



CAPÍTULO V

DISEÑO ORIENTADO A OBJETOS

5.1 DISEÑO DEL SISTEMA

Hacer un software para dar solución a requerimientos puede ser algo muy sencillo. Sin embargo, lograr que cumpla su objetivo y sea un sitio interesante para los usuarios es el gran reto de todo ingeniero en sistemas.

Parte del éxito de un software es su presentación ante el usuario. Por ejemplo, imagine que está ingresando información por un tiempo mayor a 30 minutos y empieza a ver la pantalla con dificultad por los colores fuertes que tiene el software. Es por esto que el diseño tiene una influencia increíble con el usuario final.

5.1.1 DISEÑO DE INTERFACES

El diseño de interfaces es una disciplina que estudia y trata de poner en práctica procesos orientados a construir la interfaz más usable posible, dadas ciertas condiciones de entorno

Para esto hemos tomado en cuenta los siguientes puntos

-  Utilidad
-  Facilidad de uso
-  Facilidad de Aprendizaje
-  Apreciación



5.1.1.1 UTILIDAD

La utilidad es la capacidad que tiene una herramienta para ayudar a cumplir tareas específicas.

La utilidad del sistema es el control de inventarios por medio de una filosofía de conteo, en donde podemos crear el intervalo de días para un determinado conteo y clasificar el producto a este tipo.

Así al final del mes tenemos con un 80 % de productos ya revisados y cuadrados en donde nos quedaría solo por cuadrar el último periodo.

5.1.1.2 FACILIDAD DE USO

El sistema permite la facilidad de navegación y ejecución de procesos independientes que ayudan a determinar el flujo de datos y la obtención de resultados directos y confiables.

La facilidad de uso está en relación directa con el conocimiento de alimentos y bebidas, por ejemplo, recetas, costos, eventos, compras, etc.

5.1.1.3 FACILIDAD DE APRENDIZAJE

El sistema cuenta con la ayuda en línea y cada campo está debidamente etiquetado para visualizar que es lo que se requiere que el usuario ingrese.



Adicional está con distintos colores los campos que son obligatorios de los opcionales.

5.1.1.4 ESPECIFICACIÓN DE ESTÁNDARES DE PROGRAMACIÓN

Independientemente de los algoritmos usados, hay muchas formas y estilos de programar. La legibilidad de un programa es demasiado importante como para prestarle la atención que merece por parte del programador.

Los programas, a lo largo de su vida, se van quedando obsoletos debido a cambios en su entorno. Un programa pensado para una determinada actividad, es muy normal tener que modificarlo porque cambie dicha actividad o porque decidamos incluirle nuevas posibilidades que antes no estaban previstas. De aquí las múltiples versiones que sacan al mercado las empresas de programación

Aquí utilizamos para calidad de software las Normativas ISO

- ISO 9000: Gestión y aseguramiento de calidad (conceptos y directrices Generales)
- Recomendaciones externas para aseguramiento de la calidad (ISO 9001, ISO 9002, ISO 9003)
- Recomendaciones internas para aseguramiento de la calidad (ISO 9004)



5.1.1.5 FACTORES QUE DETERMINAN LA CALIDAD DEL SOFTWARE DE ACUERDO A ESTAS NORMATIVAS

Se clasifican en tres grupos:

5.1.1.5.1 OPERACIONES DEL PRODUCTO: CARACTERÍSTICAS

OPERATIVAS

– *Corrección* (Si Hace lo que se le pide)

- El grado en que una aplicación satisface sus especificaciones y consigue los objetivos encomendados por el cliente. El sistema cumple con los módulos expuesto en los objetivos

– *Fiabilidad* (Si Lo hace de forma fiable todo el tiempo)

- El grado que se puede esperar de una aplicación lleve a cabo las operaciones especificadas y con la precisión requerida

– *Eficiencia* (Qué recursos hardware y software necesito?)

- La cantidad de recursos hardware y software que necesita una aplicación para realizar las operaciones con los tiempos de respuesta adecuados. En el capítulo II se detalla los recursos de software y hardware que se utilizan en el desarrollo del presente proyecto. Y el capítulo III en Especificación de requerimientos del software y hardware



– *Integridad* (Si Puedo controlar su uso)

- El grado con que puede controlarse el acceso al software o a los datos a personal no Autorizado. Está controlado por medio de clases de perfiles para cada usuario.

- *Facilidad de uso* (Si Es fácil y cómodo de manejar)

- El esfuerzo requerido para aprender el manejo de una aplicación, trabajar con ella, Introducir datos y conseguir resultados. El sistema está planificado para que el usuario pueda navegar de manera fácil, sin embargo es importante tener un conocimiento básico de alimentos y bebidas.

5.1.1.5.2 REVISIÓN DEL PRODUCTO: CAPACIDAD PARA SOPORTAR CAMBIOS

– *Facilidad de mantenimiento* (Si Puedo localizar los fallos)

- El esfuerzo requerido para localizar y reparar errores. Al ser nosotros los desarrolladores tenemos las fuentes y las posibilidades de realizar todo tipo de mantenimiento.

– *Flexibilidad* (Si Puedo añadir nuevas opciones)

- El esfuerzo requerido para modificar una aplicación en funcionamiento. En cada módulo se puede avanzar y desarrollar varias opciones que no están en el alcance de la tesis. Pero que sería de gran beneficio para el hotel



– *Facilidad de prueba* (Si Puedo probar todas las opciones)

- El esfuerzo requerido para probar una aplicación de forma que cumpla con lo especificado en los requisitos

5.1.1.5.3 TRANSICIÓN DEL PRODUCTO: ADAPTABILIDAD A NUEVOS ENTORNOS

– *Portabilidad* (Si Puedo usarlo en otra máquina)

- El esfuerzo requerido para transferir la aplicación a otro hardware o sistema operativo. El sistema es cliente servidor, se puede intalar en las máquinas que sean necesarios.

– *Reusabilidad* (Si Puedo utilizar alguna parte del software en otra aplicación?)

- Grado en que partes de una aplicación pueden utilizarse en otras aplicaciones
El ejemplo para esta norma es el módulo de ventas que puede trasladarse a cualquier equipo o aplicación que tenga una variedad de productos.

– *Interoperabilidad* (Si Puede comunicarse con otras aplicaciones o sistemas Informáticos)

- El esfuerzo necesario para comunicar la aplicación con otras aplicaciones o sistemas Informáticos. Al tener una base de datos es muy fácil importar datos desde Excel o base dbase que tenga el mismo formato.



5.1.1.6 FACILIDAD DEL MANTENIMIENTO

Por otra parte, también es frecuente que uno deba modificar un programa escrito por otro programador. Es entonces, cuando hay que modificar un programa escrito hace cierto tiempo o escrito por otro programador, cuando salen los problemas de legibilidad de un programa. Para poder modificarlo, primero hay que comprender su funcionamiento, y para facilitar esta tarea el programa debe estar escrito siguiendo unas normas básicas. La tarea de **mantenimiento** del software (corregir y ampliar los programas) es una de las tareas más árduas del ciclo de vida del software. Por eso, al programar debemos intentar que nuestros programas sean lo más expresivos posibles, para ahorrarnos tiempo, dinero y quebraderos de cabeza a la hora de modificarlos.

La mayor parte del tiempo de mantenimiento de un programa se emplea en estudiar y comprender el código fuente existente. Esto es especialmente importante cuando se trabaja en grupo pero no es exclusivo de este modo de trabajo. Naturalmente, esto es más importante cuantas más líneas de código tenga el programa, pero incluso en programas pequeños puede perderse mucho tiempo intentando comprender su funcionamiento.

Unas **normas de estilo** en programación, son tan importantes que todas las empresas dedicadas a programación imponen a sus empleados una mínima uniformidad, para facilitar el intercambio de programas y la modificación por cualquier empleado, sea o no el programador inicial. Por supuesto, cada



programa debe ir acompañado de una documentación adicional, que aclare detalladamente cada módulo del programa, objetivos, algoritmos usados, ficheros.

No existen un conjunto de reglas fijas para programar con legibilidad, ya que cada programador tiene su modo y sus manías y le gusta escribir de una forma determinada. Lo que sí existen son un conjunto de reglas generales, que aplicándolas, en mayor o menor medida, se consiguen programas bastante legibles.

Aquí intentaremos resumir estas reglas:

5.1.1.6.1 IDENTIFICADORES SIGNIFICATIVOS

Un identificador es un nombre asociado a un objeto de programa, que puede ser una variable, función, constante, tipo de datos. El nombre de cada identificador debe *identificar* lo más claramente posible al objeto que identifica (valga la redundancia). Normalmente los identificadores deben empezar por una letra, no pueden contener espacios (ni símbolos raros) y suelen tener una longitud máxima que puede variar, pero que no debería superar los 10-20 caracteres para evitar lecturas muy pesadas.

Un identificador debe indicar lo más breve y claramente posible el objeto al que referencia. Por ejemplo, si un objeto es un campo de texto en



punto net que va a almacenar el precio de una receta, la variable se puede llamar txt_precio_receta.

No se debe poner acentos, los cuales pueden dar problemas de compatibilidad en algunos sistemas. El carácter '_' es muy usado para separar palabras en los identificadores.

Es muy normal usar variables como i, j o k para nombres de índices de bucles (for, while...), lo cual es aceptable siempre que la variable sirva sólo para el bucle y no tenga un significado especial. En determinados casos, dentro de una función o programa pequeño, se pueden usar este tipo de variables, si no crean problemas de comprensión, pero esto no es muy recomendable.

Para los identificadores de función se suelen usar las formas de los verbos en infinitivo, seguido de algún sustantivo, para indicar claramente lo que hace. Por ejemplo, una función podría llamarse grabar_Receta, y sería más comprensible que si le hubiéramos llamado Escribir o EscrRec. Si la función devuelve un valor, su nombre debe hacer referencia a este valor, para que sea más expresivo usar la función en algunas expresiones, como:

```
Precio_Total = Precio_Total + IVA(Precio_Total,16) + Gastos_Transporte(Destino);
```



5.1.1.6.2 COMENTARIOS

El uso de comentarios en un programa escrito en un lenguaje de alto nivel es una de las ventajas más importantes con respecto a los lenguajes máquina, además de otras más obvias. Los comentarios sirven para aumentar la claridad de un programa, ayudan para la documentación y bien utilizados nos pueden ahorrar mucho tiempo.

No se debe abusar de *comentarista*, ya que esto puede causar una larga y tediosa lectura del programa, pero en caso de duda es mejor poner comentarios de más. Por ejemplo, es absurdo poner:

```
Nota = 10; /* Asignamos 10 a la variable Nota */
```

Los comentarios deben ser breves y evitando divagaciones. Se deben poner comentarios cuando se crean necesarios, y sobre todo:

- Al principio del programa o de cada clase del programa que permita seguir un poco la historia de cada forma, indicando: forma, objetivo, parámetros (si los tiene), condiciones de ejecución, módulos que lo componen, autor o autores, fecha de finalización, últimas modificaciones realizadas y sus fechas... y cualquier otra eventualidad que el programador quiera dejar constancia.



- En cada sentencia o bloque (bucle, if, switch...) que revista cierta complejidad, de forma que el comentario indique qué se realiza o cómo funciona.
- Al principio de cada función cuyo nombre no explique suficientemente su cometido. Se debe poner no sólo lo que hace sino la utilidad de cada parámetro, el valor que devuelve (si lo hubiera) y, si fuera oportuno, los requisitos necesarios para que dicha función opere correctamente.

No olvidemos que los comentarios son textos literarios, por lo que debemos cuidar el estilo, acentos y signos de puntuación.

5.1.1.6.3 SANGRADO

La indentación o sangrado consiste en marginar hacia la derecha todas las sentencias de una misma función o bloque, de forma que se vea rápidamente cuales pertenecen al bloque y cuales no. Algunos estudios indican que el indentado debe hacerse con 2, 3 ó 4 espacios. Usar más espacios no aumenta la claridad y puede originar que las líneas se salgan de la pantalla, complicando su lectura.

La indentación es muy importante para que el lector/programador no pierda la estructura del programa debido a los posibles anidamientos.



En Punto net esto se ahorra fácilmente ya que lo hace el mismo lenguaje.

5.1.1.6.4 PRESENTACIÓN

Al hacer un programa debemos tener en cuenta quien o quienes van a usarlo o pueden llegar a usarlo, de forma que el intercambio de información entre dichos usuarios y el programa sea de la forma más cómoda, clara y eficaz posible.

En general, se debe suponer que el usuario no es un experto en la materia, por lo que se debe implementar un interfaz que sea fácil de usar y de aprender, intuitivo y que permita efectuar la ejecución de la forma más rápida posible.

Para ello, se suelen usar las siguientes técnicas:

- Usar argumentos en la línea de comandos para especificar las distintas opciones posibles.
- Avisar lo más detalladamente posible de todos los errores que se produzcan, los motivos de los mismos y cómo solucionarlos.
- Incluir una ayuda en línea (*on line*) en el programa, de forma que el usuario pueda consultar, en cualquier parte del programa, cómo ejecutar cada una de las acciones posibles en cada



momento. Un estándar es usar la tecla de función F1 para solicitar esta ayuda. Es muy usual incluir el argumento ? (o con una barra o guion delante) para solicitar que el programa nos muestre una ayuda sobre su funcionamiento, opciones, ficheros usados... Si el programa requiere algunos argumentos obligatorios, se puede mostrar esa ayuda si falla alguno de esos argumentos.

- Usar teclas estándar: ESC para salir o deshacer las modificaciones hechas, RePág y AvPág para retroceder y avanzar una página, Supr (o Del) para borrar, las flechas para seleccionar o cambiar de posición, el tabulador.
- Usar las teclas de función (F1, F2...) para las operaciones más usuales y combinaciones de teclas para acceder a otras operaciones de forma rápida (como Alt-P, Ctrl-F1...).
- Cualquier programa debe ir siempre acompañado de un manual de usuario, en el que se explique detalladamente todo lo referente a la ejecución del programa: Requisitos mínimos de ejecución (de procesador, memoria RAM necesaria, espacio en disco que ocupa...), parámetros disponibles, ficheros que usa, mensajes de error, soluciones a los errores, variables de entorno (si las usa), ejemplos (si procede).



- Avisar de los cambios importantes y confirmar la acción si es necesario: Cuando el usuario elige alguna acción, si ésta es delicada o no admite vuelta atrás es necesario avisarle al usuario de lo que se va a hacer y exigirle que confirme tal acción. Por ejemplo, al borrar un registro o alterar datos importantes.

5.1.2 COMPATIBILIDAD DE RESOLUCIONES

En ciertas ocasiones hay una gran incomodidad al tener que desplazar la ventana de nuestro software hacia la derecha.

