

INSTITUTO TECNOLÒGICO DE PARRAL

Lic. en Informática

“Proyecto de Residencias Profesionales”



(Sistema de Cafeteria)

ALUMNO:
MEZA UGARTE ISABEL CRISTINA
03410017

PALMA DUARTE JITZEL DENISSE
02410916

ASESOR EXTERNO:
SRITA. GRECIA PAYAN ESPINOZA

ASESOR INTERNO:
L.S.C.A Alma Elvira Zubia Barraza

JUNIO 2007

INSTITUTO TECNOLÓGICO DE PARRAL

CARTA DE CONFORMIDAD



Asesor Interno:

L.S.C.A Alma Elvira Zubia Barraza.

Revisor:

A handwritten signature in black ink, appearing to read "JP".

Ing. Jesús Ariel Polanco Perea.

Revisor:

A handwritten signature in black ink, appearing to read "JYH".

Lic. Jesse Yamal Hernández Moreno

**El Rincón Del Café
R.F.C. BAGA-480301-LY9
CURP BAGA480301MCHRRD01
AVE. INDEPENDENCIA #89
HGO. DEL PARRAL, CHIH.**

**ING. JORGE LUÍS GARCÍA CANTÓN
DIRECTOR DE ITP PARRAL
INSTITUTO TECNOLÓGICO DE PARRAL
PRESENTE**

El Rincón del Café de la Sra. Adela Barraza García hace de su conocimiento que el alumno PALMA DUARTE JITZEL DENISSE, con número de control 02410916, que actualmente cursa el noveno semestre en esta Institución en la carrera de Licenciatura en Informática; fue liberado de su Residencia Profesional con el proyecto “SyCAF”, teniendo como asesor externo a la Srita. Grecia Payan Espinoza, y tras haber cumplido con las 640 horas correspondientes, en el periodo de Enero a Junio del 2007.

ATENTAMENTE


Sra. Adela Barraza García

**El Rincón Del Café
R.F.C. BAGA-480301-LY9
CURP BAGA480301MCHRRD01
AVE. INDEPENDENCIA #89
HGO. DEL PARRAL, CHIH.**

ING. JORGE LUÍS GARCÍA CANTÓN
DIRECTOR DE ITP PARRAL
INSTITUTO TECNOLÓGICO DE PARRAL
PRESENTE

El Rincón del Café de la Sra. Adela Barraza García hace de su conocimiento que el alumno MEZA UGARTE ISABEL CRISTINA, con número de control 03410017, que actualmente cursa el noveno semestre en esta Institución en la carrera de Licenciatura en Informática; fue liberado de su Residencia Profesional con el proyecto “SyCAF”, teniendo como asesor externo a la Srita. Grecia Payan Espinoza, y tras haber cumplido con las 640 horas correspondientes, en el periodo de Enero a Junio del 2007.

ATENTAMENTE



Sra. Adela Barraza García

ÍNDICE

Introducción	1
Justificación	3
Objetivo General	4
Caracterización del Área en que se participó	6
Organigrama de la empresa	8
Problemas a Resolver	9
Alcances y Limitaciones	10
Estudio de Factibilidad	11
Marco Teórico	13
Sistema de Información	14
Equipo Computacional	14
Planificación del Proyecto	15
Definición del Sistema	16
Recolección y Análisis de Requisitos	16
Diseño de la base de Datos	17
Selección del Manejador de Base de Datos	19
Diseño del Sistema de Información	19
Implementación	20
Conversión y Carga de Datos	21
Pruebas	21
Entrevista	23

Entrevista tipo piramidal	24
Entrevista tipo embudo	24
Entrevista tipo rombo	24
Diagramas de Flujo	25
Ventajas de los Diagramas de Flujo	25
Símbolos que emplean los diagramas de Flujo	26
Desarrollo del Diagrama de Flujo	28
Reglas de los Diagramas de Flujo	29
Diccionario de Datos	31
Herramientas de Desarrollo	32
¿Qué es Delphi?	33
Componentes	34
Eventos	35
Base de Datos	35
Borland Database Engine (BDE)	35
Desarrollo Visual	36
Entorno Integrado de Desarrollo (EID)	36
Depurador Integrado	36
Métodos utilizados	38
Descripción y Procedimientos de las Actividades Realizadas	39
Análisis del Sistema y Recopilación de Información	39
Elaboración de los Diagramas de Flujo	42
Diagrama de Flujo de Datos del Sistema Propuesto	43
Flujo del Manejo de la Comanda	44

Elección y Aprendizaje de Herramientas de Desarrollo	45
Elaboración del Diccionario de Datos	25
Diseño del Sistema	48
Principales componentes que se utilizaron para el desarrollo del sistema ..	49
Desarrollo del sistema	52
Pruebas	54
Elaboración de Manuales	54
Implantación	54
Conclusiones	55
Recomendaciones	56
Bibliografía	57
Anexos	58
Anexo 1 (Entrevista 1)	59
Anexo 2 (Entrevista 2)	60
Anexo 3 (Comanda)	61
Anexo 4 (Determinación del Hw)	62
Manual del Usuario	63
Manual Técnico	64

INTRODUCCIÓN

La globalización y las crecientes demandas de clientes cada vez más complejos han creado necesidades y oportunidades de negocio que implican retos difíciles de alcanzar para las empresas. Además el proceso tecnológico con la informática, robótica, el diseño y la fabricación asistida por computadora, sin dejar de pasar por alto las telecomunicaciones, suponen una profunda transformación en el sistema productivo de las empresas y provoca que la innovación tecnológica escape al dominio de la mayor parte de las empresas y se transforme en un imperativo para el crecimiento y la supervivencia.

Las empresas se deben mostrar favorables a los cambios, cambios que les permitirán ser más eficientes en su organización, es aquí donde la informática desempeña un papel muy importante para los beneficios de la empresa, los cuales se darán para aquellos que sepan utilizarlos bien y que desarrollen aplicaciones orientadas a explotar el conocimiento de sus empleados generando así valor añadido.

Por lo anterior mencionado el objetivo por el cual se realizó este sistema fue para poder agilizar el proceso del corte de caja, ya que se hacía de forma manual y a veces se tenía que repetir varias veces porque las cuentas no quedaban exactas; otro motivo por el cual también se desarrollo este software es porque se quiere tener un manejo de inventarios como lo es en las botellas, un control de proveedores y gastos.

Para poder llevar a cabo el desarrollo de dicho software, primero se analizó el lugar y se determinaron los requerimientos que tiene y los que serían necesarios para llevar a cabo el análisis y diseño del software.

Básicamente con el desarrollo de este proyecto se permite:

- El control de ventas, reduciendo la carga de trabajo y minimizando tiempos.
- El manejo del ticket que se le entrega al cliente al momento de pagar su cuenta, ya que anteriormente no se le daba ningún comprobante de pago.
- Se puede también manejar los diferentes proveedores que tiene la empresa y consultar de todas las ventas que se realizaron en determinada fecha.
- Control de acceso de información dependiendo a que tipo de usuario se refiera.
- Información sobre los datos personales del personal laborando.

En general, proporciona información valiosa para la administración y toma de decisiones de la empresa.

JUSTIFICACIÓN

El proyecto nace de la necesidad de llevar un control sistematizado de las operaciones del negocio.

Anteriormente el registro de las ventas se realizaban de manera manual, mediante reportes diarios de las mismas; en cuanto al inventario, no existía un procedimiento formal para el control de éste, lo que deriva en problemas tales como el desabastecimiento de algunos productos indispensables para el buen funcionamiento de la empresa.

Tomando lo anterior como referencia: La automatización permite ejecutar e integrar en forma eficiente las tareas del área comercial, gestionando y controlando el circuito de flujo de trabajo de compras, una reducción de los costos de abastecimiento, solicitando solo lo que se necesita cuando se necesita. Además será posible la evaluación de cotizaciones de proveedores y mantener el stock de los productos en niveles óptimos, reduciendo costos de inventarios, capital inmovilizado y riesgos de pérdidas por casualidad y obsolescencia.

La información generada por un sistema permite un análisis de ventas y formación de estrategias para mejorar la gestión administrativa optimizando el control de caja, ingresos y egresos; integrando los movimientos de ventas, compras y pagos. Tomando en cuenta el cierre más rápido de operaciones diarias de caja.

Las funciones del sistema permitirán destinar el tiempo a analizar la información y no a recopilarla, lo que facilitaría para una mejor toma de decisiones para mantener e incluso aumentar el éxito de la empresa.



Junio 2007

OBJETIVO GENERAL

Crear un software de control de ventas e inventarios de la empresa "El Rincón del Café", que permita lograr una mejor administración de recursos y actividades.

OBJETIVOS ESPECÍFICOS

- Reducir drásticamente el tiempo en las operaciones.
- Aumentar considerablemente la exactitud y confiabilidad.
- Ayudar a reducir pérdidas y merma de productos.
- Permitir identificar áreas de oportunidad.
- Agilizar la administración.

CARACTERIZACIÓN DEL ÁREA EN QUE SE PARTICIPÓ

Nombre: El Rincón del Café

Ubicación: Ave. Independencia No. 89

Col. Centro Hgo. Del Parral, Chih.

Horario: Martes a Domingo de 4 a 12 p.m.

El Rincón del Café se dedica principalmente a la elaboración de bebidas preparadas utilizando como base el café, por lo que se considera "Barra de Cafés". Además ofrece otras opciones tales como: botanas, platillos fuertes, postres, bebidas nacionales, cócteles, etc. Esto permite abarcar un mercado más amplio.

