tecnológico} = \frac{\Sigma}{\text{N. I.}}$$

$$\text{N. I.} = 4$$

Nivel de Impacto Tecnológico = 3 Alto positivo

ANÁLISIS:

Con el uso del programa informático en la empresa, el uso de Tecnologías de Información y la comunicación (TIC's) toma un valor de impacto alto positivo debido a que anteriormente no existía un sistema de administración que use tecnología informática, varios de los registros de información se realizaban de forma manual en hojas de papel y cuadernos de control, registros como las ventas no se almacenaban por ser una gran cantidad de información generada casi imposible de controlar. En la actualidad, el sistema de administración se apoya en herramientas tecnológicas recientes, con el uso del computador y un sistema informático que controla y registra los procesos fundamentales del hotel restaurant fogón manabita que antes no se llevaban a cabo.

La agilización de los procesos toma un valor de impacto alto positivo debido a que se mejora el proceso de atención a clientes, asignación de habitaciones, consultas de disponibilidad, reportes de ventas, entre otros procesos internos de la empresa. Dichos procesos anteriormente tomaban mucho tiempo y en ocasiones no se realizaban por lo cual este es uno de los impactos más notables del sistema actual.

Gracias a la implementación tecnológica utilizada en el proyecto, la automatización de registros toma un valor de impacto alto positivo, debido a que todos los procesos realizados con el sistema son registrados en la base de datos, junto con los atributos necesarios para identificar el usuario, la fecha en que se realizó dicho ingreso y modificación del registro.

El indicador de uso de software libre toma el valor alto positivo debido a que las herramientas utilizadas para el desarrollo e implementación del sistema hotelero son de libre uso es decir sin ningún costo de licenciamiento.

4.3. IMPACTO SOCIAL

MATRIZ DE IMPACTO

Nivel de Impacto	-3	-2	-1	0	1	2	3
Indicador							
Capacitación Personal.							X
Responder con agilidad a las necesidades del cliente.						X	
Comodidad y bienestar para el personal.							X
Mejorar intercambio cultural.						X	
TOTAL						4	6

$$\Sigma = 10$$

$$\text{Nivel de Impacto Social} = \frac{\Sigma}{\text{N. I.}}$$

$$\text{N. I.} = 4$$

$$\text{Nivel de Impacto Social} = 2.5 \approx 3 \text{ Alto Positivo}$$

ANÁLISIS:

Luego de la implementación del sistema hotelero, el indicador Capacitación de Personal tomó un valor alto positivo ya que existió un enriquecimiento del conocimiento de los trabajadores, debido a que los mismos deben tener las destrezas básicas que les permitan usar e interactuar de manera adecuada con el sistema informático desarrollado. Los niveles de capacitación fueron dependiendo a los tipos de usuarios que se relacionan con el sistema.

Responder con agilidad a las necesidades del cliente toma el calificativo de medio positivo, ya que al contar con un programa informático, se pueden tener respuestas más ágiles, rápidas y efectivas a los clientes, como consultas, y procesos como la facturación.

El indicador comodidad y bienestar para el personal corresponde a crear una posibilidad de compensación de esfuerzos con el registro adecuado de las horas laboradas, y a facilitar procesos que se registraban manualmente, ello hacen que los trabajadores se sientan cómodos y sin desánimo, visualizando claramente los anticipos, controlando mejor los préstamos realizados, obteniendo de la manera más sencilla los reportes asociados a cada trabajador, dando el valor a este indicador de alto positivo.

Al mejorar los servicios de la empresa se atraerá mayor clientela tanto al hotel como al balneario de Tonsupa, lo cual permitirá un mayor intercambio cultural y de conocimientos, construyendo puentes de fraternidad con los que visitan el balneario, por ello el indicador Mejorar intercambio cultural, toma el valor de bajo positivo.

4.4. IMPACTO ECONÓMICO

MATRIZ DE IMPACTO

Nivel de Impacto	-3	-2	-1	0	1	2	3
Indicador							
Infraestructura Tecnológica.			X				
Software Libre.							X
Mayor Clientela.						X	
Mejorar Administración del Hotel.							X
TOTAL			-1			2	6

$$\Sigma = 7$$

$$\text{Nivel de Impacto Económico} = \frac{\Sigma}{\text{N. I.}}$$

$$\text{N. I.} = 4$$

$$\text{Nivel de Impacto Económico} = 1.75 \approx 2 \text{ Medio positivo}$$

ANÁLISIS:

Económicamente, la compra de los equipos necesarios para la implementación del sistema genera un impacto bajo negativo, ya que la empresa tuvo que realizar un desembolso para la compra de los mismos. Sin embargo, dentro del mismo punto, esta adquisición y la implementación del sistema permite un mejor control y registro de los procesos llevados a cabo en la empresa, lo cual se tomaría como una inversión que controlará los ingresos y egresos, producto de las actividades que realiza el hotel.