Para evitar esto es necesario adaptar nuestras pantallas para cualquier monitor por medio de tablas que ocupen el 100% de la pantalla o tomar en cuenta la resolución promedio que utilizarán los usuarios.

Actualmente la mayoría de monitores tienen una resolución de 800X600, por lo que en si trabajamos en este formato, monitores más grandes podrán verla sin problema. Hay que tomar en cuenta que aún existen computadoras con resoluciones de 640 X 480, pero ya es una minoría menor al 5% de los usuarios

5.1.3 FACILIDAD DE NAVEGACIÓN

Todo sitio debe contar con un menú de navegación o estructura bien elaborado para que el usuario no se pierda.



Es recomendable agrupar los módulos con los procesos maestros que se utilizan en el mismo para que se desarrolle en un solo sitio todas las opciones.

5.1.4 ADMINISTRACIÓN DE RECURSOS

Para un buen desempeño del sistema hemos visto la necesidad de chequear cada uno de los recursos que afectan directa e indirectamente a nuestro sistema.

Primero hemos dividido a nuestro sistema en un pequeño número de componentes. He aquí uno de los puntos en los cuales los sistemas tienen conflictos y muchas veces la integridad de los datos no es la más exacta.

Cada uno de los componentes principales de un sistema le hemos llamado subsistema. Cada subsistema abarca aspectos del sistema que comparten alguna propiedad común.

Para nuestro caso hemos revisado los recursos que se cuenta en el hotel y son (ver figura 62):

-  Reglas de negocios
-  Hardware
-  Base de datos
-  Recursos de Red

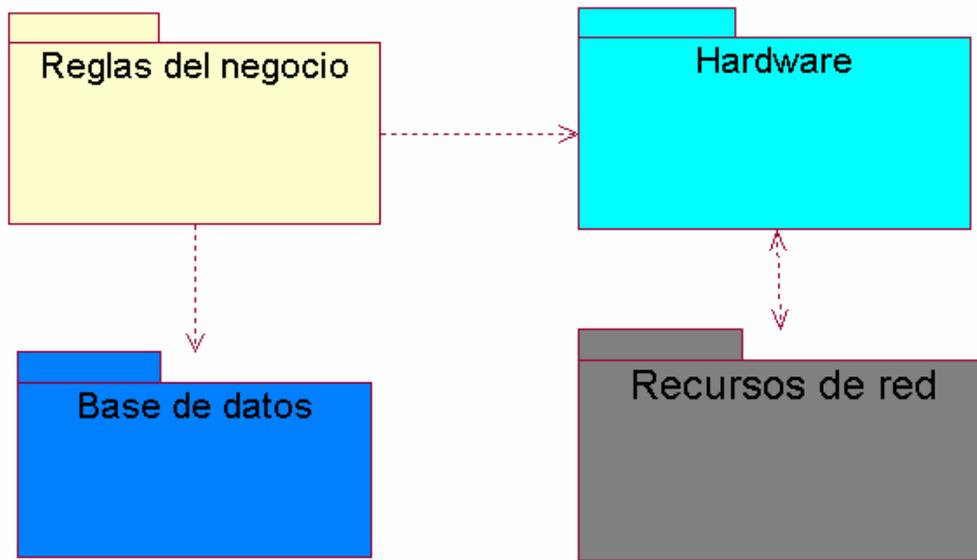


Figura No.62 (Recurso Reglas del Negocio)

5.1.4.1 RECURSO REGLAS DEL NEGOCIO

Dentro de las reglas del negocio están identificados los módulos que hemos desarrollado



5.1.4.1.1 MODULOS QUE EXISTEN EN EL SISTEMA

Referente a nuestro proyecto, hemos subdividido al sistema en los siguientes módulos, lo podemos visualizar en la figura No.63 de aconticuación:



Figura No.63 (Módulos Existentes en el Sistema)

5.1.4.1.1.1 MÓDULO BODEGA

Es una ruta por la cual el personal del hotel mantiene lo necesario para poder trabajar y cumplir son sus responsabilidades.

Es el único método dentro del hotel por el cual se puede obtener algo de Bodega.

Cada pedido que realiza el personal de Bodega tendrá un número, con el cual podrá identificar sus movimientos dentro del mes y poder tener documentado su stock.

Las etapas que se han subdivididos en este proceso son:

-  Solicitud de Pedido
-  Aprobación del Pedido
-  Entrega del Pedido



5.1.4.1.1.1 SOLICITUD DE PEDIDO

Cuando un empleado necesita suministros o alimentos (en el caso del personal de cocina), éste deberá realizar la solicitud de pedido a Bodega General.

Todo el personal deberá realizar el pedido a Bodega para poder sacar algo de la misma. Inclusive la gerencia general, ya que al momento de realizarlo podemos tener el histórico de algún producto en particular.

Referente al departamento de Cocina, es indispensable hacer este proceso, ya que permite a la gerencia de AYB y Chef Principal, revisar el movimiento de un producto para preparación de recetas y poder determina si hay cambios en marca y cantidad. Referencia figura No.64.

SOLICITUD DE PEDIDO

Nº 7
Fecha 15/06/2006

Bodega ALIMENTOS
Area COLEGIO EXPOSICION Por JOSUUE

Cantidad de Productos: 1

| FACTURA | PROD | ARTICULO | CANTIDAD |
|---------|---------|-----------|----------|
| 7 | PROD005 | NORTEÑO G | 0 |

Código PROD005 Cantidad 2 UNIDAD BOT
Artículo NORTEÑO GRANDE STOCK 9

Figura No.64 (Solicitud de Pedido)



En este proceso solo se podrá pedir los productos que están relacionados a la bodega escogida.

Cada ítem de los productos está relacionada a las distintas bodegas creadas en el sistema.

Es importante realizar los pedidos con anticipación, ya que, si en el departamento de Bodega no tienen lo solicitado, deberán realizar una solicitud de compra y, en frecuentes ocasiones –especialmente en productos importados- los proveedores no cumplen en el día de entrega. Así mismo el departamento de Bodega no trabaja fines de semana, sin embargo el hotel atiende en todos los restaurantes y habitaciones, por lo cual, si no se tiene el stock necesario (*ver en la figura No.64*) en cocina para el promedio de ventas no se podrá sacar productos de la Bodega general y poder preparar los alimentos.

5.1.4.1.1.2 APROBACIÓN DEL PEDIDO

Todas las solicitudes realizadas tendrán que ser aprobadas por el jefe departamental, no podrá ser entregada ninguna solicitud si no fue aprobada anteriormente, simplemente no saldrá como pedido para entregar.

No es necesario aprobar la solicitud de pedido en el mismo día, se puede realizar en cualquier momento después de haber sido realizada la misma.

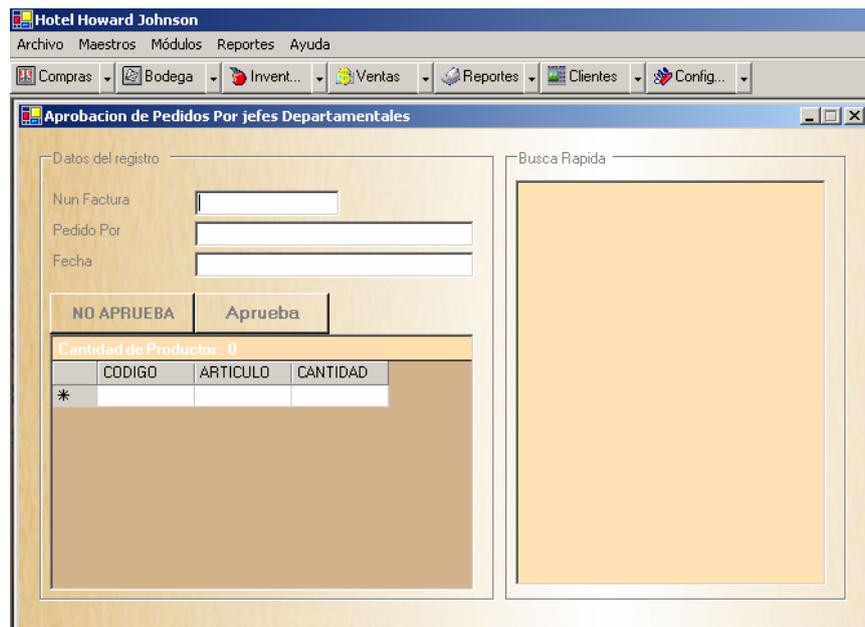


Figura No.65 (Aprobación del Pedido)

En el momento de la aprobación, solo se procede a aprobar o no, de ninguna manera se podrá realizar cambios en los productos pedidos y en la cantidad, ya que pasaría de ser proceso de aprobación a solicitud de pedido. (Ver Figura No.64).

5.1.4.1.1.3 ENTREGA DE PEDIDO

Una vez realizado el pedido y aprobado, el responsable de la Bodega General tendrá el listado de los productos pedidos y podrá despachar a las áreas respectivas.

Cabe recalcar que el departamento de bodega no podrá visualizar los pedidos que están pendientes de aprobación, por lo cual, no podrá despachar ningún pedido que no esté aprobado.

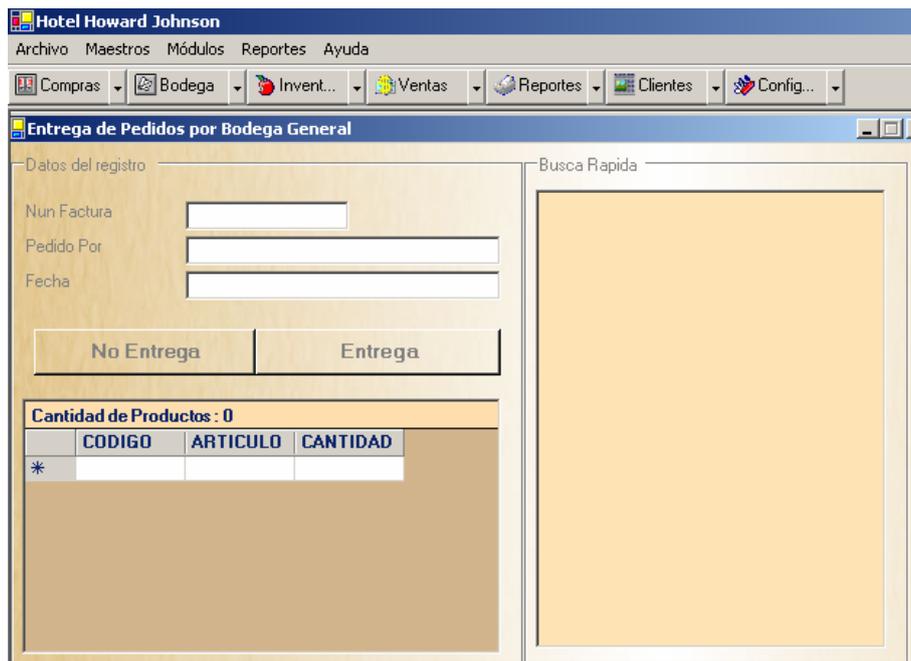


Figura No.66 (Entrega del Pedido)

Al entregar los productos se modifica el stock de los mismos, y la solicitud de pedido cambia de estado: de pedido a entregado. (Ver Figura No.66).

5.1.4.1.1.2 MÓDULO DE COMPRAS

Es un conjunto de etapas por las que pasa el consumidor para adquirir algo. Nosotros hemos tomado como importante no solo lo que vendemos, también lo que compramos. Por esto, se ha atacado a las compras para los productos, tanto de oficina como de alimentos y bebidas. Se ha tomado en cuenta, una cartera de proveedores existente en el hotel, para subir a nuestra base de datos.

Aquí cada producto está relacionado con un proveedor. Pero esto no significa que no pueda cambiar de proveedor. Hemos creado una pantalla en donde



podemos asignar el producto a cualquier proveedor. Esto debemos tomar en cuenta para realizar las compras (Ver figura 67).

| codigo_prod | descripcion | stock |
|-------------|-------------------------|-------|
| PROD-48 | ACEITE EL COCINERO 1 LI | 3 |
| PROD-47 | ACEITE LA FAVORITA 1 LI | 0 |
| PROD-51 | ACEITE LA FAVORITA 1/2 | 21 |
| PROD-192 | ACEITE LA FAVORITA CAJ | 3 |
| PROD-49 | ACEITE LA FAVORITA LIT | 2 |
| PROD-128 | ACHIOTE EN PEPA | 43 |
| PROD-55 | ACHIOTE LA FAVORITA | 2 |
| PROD-133 | ACHIOTE SIMPLE | 44 |
| PROD-38 | AGUA FONTANA 1/2 LITRO | 9 |
| PROD-37 | AGUA FONTANA 1/2 LITRO | 5 |
| PROD-10 | AJAX CLORO | 8 |
| PROD-74 | AJAX CLORO BOTELLA | 2 |
| PROD-119 | AJI | 5 |
| PROD-138 | AJINOMOTO | 1528 |
| PROD-134 | ALIÑOS FUNDA | 28 |
| PROD-131 | ANIS | 17 |
| PROD-83 | ARROZ MIL UNO FAMILIAR | 3 |
| PROD-196 | ARROZ MIL UNO MONTUBI | 99 |
| PROD-125 | ARROZ MONTUBIO | 9,5 |
| PROD-78 | ATUN REAL ABRE FACIL | 33 |
| PROD-200 | ATUN VAN CAMPS ABRE F | 1 |
| PROD-79 | ATUN VAN CAMPS | 5 |
| PROD-126 | AZUCAR ECUDOS | 45 |
| PROD-194 | AZUCAR ECUDOS QUINTA | 8,5 |
| PROD-85 | CAFÉ MINERVA 100 GRAM | 10 |

Figura No.67 (Pantalla proveedor x producto)

Las etapas en que hemos subdividido a este proceso son:

-  Solicitud de compra
-  Aprobación de Compra
-  Recepción de la mercadería

5.1.4.1.1.2.1 SOLICITUD COMPRA

La solicitud de compra se puede realizar en 2 subprocesos.

1. La primera es cuando hacemos un pedido a la bodega general y ésta no tiene en stock los productos necesarios para el departamento. Por consiguiente, el departamento de Bodega se encarga de realizar una



compra de todos los productos necesarios, cumpliendo el mínimo y el promedio del mes sobre cada producto.

Ahora este proceso es manual, el responsable de la Bodega debe validar cada 2 días los productos necesarios, de ahí, escoger por proveedor y realizar el pedido de compra, para la aprobación respectiva.

2. El segundo es cuando se necesita realizar una compra directamente, como por ejemplo, la compra de toallas, vajilla, etc, y directamente el departamento de Compras hace los trámites necesarios con el proveedor.