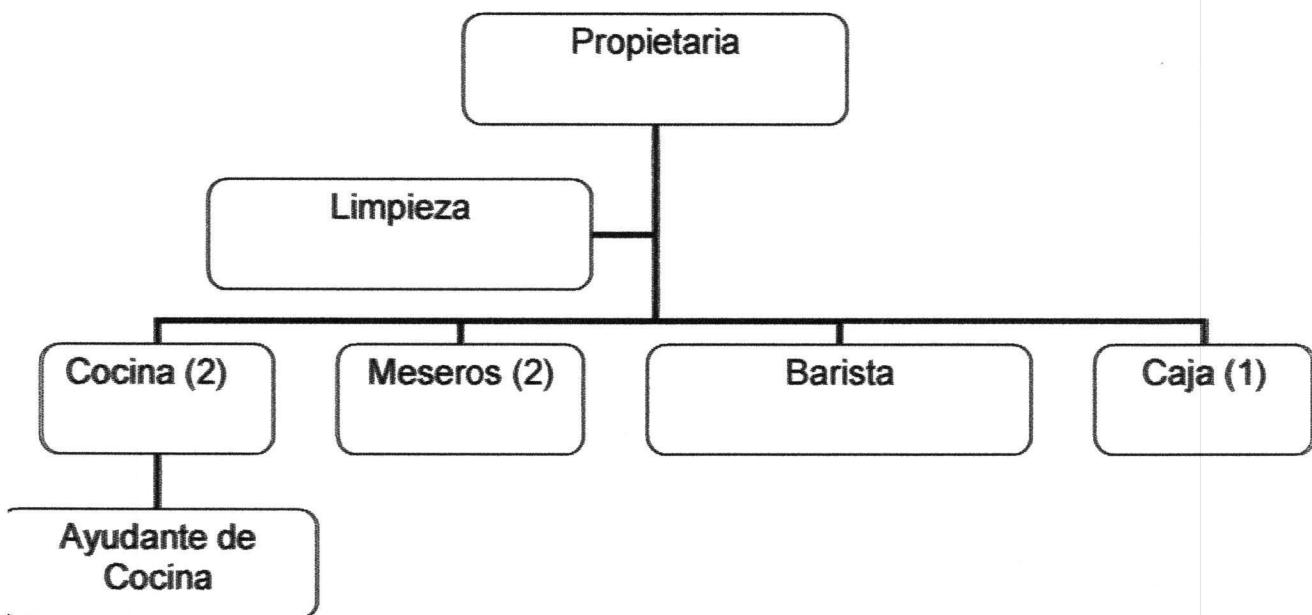
Esta microempresa se creó en Diciembre de 1998, como muchos negocios, nace alrededor de una conversación acompañada por una taza de café, donde un grupo de jóvenes vieron la necesidad de un lugar de esparcimiento cuyo producto principal fuera el café en diferentes modalidades, y en donde se pudiera pasar un rato ameno. En ese tiempo no había ninguna cafetería en la ciudad, solamente restaurantes que ofrecían otro tipo de productos.

Hoy en día ya no es una sociedad, quedó en manos de una sola persona, el concepto ha mejorado; a pesar de que hay otros lugares que dicen ser cafés aun no hay ninguno que ofrezca la variedad y sabor que se ofrece aquí.

Cuenta con un ambiente tranquilo, en donde se puede platicar sin tener que "gritar". Además se ofrecen algunas bebidas preparadas con licor, y el fin de semana hay música en vivo por las noches.

El local es una casa antigua situada en una de las principales calles la ciudad, lo que le da un aire muy especial. Los muros son muy anchos, tal y como se construían al principio del siglo pasado y en su fachada aun cuenta con un poco de cantera que enmarca la puerta principal y la ventana. Los clientes comentan que es un lugar muy "acogedor", probablemente porque conserva un estilo antiguo.

ORGANIGRAMA DE LA EMPRESA



PROBLEMAS A RESOLVER

Entre los problemas que se generaban por carecer de un manejo eficiente, ya que no se contaba con un sistema de Punto de Venta eran:

1. Cobros incorrectos por no realizar los cálculos adecuados o por no asignar precios correctos.
2. Excesivo consumo de tiempo al realizar el corte diario, ya que los cálculos se realizan manualmente y en ocasiones es necesario rectificarlos varias veces.
3. Cuando varias mesas piden la cuenta el tiempo de respuesta no es el óptimo.
4. Faltantes por entregar importes incorrectos al enviar el cambio de la cuenta al cliente.
5. Deficiente control del inventario de insumos, lo que deriva en desabastecimiento de los mismos.
6. Las mermas no son evidentes, por lo que no se conocen los importes apropiados de perdidas por este concepto.
7. En ocasiones se presentan excesos de existencias de inventarios por no tener los datos actualizados.

ALCANCES Y LIMITACIONES

ALCANCES

Es importante hacer notar, que el sistema fue desarrollado pensando en cubrir necesidades relacionadas con el área de inventario y ventas. Considerando que las ventas son parte fundamental de cualquier empresa, a partir de la instalación de "SyCAF" es posible tener un panorama más definido y claro de la situación que presentan las transacciones en este rubro; permitiendo una mejor toma de decisiones.

Por otra parte, establecer un control de inventario adecuado representa un enlace importante con las ventas; y esto se logró a través del vínculo creado entre el punto de venta y el descuento automático de todos los ingredientes utilizados en cada uno de los productos. No solamente es importante obtener datos relacionados con las existencias, sino también estar seguros de que estas son modificadas en relación a los productos vendidos.

Independientemente del contenido del sistema y de las ventajas que proporciona en las áreas mencionadas, cuenta con un valor agregado ya que a través del mismo se otorga un mejor servicio al cliente; y esto le permite a la empresa tener una ventaja competitiva.

Cabe señalar que al realizar un proyecto de este tipo permitió ampliar el panorama respecto a las tecnologías de información, conocer y utilizar herramientas diferentes a las comúnmente conocidas.

LIMITACIONES

El sistema fue creado considerando que para su utilización solamente trabajará de modo monousuario, ya que las posibilidades actuales de la empresa solo permitieron realizarlo de esta manera.

En relación a los requerimientos mínimos del sistema, fue necesario adaptarse al equipo con el que contaba la empresa; cabe señalar que afortunadamente no hubo problema en este sentido.

Otra de las limitantes de este sistema es que no va a estar conectado en red, ya que solo cuenta con un local. Inicialmente se contempló la posibilidad de instalar una impresora en la cocina, con el fin de que el pedido fuera surtido sin la necesidad de llevar la comanda a las cocineras; sin embargo esto no fue posible por varias razones una de las cuales era la inversión requerida de equipo.

ESTUDIO DE FACTIBILIDAD

Es importante valorar el proyecto desde tres expectativas:

- Factibilidad Operacional.
- Factibilidad Económica.
- Factibilidad Técnica.

Es necesario identificar los objetivos de la empresa y determinar si el proyecto está orientado hacia el alcance de esos objetivos.

La factibilidad técnica permite determinar si la empresa cuenta con los recursos tecnológicos necesarios para satisfacer los requerimientos de lo que se está solicitando en el proyecto.

Si los recursos no existen, tal vez sea posible que la empresa los adquiera, es aquí donde el estudio de factibilidad económica aparece, pero incluye aspectos de otros costos aparte del costo del hardware y software: costos como el de realizar el análisis completo del sistema, el costo de tiempo de los empleados del negocio.

Finalmente, después de que la opinión respecto de los recursos técnicos y económicos sea positiva en cuanto factibilidad se refiere, existe la factibilidad operacional, la cual depende de los recursos humanos, tal vez la parte más difícil de administrar en un proyecto. La factibilidad operacional del proyecto involucra aspectos como el analizar si el sistema será utilizado una vez implantado, así como el grado de resistencia al cambio que puede existir ante los posibles usuarios.

MARCO TEÓRICO

Investigar es un proceso sistemático y ordenado que no puede hacerse a la ligera. Para que una investigación sea exitosa, se deben seguir ciertos procedimientos y técnicas que facilitan y ayudan a obtener el máximo provecho y éxito en cada uno de los estudios que se realizan.

Cada investigación tiene sus particularidades, de modo que no se deben imponer estrictas reglas que limiten la libertad a la hora de investigar; sin embargo, es necesario seguir ciertas normas y recomendaciones que son herramientas para desarrollar estudios ordenados y útiles ya sea en el ámbito de las ciencias exactas o en el ámbito social.

Elaborar correctamente el marco teórico es fundamental para realizar una buena investigación, ya que el marco teórico (también llamado marco referencial o marco conceptual) tiene el propósito de dar a la investigación un sistema coordinado y coherente de conceptos y proposiciones que permitan abordar el problema. De éste dependerá el resultado del trabajo.

Es también de suma importancia definir el alcance que tendrá la investigación que se va a realizar, que es precisamente el tipo de estudio a desarrollar y puede tratarse de una investigación exploratoria, descriptiva, correlacional o explicativa.

Para la realización de sistema de información es necesario saber los elementos que lo componen y ver que es en sí un sistema de información.

SISTEMA DE INFORMACIÓN: Conjunto de funciones o componentes interrelacionados que forman un todo, es decir, obtiene, procesa, almacena y distribuye información para apoyar la toma de decisiones y el control en una organización.

Las principales actividades que un sistema de información realiza son las siguientes:

- Entradas
- Procesamiento
- Almacenamiento
- Salidas

EQUIPO COMPUTACIONAL: El hardware es esencial para que un sistema de información automatizado pueda ser operado por los usuarios. El equipo permite la comunicación dinámica usuario-sistema logrando así que el sistema logre su objetivo para el que fue creado.