El uso de software libre tiene un impacto alto positivo en el desarrollo del sistema ya que gracias a la tecnología utilizada, los costos de licenciamiento por el uso del lenguaje de programación como del uso del sistema gestor de base de datos, no afectan el factor económico de la empresa, ya que estas herramientas según la *Free Software Foundation* pueden ser usadas, modificadas y distribuidas de manera libre y sin ningún costo económico, disponibles gratuitamente desde las páginas de sus autores.

Gracias a la implementación del sistema hotelero, y al aporte que da el mismo en el desempeño de las actividades realizadas en la empresa, una vez que se mejoran los procesos de atención a clientes, el incremento de clientes afecta proporcionalmente el factor económico de la empresa, ya que el ingreso principal de la empresa proviene de los servicios que se le dan al mismo. Por ello, el indicador mayor clientela, tiene un valor de impacto medio positivo.

El Mejorar la Administración del Hotel, tiene un valor alto positivo, debido a que el sistema mejora e implementa procesos de administración que permite asignar mejor los recursos con los que se cuenta en la empresa, permitiendo obtener reportes que ayudan a la toma de decisiones al personal administrativo, ello implica que el factor económico mejore y que la empresa se proyecte mejor en el entorno que se desenvuelve.

4.5. IMPACTO AMBIENTAL

MATRIZ DE IMPACTO

Nivel de Impacto	-3	-2	-1	0	1	2	3
Indicador							
Diminución de uso de papel.						X	
Uso de Equipos Informáticos.			X				
TOTAL			-1			2	

$$\Sigma = 1$$

$$\text{Nivel de Impacto Ambiental} = \frac{\Sigma}{\text{N. I.}}$$

$$\text{N. I.} = 2$$

$$\text{Nivel de Impacto Ambiental} = 0.5 \approx 1 \text{ Bajo Positivo}$$

ANÁLISIS:

Disminución de uso de papel, es un impacto medio positivo, debido a que el sistema implementado disminuye al máximo el uso del papel, gracias a que trabaja con plantillas de impresión, lo que permite tener varios tamaños de un comprobante según la cantidad de información a imprimir, optimizando el uso del papel. También se da la opción de generar archivos en varios formatos para que estos sean guardados de forma digital, y enviados a los clientes en caso de ser requerido mediante correo electrónico. Así mismo, el presente sistema deja a un lado los procesos que implicaban el uso de registros manuales en papel.

El uso de equipos informáticos tiene un impacto bajo negativo, afectando al factor ambiental ya que los mismos producen una demanda de consumo de energía eléctrica, cosa que no sucede con el sistema manual, además que los mismos contaminan el ambiente cuando cumplen su vida útil, si no se reciclan correctamente.

4.6. MATRIZ DE IMPACTO GENERAL

Nivel de Impacto	-3	-2	-1	0	1	2	3
Impacto Administrativo.							X
Impacto Tecnológico.							X
Impacto Social.							X
Impacto Económico.						X	
Impacto Ambiental.					X		
TOTAL					1	2	9

$$\sum = 12$$

$$\text{Nivel de impacto general} = \frac{\sum}{\text{N. I.}}$$

$$\text{N. I.} = 5$$

$$\text{Nivel de impacto general} = 2.4 \approx 2 \text{ Medio Positivo}$$

ANÁLISIS GENERAL:

El sistema informático implementado tuvo un impacto general medio positivo, esto es gracias a que el estudio realizado permitió determinar cuáles eran los procesos más comunes para poder implementarlos.

De esta manera, el aporte que la herramienta brinda se dirige a varios ámbitos entre ellos el administrativo, el cual fue mejorado en un nivel alto positivo debido a que los procesos de tomas de decisiones se ven apoyados con los reportes en tiempo real que el sistema ofrece, aportando con la administración de la empresa de forma más eficiente y efectiva, así también la información se centraliza para ser tomada de un mismo lugar con la automatización de los procesos de gestión que se realizan diariamente como facturación y gestión de reservaciones.