Aquí el departamento de compras realiza la solicitud de compra y espera a la aprobación únicamente de la gerencia y no de jefes departamentales. (ver figura No.68)

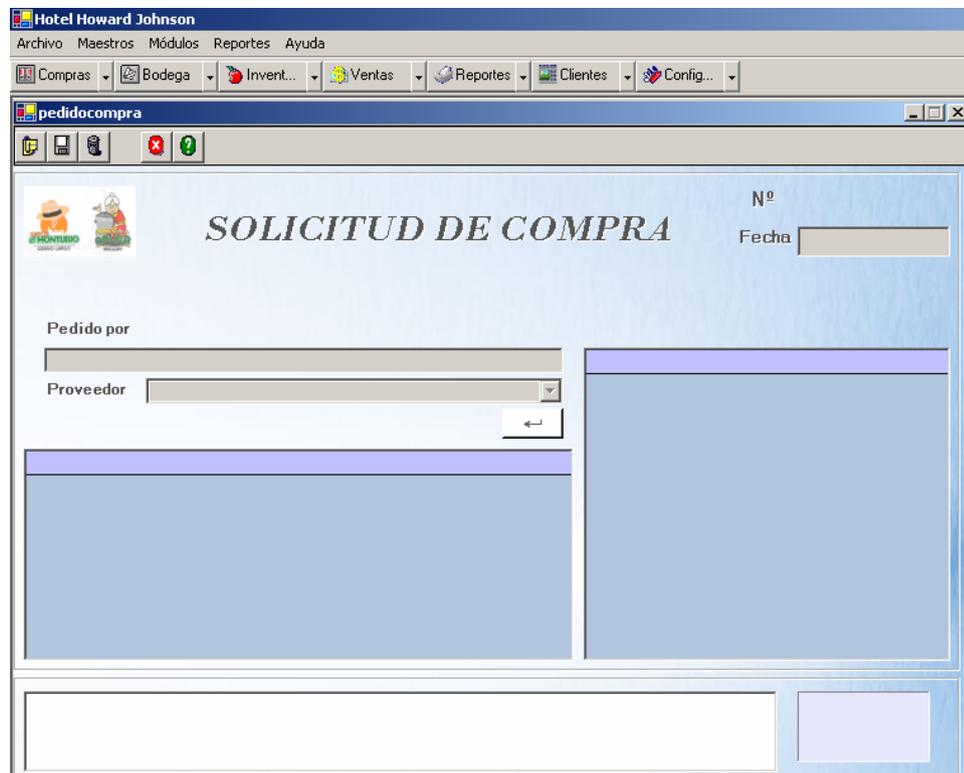


Figura No.68 (Pantalla Solicitud De Compra)



5.1.4.1.1.2.2 APROBACIÓN DE COMPRA

Al igual que la aprobación de pedidos a bodega, solo se cumple un proceso de aprobar o no aprobar, no se puede modificar ni cantidad, ni productos, y menos aún el precio de la factura.

Hay que indicar que, si el gerente, en este caso no está de acuerdo con el total de la factura, simplemente no aprueba y tendrá que pedir al departamento de Compras que vuelva a realizar una solicitud de compra. (Ver Figura No.69)

| ARTICULO | DESCRIPCIO | CANTIDAD | PRECIO |
|-----------|------------|----------|--------|
| ▶ PROD007 | TROPICO | 5 | 1,0000 |
| PROD006 | TROPICO PE | 4 | 0,5000 |
| PROD005 | NORTEÑO G | 3 | 1,0000 |
| * | | | |

Figura No.69 (Pantalla Aprobación de Compra)

5.1.4.1.1.2.3 RECEPCIÓN DE LA MERCADERÍA

Ya realizada la aprobación respectiva por parte del gerente general para las compras pendientes, el responsable del departamento tendrá que ponerse de acuerdo con el proveedor y coordinar la entrega de los productos.



La recepción de la mercadería se realiza de manera completa y no parcial, o al menos en el sistema se debe realizar este proceso de la forma mencionada.

Al recibir la mercadería se procede a modificar el stock y el costo promedio de los productos.

En la recepción de la mercadería, solo se limita a recibir más no a cambiar productos ni precios. (Ver figura No.70)

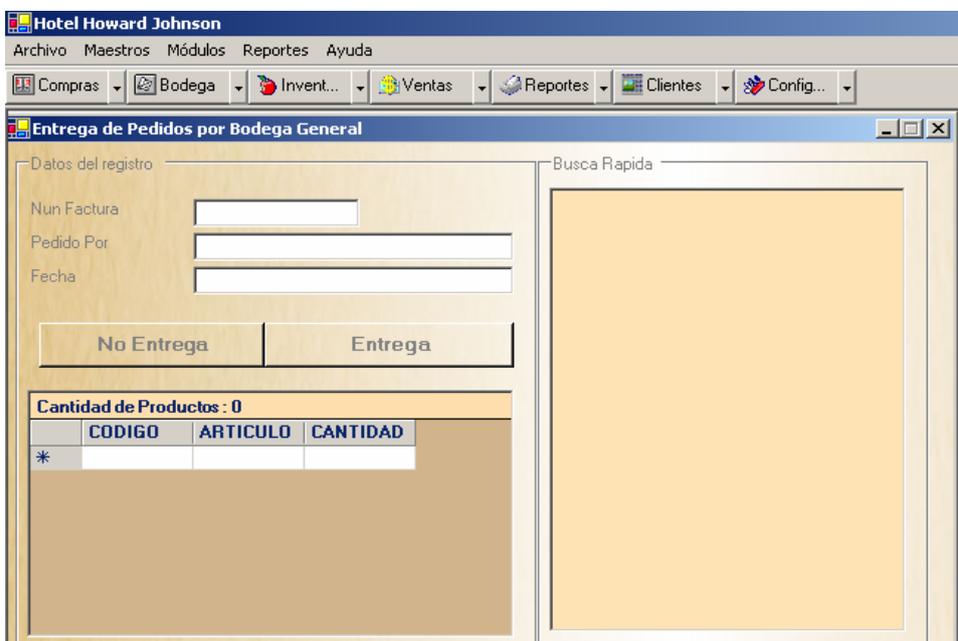


Figura No.70 (Pantalla Entrega de Compra)

5.1.4.1.1.3 MÓDULO INVENTARIO

Este proceso contiene la relación pormenorizada de cada uno de los productos del hotel, en la cual se va detallado el nombre y código, características, costo, costo promedio, etc.



La base de todo hotel, aparte de las habitaciones, es el servicio de sus restaurantes, la compra y venta de alimentos; de aquí la importancia del manejo del inventario por parte del mismo.

Este manejo permite a la empresa mantener el control oportunamente, así como también, conocer al final de un periodo el stock y valor de los productos.

Para nuestro caso, en lo que respecta al sistema de control del inventario, utilizaremos en forma mezclada los modelos de sistema perpetuo y periódico.

- ✚ El sistema Perpetuo mantiene un registro continuo para cada artículo del inventario. Los registros muestran, por lo tanto, el inventario disponible todo el tiempo.

El sistema perpetuo ofrece un alto grado de control, porque los registros de inventario están siempre actualizados. Anteriormente, los negocios utilizaban el sistema perpetuo, principalmente para los inventarios de alto costo unitario, como las joyas y los automóviles; hoy día con este método, los administradores pueden tomar mejores decisiones acerca de las cantidades a comprar, los precios a pagar por el inventario, la fijación de precios al cliente y los términos de venta a ofrecer.

Para determinar el saldo final en nuestro sistema utilizamos la siguiente fórmula:

Saldo Final =

Saldo Inicial + Incrementos (Compras) - Disminuciones (ventas de recetas)



- ✚ El sistema Periódico lo vamos a utilizar exclusivamente para el fin de mes. El hotel hace un conteo físico del inventario disponible y aplica los costos unitarios para determinar el costo del inventario final.

Esta cifra de inventario, que aparece en este conteo, debe coincidir con el listado que nos ofrece el sistema; esta comparación se llamará faltantes y sobrantes.

Una vez que ingresemos al sistema el conteo final de los productos, éste producirá el reporte de faltantes y sobrantes. Con lo cual el, responsable de la Bodega General, tendrá que justificarlo.

Aquí nos valdremos de todas las solicitudes de pedidos, solicitudes de compras y sus respectivas aprobaciones.

Los productos de bajo costo pueden no ser lo suficientemente valiosos para garantizar el costo de llevar un registro al día del inventario disponible. Por lo cual, el Bodeguero tendrá la capacidad de parametrizar la clase de cada producto para un conteo diario – semanal – quincenal – mensual – o como crea prudente; por ejemplo esto sucede con productos como la sal, pimienta, cominos, etc.

Para usar el sistema periódico con efectividad, el bodeguero debe tener la capacidad de controlar el inventario mediante la inspección visual.



Para estos dos procesos tenemos en el sistema el detalle rápido de un producto. (kardex, ver en la figuraNo.71), donde podemos ver en grid el detalle de las compras, pedidos a bodega y las ventas que involucra este producto.

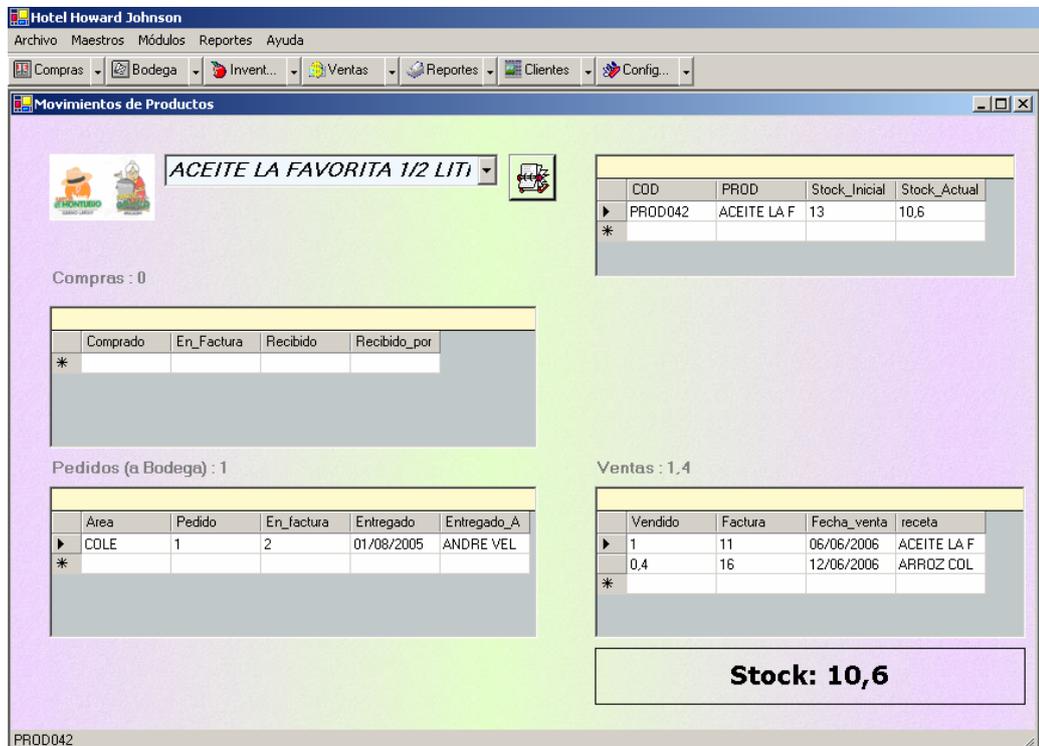


Figura No.71 (Pantalla Stock x producto)

Adicional a este proceso, hemos incluido dentro del mismo proceso, la preparación de recetas y subrecetas, preparación de las subrecetas involucra hasta 10 items.

Aquí se debe especificar la unidad de preparación de toda la subreceta, así como de los productos que conforman la subreceta con la respectiva cantidad.

La subreceta es un conjunto de productos que se conforman para utilizarlos en distintas recetas con frecuencia. Esta en ningún momento tiene precio final de venta. (ver figura No.72)

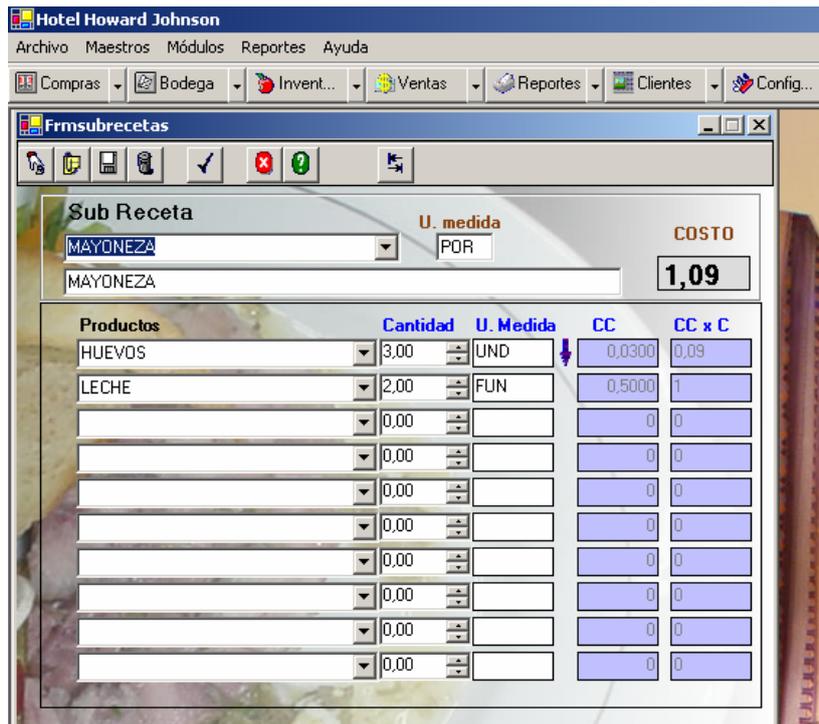


Figura No. 72 (Pantalla Subrecetas)

Maneja un tipo de medida y el costo que da el sistema de acuerdo al costo de compra que tiene cada producto multiplicado por la cantidad que se utiliza. La subreceta solo está conformada por productos y de acuerdo al producto hará el cálculo de acuerdo al costo de compra que tiene cada una de ellos, este costo de compra es ingresado en el sistema, y no es el costo promedio sino el que da el proveedor. (Ver Figura No73)



Figura No. 73 (Pantalla Productos)



La Receta en cambio es el producto de venta al cliente; es el plato de comida que se vende y el cual está en la carta.

La receta está conformado por subrecetas o productos, en conjuntos de 5 items, con costo real y precio final. Adicional a esto, está la ruta de la imagen de la receta (ver figura 74).