Comprendiendo así lo que es un sistema de información, para desarrollar el sistema propuesto en este proyecto, se utilizó una metodología la cual es llamada **Método del Ciclo de Vida**.

¹“El método del ciclo de vida para el desarrollo de sistemas (SDLC) es el conjunto de actividades que los analistas, diseñadores y usuarios realizan para desarrollar e implantar un sistema de información.”

Las etapas del ciclo de vida de un sistema de información son las siguientes:

¹ Análisis y Diseño de Sistemas
Kenneth E. Kendall
Págs. 8-13

1. Planificación del proyecto
2. Definición del sistema
3. Recolección y Análisis de requisitos
4. Diseño de la base de datos
5. Selección de manejador de base de datos
6. Diseño del sistema
7. Implementación
8. Conversión y Carga de datos
9. Prueba

1). PLANIFICACIÓN DEL PROYECTO.

Esta etapa conlleva la planificación de cómo se pueden llevar a cabo las etapas del ciclo de vida de la manera más eficiente. Hay tres componentes principales:

- El trabajo que se realizará.
- Los recursos para llevarlo a cabo.
- El capital para cubrir lo anterior.

Como apoyo a esta etapa, se necesita un modelo de datos corporativo en donde se muestran las actividades principales de la empresa, sus relaciones y lugar

donde se identifica las áreas funcionales. Normalmente, este modelo de datos se representa mediante un diagrama entidad relación. En este modelo se tiene que mostrar también que datos comparten las distintas áreas funcionales de la empresa.

La planificación de la base de datos también incluye el desarrollo de estándares que especifiquen como realizar la recolección de datos, como especificar su formato, que documentación será necesaria y como se va a llevar a cabo el diseño y la implementación. El desarrollo y el mantenimiento de los estándares puede llevar bastante tiempo, pero si están bien diseñados, son una base para el personal informático en formación y para medir calidad, además, garantizar que el trabajo se ajusta a unos patrones, independientemente de las habilidades y experiencia del diseñador.

2). DEFINICIÓN DEL SISTEMA.

En esta etapa se especifica el ámbito y los límites de la aplicación de la base de datos, así como con que otros sistemas interactúan. También se requiere la tarea de determinar quienes son los usuarios y las áreas de aplicación.

3). RECOLECCIÓN Y ANÁLISIS DE REQUISITOS.

En esta etapa se recogen y analizan los requerimientos de los usuarios y de las áreas de aplicación. Esta información se puede recoger en varias formas:

- Entrevistando al personal de la empresa, concretamente, a aquellos que son considerados expertos en las áreas de interés.
- Observando el funcionamiento de la empresa
- Examinando documentación, sobre todo aquella que se utiliza para obtener o visualizar información
- Utilizando cuestionarios para recoger información de grandes grupos de usuarios
- Utilizando la experiencia adquirida en el diseño de sistemas similares

Esta etapa tiene como resultado un conjunto de documentos con las especificaciones de requisitos de los usuarios, en donde se describen las operaciones que se realizan en la empresa desde distintos puntos de vista.

La información recogida se debe estructurar utilizando técnicas de especificación de requisitos, como por ejemplo técnicas de análisis, diseño estructurado y diagrama de flujo de datos. También las herramientas CASE pueden proporcionar una asistencia automatizada que garantice que los requisitos son completos y consistentes.

4). DISEÑO DE LA BASE DE DATOS.

Esta etapa consta de 3 fases:

1. Consiste en la producción de un esquema conceptual, que es independiente de todas las consideraciones físicas.
2. Este modelo se refina después en un sistema lógico eliminando las construcciones que no se pueden representar en el modelo de base de datos escogido.

3. El esquema lógico se traduce en un esquema físico para el manejador de base de datos elegido. La fase de diseño físico considera las estructuras de almacenamiento y los métodos de acceso necesarios para proporcionar un acceso eficiente de la base de datos.

Los objetivos del diseño de la base de datos son:

- Representar los datos que requieren los principales áreas de aplicación, usuarios y representar las relaciones entre dichos datos.
- Proporcionar un modelo de datos que soporte las transacciones que se vaya a realizar sobre los datos.
- Especificar un esquema que alcance las prestaciones requeridas para el sistema.

Hay varias estrategias a seguir para realizar el diseño:

1. De abajo a arriba.
2. De arriba a abajo.
3. De adentro a afuera.
4. Estrategia mixta.

1. Estrategia de Abajo a Arriba: parte de todos los atributos y los va agrupando en entidades y relaciones. Es apropiada cuando la base de datos es simple y con pocos atributos.

2. Estrategia de Arriba a Abajo: es más apropiada cuando se trata de base de datos complejas. Se comienza con un esquema con entidades de bajo nivel, atributos y relaciones.

3. Estrategia de Adentro a Afuera: es similar a la primera, pero difiere en su parte de los conceptos principales y se va extendiendo el esquema para considerar también otros conceptos asociados como los que se han identificado en primer lugar.

4. Estrategia Mixta: utiliza ambas estrategias. Se obtiene un esquema inicial de alto nivel, se divide en partes, y de cada parte se obtiene un subesquema, estos subesquemas se integran después para obtener el modelo final.

5). SELECCIÓN DEL MANEJADOR DE BASE DE DATOS.

Si no se dispone de un manejador de base de datos o el que ya se encuentra obsoleto, se debe escoger un manejador de base de datos que sea adecuado para el sistema de información. Esta selección se debe hacer en cualquier momento ante el diseño lógico.

6). DISEÑO DEL SISTEMA DE INFORMACIÓN.

En esta etapa se diseñan los módulos del sistema de información que usara y procesara la base de datos. Esta sección y el diseño de la base de datos casi siempre son paralelos. En mayor parte de los casos no se puede finalizar el diseño de los módulos hasta que sea terminado con el diseño de la base de datos. También hay que asegurarse de que toda la funcionalidad especificada en los requisitos del usuario esté correcta, además habrá que diseñar las interfaces, aspecto muy importante que se suele ignorar. El sistema debe ser fácil de aprender, fácil de usar, ser directo y disponible a corregir en caso de errores de usuarios. Si la interfaz no tiene estas características el sistema ocasionará problemas.

El diseño de un sistema de información produce los detalles que establecen la forma en la que el sistema cumplirá con los requerimientos identificados durante la fase de análisis. Los especialistas en sistemas se refieren, con frecuencia, a esta etapa como diseño lógico en contraste con la del desarrollo del software, a la que denominan diseño físico.

Por otro lado la base de datos existe para dar soporte a las aplicaciones, por lo que habrá una realimentación desde el diseño de las aplicaciones al diseño de la base de datos.

Este proceso permite que quienes diseñan, implementan el sistema sepan si han interpretado correctamente las necesidades de los usuarios. Otra ventaja de los prototipos es que se construyen rápidamente.

Esta etapa es imprescindible cuando el sistema que se va a implementar tiene un gran costo, alto riesgo o utiliza nuevas tecnologías.

7). IMPLEMENTACIÓN.

En esta etapa se crean las definiciones de la base de datos a nivel conceptual, externo e interno, así como programas de aplicación. La implementación de la base de datos se realiza mediante las sentencias del lenguaje de definición de datos (LLD) del manejador de base de datos escogido. Estas sentencias se encargan de crear el esquema de la base de datos, los ficheros en donde se almacenaran los datos y las vistas de los usuarios.

Los programas de aplicación se implementan utilizando lenguajes de tercera o cuarta generación. Partes de estas aplicaciones son transaccionales sobre la base de datos, que se implementan mediante el lenguaje de manejo de datos (LMD) del manejador de base de datos. Las sentencias de este lenguaje se

pueden embeber en un lenguaje de programación anfitrión como Visual Basic, Delphi, C, C++, Java, C Builder.

En esta etapa, también se implementan los menús, los formularios para la introducción de datos y los informes de visualización de datos. Para ello, el manejador de base de datos puede disponer de lenguajes de cuarta generación que permiten el desarrollo rápido de aplicaciones mediante lenguajes de consultas no procedimentales, generadores de informes, generador de formularios, generadores de gráficos y aplicaciones.

También se implementan en esta etapa todos los controles de seguridad e integridad. Algunos de estos controles se pueden implementar mediante el LLD y otros, puede que haya que implementarlos mediante utilidades de manejador de base de datos o en su defecto por programas de aplicación.

8). CONVERSIÓN Y CARGA DE DATOS.

Esta etapa es necesaria cuando se está reemplazando un sistema antiguo. Los datos se cargan desde el viejo al nuevo directamente o, si es necesario, se convierten al formato que requiera en un nuevo manejador de base de datos y posteriormente se cargan. Si es posible, los programas de aplicación del sistema antiguo también se convierten para que se puedan utilizar en el sistema nuevo.

9. PRUEBAS.

En esta etapa se prueba y valida el sistema con los requisitos especificados por el usuario. Para ello se debe diseñar una batería de test

con datos reales, que se pueden llevar a cabo de manera metódica y rigurosa. Es importante darse cuenta que la fase de prueba no sirve para demostrar que no hay fallos, sirve para encontrarlos. Si la fase de prueba se lleva a cabo correctamente, se cubrirá los errores en los programas de aplicación y en la estructura de la base de datos. Además, se demostrará que los programas parecen trabajar tal y como se especificaba en los requisitos y que las prestaciones deseadas parecen obtenerse.

Se alimentan como entradas conjunto de datos de prueba para su procesamiento y después se examinan los resultados.

Por último, en las pruebas de podrá hacer una medida de la fiabilidad y la calidad del software desarrollado.