Otro ámbito implicado es el social, con un impacto alto positivo gracias a que el uso de tecnología implementada en los procesos a cargo del recurso humano se facilita, haciendo que estas sean de menos esfuerzos a las personas encargadas de dichas labores.

**CONCLUSIONES
Y
RECOMENDACIONES**

CONCLUSIONES

- Un software empresarial es el aquel tipo de software que permite automatizar procesos para mejorar la administración de la empresa, permitiendo tener un mejor control y registro de los datos generados con la finalidad de tener una herramienta de apoyo para la toma de decisiones y optimización de recursos.
- El sistema se ha desarrollado empleando herramientas de Software libre lo cual evita la inversión en la compra de licencias ya que este tipo de software se puede utilizar libremente, el mismo es multiplataforma, lo cual hace que el programa sea independiente del sistema operativo que en su momento decida utilizar el hotel.
- El sistema necesita los recursos básicos para el correcto funcionamiento del mismo, así como con el compromiso serio de los dueños y trabajadores del hotel, lo cual hace factible su utilización.
- Con la implementación del sistema se ha mejorado el proceso de gestión del Restaurant-Hotel Fogón Manabita, debido a que toda la información que se genera en el mismo actualmente se encuentra digitalizada y en tiempo real, facilitando la toma de decisiones por parte del Administrador.
- Al emplear un sistema informático, el hotel se pone a la par de sus competidores que han visto en la tecnología una oportunidad para mejorar sus procesos cotidianos de control y gestión, satisfaciendo las necesidades de sus clientes.
- El proyecto presenta un impacto medio positivo debido a que utiliza tecnología de última generación, esto permitió que los procesos se automaticen de la mejor manera, haciendo que la sociedad le de una buena acogida, de la misma forma los administradores de la empresa ya que al usar software libre los costos sólo se ven reflejados en los equipos informáticos.

RECOMENDACIONES

- En la actualidad los recursos informáticos juegan un papel muy importante en las actividades de una empresa para mejorar sus procesos de administración y control por ello es importante implementar sistemas informáticos que faciliten las gestiones y tomas de decisiones por parte de la gerencia.
- Es importante que cuando se contrate a personal nuevo, el cual va a realizar una de las labores contempladas dentro del programa informático, se debe dar la capacitación correspondiente para que pueda interactuar correctamente con el sistema, especialmente con las personas que ingresan información para que ésta se mantenga íntegra y sea útil al momento de utilizarla.
- Se recomienda respaldar constantemente la información de la base de datos del sistema en una unidad de almacenamiento externo, de manera organizada, para en caso de suceso de algún riesgo, se pueda realizar una restauración de la información al punto más reciente para así reanudar el servicio lo más pronto posible y no afecte los procesos y actividades del hotel disminuyendo pérdidas económicas y de información.
- Incrementar el número de terminales para que los recursos que brinda el sistema de una estructura cliente servidor sea mejor usada, así se dividan los procesos internos y que no haya una sobrecarga de tareas al terminal existente.
- Usar monitores táctiles en los terminales determinados, específicamente a la realización de pedidos y facturación en el restaurant, ya que estos procesos deben de hacerse lo más rápido posible, para brindar al cliente un mejor servicio.

FUENTE DE INFORMACIÓN

BIBLIOGRAFÍA

POSSO, M. (2006): Metodología para el trabajo de grado (tesis y proyectos), (3^{ra} edición), Ibarra – Ecuador.

GERALD, W. (1975): Administración moderna de hoteles y moteles (Modern hotel and motel management), (1^{era} Edición), EE.UU.

CARLOS, G. (1996): Enciclopedia práctica profesional de turismo, hoteles y restaurantes, MMIII editorial océano, MÉXICO.

KEN, A., JAMES G., David, H. (2005): Java(TM) Programming Language (4^{ta} Edición), Addison-Wesley Professional 2005.

VARTAN, P. (1999): Java GUI development (1^{ra} Edición), Sams Indianapolis, USA.

Korry, D. (2005): PostgreSQL (2da Edición), Sams Indianapolis, USA.