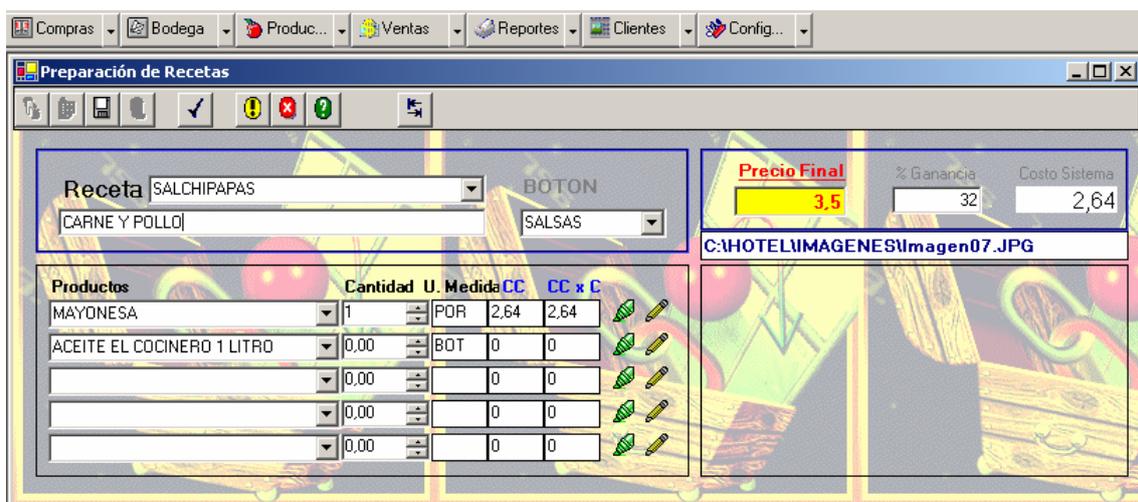


Figura No. 74 (Pantalla Recetas)

En la preparación de la receta se maneja un campo adicional que es el botón, esto significa el lugar donde aparecerá la receta para ser vendido. Así mismo el precio final es ingresado por el usuario, el sistema calcula el costo actual y el porcentaje de ganancia.

Hay que tener el suficiente conocimiento para la creación de las recetas, es por eso que se recomienda que este módulo lo realice directamente el Chef Principal ya que involucra un sinnúmero de productos en el cual se maneja stock.



El sistema proveerá una gran cantidad de informes destinados a controlar la gestión de stock a nivel integrado y contable, más todos aquellos referidos a los movimientos estadísticos de productos.

En el mismo módulo de inventarios hemos involucrado el conteo de los productos. Para esto se desarrolla un conteo manual que lo debe hacer el encargado de la bodega y compararlo con el que da el sistema.

Aquí debemos tener en claro la clasificación que recomienda el sistema para el conteo de productos.

El usuario puede crear el tipo de conteo para relacionarlo al producto, debe tener el intervalo de días para poder hacer el conteo en las fechas respectivas. Aquí podemos ver los distintos tipos de conteo con sus respectivos intervalos (Ver figura No 75)

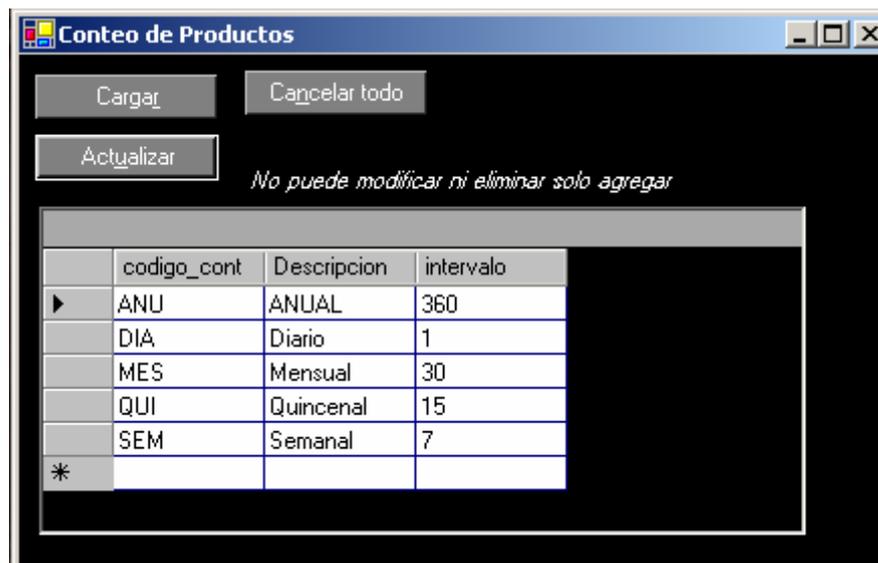


Figura No.75 (Pantalla de conteo de productos)



Entonces para realizar el conteo debemos hacer el siguiente proceso:

- 1) Ejecutar la creación de las tablas respectivas para el tipo de conteo. En este proceso se selecciona todos los productos de acuerdo al conteo, se crea dinámicamente una tabla y se actualiza la fecha de creación de la misma. (ver figura No.76)

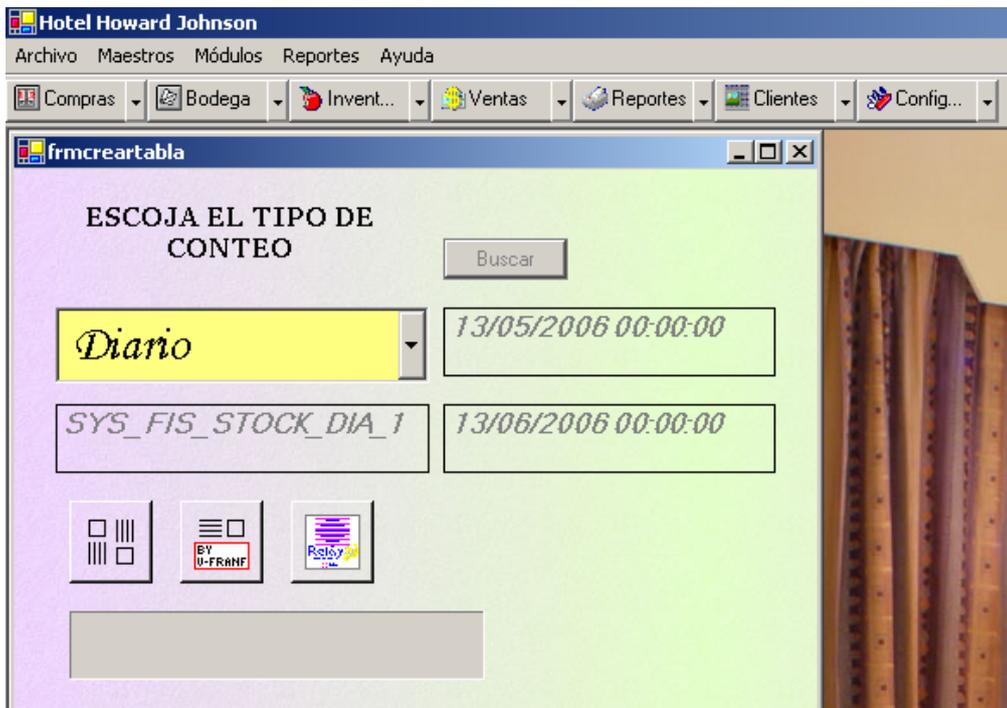


Figura No.76 (Pantalla de creación de tablas)

- 2) Luego podemos generar la pantalla de conteo físico en donde el responsable de la bodega deberá ingresar el stock físico que encontró, éste no debe saber cuanto existe en el sistema, solo debe ingresar acorde al tipo de conteo predeterminado.

En esta pantalla podemos encontrar también la fecha de la creación de la tabla, que vendría a ser la fecha del conteo. (ver figura No.77)

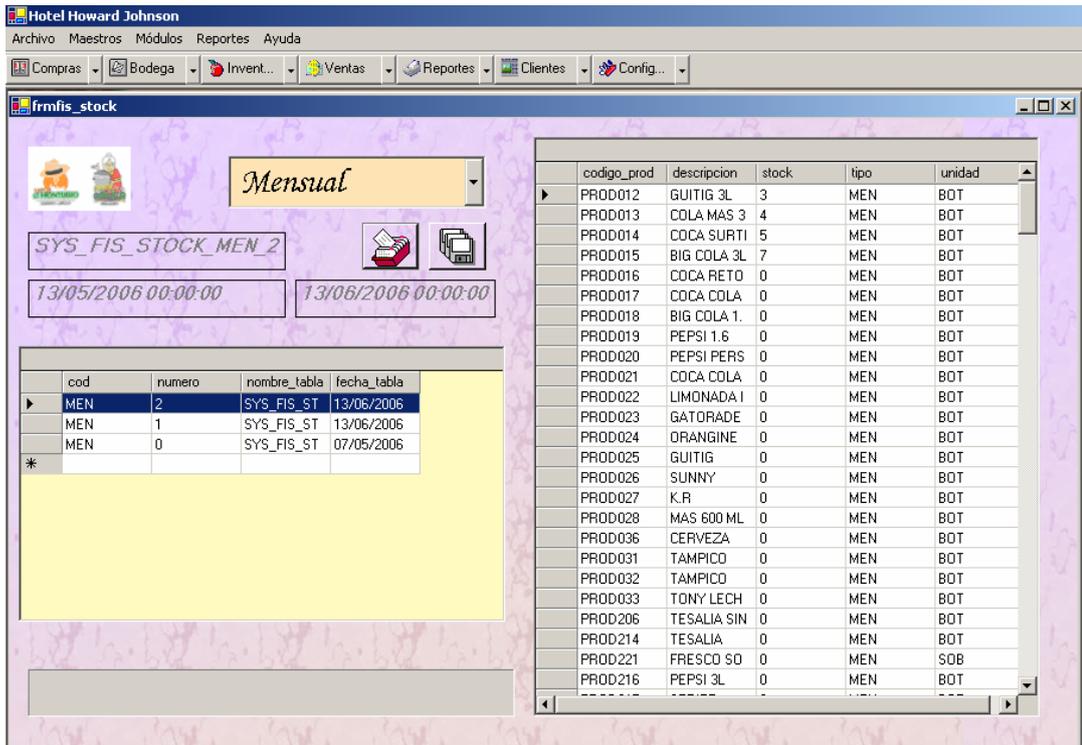


Figura No. 77 (Pantalla conteo físico)

- 3) Una vez ya ingresado el stock podemos generar la pantalla de faltantes y sobrantes. El sistema revisa el stock actual de cada producto del tipo de conteo y lo compara con el ingresado por el usuario. (ver figura No.78)



| codigo_prod | descripcion | Sistema | Fisico |
|-------------|-------------|---------|--------|
| PROD001 | GRAN DUVA | 18 | 0 |
| PROD002 | CLAN MAG | 1 | 0 |
| PROD003 | RON ESTEL | 1 | 0 |
| PROD004 | BARDAY | 1 | 0 |
| PROD005 | NORTEÑO G | 9 | 0 |
| PROD006 | TROPICO PE | 15 | 0 |
| PROD007 | TROPICO | 5 | 0 |
| PROD008 | EL JAGUAR | 1 | 0 |
| PROD009 | SHUMIR PE | 6 | 0 |
| PROD010 | TROPICO SE | 2 | 0 |
| PROD011 | VINO ALEGR | 3 | 0 |
| PROD012 | GUITIG 3L | 1 | 3 |
| PROD014 | COCA SURTI | 7 | 5 |
| PROD015 | BIG COLA 3L | 9 | 7 |
| PROD016 | COCA RETO | 2 | 0 |
| PROD017 | COCA COLA | 2 | 0 |
| PROD019 | PEPSI 1.6 | 7 | 0 |
| PROD020 | PEPSI PERS | 12 | 0 |
| PROD021 | COCA COLA | 17 | 0 |
| PROD022 | LIMONADA I | 11 | 0 |
| PROD023 | GATORADE | 3 | 0 |

Figura No.78 (Pantalla Sobrantes y faltantes)

Esta pantalla debe estar sin ningún producto. En caso de presentar un o varios ítems debe el responsable de justificar esas diferencias por medio de compras, pedidos o ventas.

5.1.4.1.1.4 MODULO DE POS

5.1.4.1.1.4.1 PROCESO POS

El módulo de POS⁵ se refiere al proceso de venta en los restaurantes Tradiciones y Sport Bar.

⁵ (Point of Sales)



Aquí se recoge todas las recetas preparadas ya asignadas a un botón específico para la venta. Son tal como están en la carta.

En el módulo de inventarios revisamos que cada receta debe estar asociada a un botón. Este botón se visualiza en el grupo de esta pantalla. Es totalmente dinámico. (ver figura No.79)

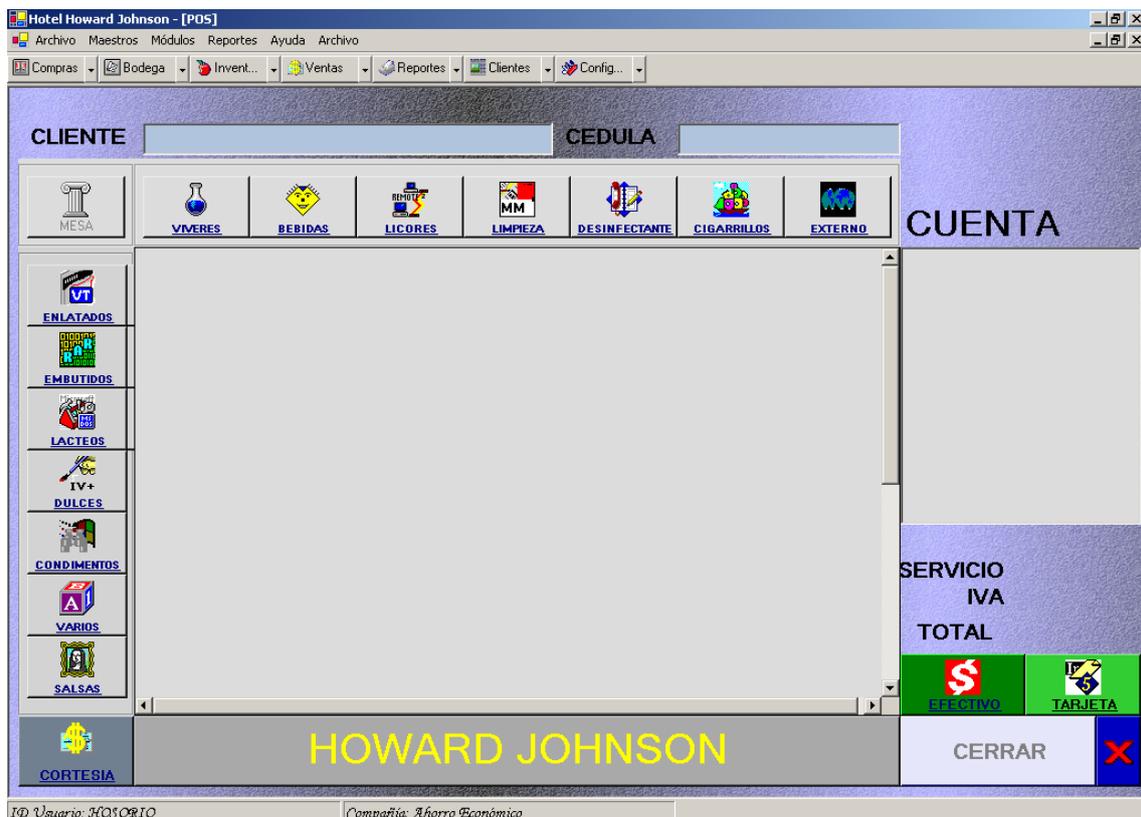


Figura No.79 (Pantalla POS)

El proceso de venta es por medio de un salonerero que recibe el pedido del cliente en la mesa, éste es enviado a la cocina para su preparación y servicio al cliente. Al momento del pago el salonerero se dirige al cajero y le entrega las comandas que fueron dadas a cocina para su respectiva facturación.