ENTREVISTA

²La entrevista tiene como objetivo obtener la opinión del entrevistado y la debilidad acerca del sistema actual, los objetivos de la organización, los personales y procedimientos informales.

Para planear la entrevista se deben tomar en cuenta los 5 pasos que consisten en: recopilar e investigar todo acerca de la empresa y la persona a la que se va a entrevistar, también establecer los objetivos de la entrevista así como decidir a quienes son las personas a las cuales se debe entrevistar, preparar al entrevistado, es decir, hacer una cita previa antes de la entrevista, y por último decidir por el tipo de preguntas y su estructura.

Las entrevistas pueden clasificarse como estructuradas o no estructuradas. Las entrevistas no estructuradas utilizan un formato pregunta-respuesta y son apropiadas cuando el analista desea adquirir información general acerca de un sistema. Este formato anima a los entrevistados a compartir sus sentimientos, ideas y creencias.

Por otro lado, las entrevistas estructuradas utilizan preguntas estándar es un formato de respuesta abierta o cerrada. El primero permite que el entrevistado de respuesta a las preguntas con sus propias palabras; el segundo utiliza un conjunto anticipado de respuestas.²

Las entrevistas se pueden usar de acuerdo a su estructura y pueden ser 3 tipos:

² Análisis y Diseño de Sistemas de Información
James A. Senn
Págs. 133-140

1. ENTREVISTA TIPO PIRAMIDAL.

Se debe usar una estructura de pirámide si se considera que el entrevistador necesita ambientarse en el tema. También es útil si el entrevistador parece ser que resiste entrar al tema.

La utilización de una estructura piramidal para la secuencia de las preguntas también es útil cuando se quiere una determinación final acerca del tema.

2. ENTREVISTA DE TIPO EMBUDO.

El entrevistador toma un enfoque deductivo, comenzando con preguntas generales y abiertas y estrechando las respuestas posibles usando preguntas abiertas.

La estructura de embudo proporciona una forma fácil y no intimidante para comenzar una entrevista. Un beneficio del uso de una estructura de embudo es que el organizar la entrevista en tal forma puede elucidar tanta información detallada que son innecesarias las secundas largas de preguntas cerradas y averiguaciones.

3. ENTREVISTA TIPO ROMBO.

Comienza en una forma muy específica, luego examina temas generales y por último llegar a una conclusión muy específica.

La estructura de rombo combina la fuerza de los dos enfoques. La ventaja principal del uso de una estructura de rombo es conservar el interés y la atención del entrevistado por medio de una diversidad de preguntas

DIAGRAMAS DE FLUJO

El diagrama de flujo representa la forma más tradicional y duradera para especificar los detalles algorítmicos de un proceso. Se utiliza principalmente en programación, economía y procesos industriales; estos diagramas utilizan una serie de símbolos con significados especiales. Son la representación gráfica de los pasos de un proceso, que se realiza para entender mejor al mismo. Son modelos tecnológicos utilizados para comprender los rudimentos de la programación lineal.

Otra definición del diagrama de flujo es:

"Es un esquema para representar gráficamente un algoritmo. Se basan en la utilización de diversos símbolos para representar operaciones específicas. Se les llama diagramas de flujo porque los símbolos utilizados se conectan por medio de flechas para indicar la secuencia de operación. Para hacer comprensibles los diagramas a todas las personas, los símbolos se someten a una normalización; es decir, se hicieron símbolos casi universales, ya que, en un principio cada usuario podría tener sus propios símbolos para representar sus procesos en forma de Diagrama de flujo. Esto trajo como consecuencia que sólo aquel que conocía sus símbolos, los podía interpretar. La simbología utilizada para la elaboración de diagramas de flujo es variable y debe ajustarse a un patrón definido previamente."

VENTAJAS DE LOS DIAGRAMAS DE FLUJO

- Favorecen la comprensión del proceso a través de mostrarlo como un dibujo. El cerebro humano reconoce fácilmente los dibujos. Un buen diagrama de flujo reemplaza varias páginas de texto.

- Permiten identificar los problemas y las oportunidades de mejora del proceso.
Se identifican los pasos redundantes, los flujos de los reprocesso, los conflictos de autoridad, las responsabilidades, los cuellos de botella, y los puntos de decisión.
- Muestran las interfaces cliente-proveedor y las transacciones que en ellas se realizan, facilitando a los empleados el análisis de las mismas.
- Son una excelente herramienta para capacitar a los nuevos empleados y también a los que desarrollan la tarea, cuando se realizan mejoras en el proceso.

SÍMBOLOS SE EMPLEAN EN LOS DIAGRAMAS DE FLUJO

Los símbolos tienen significados específicos y se conectan por medio de flechas que indican el flujo entre los distintos pasos o etapas.

No es indispensable usar un tipo especial de símbolos para crear un diagrama de flujo, pero existen algunos ampliamente utilizados por lo que es adecuado conocerlos y utilizarlos, ampliando así las posibilidades de crear un diagrama más claro y comprensible para crear un proceso lógico y con opciones múltiples adecuadas.

NOMBRE	SÍMBOLO	FUNCIÓN
Terminal		Representa el inicio y fin de un programa. También puede representar una parada o interrupción programada que sea necesaria realizar en un programa.
Entrada / salida		Cualquier tipo de introducción de datos en la memoria desde los periféricos o registro de información procesada en un periférico.
Proceso		Cualquier tipo de operación que pueda originar cambio de valor, formato o posición de la información almacenada en memoria, operaciones aritméticas, de transformaciones, etc.
Decisión		Indica operaciones lógicas o de comparación entre datos (normalmente dos) y en función del resultado de la misma determina (normalmente si y no) cual de los distintos caminos alternativos del programa se debe seguir

Indicador de dirección o línea de flujo		Indica el sentido de la ejecución de las operaciones
Salida		Se utiliza en ocasiones en lugar del símbolo de salida. El dibujo representa un pedazo de hoja. Es usado para mostrar datos o resultados.

DESARROLLO DEL DIAGRAMA DE FLUJO

Las siguientes son acciones previas a la realización del diagrama de flujo:

- Identificar a los participantes de la reunión donde se desarrollará el diagrama de flujo. Deben estar presentes el dueño o responsable del proceso, los dueños o responsables del proceso anterior y posterior y de otros procesos interrelacionados, otras partes interesadas.
- Definir que se espera obtener del diagrama de flujo.
- Identificar quién lo empleará y cómo.
- Establecer el nivel de detalle requerido.
- Determinar los límites del proceso a describir.

Los pasos a seguir para construir el diagrama de flujo son:

- Establecer el alcance del proceso a describir. De esta manera quedará fijado el comienzo y el final del diagrama. Frecuentemente el comienzo es la salida del proceso previo y el final la entrada al proceso siguiente.
- Identificar y listar las principales actividades/subprocesos que están incluidos en el proceso a describir y su orden cronológico.
- Si el nivel de detalle definido incluye actividades menores, listarlas también.
- Identificar y listar los puntos de decisión.
- Construir el diagrama respetando la secuencia cronológica y asignando los correspondientes símbolos.
- Asignar un título al diagrama y verificar que esté completo y describa con exactitud el proceso elegido.

REGLAS DE LOS DIAGRAMAS DE FLUJO

- Debe de indicar claramente dónde inicia y dónde termina el diagrama.
- Cualquier camino del diagrama debe de llevarte siempre a la terminal de fin.
- Organizar los símbolos de tal forma que siga visualmente el flujo de arriba hacia abajo y de izquierda a derecha.

- No usar lenguaje de programación dentro de los símbolos.
- Las líneas deben ser verticales u horizontales, nunca diagonales.

DICCIONARIO DE DATOS

Un diccionario de datos es un conjunto de metadatos que contiene las características lógicas de los datos que se van a utilizar en el sistema que se programa, incluyendo nombre, descripción, alias, contenido y organización.

Estos diccionarios se desarrollan durante el análisis de flujo de datos y ayuda a los analistas que participan en la determinación de los requerimientos del sistema, su contenido también se emplea durante el diseño del proyecto.

Identifica los procesos donde se emplean los datos y los sitios donde se necesita el acceso inmediato a la información, se desarrolla durante el análisis de flujo de datos y auxilia a los analistas que participan en la determinación de los requerimientos del sistema, su contenido también se emplea durante el diseño.

En un diccionario de datos se encuentra la lista de todos los elementos que forman parte del flujo de datos de todo el sistema. Los elementos más importantes son flujos de datos, almacenes de datos y procesos. El diccionario de datos guarda los detalles y descripción de todos estos elementos.

HERRAMIENTAS DE DESARROLLO

Las herramientas de desarrollo, permiten a los desarrolladores realizar sus proyectos en poco tiempo. También proporcionan un alto grado de personalización, lo que da lugar a un conjunto mayor de soluciones escalables. Asimismo, las avanzadas funciones de depuración de errores permiten a los desarrolladores crear un producto de calidad mediante una sola interfaz.

¿QUÉ ES DELPHI?

³Delphi es un entorno de desarrollo de software diseñado para la programación de propósito general con énfasis en la programación visual. En Delphi se utiliza como lenguaje de programación una versión moderna de Pascal llamada Object Pascal. En sus diferentes variantes, permite producir archivos ejecutables para Windows, Linux y la plataforma .NET".