PÁGINAS Y SITIOS WEB

[http://es.wikipedia.org/wiki/Sistema de informaci%c3%b3n](http://es.wikipedia.org/wiki/Sistema_de_informaci%c3%b3n)

<http://www.ecured.cu/index.php/portal:inform%c3%a1tica/software>

[http://es.wikipedia.org/wiki/herramientas de gesti%c3%b3n](http://es.wikipedia.org/wiki/herramientas_de_gesti%c3%b3n)

<http://www.monografias.com/trabajos29/beneficios-erp/beneficios-erp.shtml>

[http://es.wikipedia.org/wiki/planificaci%c3%b3n de recursos empresariales](http://es.wikipedia.org/wiki/planificaci%c3%b3n_de_recursos_empresariales)

http://www.cepeu.edu.py/libros_electronicos_3/lpcu097%20-%2001.pdf

<http://www.java.com/es/about/>

<http://html.rincondelvago.com/adminispucese>

<tracion-y-ventas-en-un-hotel.html>

<http://es.wikipedia.org/wiki/hotel>

<http://www.monografias.com/trabajos150/hoteles/administracion-hotelera.shtml>

[http://www.trabajodehoteles.com /clasificaci%c3%b3n de hoteles](http://www.trabajodehoteles.com /clasificaci%c3%b3n_de_hoteles)

[http://es.wikipedia.org/wiki/planificaci%c3%b3n de recursos empresariales](http://es.wikipedia.org/wiki/planificaci%c3%b3n_de_recursos_empresariales)

[http://es.wikipedia.org/wiki/customer relationship management](http://es.wikipedia.org/wiki/customer_relationship_management)

[http://es.wikipedia.org/wiki/Sistema de informaci%C3%B3n](http://es.wikipedia.org/wiki/Sistema_de_informaci%C3%B3n)

http://html.rincondelvago.com/herramientas-case_4.html

<http://members.tripod.com/~gепsea/sistema.html>

[http://es.wikipedia.org/wiki/Teoria-de los sistemas de desarrollo](http://es.wikipedia.org/wiki/Teoria-de_los_sistemas_de_desarrollo)

GLOSARIO

Nodo: En redes de computadoras cada una de las máquinas es un nodo, y si la red es Internet, cada servidor constituye también un nodo.

Software: Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación.

ERP (Planificación de Recursos Empresariales): son sistemas de información gerenciales que integran y manejan muchos de los negocios asociados con las operaciones de producción y de los aspectos de distribución de una compañía en la producción de bienes o servicios.

CRM (Software para la administración de la relación con los clientes) Sistemas informáticos de apoyo a la gestión de las relaciones con los clientes, a la venta y al marketing.

TIC: Acrónimo de Tecnologías de la Información y las Comunicaciones es el conjunto de tecnologías que permiten la adquisición, producción, almacenamiento, tratamiento, comunicación, registro y presentación de informaciones, en forma de voz, imágenes y datos contenidos en señales de naturaleza acústica, óptica o electromagnética.

Free Software Foundation (Fundación para el software libre): es una organización creada en octubre de 1985 por Richard Stallman y otros entusiastas del software libre con el propósito de difundir este movimiento.

Debugging o Depuración de programas, es el proceso de identificar y corregir errores de programación. En inglés se le conoce comodebugging, ya que se asemeja a la eliminación de bichos (bugs), manera en que se conoce informalmente a los errores de programación.

Abstracción: es lo que permite analizar y desarrollar las características esenciales de un objeto (requerimiento), despreocupándonos de las menos relevantes.

Java: es un lenguaje de programación multiplataforma orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90.

SQL: El lenguaje de consulta estructurado o SQL (por sus siglas en inglés structured query language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas.

PostgreSql: Es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD.

Clase: Es una construcción que se utiliza como un modelo (o plantilla) para crear objetos de ese tipo.

Interfaz de usuario: La interfaz de usuario es el medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo, normalmente suelen ser fáciles de entender y fáciles de accionar.

CASE: (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son

Licencia: Es el permiso o autorización para realizar, utilizar u domificar un sistema o programa informático con condiciones que se aceptan al momento de adquirir dicho contrato.

ANEXOS

ANEXO (1)

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR SEDE ESMERALDAS

Encuesta dirigida a los Trabajadores del Hotel Fogón Manabita.

Estimado encuestado sírvase a contestar las siguientes preguntas, las cuales servirán para determinar el grado de aceptabilidad de un Sistema Hotelero para la automatización de procesos del Hotel Fogón Manabita – Tonsupa.