Aquí el cajero ingresará todos los productos vendidos y podrá escoger la forma de pago entre efectivo o tarjeta (ver figura 80). Existe para mayor facilidad del cajero la opción de dar vuelto. De igual manera se puede pagar como cortesía una cuenta que haya sido aprobada por la gerencia de AyB.

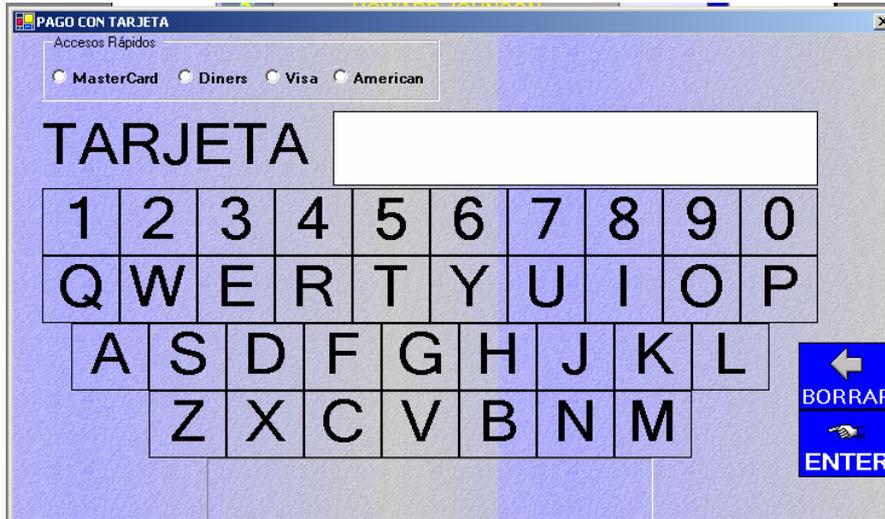


Figura No.80 (Pantalla Pago Tarjeta)

El precio total de la factura esta incluido el porcentaje de Servicio e IVA, esto de acuerdo a las políticas de hotelería y gubernamental. (Ver figura No.81)

| | |
|----------|-------|
| | 21,00 |
| SERVICIO | 2,10 |
| IVA | 2,52 |
| TOTAL | 25,62 |

Figura No. 81 (Cálculo de la Factura)

Cabe recalcar que al momento de realizar la cuenta y cerrarla se imprimirá la factura en una impresora Epson TM 200 con totales y productos consumidos, en cambio en la cocina principal se imprimirá solamente la comanda con los productos pedidos, más no detalles de pago y cliente. Esto le permitirá al fin del



turno al cocinero poder cuadrar los platos preparados y vendidos con las facturas que tenga el cajero.

El cajero al momento de cerrar su caja imprimirá el reporte de cuentas cerradas para poder comparar con el dinero en efectivo y en vouchers. (ver figura No.82)

| FECHA 26-Abril-2006 | | | |
|----------------------------|-------------|----------------------|----------------|
| CEDULA | MESA | NOMBRE | FACTURA |
| | 11 | 1 HUGO | 12 |
| 1 | | GALLETAS RICAS | 0,2 |
| 1 | | GALLETAS RICAS | 0,2 |
| | | <i>SUBTOTAL</i> | 0,40 |
| | | <i>IVA</i> | 0,05 |
| | | <i>SERVICIO</i> | 0,05 |
| | | <i>TOTAL</i> | 0,5 |
| | | <i>FORMA DE PAGO</i> | EFFECTIVO |

Figura No. 82 (Reporte de Factura)

Cuando un plato sea devuelto por el salonerero a pedido de un cliente éste tendrá primero que autorizarlo el chef y luego reportarlo al cajero para realizar el mismo proceso de venta el producto pero como devolución para no tener descuadres de stock, ya que un plato devuelto no significa que se podrá devolver los productos que lo conforman, por el contrario es una venta en cero ya que los productos si serán afectados en stock.

Los datos de los productos vendidos se almacenarán en las distintas tablas que conforman este módulo, sin embargo no se afectará el stock en línea ni tendrá efecto de no venta al no tener en stock los productos que lo conforman en el sistema.



El por qué de esto se detalla a continuación. El módulo POS se utiliza independiente de todo el sistema, este podrá ser instalado en un punto en los restaurantes y realizará las transacciones de ventas. Ya que en ciertas ocasiones cuando se necesita comprar productos para la preparación de tales recetas y sucede que es fin de semana o un momento en que la bodega ésta cerrada, se procede a sacar los productos directamente de la bodega realizando una solicitud manual o comprar y proceder a vender. Luego al momento que esté el responsable de la Bodega general o del departamento de Compras, éste ingresará las solictudes y modificará el stock de los productos sumando a las ventas realizadas en el módulo POS.

No podemos impedir la venta en los restaurantes por no estar el encargado de compras o bodega.

Lo importante es que al momento de realizar el proceso de faltantes y sobrantes cuadre las ventas, compras y por supuesto el stock final de cada uno de los productos.

5.1.4.1.1.4.2 ACTUALIZACIÓN STOCK

Este módulo es una pantalla donde se procede a buscar todas las facturas que se han producido y aún no se han actualizado su stock.

Lo que hace es buscar las recetas de la factura, una vez que tiene la receta por ende busca las subrecetas y los productos que lo conforman. Aquí si es



producto manda a descargar directamente el stock. Si es subreceta manda a buscar los productos de la subreceta y descarga del stock también.

Una vez realizado este proceso se cambia de estado la factura para no volver a actualizar un stock de un producto ya afectado.

Este proceso realiza varias búsquedas y actualizaciones por lo que se hace dependiendo de la venta una carga de tareas para el sistema considerables. Sin embargo se lo puede realizar en cualquier momento. Lo ideal es hacerlo a diario. (ver figura No.83)

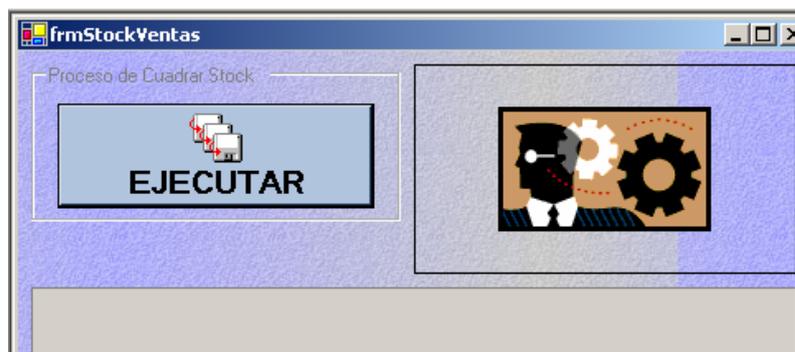


Figura No. 83 (Actualización de stock)

5.1.4.1.1.5 MODULO DE RESERVA DE SALONES

El proceso de reserva de salones tiene que ver específicamente sobre la presentación vía web de los distintos montajes que se pueden llevar a cabo en el hotel.

Tendremos que describir la ubicación de cada salón, los tipos de montajes que se pueden agregar, las reservaciones de los mismos.



También ofrece la posibilidad de ver los distintos menús y ofrecer el precio final por x cantidad de pax⁶ que deseen. (ver figuras No. 84, 85, 86).



Figura No. 84 (Pantalla inicial Web)

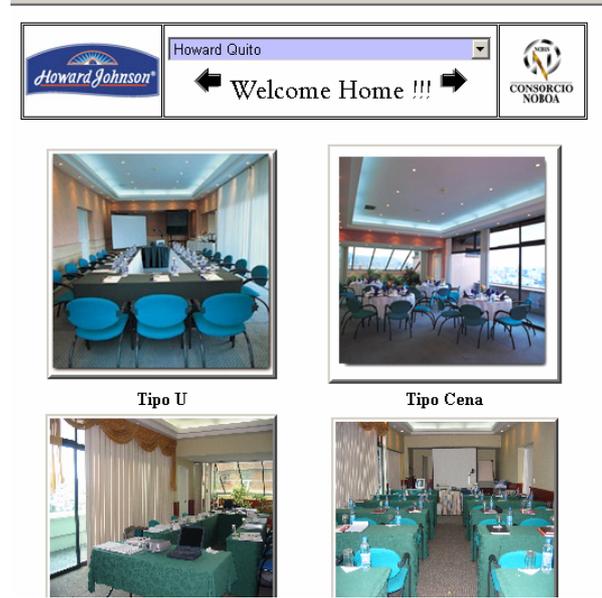


Figura No. 85 (Pantalla Montaje de salones)

| | | | |
|--|---|---|--------------------------------------|
| | | Howard Quito | |
| | | ← | → |
| Bienvenidos a Nuestro Sistema de Reservas | | | |
| Datos del Huésped | | | |
| Nombres | | Apellidos | |
| <input type="text" value="Hugo"/> | | <input type="text" value="Osorio"/> | |
| Ciudad | <input type="text" value="Quito"/> | País | <input type="text" value="Ecuador"/> |
| C.I. / R.U.C. | <input type="text" value="1308808490"/> | Teléfono | |
| Email | | <input type="text" value="hosorio@qtgc.com"/> | |
| Datos de la Estadía | | | |
| Hotel: Howard Quito | | | |
| Huésped | | | |
| <input type="text" value="Hugo Osorio"/> | | | |
| Pax | <input type="text" value="46"/> | | |
| Horario | | | |
| <input type="text"/> | | | |

Figura No. 86 (Pantalla Ingreso de Datos)

⁶ Número de personas que reservan el salón.



Esta página Web estará subida a un servidor Web apuntando a la base de datos para realizar las distintas reservas.

Si bien la reserva puede grabarse no significa que ya se reservó el salón escogido. Entra un proceso del responsable de AYB⁷ donde tendrá que chequear y llamar a asegurarse que realmente se desea el salón para tal fecha y para tal número de personas. (Ver figura No.87).

The screenshot shows a web-based reservation form titled 'frmevento' within the 'Hotel Howard Johnson' application. The interface includes a menu bar with options like 'Compras', 'Bodega', 'Invent...', 'Ventas', 'Reportes', 'Clientes', and 'Config...'. The form itself has several input fields for personal information and event details. A tabbed interface is used for different sections of the reservation, with 'Cuándo' (When) currently selected. The 'Cuándo' tab contains fields for the number of guests ('Pax'), start and end dates ('Fecha Inicio', 'Fecha Final'), time ('Horario'), email, city ('Ciudad'), country ('Pais'), phone number ('Teléfono'), and identification number ('C.I.'). There are also icons for a telephone, a printer, and a 'UR' logo on the right side of the form.

Figura No.87 (Pantalla reserva eventos)

Incluso tiene un campo donde puede ingresar todo el acuerdo que llegaron.

Para confirmar una reserva el responsable de AYB tendrá a la mano un reporte en donde pueda verificar si el salón está siendo usado para esa mismas fecha y hora.

⁷ Alimentos y Bebidas



Se visualizará un seguimiento a la reserva para poder realizar el cronograma de eventos, esto de ninguna manera exigirá un seguimiento como sistema CRM pero si ayudará a mantener una base de clientes por eventos, fechas y menús. (Ver figura No.88)

Howard Johnson
HOTEL LA CAROLINA

Lunes, Mayo 1, 2006

5 - May **6 - May** **9 - May**

Tabla cruzada

| Total | 5 - May | 6 - May | 9 - May |
|--------------|---------|---------|---------|
| 12:00 | | | |
| 1 | 1 | 0 | 0 |
| 2 | 0 | 0 | 1 |
| 21:00 | | | |
| 1 | 0 | 1 | 0 |

Figura No. 88 (Reporte Eventos)

Es importante poder tener información de lo realizado ya que permite crear un estándar de logística y tipos de eventos.

5.1.4.2 RECURSO DE HARDWARE

El servidor de la base de datos y la aplicación estas ubicados en un HP Proliant ML 350 G2 de características:

- ✚ Dos procesadores Xeon de 2.4 Ghz.
- ✚ Dos Gigas de memoria
- ✚ Dos discos de 60 gb.
- ✚ Quemador de DVD.



Los discos se encuentran particionado en 20 y 40 gb y hechos espejos. Hay una partición para la aplicación y otra la base de datos.

5.1.4.3 RECURSO DE BASE DE DATOS

Hemos buscado un tipo de base de datos que sea administrada por medio de un sistema de gestión de base de datos, esta es MSDE⁸.

Esta base reserva los datos de acceso frecuente en memoria, con objeto de alcanzar la mejor combinación posible de costo y rendimiento desde y hacia la memoria y el almacenamiento en disco. Las bases de datos son potentes y hacen que las aplicaciones sean más fáciles de transportar a sistemas operativos y a distintas plataformas, en el capítulo 2 subtítulo 2.11 describimos la base de datos MSDE.

En esta fase nos vamos a centrar a explicar sobre la conexión y las tablas principales del sistema.

El software es un sistema Multicompañía en todo aspecto. Maneja distintas bases y por ende distinta información. (ver figura No.89)

⁸ MICROSOFT SQL SERVER 2000 DESKTOP ENGINE

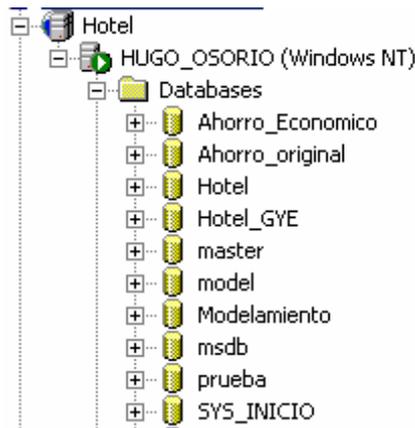


Figura No. 89 (Esquema de base de datos)

La definición de los nombres de las bases están en la tabla SYS_SYS, la cual está en la base sys_inicio.(ver figura No.90)

| | Column Name | Data Type | Length | Allow Nulls |
|----|---------------|-----------|--------|-------------|
| PK | ID | nvarchar | 8 | |
| | Servidor | nvarchar | 30 | ✓ |
| | Version | nvarchar | 15 | ✓ |
| | Autor | nvarchar | 30 | ✓ |
| | Empresa | nvarchar | 30 | ✓ |
| | Ruc | nvarchar | 13 | ✓ |
| | Direccion | nvarchar | 50 | ✓ |
| | Telefono | nvarchar | 15 | ✓ |
| | Celular | nvarchar | 15 | ✓ |
| | Representante | nvarchar | 50 | ✓ |

Figura No. 90 (Atributos de la tabla SYS_SYS)

En esta base tenemos únicamente 3 tablas, la de bases (sys_sys), la de usuarios (sys_user), y la de perfiles (sys_perfil).