USO Y VARIANTES

Un uso habitual de Delphi es el desarrollo de aplicaciones visuales y de bases de datos cliente-servidor y multicapas. Debido a que es una herramienta de propósito múltiple, se usa también para proyectos de casi cualquier tipo, incluyendo aplicaciones de consola, aplicaciones de web. En la última versión Developer Studio 2006 incluye en el mismo entorno de desarrollo los lenguajes:

EL LENGUAJE DE PROGRAMACIÓN

Object Pascal expande las funcionalidades del Pascal estándar:

1. Soporte para la programación orientada a objetos (POO) también existente desde Turbo Pascal 5.5, pero más evolucionada en cuanto a:
 - Encapsulación: declarando partes privadas, protegidas, públicas y publicadas de las clases
 - Propiedades: concepto nuevo que luego han adaptado muchos otros lenguajes. Las propiedades permiten usar la sintaxis de asignación para setters y getters.
 - Simplificación de la sintaxis de referencias a clases y punteros.

³ <http://es.wikipedia.org/wiki/Delphi>

2. Soporte para manejo estructurado de excepciones, mejorando sensiblemente el control de errores de usuario y del sistema.
3. Programación activada por eventos (event-driven), posible debido a la técnica de delegación de eventos. Esta técnica permite asignar el método de un objeto para responder a un evento lanzado sobre otro objeto.

COMPONENTES

Delphi dio una implementación muy buena a la idea del uso de componentes, que son piezas reutilizables de código (clases) que pueden interactuar con el entorno integrado de desarrollo (EID) en tiempo de diseño y de ejecución. Desde un enfoque más específico de la herramienta, se catalogan como componentes todos aquellos objetos que heredan de la clase `TComponent`, donde se implementa la funcionalidad necesaria para interactuar con el entorno de desarrollo, la carga dinámica desde streams y la liberación de memoria mediante una jerarquía. Una gran parte de los componentes disponibles para Delphi son controles (derivados de `TControl`), que encapsulan los elementos de interacción con el usuario como botones, menús, barras de desplazamiento, etcétera.

Delphi incluye una biblioteca de clases bien diseñada denominada Biblioteca de Componentes Visuales (VCL) y, en sus versiones 6 y 7, una jerarquía multiplataforma paralela denominada CLX.

Además de poder utilizar en un programa estos componentes estándar (botones, conjuntos de datos, etc.), es posible crear nuevos componentes o mejorar los ya existentes, extendiendo la funcionalidad de la herramienta.

EVENTOS

Delphi permite de manera sencilla ejecutar segmentos de código en respuesta a acciones o eventos (sucesos) que ocurren durante el tiempo que un programa se ejecuta.

Los eventos pueden generarse debido a la recepción de señales desde elementos de hardware como el ratón o el teclado, o pueden producirse al realizar alguna operación sobre un elemento de la propia aplicación.

BASE DE DATOS

Una de las principales características y ventajas de Delphi es su capacidad para desarrollar aplicaciones con conectividad a bases de datos de diferentes fabricantes. El programador de Delphi cuenta con una gran cantidad de componentes para realizar la conexión, manipulación, presentación y captura de los datos, algunos de ellos liberados bajo licencias de código abierto o gratuito. Estos componentes de acceso a datos pueden enlazarse a una gran variedad de controles visuales, aprovechando las características del lenguaje orientado a objetos, gracias al polimorfismo.

En la paleta de componentes pueden encontrarse varias pestañas para realizar una conexión a bases de datos usando diferentes capas o motores de conexión.

Hay motores que permiten conectarse a bases de datos de diferentes fabricantes tales como BDE, DBExpress o ADO, que cuentan con manejadores para los formatos más extendidos.

BORLAND DATABASE ENGINE (BDE)

Es un motor de conexión a bases de datos de uso bastante amplio y que permite manejar bases de datos de escritorio como dBase, Foxpro y Paradox, además de ofrecer la capacidad para conectarse a servidores SQL locales y remotos. Su uso, va siendo cada vez menor, debido a la pobre gestión de memoria que realiza,

sustituyéndolo por componentes más actualizados y especializados como Direct Oracle Access Components (DOAC) o DBExpress, esto sumado a la fiabilidad que están presentando.

DESARROLLO VISUAL

Como entorno visual, la programación en Delphi consiste en diseñar los formularios que componen al programa colocando todos sus controles (botones, etiquetas, campos de texto, etc.) en las posiciones deseadas, normalmente usando un ratón. Luego se asocia código a los eventos de dichos controles y también se pueden crear módulos de datos, que regularmente contienen los componentes de acceso a datos y las reglas de negocio de una aplicación.

ENTORNO INTEGRADO DE DESARROLLO (EID)

Es el ambiente de desarrollo de programas de Delphi. Se trata de un editor de formularios (que permite el desarrollo visual), un potente editor de textos que resalta la sintaxis del código fuente, la paleta de componentes y el depurador integrado, además de una barra de botones y un menú que nos permite la configuración de la herramienta y la gestión de proyectos. En las ediciones Client/Server y Enterprise el EID también ofrece integración con una herramienta de control de versiones (PVCS).

DEPURADOR INTEGRADO

Es una potente característica que nos permite establecer puntos de ruptura (breakpoints), la ejecución paso a paso de un programa, el seguimiento de los valores de las variables y de la pila de ejecución, así como la evaluación de expresiones con datos de la ejecución del programa. Con su uso, un programador experimentado puede detectar y resolver errores lógicos en el funcionamiento de un aplicativo desarrollado con Delphi. En las ediciones Client/Server y Enterprise se

añade la opción de depuración a programas corriendo en equipos remotos (remote debugging), lo que posibilita el uso de todas las características del depurador con un programa ejecutándose en su entorno normal de trabajo y no en el ordenador del programador (en donde muchas veces no ocurren los errores).

MÉTODOS UTILIZADOS

Para la adquisición de información acerca del modulo de ventas se llevó a cabo una técnica que es de vital importancia para el desarrollo de este sistema:

ENTREVISTA. Esta técnica fue la parte modular de la recopilación de información, ya que mediante ella, se logró obtener la mayor parte de esta. De hecho en este método se entrevistó únicamente a los meseros y a la encargada del lugar, por la razón de que ellos se comisionan de realizar las tareas antes de la implantación de dicho sistema.

Para obtener la información fue necesario realizar los siguientes cuestionamientos:

¿Qué es lo que hace?

¿Cómo se hace?

¿Con qué frecuencia se presenta?

¿Qué tan grande es el volumen de transacciones o decisiones?

¿Cuál es el grado de eficiencia con el que se efectúan las tareas?

¿Existe algún problema? ¿Qué tan serio es? ¿Cuál es la causa que lo origina?

DESCRIPCIÓN Y PROCEDIMIENTOS DE LAS ACTIVIDADES REALIZADAS

Las actividades que se llevaron a cabo durante la elaboración del desarrollo del sistema fueron las siguientes.

1. Análisis del sistema y recopilación de la información.
 - a) Diagramas de flujo
 - b) Elección y aprendizaje de herramientas de desarrollo
 - c) Diccionario de datos
2. Diseño del sistema
3. Desarrollo del sistema
4. Pruebas
5. Elaboración de manuales
6. Implantación

1. ANÁLISIS DEL SISTEMA Y RECOPIACIÓN DE INFORMACIÓN

El propósito de esta actividad fue identificar las técnicas utilizadas para recolectar la información que se requirió para elaborar el sistema, para esto se utilizó la entrevista, debido a que ya se conocían algunos datos de la empresa, a las personas responsables y sobre todo, acerca del proyecto planteado como solución a los problemas de información que se presentaban en la empresa. (Ver anexo 1)

Entrevista de preguntas generales.

Tal vez la forma más utilizada para obtener información es la entrevista, solo es necesario determinar el tipo de entrevista que será aplicada en la empresa y la

persona que será entrevista, la cual debe ser una persona representativa dentro de la empresa, es decir, una persona clave.

Las entrevistas de preguntas abiertas y generales dan pauta para planear una entrevista en las que las preguntas sean acerca de detalles o información más específica que en la pregunta abierta pudieron escapar a la respuesta del entrevistado, es decir, preguntas más concretas acerca de algún tema, proceso o hecho de la empresa.

En la entrevista que se le hizo a la encargada del lugar se pretendía conocer como se estaba manejando el inventario, y notar como se llevaba a cabo la elaboración de dichos productos.

En la segunda entrevista que fue enfocada hacia los meseros se desea saber como toman la orden del cliente, que mecanismo usan ellos para hacer dicha actividad, y como logran ellos tener un control sobre las mesas para poder llevar acabo el orden de las mesas. (Ver anexo 2)

De esta forma se detectó la problemática existente, al haber duplicidad en los pedidos, perdida de dinero, que las comandas lleguen a perderse, no poder realizar el corte de caja bien y no contar con reportes para mostrar las ventas que se realizan.

El objetivo de haber realizado dicha entrevista fue:

- Conocer cual es la problemática del lugar.
- Cómo se llevan a cabo los procesos de cobro.
- Determinar la importancia de los reportes que se generan con la realización de las actividades para mejorar la administración del negocio.

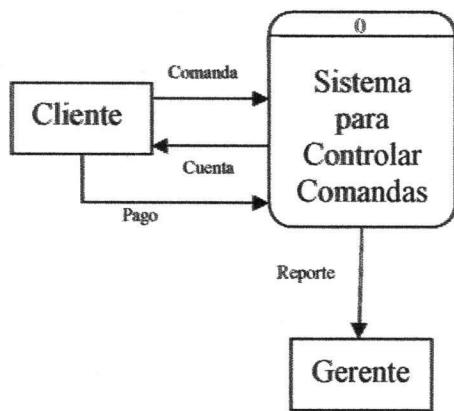
- Detectar la problemática que se tiene.
- Determinar si es necesario instalar un sistema de información que cubra las necesidades que se tienen.