1. ¿Cuántas horas de servicio presta al hotel aproximadamente por día?
 De 1 a 4 hrs.
 De 4 a 8 hrs.
 De 8 a 12 hrs.
 Más de 12 hrs.
2. ¿En qué medida maneja usted programas informáticos como Microsoft Office Word y Excel?
 Nada Medianamente Mucho
3. ¿De implementarse un Sistema Informático, estaría dispuesto a capacitarse para el uso del mismo?
 Si No

¿Por qué? _____

4. ¿Cree que con el uso de un Sistema Informáticos los procesos de Administración (reservaciones, restaurant, bar, facturación) serían más eficientes?
 Si No No lo sé

¿Por qué? _____

5. ¿Estaría de acuerdo con que sus roles de pago y anticipos se registren en un sistema de aplicación para el mejor control de los mismos?
 Si No No lo sé

¿Por qué? _____

6. ¿Cree que son óptimos los procesos que se realizan dentro del Hotel, para lograr una buena atención a los clientes?
 Si No Más o menos

¿Por qué? _____

7. En qué medida el sistema manual de administración actual del hotel satisface todas las necesidades del mismo.
Excelente
Muy Bueno
Bueno
Regular
Malo

¿Por qué? _____

ANEXO (2)
PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
SEDE ESMERALDAS

Encuesta dirigida a los Clientes del Hotel Fogón Manabita.

Estimado encuestado sírvase a contestar las siguientes preguntas, las cuales servirán para determinar el grado de aceptabilidad de un Sistema Hotelero para la automatización de procesos del Hotel Fogón Manabita – Tonsupa.

1. ¿Se siente satisfecho con el proceso de facturación de los servicios que se lleva actualmente en el Hotel Fogón Manabita?

Más o menos Si No

¿Por qué? _____

2. ¿Está satisfecho con el proceso de asignación de habitaciones?

Si No

¿Por qué? _____

3. ¿Le gustaría que el Hotel cuente con un mejor control para la devolución de objetos olvidados por los clientes?

Si No

4. ¿Está conforme con la manera con que se administran los pedidos en el Hotel (existieron equivocaciones, ilegibilidad y extravío)?

Si No

¿Por qué? _____

5. ¿Cree que hay un buen control de parte del Hotel para no caer en el error de atender primero a Clientes que llegaron después de otros?

Si No

¿Por qué? _____

6. ¿Le gustaría tener la posibilidad de separar una habitación y reservarla para planificar sus vacaciones a tiempo?

Si No

ANEXO (3)
PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
SEDE ESMERALDAS

Entrevista dirigida al Administrador del Hotel.

Nombre del Entrevistado:

Fecha:

- 1) ¿Cuál es la Misión y la Visión de su hotel?

- 2) ¿Qué servicios Brinda el Hotel a los turistas Nacionales y Extranjeros que visitan el balneario Tonsupa?

- 3) ¿Cuál es el proceso que se realiza normalmente al momento que llega un Cliente al Hotel?

- 4) ¿Cuáles son las obligaciones legales y/o leyes que inciden en el funcionamiento del Hotel?

- 5) ¿Cuál es el número de personas que trabajan en su empresa aproximadamente por áreas y quiénes se encargan del sistema de administración actual?

- 6) Cuáles son las dificultades al momento de conocer el total de ingresos y egreso que se realizan en el hotel en un momento dado