Los usuarios se crean una sola vez y solo una por sistema, es decir no es necesario crear cada usuario para cada empresa.



La creación de la base es dinámica pero debe hacerlo un administrador de base de datos.

Adicional en MSDSQL tenemos los siguientes login con los cuales podemos conectar a la base de datos (ver figura No 91).

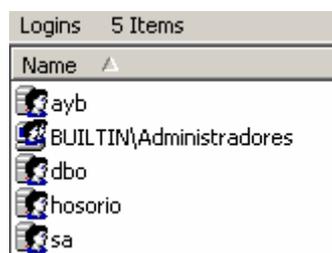


Figura No.91 (Conexiones)

Estos login nos sirven para la conexión que se realiza por medio de los ODBC



Figura No. 92 (Conexiones ODBC)

Debe tener el ODBC el nombre Hotel y conectarse con el usuario “sa”. (ver figura No.91)

Una vez ya realizado todos estos pasos podremos escoger en el sistema cualquier empresa a la cual tengamos acceso. (ver figura No.93)



Figura No. 93 (Pantalla Inicial del Sistema)

5.1.4.4 RECURSO DE RED

El hotel cuenta con una red 100 mbps con 3 switch en un panel de datos donde es depositario de todos los puntos de red.

Actualmente tiene 78 puntos de red a 100 mbps.

5.2 DISEÑO DE OBJETOS

5.2.1 DEFINICIÓN DE CLASES.

Las principales clases tenemos:

5.2.1.1 FISICO:

```

*****
' Module: SYS_FISICO.vb
' Author: hosorio
' Modified: Lunes, 08 de Mayo de 2006 16:33:30
' Purpose: Definition of the Class SYS_FISICO
*****

```

```

Public Class SYS_FISICO

    ' Attributes
    Public descripcion As String
    Public unidad As String

```



```

Public stock_inicial As Float
Public Stock As Float
Public Costo_inicial As Double
Public Costo_promedio As Double
Public costo_compra As Double
Public costo As Double
Public unidad_compra As String
Public tipo_impuesto As String
Public CODIGO_PROVEEDOR As String
Public maximos As Integer
Public minimos As Integer
Public producto_subreceta() As DIAGRAMAFISICO.SYS_SUBRECETA
Public producto_receta() As DIAGRAMAFISICO.SYS_RECETA
Public pedidos_detalle_producto() As DIAGRAMAFISICO.SYS_DET_PED
Public compras_detalle_producto() As DIAGRAMAFISICO.SYS_DET_COM
Public conteo_fisico_semanal As DIAGRAMAFISICO.SYS_FIS_STOCK_SEM_1
Public conteo_fisico_quincenal As DIAGRAMAFISICO.SYS_FIS_STOCK_QUI_1
Public conteo_fisico_mensual As DIAGRAMAFISICO.SYS_FIS_STOCK_MEN_1
Public conteo_fisico_diario As DIAGRAMAFISICO.SYS_FIS_STOCK_DIA_1
Public producto_conteo As DIAGRAMAFISICO.SYS_CONTEO
Public producto_clas As DIAGRAMAFISICO.SYS_CLASES
Public productos_bodega As DIAGRAMAFISICO.SYS_BODEGAS

```

End Class

```

*****
' Module: SYS_POS_CAB.vb
' Author: hosorio
' Modified: Lunes, 08 de Mayo de 2006 16:33:30
' Purpose: Definition of the Class SYS_POS_CAB
*****

```

Public Class SYS_POS_CAB

```

' Attributes
Public chk As Integer
Public fecha As DT
Public nombre As String
Public cedula As String
Public mesa As Integer
Public subtotal As String
Public iva As String
Public servicio As String
Public total As String
Public numero_productos As Integer
Public forma_pago As String
Public vuelto As String
Public pago_cliente As String
Public nombre_tarjeta As String
Public numero_tarjeta As String
Public numero_aprobacion As String
Public stock As String
Public fecha_stock As DT
Public pos_cabecea_stock() As DIAGRAMAFISICO.SYS_STOCK_TEMP
Public pos_cabecera_detalle() As DIAGRAMAFISICO.SYS_POS_DET
Public cc_detalle_venta() As DIAGRAMAFISICO.SYS_DET_CC

```

End Class

```

*****
' Module: SYS_POS_DET.vb
' Author: hosorio
' Modified: Lunes, 08 de Mayo de 2006 16:33:30
' Purpose: Definition of the Class SYS_POS_CAB
*****

```

Public Class SYS_POS_DET



```
' Attributes
Public NOMBRE As String
Public PRECIO As String
Public CANTIDAD As Integer
Public pos_detalle_receta As DIAGRAMAFISICO.SYS_RECETA
Public pos_cabecera_detalle As DIAGRAMAFISICO.SYS_POS_CAB
```

End Class

```
*****
' Module: SYS_EVENTOS_WEB.vb
' Author: hosorio
' Modified: Lunes, 08 de Mayo de 2006 16:33:30
' Purpose: Definition of the Class SYS_POS_CAB
*****
```

Public Class SYS_EVENTOS_WEB

```
' Attributes
Public codigo As String
Public nombre As String
Public apellidos As String
Public ciudad As String
Public pais As String
Public ci As String
Public telefono As String
Public email As String
Public huesped As String
Public horario As String
Public fecha_inicio As DT
Public fecha_final As DT
Public adicionales As String
Public pax As Integer
Public conversado As String
Public desc_conver As String
Public reservado As String
Public terminado As String
Public desc_termi As String
Public division As String
Public codigo_real As String
Public reserva_web_menu As DIAGRAMAFISICO.SYS_TIP_MENU
Public reserva_web_evento As DIAGRAMAFISICO.SYS_TIP_EVENTOS
```

End Class

```
*****
' Module: SYS_RECETA.vb
' Author: hosorio
' Modified: Lunes, 08 de Mayo de 2006 16:33:30
' Purpose: Definition of the Class SYS_POS_CAB
*****
```

```
' Name: SYS_RECETA
'-----
Public Class SYS_RECETA
```

```
' Attributes
Public codigo As String
Public nombre As String
Public descripcion As String
Public nombre_1 As String
Public cantidad_1 As String
Public unidad_1 As String
Public costo_1 As String
Public producto_2 As String
```



```
Public nombre_2 As String
Public cantidad_2 As String
Public unidad_2 As String
Public costo_2 As String
Public producto_3 As String
Public nombre_3 As String
Public cantidad_3 As String
Public unidad_3 As String
Public costo_3 As String
Public producto_4 As String
Public nombre_4 As String
Public cantidad_4 As String
Public unidad_4 As String
Public costo_4 As String
Public producto_5 As String
Public nombre_5 As String
Public cantidad_5 As String
Public unidad_5 As String
Public costo_5 As String
Public ruta As String
Public costo_sistema As String
Public costo_usuario As String
Public precio_final As Double
Public pos_detalle_receta() As DIAGRAMAFISICO.SYS_POS_DET
Public receta_subreceta As DIAGRAMAFISICO.SYS_SUBRECETA
Public producto_receta As DIAGRAMAFISICO.SYS_FISICO
Public pos_producto_boton As DIAGRAMAFISICO.SYS_BOTON
```

```
End Class
```

```
End Namespace ' DIAGRAMAFISICO
```

```
'
```

5.2.2 DISEÑO DE ALGORITMOS

Las operaciones más importantes que indican las reglas del negocio de la aplicación están formuladas como un algoritmo.

Los algoritmos muestran cómo se hace lo que queremos obtener. Un algoritmo se puede subdividir en llamadas a operaciones más sencillas y así sucesivamente, hasta que las operaciones del nivel más bajo sean suficientemente sencillas.

Aquí vamos a presentar los principales algoritmos.

5.2.2.1 COMPRAS



Estos procesos son para la solicitud de compras

```
' ++++++ Guarda Factura ++++++
```

```
Private Sub btnguardarfactura_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)  
Handles btnguardarfactura.Click
```

```
    incluir_factura()  
    incluir_detalle_actualizado()  
    incluir_detalle()  
    incluir_numero_secuencial()  
    borra_detalle()  
    arma_detalle()  
    MessageBox.Show("Se grabó Satisfactoriamente su Pedido", "Grabar  
    Pedido", MessageBoxButtons.OK)  
    nuevo = False
```

```
End Sub
```

```
' ++++++ Incluir Factura (cabecera) ++++++
```

```
Public Sub incluir_factura()
```

```
    Try
```

```
        Dim conn As New SqlConnection(g_conexion.conexion)  
        Dim cmd As SqlCommand  
        Dim sqlbuilder As New System.Text.StringBuilder
```

```
        Dim numero_fact As String = "COM-"  
        numero_fact = numero_fact + lbnumero.Text  
        With sqlbuilder  
            .Append("insert into SYS_CAB_COM")  
            .Append("(numero,codigo_real,cod_depart,pedido_por,aprobado,entregado,total,fecha_  
            pedido,detalle,cod_proveedor)")  
            .Append(" values ("")  
            .Append(numero_fact & ",")  
            .Append(lbnumero.Text & ",")  
            .Append("B" & ",")  
            .Append(txtpedidoopor.Text & ",")  
            .Append("B" & ",")  
            .Append("B" & ",")  
            .Append(total & ",")  
            .Append(txtfecha.Text & ",")  
            .Append(Me.txt_detalle.Text.ToUpper & ",")  
            .Append(Me.cbproveedor.SelectedValue & ")")  
        End With
```

```
        conn = New SqlConnection(g_conexion.conexion)  
        conn.Open()  
        cmd = New SqlCommand(sqlbuilder.ToString, conn)  
        cmd.ExecuteNonQuery()  
        conn.Close()  
        cmd.Dispose()  
        conn.Dispose()
```

```
    Catch err As SqlException  
        Throw err  
    Catch err As Exception  
        Throw err
```

```
    End Try  
End Sub
```

```
' ++++++ incluir Detalle (detalle) ++++++
```

```
Public Sub incluir_detalle()
```



```
Try
Dim conn As New SqlConnection(g_conexion.conexion)
Dim cmd As SqlCommand
Dim sqlbuilder As New System.Text.StringBuilder

Dim aux As String

aux = gridpedidos.Item(0, 4).ToString.Trim
With sqlbuilder
.Append("insert into SYS_DET_COM ")
.Append("(codigo,numero_cab,codigo_item,cantidad_pedida,nombre_articulo,precio) ")
.Append(" SELECT
codigo,numero_cab,codigo_item,cantidad_pedida,nombre_articulo,precio from
TEMP_DET_COM ")
End With

conn = New SqlConnection(g_conexion.conexion)
conn.Open()
cmd = New SqlCommand(sqlbuilder.ToString, conn)
cmd.ExecuteNonQuery()
conn.Close()
cmd.Dispose()
conn.Dispose()

Catch err As SqlException
Throw err
Catch err As Exception
Throw err
End Try

End Sub

'+++++ incluir número secuencial +++++
Public Sub incluir_numero_secuencial()
Try
Dim conn As New SqlConnection(g_conexion.conexion)
Dim cmd As SqlCommand
Dim sqlbuilder As New System.Text.StringBuilder

With sqlbuilder
.Append("update SYS_SECUENCIAL set ")
.Append("numero = " & numero_secuencial & " ")
.Append(" where tabla = " & nombre_tabla & " ")
End With

conn = New SqlConnection(g_conexion.conexion)
conn.Open()
cmd = New SqlCommand(sqlbuilder.ToString, conn)
cmd.ExecuteNonQuery()
conn.Close()
cmd.Dispose()
conn.Dispose()

Catch err As SqlException
Throw err
Catch err As Exception
Throw err

End Try
End Sub

'+++++ arma número +++++
Sub arma_numero()
Try
Dim g_conexion As New cls_conexion
```



```
Dim drnumero As SqlDataReader
Dim conn As New SqlConnection(g_conexion.conexion)
Dim sql As String = "Select numero from SYS_SECUENCIAL where tabla = " &
nombre_tabla & " "

Dim cmd As SqlCommand
cmd = New SqlCommand(sql, conn)

conn.Open()
drnumero = cmd.ExecuteReader

If drnumero.Read() Then
    lnumero.Text = drnumero.Item("numero").ToString
End If
numero_secuencial = Int(lnumero.Text) + 1
conn.Close()
contador = 0
Catch err As SqlException
    MessageBox.Show(err.Message)
Catch err As Exception
    MessageBox.Show(err.Message)

End Try
End Sub
```

'+++++ borra detalle +++++'

```
Public Sub borra_detalle()
    Try
        Dim conn As New SqlConnection(g_conexion.conexion)
        Dim cmd As SqlCommand
        Dim sqlbuilder As New System.Text.StringBuilder

        With sqlbuilder
            Append("delete from TEMP_DET_COM ")
            .Append(" where numero_cab= " & lnumero.Text.ToString.Trim & "")
        End With

        With sqlbuilder
            Append("delete from TEMP_DET_COM1 ")
            .Append(" where numero_cab= " & lnumero.Text.ToString.Trim & "")
        End With

        conn = New SqlConnection(g_conexion.conexion)
        conn.Open()
        cmd = New SqlCommand(sqlbuilder.ToString, conn)
        cmd.ExecuteNonQuery()
        conn.Close()
        cmd.Dispose()
        conn.Dispose()

        Catch err As SqlException
            Throw err
        Catch err As Exception
            Throw err

    End Try
End Sub
```