A) ELABORACIÓN DE DIAGRAMAS DE FLUJO

Para presentar el flujo de la información de acuerdo a las actividades se realizaron los diagramas de: (Ver anexo 3).

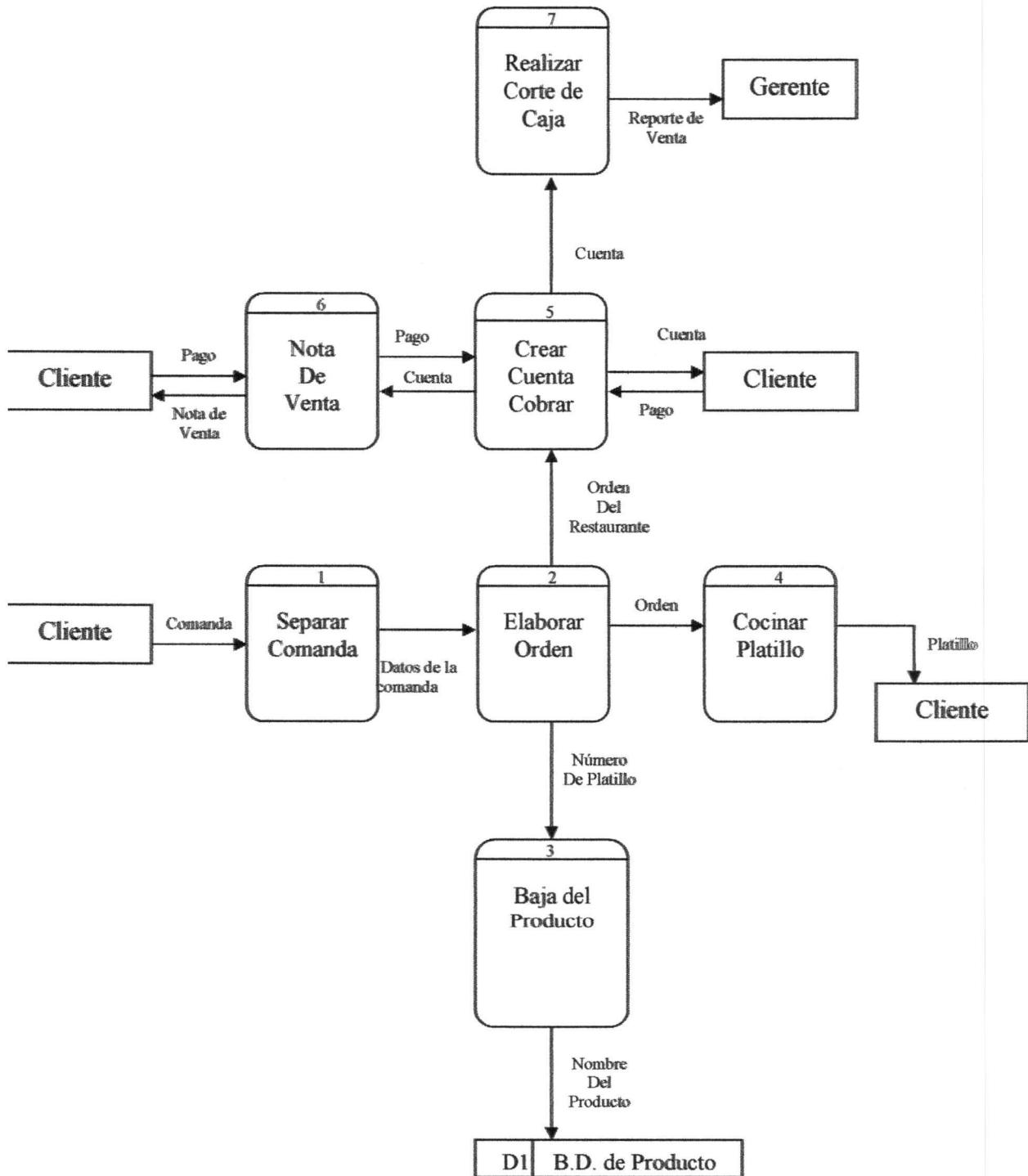
NIVEL CONTEXTUAL. La relación entre el sistema, el cliente y la gerencia.

NIVEL CERO. Simboliza todo el proceso que se lleva a cabo desde que llega el cliente hasta que paga la cuenta, de cómo se elabora la "comanda" (Ver anexo 3) y después de haber levantado el pedido qué es lo que se hace con esa comanda; a donde se lleva, cómo la manejan después de haber levantado dicho pedido y cómo realizan la cuenta para el cliente.

DIAGRAMA DE FLUJO DE DATOS DEL SISTEMA PROPUESTO**NIVEL CONTEXTUAL**

FLUJO DEL MANEJO DE LA COMANDA

NIVEL 0



B) ELECCIÓN Y APRENDIZAJE DE HERRAMIENTAS DE DESARROLLO

Teniendo en cuenta esta metodología para desarrollar sistemas de información es necesario especificar las herramientas con las que se desarrolló lo que es Punto de Venta. El sistema se elaboró bajo un manejador de base de datos y el lenguaje de programación como lo es Delphi 7 los cuales se describieron anteriormente.

La metodología de desarrollo de Borland Delphi es que proporciona un mecanismo integrado para definir la funcionalidad del lado del cliente para:

- Validar campos
- Actualizar automáticamente los datos
- Realizar cambios automáticos de interfaz
- Actualizar directamente la visión de los datos

C) ELABORACIÓN DEL DICCIONARIO DE DATOS

El diccionario de datos nos permitió de alguna forma manejar los tipos de datos que se tenían que utilizar para elaborar los que fueron las bases de datos.

DATOS DE PRODUCTO = No_producto + Nombre + Cantidad

CUENTA = Fecha + No_Producto(s) + Nombre (s) + Cantidad + Total_venta
+ No_mesa

REPORTE DE CONSUMO= cve_prod + nombre + cantidad

CORTE DE CAJA= Total de las ventas

COMANDA

Levanta el pedido del cliente

DATOS A UTILIZAR	TIPO DE DATO	LONGITUD
FOLIO	Numérico	15
FECHA	Date	
HORA	Carácter	11
N_MESAS	Numérico	2
N_PNAS	Numérico	10
N_MESERO	Numérico	2
PRE_COM	Float	10
CTO_COM	Float	10
NOM_USU	Numérico	10
DESC_COM	Flota	10
HR_ENT	Carácter	11
HR_SAL	Carácter	11

DETALLE DE LA COMANDA

Datos necesarios que debe de tener la comanda.

DATOS A UTILIZAR	TIPO DE DATO	LONGITUD
FOLIO	Numérico	15
CVE_PROD	Numérico	20
CTO_PROD	Float	10
PRE_PROD	Float	10
CANT_PROD	Numérico	10

2. DISEÑO DEL SISTEMA

Se definieron los módulos del sistema de información y la manera en que van a interactuar unos con otros, intentando que cada módulo trate total o parcialmente un proceso específico y tenga una interfaz sencilla.

El diseño de los módulos se fue realizando de una manera visual al usuario, en el cual se agregaron imágenes las cuales son de gran ayuda para el usuario ya que se pueden evitar errores al momento de capturar datos; se relacionaron las imágenes con el nombre de los productos.

Conforme a la información que se recabó del análisis, se fue diseñando el sistema y se cubrieron las necesidades con las que presenta la empresa, logrando que al usuario se le facilite el trabajo que realiza, ahorrar tiempo, manejar bien el control de inventarios, administración de gastos, etc.

Durante el diseño de módulos, se identifican características o comportamientos comunes relacionados con accesos a las bases de datos, en las cuales se están guardando los datos que se tienen tanto del personal, usuarios, artículos, proveedores, productos, etc.

PRINCIPALES COMPONENTES QUE SE UTILIZARON PARA DESARROLLAR EL SISTEMA

Para el desarrollo del sistema fue necesario auxiliarse de algunos componentes que facilitan ciertas tareas, a continuación se presentan algunos de ellos.

ESTANDAR:



Label



Edit



Botón



Panel

ADICIONAL



SpeedButton



Image



Junio 2007

SYSTEM



Timer

DATASOURCE



DATA CONTROLS



DBGrid



DBText

BDE



Table



Query



Junio 2007

DIALOGS



Open Picture Dialogs

FORMAS



Forms

3. DESARROLLO DEL SISTEMA

Después de recopilar, analizar y diseñar los módulos necesarios del sistema, se empezó a programar en un lenguaje de programación orientada a objetos como lo es Delphi; se manejaron y almacenaron datos necesarios para dicha aplicación.

Para poder realizar el sistema primero se elaboraron las bases de datos en las que se va a almacenar la información de usuarios, meseros, artículos, familias, productos, gastos, etc.; ya teniendo las tablas bien hechas y normalizadas se programó lo que son altas, bajas, consultas y modificaciones lo que se conoce como un (A,B,C y M).

Una vez hecho las A,B,C y M se prosiguió a lo que es punto de venta que se desarrolló para que se le pudiera asignar la mesa que se va a ocupar, y también a desocupar al momento de pagar la cuenta, y el mesero que está atendiendo al cliente.

LOS EVENTOS QUE SE MANEJARON COMÚNMENTE FUERON LOS DE:

OnClick: Cuando se da clic con el mouse sobre el componente realiza la función programada con anterioridad

OnKeyPress: A cada componente se le programan las validaciones necesarias, y cuando se le de algún dato, esta función se activa.

OnActivate: Cuando las formas en las que se programa se activan, dependiendo que es lo que se requiera se aplica esa función.

OnClose: El Formulario se cierra, cancelando así todos sus componentes que se utilizaron y todo el sistema.