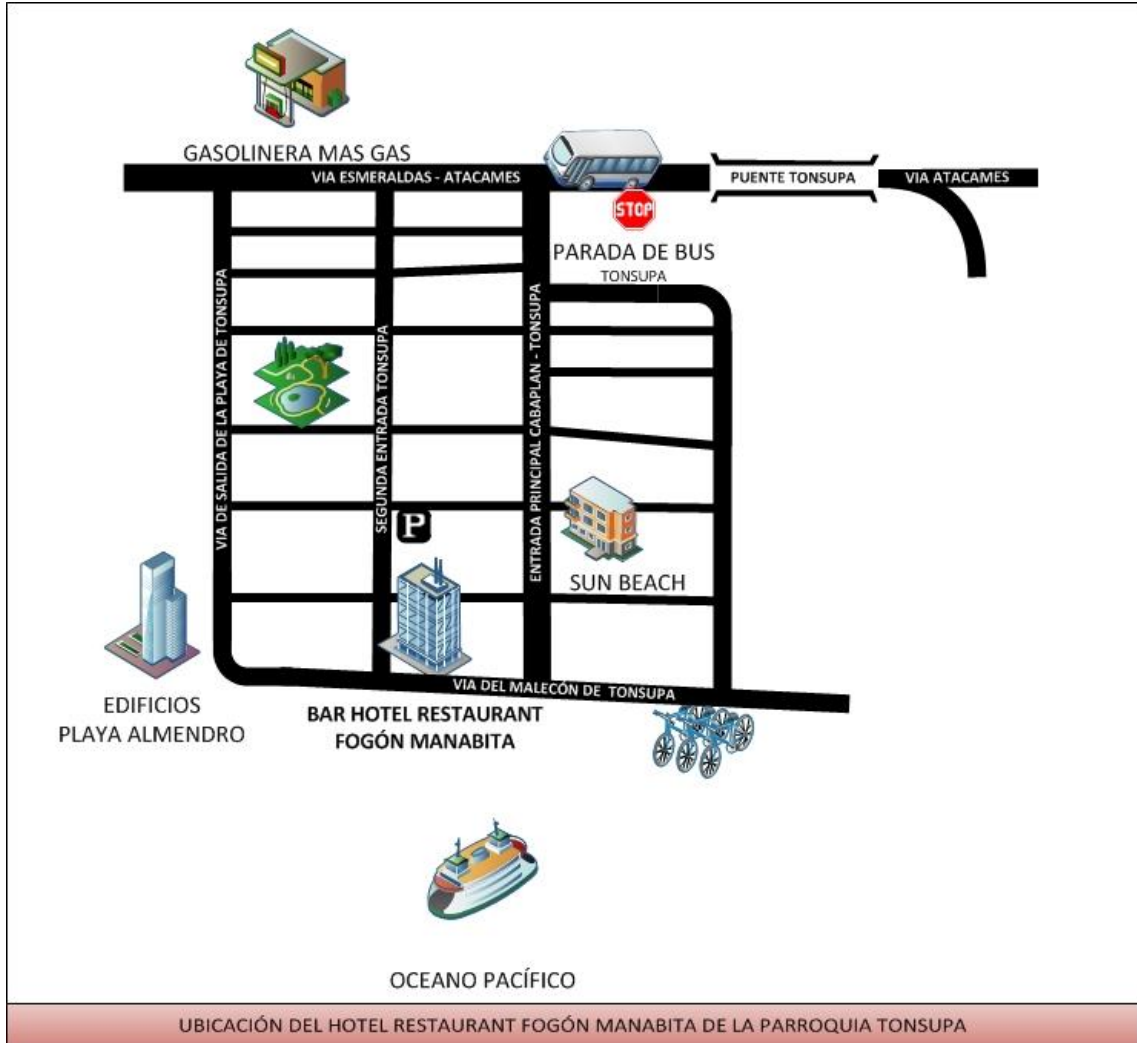
- 7) ¿Qué debilidades cree que tiene el sistema de administración de su hotel frente a la competencia?

- 8) ¿Qué fortalezas cree que tiene el sistema de administración de su hotel frente a la competencia?

- 9) ¿Cuál es el principal problema que evidencian los clientes del hotel respecto al sistema de administración actual?

- 10) ¿Actualmente, cómo es está organizada internamente su empresa?

ANEXO (4)
CROQUIS DE UBICACIÓN DEL HOTEL RESTAURANT FOGÓN
MANABITA.



ANEXO (5)
BAR HOTEL RESTAURANT FOGÓN MANABITA.



ANEXO (6)

INSTALACIÓN DE LOS EQUIPOS INFORMÁTICOS PARA LA CAJA/RECEPCIÓN EN EL BAR HOTEL RESTAURANT FOGÓN MANABITA.