Estos procesos son para la aprobación y recepción de la compra

'+++++ Actualizar Compra +++++'

```
Sub actualizarcompra(ByVal condicion As String, ByVal mensaje As String)
```



```

Try
    Dim objactualizar As New clscustomer
    Dim id As String = Me.txttempid.Text
    objactualizar.Aprobar_compra(id, g_conexion.conexion, condicion)
    armafacturas()
    btnsalvar.Enabled = True
    btnincluir.Enabled = True
    btnexcluir.Enabled = True
    MessageBox.Show("'" & mensaje & "' se ha aprobado su pedido", "Aprobacion",
    MessageBoxButtons.OK)
Catch err As Exception
    MessageBox.Show(err.Message)
End Try

End Sub

'+++++ Arma Facturas +++++
Private Sub armafacturas()
    Dim clientes As New clscustomer
    With Me.lstfacturas

        .DisplayMember = "numero"
        .ValueMember = "codigo_real"
        .DataSource = clientes.ArmaClientes_Apruebacompra(
            g_conexion.conexion, "B").Tables("SYS_CAB_COM").DefaultView
    If Not (Me.lstfacturas.Items.Count = 0) Then
        .SelectedIndex = 0
    End If
    End With
End Sub

'+++++ Aprueba Compra +++++

Public Function ArmaClientes_Apruebacompra(ByVal conexion As String, ByVal bandera As String) As
DataSet
    Dim no As String = bandera
    Try
        Dim conn As New SqlConnection(conexion)
        Dim sql As String = "Select numero,codigo_real,fecha_pedido from SYS_CAB_com where
aprobado = '" & no & "'order by fecha_pedido"
        Dim dscustomers As New DataSet
        Dim dacustomers As New SqlDataAdapter(sql, conn)
        conn.Open()
        dacustomers.Fill(dscustomers, "SYS_CAB_COM")
        conn.Close()
        Return dscustomers

    Catch err As SqlException
        Throw err
    Catch err As Exception
        Throw err

    End Try
End Function

```

5.2.2.2 PEDIDOS



Estos procesos son para la solicitud de Pedidos las demás funciones son similares a las de compras

```
' ++++++ Guarda Pedidos ++++++
```

```
Public Sub GUARDAR()
```

```
    incluir_detalle()
    incluir_factura()
    incluir_numero_secuencial()
    borra_detalle()
    arma_detalle()
    limpia_textos()
    MessageBox.Show("Se grabó Satisfactoriamente su Pedido",
    "Grabar Pedido", MessageBoxButtons.OK)
    nuevo = False
    Me.btn_guardar.Pushed = False
End Sub
```

5.2.2.3 INVENTARIOS

Aquí vemos como generamos dinámicamente las tablas de conteo, ingresamos producto y actualizamos las fechas.

```
' ++++++ Crea Tablas ++++++
```

```
Private Sub crea_tablas(ByVal condicion As String)
```

```
    Try
        Dim conn As New SqlConnection(g_conexion.conexion)

        Dim sql As String = _
            " CREATE TABLE [SYS_FIS_STOCK_" & condicion & "_" & inicio & "]" (" & _
            "[codigo_prod] [nvarchar] (8) COLLATE Modern_Spanish_CI_AS NOT NULL ," & _
            "[descripcion] [nvarchar] (50) COLLATE Modern_Spanish_CI_AS NOT NULL ," & _
            "[tipo] [char] (4) COLLATE Modern_Spanish_CI_AS NOT NULL ," & _
            "[unidad] [nvarchar] (50) COLLATE Modern_Spanish_CI_AS NOT NULL ," & _
            "[stock_inicial] [int] NULL ," & _
            "[Stock] [nvarchar] (8) ," & _
            "[codigo_clase] [nvarchar] (3) COLLATE Modern_Spanish_CI_AS NOT NULL ," & _
            "[CODIGO_PROVEEDOR] [nvarchar] (10) COLLATE Modern_Spanish_CI_AS " & _
            "NULL ," & _
            " CONSTRAINT [PK_SYS_FIS_STOCK_" & condicion & "_" & inicio & "] PRIMARY KEY " & _
            " CLUSTERED " & _
            " ([codigo_prod]) ON [PRIMARY] ) ON [PRIMARY] "

        Dim dsreceta As New DataSet
        Dim drreceta As SqlDataReader
        Dim cmd As SqlCommand
        cmd = New SqlCommand(sql, conn)
        conn.Open()
        drreceta = cmd.ExecuteReader
        drreceta.Close()
        conn.Close()
        cmd.Dispose()
        conn.Dispose()
    End Try
End Sub
```



```

Catch err As SqlException
    MessageBox.Show(err.Message)
Catch err As Exception
    MessageBox.Show(err.Message)

End Try
End Sub

' ++++++ Inserta Productos ++++++
Private Sub inserta_productos(ByVal condicion As String)
    Try
        Dim conn As New SqlConnection(g_conexion.conexion)
        Dim sql As String = _
            " INSERT INTO SYS_FIS_STOCK_" & condicion & "_" & inicio & "
            (CODIGO_PROD,DESCRIPCION,TIPO,UNIDAD,STOCK_INICIAL,CODIGO_CLASE,
            CODIGO_PROVEEDOR,stock)" & _
            " SELECT
            CODIGO_PROD,DESCRIPCION,TIPO,UNIDAD,STOCK_INICIAL,CODIGO_CLASE,
            CODIGO_PROVEEDOR," & 0 & " & _
            " FROM SYS_FISICO WHERE TIPO = " & condicion & ""
        Dim dsreceta As New DataSet
        Dim drreceta As SqlDataReader
        Dim cmd As SqlCommand
        cmd = New SqlCommand(sql, conn)
        conn.Open()
        drreceta = cmd.ExecuteReader
        drreceta.Close()
        conn.Close()
        cmd.Dispose()
        conn.Dispose()
        Catch err As SqlException
            MessageBox.Show(err.Message)
        Catch err As Exception
            MessageBox.Show(err.Message)
        End Try
    End Sub

```

Aquí vemos como generamos las recetas.

```

' ++++++ Calcula Fisico ++++++
Sub calcula_fisico()
    If Me.cmb_sub1.Text.Trim <> "" And Me.txt_cod_1.Text.IndexOf("_") = 1 Then
        recalcula_producto1("Select costo from SYS_subreceta where codigo = " & Me.txt_cod_1.Text &
        "", "costo")
    Else
        If Me.cmb_sub1.Text.Trim <> "" And Me.txt_cod_1.Text.IndexOf("-") = 4 Then
            recalcula_producto1("Select costo_compra from SYS_fisico where codigo_prod = " &
            Me.txt_cod_1.Text & "", "costo_compra")
        End If
    End If

    If Me.cmb_sub2.Text.Trim <> "" And Me.txt_cod_2.Text.IndexOf("_") = 1 Then
        recalcula_producto2("Select costo from SYS_subreceta where codigo = " & Me.txt_cod_2.Text &
        "", "costo")
    Else
        If Me.cmb_sub2.Text.Trim <> "" And Me.txt_cod_2.Text.IndexOf("-") = 4 Then
            recalcula_producto2("Select costo_compra from SYS_fisico where codigo_prod = " &
            Me.txt_cod_2.Text & "", "costo_compra")
        End If
    End If

    If Me.cmb_sub3.Text.Trim <> "" And Me.txt_cod_3.Text.IndexOf("_") = 1 Then

```



```

        recalcula_producto3("Select costo from SYS_subreceta where codigo = " & Me.txt_cod_3.Text &
        "", "costo")
    Else
        If Me.cmb_sub3.Text.Trim <> "" And Me.txt_cod_3.Text.IndexOf("-") = 4 Then
            recalcula_producto3("Select costo_compra from SYS_fisico where codigo_prod = " &
            Me.txt_cod_3.Text & "", "costo_compra")
        End If
    End If

    If Me.cmb_sub4.Text.Trim <> "" And Me.txt_cod_4.Text.IndexOf("-") = 1 Then
        recalcula_producto4("Select costo from SYS_subreceta where codigo = " & Me.txt_cod_4.Text &
        "", "costo")
    Else
        If Me.cmb_sub4.Text.Trim <> "" And Me.txt_cod_4.Text.IndexOf("-") = 4 Then
            recalcula_producto4("Select costo_compra from SYS_fisico where codigo_prod = " &
            Me.txt_cod_4.Text & "", "costo_compra")
        End If
    End If

    If Me.cmb_sub5.Text.Trim <> "" And Me.txt_cod_5.Text.IndexOf("-") = 1 Then
        recalcula_producto5("Select costo from SYS_subreceta where codigo = " & Me.txt_cod_5.Text &
        "", "costo")
    Else
        If Me.cmb_sub5.Text.Trim <> "" And Me.txt_cod_5.Text.IndexOf("-") = 4 Then
            recalcula_producto5("Select costo_compra from SYS_fisico where codigo_prod = " &
            Me.txt_cod_5.Text & "", "costo_compra")
        End If
    End If
End Sub

'+++++ Busca Subreceta +++++
Sub buscasubreceta(ByVal strfiltro As String)

Try
    Dim conn As New SqlConnection(g_conexion.conexion)
    Dim sql As String = "Select * from SYS_RECETA where codigo = " & strfiltro & ""
    Dim dscustomer As New DataSet
    Dim drcustomer As SqlDataReader
    Dim cmd As SqlCommand
    cmd = New SqlCommand(sql, conn)
    conn.Open()
    drcustomer = cmd.ExecuteReader
    If drcustomer.Read() Then
        Me.txt_descripcion.Text = drcustomer.Item("descripcion").ToString

        Me.cmb_sub1.Text = drcustomer.Item("nombre_1").ToString
        Me.txt_cantidad1.Text = drcustomer.Item("cantidad_1").ToString
        Me.txt_medida1.Text = drcustomer.Item("unidad_1").ToString
        Me.txt_costo_1.Text = drcustomer.Item("COSTO_1").ToString
        Me.txt_cod_1.Text = drcustomer.Item("producto_1").ToString

        Me.cmb_sub2.Text = drcustomer.Item("nombre_2").ToString
        Me.txt_cantidad2.Text = drcustomer.Item("cantidad_2").ToString
        Me.txt_medida2.Text = drcustomer.Item("unidad_2").ToString
        Me.txt_costo_2.Text = drcustomer.Item("COSTO_2").ToString
        Me.txt_cod_2.Text = drcustomer.Item("producto_2").ToString

        Me.cmb_sub3.Text = drcustomer.Item("nombre_3").ToString
        Me.txt_cantidad3.Text = drcustomer.Item("cantidad_3").ToString
        Me.txt_medida3.Text = drcustomer.Item("unidad_3").ToString
        Me.txt_costo_3.Text = drcustomer.Item("COSTO_3").ToString
        Me.txt_cod_3.Text = drcustomer.Item("producto_3").ToString

        Me.cmb_sub4.Text = drcustomer.Item("nombre_4").ToString
        Me.txt_cantidad4.Text = drcustomer.Item("cantidad_4").ToString
        Me.txt_medida4.Text = drcustomer.Item("unidad_4").ToString
        Me.txt_costo_4.Text = drcustomer.Item("COSTO_4").ToString
        Me.txt_cod_4.Text = drcustomer.Item("producto_4").ToString
    End If
End Try

```



```

Me.cmb_sub5.Text = drcustomer.Item("nombre_5").ToString
Me.txt_cantidad5.Text = drcustomer.Item("cantidad_5").ToString
Me.txt_medida5.Text = drcustomer.Item("unidad_5").ToString
Me.txt_costo_5.Text = drcustomer.Item("COSTO_5").ToString
Me.txt_cod_5.Text = drcustomer.Item("producto_5").ToString

Me.TXT_RUTA.Text = drcustomer.Item("ruta").ToString
Me.CMB_BOTON.Text = drcustomer.Item("boton").ToString
Me.txt_costo_sistema.Text = drcustomer.Item("costo_sistema").ToString
Me.txt_porcentaje.Text = drcustomer.Item("costo_usuario").ToString
Me.txt_precio_final.Text = drcustomer.Item("precio_final").ToString
End If
If Me.TXT_RUTA.Text <> "" Then
    Me.FOTO_RECETA.Image = Image.FromFile(Me.TXT_RUTA.Text.ToString.ToUpper)
Else
    Me.FOTO_RECETA.Image =
Image.FromFile("C:\HOTEL\HOWJLC\HOWJLC\Logos\hotel\ninguno.GIF")
End If
drcustomer.Close()
conn.Close()
cmd.Dispose()
conn.Dispose()
Catch err As SqlException
    MessageBox.Show(err.message)
Catch err As Exception
    MessageBox.Show(err.message)
End Try
End Sub

```

5.2.2.4 VENTAS

Describimos en este punto las funciones de ventas

```

' ++++++ Crea Botones ++++++
Sub crea_botones()
    Try
        Dim g_conexion As New cls_conexion
        Dim drnumero As SqlDataReader
        Dim conn As New SqlConnection(g_conexion.conexion)

        Dim n As Int16

        For n = 1 To 14
            Dim sql As String = "Select nombre from SYS_BOTON where codigo = 'BTN' & n & "
            Dim cmd As SqlCommand
            cmd = New SqlCommand(sql, conn)
            conn.Open()
            drnumero = cmd.ExecuteReader
            If drnumero.Read() Then
                Select Case n
                    Case 1
                        Me.BTN_1.Text = drnumero.Item("nombre").ToString
                    Case 2
                        Me.BTN_2.Text = drnumero.Item("nombre").ToString
                    Case 3
                        Me.BTN_3.Text = drnumero.Item("nombre").ToString
                    Case 4
                        Me.BTN_4.Text = drnumero.Item("nombre").ToString
                    Case 5
                        Me.BTN_5.Text = drnumero.Item("nombre").ToString
                    Case 6
                        Me.BTN_6.Text = drnumero.Item("nombre").ToString
                    Case 7