OnCreate: Este evento se puede programar en tiempo de diseño y ejecución, dependiendo de los componentes utilizados, se le asigna dicho código.

4. PRUEBAS

Durante las pruebas se implementaron todas las estrategias posibles para garantizar el buen funcionamiento del sistema, y que éste se encuentre libre de problemas lo cual se fue descubriendo durante el proceso y así realizar las correcciones necesarias.

Primero se dieron de alta datos ficticios, para identificar los errores y posteriormente se revisó el funcionamiento con datos reales observando que es necesario un equipo con ciertas características que pueda soportar el sw. (Ver anexo 4)

5. ELABORACIÓN DE MANUALES

En el desarrollo de manuales se maneja la descripción del sistema y las actividades que se deben de seguir en la realización de las funciones de cada uno de los módulos, para esto se desarrollaron manuales de usuario así como el manual técnico.

6. IMPLANTACIÓN

Se instaló el sw en el equipo de la empresa, y además se otorgó la capacitación necesaria para su uso, al personal que operará el sistema.

CONCLUSIONES

Al concluir el proyecto es posible decir que se alcanzó el objetivo principal del mismo, que era el de realizar un sistema de información basado en las necesidades específicas de la empresa. Si bien se plantearon otros objetivos, estos podrán ser alcanzados conforme la empresa proporcione los recursos y las necesidades vayan surgiendo.

El llevar a cabo un análisis previo del funcionamiento de la empresa que permita conocer a detalle sus necesidades específicas, representa un gran reto para el que se necesita un verdadero compromiso y mucha responsabilidad; se trata de un proceso arduo que requiere una gran cantidad de trabajo, paciencia y dedicación.

Trabajar de la mano de las personas que utilizarán el sistema resulta una tarea complicada pero muy agradable. Es posible llegar a entender al usuario y al personal de la empresa a un nivel tan extremo que te conviertes en parte de su empresa y alguien mas en su área de trabajo.

Desarrollar software de calidad es un objetivo que todo equipo de desarrolladores debe tener presente constantemente, ya que traduce en una carta de presentación, a mayor satisfacción del cliente, se obtienen mejores recomendaciones para futuros trabajos. Pero lo más motivante es la satisfacción del cliente cuando el producto está terminado y le es útil en sus labores.

El objetivo esencial de desarrollar las residencias profesionales es adquirir experiencia al solucionar problemas reales dentro de las empresas que representan una fuente de empleo en un futuro muy cercano.

RECOMENDACIONES

Las recomendaciones que se mencionan a continuación están dirigidas a la empresa y basadas en la implantación del sistema de información. Se recomienda:

- Hacer uso del sistema. Después de la fase de implantación y de que las personas clave reciban la capacitación, el sistema estará completamente listo para brindar el servicio que de él se espera, por lo que debe ser utilizado para el propósito que fue creado y de manera correcta para obtener resultados positivos que apoyen el proceso de toma de decisiones.
- Tener en cuenta el crecimiento y la actualización constante. El sistema de información no debe, por ningún motivo, limitar las expectativas de crecimiento de la empresa, sino debe alentarlos y estar preparado para enfrentarlos.
- Realizar el mantenimiento preventivo pertinente para evitar saturación de bases de datos.

BIBLIOGRAFÍA

1. Análisis y Diseño de Sistemas 3ra. Edición

Kenneth E. Kendall

Julie E. Kendall

Editorial: Prentice Hall Hispanoamericana, S.A.

1997

2. Análisis y Diseño de Sistemas de Información 2da. Edición

James A. Senn

Editorial Mc Graw-Hill Interamericana de México, S.A. de C.V.

1992

3. Moral Víctor

Delphi 4

Prentice Hall Iberia, S.R.L.

1999

Otras fuentes bibliográficas son las siguientes:

<http://www.monografias.com/trabajos29/ciclo-sistema/ciclo-sistema.shtml>

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

ANEXOS

ANEXO. 1**FORMATO DE LA ENTREVISTA**

Dirigida a: Srita. Grecia Payan Espinoza Encargada del local.
Lugar: El Rincón del Café

ENTREVISTA 1

1. ¿Quién elabora el pedido y realiza la compra de mercancías?
2. ¿Cada cuando se hace un pedido?
3. ¿Cuál es el procedimiento para realizar el pedido?
4. ¿Cada cuando se realiza inventario?
5. ¿Existe alguna medida aproximada en cuanto a la cantidad del producto utilizado para la elaboración del platillo o bebida (lts., grs.)?
6. ¿Qué tipo de comida preparan?

ANEXO 2.

Dirigida a: Los meseros del local

Lugar: El Rincón del Café

ENTREVISTA 2

1. ¿Cómo se toma el pedido al cliente?
2. ¿Cómo se organizan los meseros para atender las mesas?
3. ¿Con cuantas mesas dispone en el Café?
4. ¿Cómo están organizadas o identificadas las mesas?
5. ¿Cómo se lleva a cabo el registro de las comandas realizadas?
6. ¿Qué sucede si se extravía alguna comanda?
7. ¿Dónde se realiza el cobro de las comandas?
8. ¿Encuentra algún problema en las actividades realizadas en el área del café?
9. ¿Han tenido problemas para calcular la cuenta del cliente?
10. ¿Se otorgan créditos o descuento a los clientes?
11. ¿Considera que se agilizaría el servicio en el café con un sistema automatizado que controle las comandas?

ANEXO 4.**DETERMINACIÓN DEL HW**

Para el adecuado funcionamiento del sistema, se requiere por lo menos:

- Procesador Intel Celaron LGAa 2.6 Ghz
- Disco duro de 80 Gb Serial ATA
- Memoria RAM DDR 512 Mb
- Monitor 17" CRT
- Unidad Floppy 3 ½
- Unidad Cd Rom
- Unidad Combo CD-RW/DVD
- Teclado ps/2
- Mouse óptico
- Regulador
- Impresora hp D395 deskjet 600
- Miniprint Epson 7000 TmU-200 PD

MANUAL TÉCNICO

Forma Punto de Venta

```
procedure TFmPuntoVta.FormActivate(Sender: TObject);
var k: Integer;
begin
  FmContra.ShowModal;
  if (vend) then begin
    activaComp;
  end else
    if TipoU='Administrador' then
      desactivaComp
    else
      activaComp;
  TbMesas.Open;
  TbComanda.Open;
  tbMeseros.Open;
  tbExtras.Open;
  tbArticulos.Open;
  TbDetalleComanda.Open;
  TbProductos.Open;
  TbFamilias.Open;
  Tbproductos.IndexName:= 'DESP_PRO_C';
  Tbproductos.First;
  PaFamilias1.Visible:=False;
  TbDetalleComanda.Filtered:=FALSE;
  TbDetalleComanda.Filter:="";
  TbDetalleComanda.Filter:='Folio=0';
  TbDetalleComanda.Filtered:=TRUE;

  TbMesas.First;
  for K:=0 to TbMesas.RecordCount do
  begin
    if TbMesas.FieldValues['Ocupada']=1 then
      case TbMesas.FieldValues['N_Mesa'] of
        1:SpEdB1.Glyph.LoadFromFile('C:\sycaf\imagenes SyCAF\mesas\mesa1o.bmp');
        2:SpEdB2.Glyph.LoadFromFile('C:\sycaf\imagenes SyCAF\mesas\mesa2o.bmp');
        3:SpeedButton11.Glyph.LoadFromFile('C:\sycaf\imagenes SyCAF\mesas\mesa3o.bmp');
        4:SpeedButton16.Glyph.LoadFromFile('C:\sycaf\imagenes SyCAF\mesas\mesa4o.bmp');
        5:SpeedButton18.Glyph.LoadFromFile('C:\sycaf\imagenes SyCAF\mesas\mesa5o.bmp');
        6:SpeedButton19.Glyph.LoadFromFile('C:\sycaf\imagenes SyCAF\mesas\mesa6o.bmp');
        7:SpeedButton20.Glyph.LoadFromFile('C:\sycaf\imagenes SyCAF\mesas\mesa7o.bmp');
        8:SpeedButton21.Glyph.LoadFromFile('C:\sycaf\imagenes SyCAF\mesas\mesa8o.bmp');
        9:SpeedButton22.Glyph.LoadFromFile('C:\sycaf\imagenes SyCAF\mesas\mesa9o.bmp');
        10:SpeedButton23.Glyph.LoadFromFile('C:\sycaf\imagenes SyCAF\mesas\mesa10o.bmp');
```

```
11:SpeedButton25.Glyph.LoadFromFile('C:\sycaf\imagenesSyCAF\mesas\mesa11o.bmp'
12:SpeedButton24.Glyph.LoadFromFile('C:\sycaf\imagenesSyCAF\mesas\mesa12o.bmp')
13:SpeedButton26.Glyph.LoadFromFile('C:\sycaf\imagenes
    end;
    TbMesas.Next;
end;
end;
```

```
procedure TFmPuntoVta.FormClose(Sender: TObject; var Action: TCloseAction);
begin
TbProductos.Close;
TbFamilias.Close;
tbExtras.Close;
tbMeseros.Close;
```