ANEXO (7)

**INSTALACIÓN DEL SISTEMA Y CONFIGURACIÓN DE LOS EQUIPOS
INFORMÁTICOS PARA LA CAJA/RECEPCIÓN EN EL BAR HOTEL
RESTAURANT FOGÓN MANABITA.**



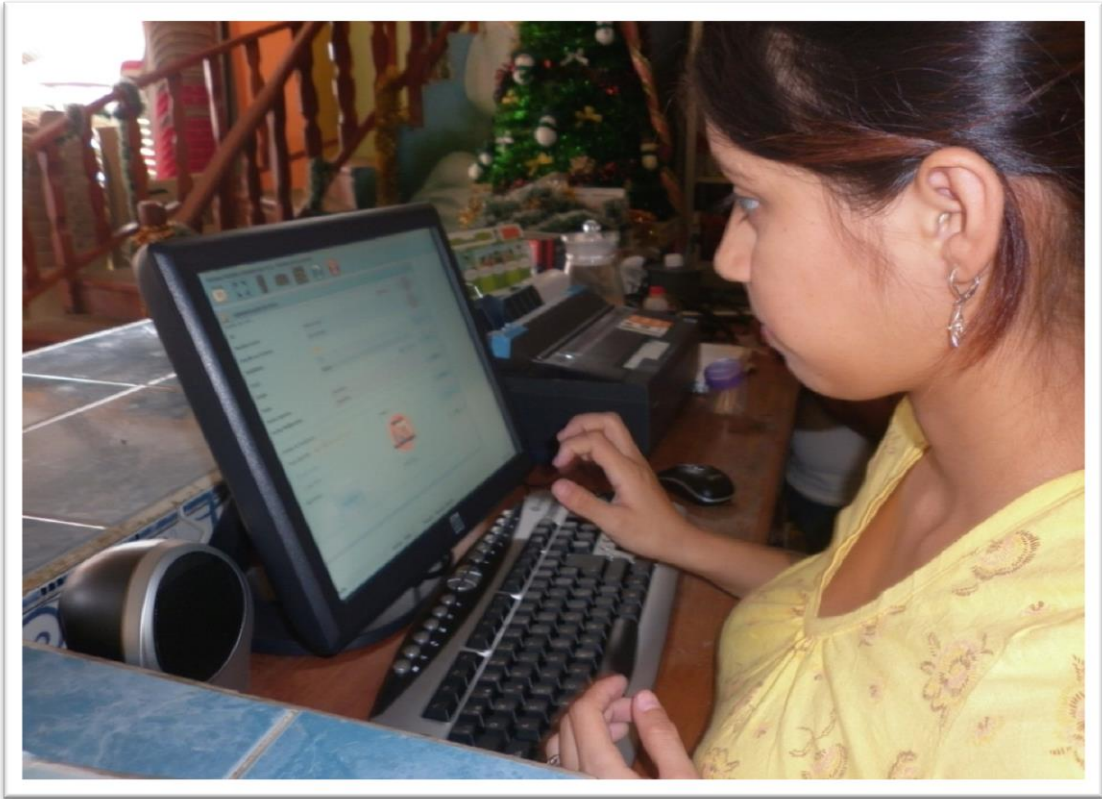
ANEXO (8)