```



```
        Me.BTN_7.Text = drnumero.Item("nombre").ToString
    Case 8
        Me.BTN_8.Text = drnumero.Item("nombre").ToString
    Case 9
        Me.BTN_9.Text = drnumero.Item("nombre").ToString
    Case 10
        Me.BTN_10.Text = drnumero.Item("nombre").ToString
    Case 11
        Me.BTN_11.Text = drnumero.Item("nombre").ToString
    Case 12
        Me.BTN_12.Text = drnumero.Item("nombre").ToString
    Case 13
        Me.BTN_13.Text = drnumero.Item("nombre").ToString
    Case 14
        Me.BTN_14.Text = drnumero.Item("nombre").ToString

    End Select
End If
conn.Close()
Next

Catch err As SqlException
    MessageBox.Show(err.Message)
Catch err As Exception
    MessageBox.Show(err.Message)

End Try

End Sub

' +++++ Arma Recetas +++++
Sub armarecetas(ByVal condicion As String)

    BORRA_LISTA()
    CREA_LISTA()
    Try
        Dim conn As New SqlConnection(g_conexion.conexion)
        Dim sql As String = "Select * from SYS_receta where boton = '" & condicion.ToUpper.Trim &
"ORDER BY NOMBRE "
        Dim dsreceta As New DataSet
        Dim drreceta As SqlDataReader
        Dim cmd As SqlCommand
        cmd = New SqlCommand(sql, conn)

        Dim n As Integer = 0
        conn.Open()
        drreceta = cmd.ExecuteReader
        While drreceta.Read() And n <= 90
            lista_productos.Items(n).Text = drreceta.Item("NOMBRE")
            lista_productos.Items(n).Text = lista_productos.Items(n).Text + " $ " &
Val(drreceta.Item("PRECIO_FINAL")) & ""
            n = n + 1
        End While

        drreceta.Close()
        conn.Close()
        cmd.Dispose()
        conn.Dispose()

        Catch err As SqlException
            MessageBox.Show(err.Message)
        Catch err As Exception
            MessageBox.Show(err.Message)

    End Try
End Sub
```



```
' ++++++ Graba Cabecera ++++++
```

```
Sub GRABAR_CAB()
```

```
Try
```

```
Dim conn As New SqlConnection(g_conexion.conexion)
```

```
Dim cmd As SqlCommand
```

```
Dim sqlbuilder As New System.Text.StringBuilder
```

```
With sqlbuilder
```

```
.Append("insert into SYS_POS_CAB ")
```

```
.Append("(CHK,fecha,NOMBRE,CEDULA,MESA,SUBTOTAL,iva,servicio,total,numero_productos,forma_pago,vuelto,pago_cliente,nombre_tarjeta,numero_tarjeta,STOCK,numero_aprobacion)")
```

```
.Append(" values ("
```

```
.Append(numero_secuencial & ",")
```

```
.Append(Now & ",")
```

```
.Append(Me.TXT_CLIENTE.Text.ToUpper & ",")
```

```
.Append(Me.TXT_CEDULA.Text.Trim & ",")
```

```
.Append(MESA & ",")
```

```
.Append(Me.LB_SUBTOTAL.Text & ",")
```

```
.Append(Me.lb_iva.Text & ",")
```

```
.Append(Me.lb_porcentaje.Text & ",")
```

```
.Append((Me.LB_TOTAL.Text) & ",")
```

```
.Append(Me.LST_CUENTA.Items.Count & ",")
```

```
.Append(objeto.forma_pago & ",")
```

```
.Append(objeto.valor_vuelto & ",")
```

```
.Append(objeto.valor_pago_cliente & ",")
```

```
.Append(objeto.tipo_tarjeta & ",")
```

```
.Append(objeto.tarjeta_credito & ",")
```

```
.Append("N" & ",")
```

```
.Append(objeto.numero_aprobacion & ")")
```

```
End With
```

```
conn = New SqlConnection(g_conexion.conexion)
```

```
conn.Open()
```

```
cmd = New SqlCommand(sqlbuilder.ToString, conn)
```

```
cmd.ExecuteNonQuery()
```

```
conn.Close()
```

```
cmd.Dispose()
```

```
conn.Dispose()
```

```
Catch err As SqlException
```

```
Throw err
```

```
Catch err As Exception
```

```
Throw err
```

```
End Try
```

```
End Sub
```

5.2.2.5 EVENTOS

Describimos en este punto las funciones de eventos

```
' ++++++ Busca montaje ++++++
```

```
Sub busca_tipo_evento()
```

```
Try
```

```
Dim g_conexion As New cls_conexion
```



```

Dim drnumero As SqlDataReader
Dim conn As New SqlConnection(g_conexion.conexion)
Dim sql As String = " select * from sys_tip_eventos where codigo = " & Me.edittipo_evento.Text &
"" ""

Dim cmd As SqlCommand
cmd = New SqlCommand(sql, conn)
conn.Open()
drnumero = cmd.ExecuteReader
If drnumero.Read() Then
    Me.txt_evento.Text = drnumero("tipo_evento")
    Dim ruta As String = drnumero("ruta")
    ruta = ruta.Substring(8, ruta.Length - 8)
    Me.pict_evento.Image = Image.FromFile(ruta)
End If
conn.Close()
Catch err As SqlException
    MessageBox.Show(err.Message)
Catch err As Exception
    MessageBox.Show(err.Message)
End Try

' +++++ Busca Menú+++++
Sub busca_tipo_menu()

    Try
        Dim g_conexion As New cls_conexion
        Dim drnumero As SqlDataReader
        Dim conn As New SqlConnection(g_conexion.conexion)
        Dim sql As String = " select * from sys_tip_menus where codigo = " & Me.edittipo_menu.Text & "" ""

        Dim cmd As SqlCommand
        cmd = New SqlCommand(sql, conn)
        conn.Open()
        drnumero = cmd.ExecuteReader
        If drnumero.Read() Then
            Me.txt_menu.Text = drnumero("nombre")
            Me.txt_menu_precio.Text = drnumero("costo")
            Me.lb_total.Text = ""
            Me.lb_dias.Text = ""
            Me.lb_total.Text = (Cdbl(Me.txt_menu_precio.Text) * Cdbl(Me.editpax.Text)) * (DateDiff("d",
Me.editfecha_inicio.Text, Me.editfecha_final.Text) + 1)
            Me.lb_total.Text = Format(Cdbl(Me.lb_total.Text), "##,##0.00")
            Me.lb_dias.Text = (DateDiff("d", Me.editfecha_inicio.Text, Me.editfecha_final.Text) + 1)

            Me.lb_dias.Text = Me.lb_dias.Text + " días de evento"
            Me.txt_entrada.Text = drnumero("entrada")
            Me.txt_fuerte.Text = drnumero("fuerte")
            Me.txt_postre.Text = drnumero("postre")
            Me.txt_descripcion.Text = drnumero("descripcion")
            Dim ruta As String = drnumero("ruta")
            ruta = ruta.Substring(8, ruta.Length - 8)
            Me.pict_menu.Image = Image.FromFile(ruta)
        End If
        conn.Close()
        Catch err As SqlException
            MessageBox.Show(err.Message)
        Catch err As Exception
            MessageBox.Show(err.Message)

    End Try
End Sub

' +++++ Graba lo conversado+++++
Sub conversado()

```



```
Try
    Dim g_conexion As New cls_conexion
    Dim drnumero As SqlDataReader
    Dim conn As New SqlConnection(g_conexion.conexion)
    Dim sql As String = "update SYS_eventos_web set desc_conver = " & Me.editdesc_conver.Text &
    ", conversado = " & "S" & " " where codigo = " & Me.editcodigo.Text & " "

    Dim cmd As SqlCommand
    cmd = New SqlCommand(sql, conn)
    conn.Open()
    drnumero = cmd.ExecuteReader
    conn.Close()
    MessageBox.Show("Se actualizó la conversación con el cliente", "Conversación",
    MessageBoxButtons.OK)
    recupera()

Catch err As SqlException
    MessageBox.Show(err.Message)
Catch err As Exception
    MessageBox.Show(err.Message)

End Try

End Sub
```

' ++++++ Crea División++++++

```
Sub crea_division()
    Try
        Dim g_conexion As New cls_conexion
        Dim drnumero As SqlDataReader
        Dim conn As New SqlConnection(g_conexion.conexion)
        Dim sql As String = "select DATEDIFF(" & "d" & ", fecha_inicio, fecha_final) as diferencia from
        sys_eventos_web where codigo = " & Me.editcodigo.Text & " "
        Dim n As Int16
        Dim cmd As SqlCommand
        cmd = New SqlCommand(sql, conn)
        conn.Open()
        drnumero = cmd.ExecuteReader
        If drnumero.Read Then
            If Val(drnumero("diferencia")) > 0 Then
                For n = 0 To Val(drnumero("diferencia"))
                    graba_division(n, Me.editfecha_inicio.Text, Me.editcodigo.Text)
                Next
            Else
                reservado()
            End If
        End If
        MessageBox.Show("Se generaron " & Val(drnumero("diferencia")) + 1 & " divisiones de reserva",
        "RESERVA", MessageBoxButtons.OK)
        conn.Close()
        recupera()
        Catch err As SqlException
            MessageBox.Show(err.Message)
        Catch err As Exception
            MessageBox.Show(err.Message)
        End Try
    End Sub
```

' ++++++ Busca numero++++++

```
Function busca_numero() As Int16
```



```

Try
    Dim g_conexion As New cls_conexion
    Dim drnumero As SqlDataReader
    Dim conn As New SqlConnection(g_conexion.conexion)
    Dim sql As String = "select count(codigo) as codigo from sys_eventos_web "
    Dim n As Int16

    Dim cmd As SqlCommand
    cmd = New SqlCommand(sql, conn)
    conn.Open()
    drnumero = cmd.ExecuteReader
    If drnumero.Read Then
        n = drnumero("codigo")
    End If

    conn.Close()
    Return n
Catch err As SqlException
    MessageBox.Show(err.Message)
Catch err As Exception
    MessageBox.Show(err.Message)

End Try
End Function

'+++++Graba División+++++
Sub graba_division(ByVal valor As Int16, ByVal fecha As Date, ByVal codigo As String)
    Try
        Dim g_conexion As New cls_conexion
        Dim drnumero As SqlDataReader
        Dim conn As New SqlConnection(g_conexion.conexion)

        Dim cmd As SqlCommand
        Dim sqlbuilder As New System.Text.StringBuilder
        fecha = DateAdd("d", valor, fecha)
        Dim cod As Int16 = busca_numero() + 1
        Dim variable As String = "S"
        Dim variable1 As String = "N"

        With sqlbuilder
            .Append("insert into SYS_eventos_web ")
            .Append("
(codigo,horario,fecha_inicio,fecha_final,huesped,codigo_real,reservado,terminado,nombre,apellidos,ciudad,pais,")
            .Append("ci,
telefono,email,tipo_evento,tipo_menu,adicionales,pax,conversado,desc_conver,desc_termi,division)")
            .Append(" values (")
            .Append(cod & ",")
            .Append(Me.edithorario.Text & ",")
            .Append(fecha & ",")
            .Append(fecha & ",")
            .Append(Me.edithuesped.Text & ",")
            .Append(Me.editcodigo.Text & ",")
            .Append(variable & ",")
            .Append(variable1 & ",")

            .Append(Me.editnombre.Text & ",")
            .Append(Me.editapellidos.Text & ",")
            .Append(Me.editciudad.Text & ",")
            .Append(Me.editpais.Text & ",")
            .Append(Me.editci.Text & ",")
            .Append(Me.editelefono.Text & ",")
        End With
    End Try
End Sub

```



```
.Append(Me.editemail.Text & ",")  
.Append(Me.edittipo_evento.Text & ",")  
.Append(Me.edittipo_menu.Text & ",")  
.Append(Me.editadicionales.Text & ",")  
.Append(Me.editpax.Text & ",")  
.Append(Me.editconversado.Text & ",")  
.Append(Me.editdesc_conver.Text & ",")  
.Append(Me.editdesc_termi.Text & ",")  
.Append(variable & ",")
```

End With

```
'conn = New SqlConnection(g_conexion)  
conn.Open()  
cmd = New SqlCommand(sqlbuilder.ToString, conn)  
cmd.ExecuteNonQuery()  
conn.Close()  
cmd.Dispose()  
conn.Dispose()
```

```
Catch err As SqlException  
    MessageBox.Show(err.Message)
```

```
Catch err As Exception  
    MessageBox.Show(err.Message)
```

```
End Try  
End Sub
```



5.2.3 DISEÑO DE ASOCIACIONES

| Name | Code | Class B | Class A | Multiplicity A | Multiplicity B |
|-----------------------------------|-----------------------------------|-----------------|---------------------|----------------|----------------|
| pedidos_cabecera_detalle_temporal | Pedidos_cabecera_detalle_temporal | TEMP_DET_PED | SYS_CAB_PED | 1..1 | 0..* |
| compra_cabecera_detalle_temporal | Compra_cabecera_detalle_temporal | TEMP_DET_COM | SYS_CAB_COM | 1..1 | 0..* |
| usuario_perfil | Usuario_perfil | SYS_USER | SYS_PERFIL | 0..1 | 0..* |
| producto_subreceta | producto_subreceta | SYS_SUBRECETA | SYS_FISICO | 0..1 | 0..* |
| pos_cabecea_stock | pos_cabecea_stock | SYS_STOCK_TEMP | SYS_POS_CAB | 1..1 | 0..* |
| receta_subreceta | receta_subreceta | SYS_RECETA | SYS_SUBRECETA | 0..1 | 0..* |
| producto_receta | producto_receta | SYS_RECETA | SYS_FISICO | 0..1 | 0..* |
| pos_producto_boton | pos_producto_boton | SYS_RECETA | SYS_BOTON | 0..1 | 0..* |
| pos_detalle_receta | pos_detalle_receta | SYS_POS_DET | SYS_RECETA | 1..1 | 0..* |
| pos_cabecera_detalle | pos_cabecera_detalle | SYS_POS_DET | SYS_POS_CAB | 1..1 | 0..* |
| conteo_fisico_semanal | conteo_fisico_semanal | SYS_FISICO | SYS_FIS_STOCK_SEM_1 | 1..1 | 0..* |
| conteo_fisico_quincenal | conteo_fisico_quincenal | SYS_FISICO | SYS_FIS_STOCK QUI_1 | 1..1 | 0..* |
| conteo_fisico_mensual | conteo_fisico_mensual | SYS_FISICO | SYS_FIS_STOCK_MEN_1 | 1..1 | 0..* |
| conteo_fisico_diario | conteo_fisico_diario | SYS_FISICO | SYS_FIS_STOCK DIA_1 | 1..1 | 0..* |
| producto_conteo | producto_conteo | SYS_FISICO | SYS_CONTEO | 1..1 | 0..* |
| productos_bodega | productos_bodega | SYS_FISICO | SYS_BODEGAS | 0..1 | 0..* |
| reserva_web_menu | Reserva_web_menu | SYS_EVENTOS_WEB | SYS_TIP_MENU | 0..1 | 0..* |
| reserva_web_evento | Reserva_web_evento | SYS_EVENTOS_WEB | SYS_TIP_EVENTOS | 0..1 | 0..* |
| pedidos_detalle_producto | Pedidos_detalle_producto | SYS_DET_PED | SYS_FISICO | 1..1 | 0..* |
| Pedidos_cabecera_detalle | Pedidos_cabecera_detalle | SYS_DET_PED | SYS_CAB_PED | 1..1 | 0..* |
| compras_detalle_producto | compras_detalle_producto | SYS_DET_COM | SYS_FISICO | 1..1 | 0..* |
| compras_cabecera_detalle | compras_cabecera_detalle | SYS_DET_COM | SYS_CAB_COM | 1..1 | 0..* |
| cc_detalle_venta | cc_detalle_venta | SYS_DET_CC | SYS_POS_CAB | 1..1 | 0..* |
| cc_cabecera_detalle | cc_cabecera_detalle | SYS_DET_CC | SYS_CAB_CC | 1..1 | 0..* |
| pedidos_cabecera_areas | Pedidos_cabecera_areas | SYS_CAB_PED | SYS_AREAS | 0..1 | 0..* |
| cxc_cabecera_clientes | cxc_cabecera_clientes | SYS_CAB_CC | SYS_CLIENTES | 0..1 | 0..* |
| compras_cabecera_proveedor | compras_cabecera_proveedor | SYS_CAB_COM | SYS_PROVEEDOR | 0..1 | 0..* |

Tabla 5.1 (Diseño de Asociaciones)