**CAPACITACIÓN TEÓRICA SOBRE LA AUTOMATIZACIÓN DE PROCESOS
CON EL SISTEMA INFORMÁTICO POSEIDÓN SYS V1.0.**



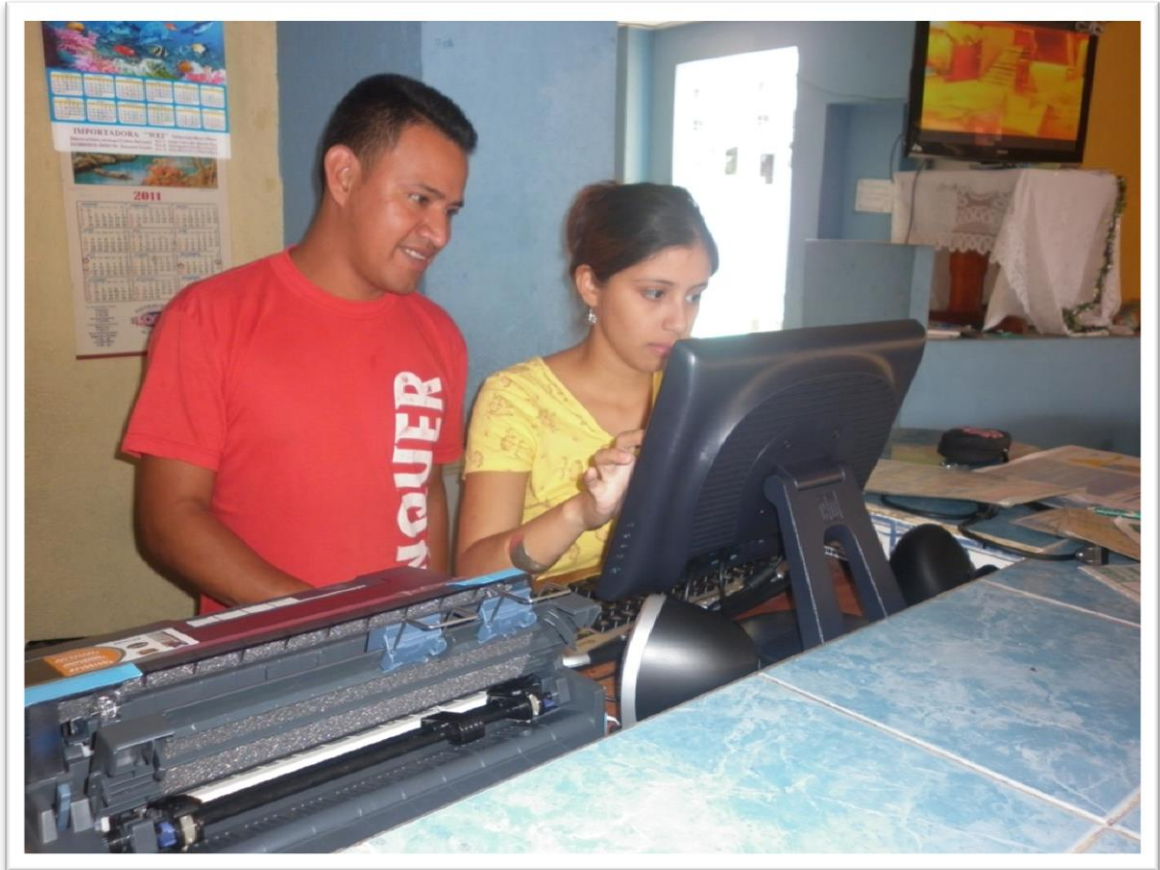
ANEXO (9)

**CAPACITACIÓN PRÁCTICA CON EL SISTEMA INFORMÁTICO POSEIDÓN
SYS V1.0 EN EL BAR HOTEL RESTAURANT FOGÓN MANABITA.**



ANEXO (10)

**USO DEL SISTEMA INFORMÁTICO POSEIDÓN SYS V1.0 EN EL BAR
HOTEL RESTAURANT FOGÓN MANABITA.**



ANEXO (11)

**CERTIFICADO DE IMPLEMENTACIÓN DEL SISTEMA INFORMÁTICO
POSEIDÓN SYS V1.0 DEL BAR HOTEL RESTAURANT FOGÓN MANABITA.**

Tonsupa, 21 de diciembre del 2011.

Srs.

Pontificia Universidad Católica del Ecuador Sede Esmeraldas.

Presente.

De mis consideraciones,

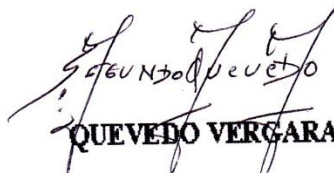
Mediante la presente hago extensivo mis más sinceros saludos, y agradecimiento por la labor comunitaria que vienen realizando en los diferentes ámbitos a favor de la colectividad Esmeraldeña.

Yo Sr. Quevedo Vergara Segundo con cedula de identidad 1303395451, gerente propietario del "Bar Hotel Restaurant Fogón Manabita" de la ciudad de Tonsupa, doy fe de que el Sr. Quevedo Garay Dubal egresado de la carrera de Sistemas de su institución, ha venido trabajando en la implementación del Sistema Hotelero Poseidón Sys V1.0 el mismo que es su proyecto de Tesis para la Obtención del Título de Ingeniero de Sistemas y Computación.

La implementación del Sistema se llevó a cabo en el Hotel antes mencionado entre el 1 y el 15 de diciembre del presente año, culminando con la capacitación y las pruebas supervisadas por el personal administrativo, asegurando el correcto funcionamiento de cada uno de los módulos que el sistema dispone. Cabe mencionar que todos los trabajos realizados fueron sin fin de lucro por parte del autor del Sistema.

El señor Dubal Quevedo Garay puede hacer uso de este documento como estime conveniente.

Atentamente,


QUEVEDO VERGARA SEGUNDO.

Gerente Propietario del Bar Hotel Restaurant Fogón Manabita.