UTILIDADES

```
procedure TFmUtilidades.FormActivate(Sender: TObject);
begin
if {FmContra}FmPuntoVta.TipoU = 'Administrador' then
begin
SpEdBUsuarios.Enabled:= true;
SpEdBMeseros.Enabled:= True;
SpEdBFamilias.Enabled:= True;
SpEdBProductos.Enabled:= True;
SpEdBArticulos.Enabled:= True;
SpEdBProveedores.Enabled:= True;
SpEdBGastos.Enabled:= true;
SpEdBGM.Enabled:= True;
SpEdReportes.Enabled:= True;
SpEdBmermas.Enabled:= True;
SpEdBCorteC.Enabled:= True;
end
else
begin
SpEdBUsuarios.Enabled:= False;
SpEdBMeseros.Enabled:= False;
SpEdBFamilias.Enabled:= False;
SpEdBProductos.Enabled:= False;
SpEdBArticulos.Enabled:= True;
SpEdBProveedores.Enabled:= True;
SpEdBGastos.Enabled:= true;
```

```
SpEdBGM.Enabled:= false;
SpEdReportes.Enabled:= false;
SpEdBmermas.Enabled:= false;
SpEdBCorteC.Enabled:= True;
end;
end;

procedure TFmUtilidades.SpEdReportesClick(Sender: TObject);
begin
FmReportesTodos.Showmodal;
end;

procedure TFmUtilidades.SpEdBGastosClick(Sender: TObject);
begin
FmGastos.tipo:=true;
FmGastos.Showmodal;
end;

procedure TFmUtilidades.SpEdBGMClick(Sender: TObject);
begin
FmGastos.tipo:=false;
FmGastos.Showmodal;
end;

procedure TFmUtilidades.SpEdBCorteCClick(Sender: TObject);
begin
FmCorteFinal.Showmodal;
end;

procedure TFmUtilidades.SpEdBmermasClick(Sender: TObject);
begin
Fmmemas.showmodal;
end;
```

ALTAS-BAJAS-CAMBIOS

```
procedure TFmArticulos.SpEdBGuardarClick(Sender: TObject);
begin
if CoBoUnidad.Text='Elija una Opcion' then
Application.MessageBox('Seleccione un tipo de Unidad','Advertencia',64)
else
begin
```

```
if TbArticulos.FindKey([LbClaveArt.Caption]) then
    TbArticulos.Edit
else
    TbArticulos.Append;

//Paso los datos del edit a la tabla
TbArticulosCVE_ART.AsString:= LbClaveArt.Caption;
TbArticulosDes_Art_L.AsString:=EdDescCart.Text;
TbArticulosDes_Art_C.AsString:=EdDescLArt.Text;
TbArticulosPrec_Prov.AsString:=EdPreProv.Text;
TbArticulosFECH_ULT_C.AsDateTime:=now;
TbArticulosUnidad.AsString:=CoBoUnidad.Text;
if EdCapacidad.Enabled then
    TbArticulosCapacidad.AsString:=EdCapacidad.Text;
{else
    TbArticulosCapacidad.AsString:='1';}

TbArticulosCTO_EX.AsFloat:=0;

TbArticulosTOT_CAP.AsFloat:=TbArticulosTOT_CAP.AsFloat+
    (StrToInt(EdExistencia.Text)*TbArticulosCAPACIDAD.AsFloat);

aux:=0;
aux_2:=0;

aux:=trunc(TbArticulosTOT_CAP.AsFloat/TbArticulosCAPACIDAD.AsFloat);
aux_2:=(TbArticulosTOT_CAP.AsFloat/TbArticulosCAPACIDAD.AsFloat)-aux;
aux_2:=aux_2*TbArticulosCAPACIDAD.AsFloat;
if TbArticulosUNIDADAsString='oz' then
    edExistAct.Text:=inttostr(aux)+' botellas '+
        (floattostr(aux_2))+' onzas'
else
    if TbArticulosUNIDADAsString='g' then
        edExistAct.Text:=inttostr(aux)+' Kilos '+
            (floattostr(aux_2))+' gramos'
    else
        edExistAct.Text:=inttostr(aux)+' Piezas ';

//Paso los datos a la tabla
TbArticulos.Post;
RegistrodeCompra;
```

```
TbArticulos.Refresh;
TbArticulos.Last;

SpEdBGuardar.Enabled:=FALSE;

//limpio los edits

//limpio los edits
EdDescCArt.Clear;
EdDescLArt.Clear;
CoBoUnidad.Text:='Elija una Opcion';
EdCapacidad.Clear;
EdExistencia.Clear;
EdStock.Clear;
procedure TFmArticulos.SpEdBEliminarClick(Sender: TObject);
begin

if MessageDlg('Desea Eliminar a: '+ LbClaveArt.Caption,
  mtConfirmation, [mbOk, mbCANCEL], 0) = mrOk then
TbArticulos.Delete;

Ti1.Enabled:=TRUE;

//limpio los edits
//limpio los edits
EdDescCArt.Clear;
EdDescLArt.Clear;
CoBoUnidad.Text:='Elija una Opcion';
EdCapacidad.Clear;
EdExistencia.Clear;
EdStock.Clear;
EdCveProv.Clear;
EdPreProv.Clear;
EdCostooz.Clear;
//EdDescCArt.SetFocus;
LbNomProv.Caption:="";
EdDescCArt.Enabled:=false;
EdDescLArt.Enabled:=false;
CoBoUnidad.Enabled:=false;
EdCapacidad.Enabled:=false;
EdExistencia.Enabled:=false;
EdStock.Enabled:=false;
EdCveProv.Enabled:=false;
EdPreProv.Enabled:=false;
EdCostooz.Enabled:=false;
SpEdBConvertir.Enabled:=false;
```

```
BtMultiplicar.Enabled:=false;
  SpeedButton1.Enabled:=false;
  Maximo;

end;

procedure TFmArticulos.SpEdBNuevoClick(Sender: TObject);
begin
  //limpio los edits
  EdDescCArt.Clear;
  EdDescLArt.Clear;
  CoBoUnidad.Text:='Elija una Opcion';
  EdCapacidad.Clear;
  EdExistencia.Clear;
  EdStock.Clear;
  EdCveProv.Clear;
  EdPreProv.Clear;
  //EdDescCArt.SetFocus;
  EdCostooz.Clear;
  LbNomProv.Caption:="";
  TiGuardar.Enabled:=true;
  SpEdBModificar.Enabled:=false;
  SpEdBEliminar.Enabled:=False;
  SpEdBcomprar.Enabled:=false;

  EdDescCArt.Enabled:=true;
  EdDescLArt.Enabled:=true;
  CoBoUnidad.Enabled:=true;
  EdCapacidad.Enabled:=true;
  EdExistencia.Enabled:=true;
  EdStock.Enabled:=true;
  EdCveProv.Enabled:=true;
  EdPreProv.Enabled:=true;
  EdCostooz.Enabled:=true;
  SpEdBConvertir.Enabled:=true;
  BtMultiplicar.Enabled:=true;
  SpeedButton1.Enabled:=true;

  Maximo;
end;

procedure TFmArticulos.FormActivate(Sender: TObject);
begin
  TbArticulos.Open;
  TbProveedores.Open;
  SpEdBcomprar.Enabled:=false;
  cve_art:=TbArticulosCve_Art.AsInteger;
  cve_prov:=TbArticulosCVE_PROV.AsInteger;
```

```

Ti1.Enabled:=false;
TiGuardar.Enabled:=FALSE;
TbArticulos.First;
//aux1:=TbArticulosEXISTENCIA.AsFloat;
CoBoUnidad.Text:= TbArticulosUnidad.DisplayText;
LbClaveArt.Caption:=TbArticulosCve_Art.DisplayText;
EdDescCArt.Text:=TbArticulosDes_Art_C.DisplayText;
EdDescLArt.Text:=TbArticulosDes_Art_L.DisplayText;
EdPreProv.Text:=TbArticulosPrec_Prov.DisplayText;
EdCapacidad.Text:=TbArticulosCapacidad.DisplayText;
EdExistencia.Text:=TbArticulosExistencia.DisplayText;
EdStock.Text:=TbArticulosStock.DisplayText;
EdCveProv.Text:= TbArticulosCVE_PROV.DisplayText;
EdCostooz.Text:=TbArticulosCTO_EXAsString;
aux:=0;
aux_2:=0;

aux_2:=aux_2*TbArticulosCAPACIDAD.AsFloat;
edExistAct.Text:=inttostr(aux)+' botellas (floattostr(aux_2))+' onzas';
//FmArticulos.LbNomP.Caption:=
FmConsultaProveedores.TbProveedoresNom_Prov.DisplayText;
if TbProveedores.FindKey([EdCveProv.Text]) then
  LbNomProv.Caption:= TbProveedoresNOM_PROV.DisplayText
else
  LbNomProv.Caption:= "";

if TbArticulosUNIDADAsString='pza' then
  Label13.Caption:='Stock mínimo en piezas'
else if TbArticulosUNIDADAsString='oz' then
  Label13.Caption:='Stock mínimo en onzas'
else
  Label13.Caption:='Stock mínimo en gramos';
end;

procedure TFmArticulos.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
  TbArticulos.Close;
  TbProveedores.Close;
  SpEdBModificar.Enabled:=FALSE;
  SpEdBEliminar.Enabled:=FALSE;
end;

procedure TFmArticulos.SpEdBDeshacerClick(Sender: TObject);
begin
  //lo que hago en el SpEdBNuevo lo hago en deshacer
  SpEdBNuevo.OnClick(SpEdBNuevo);

```

```
procedure TFmArticulos.SpEdBModificarClick(Sender: TObject);
begin
  if TbArticulos.FindKey([LbClaveArt.Caption]) then
  begin
    //pongo en edicion la tabla para poder modificarla
    //aux1:=TbArticulosEXISTENCIA.AsFloat;
    TbArticulos.Edit;
    TbArticulosDes_Art_L.AsString:=EdDescLArt.Text;
    TbArticulosDes_Art_C.AsString:=EdDescCArt.Text;
    TbArticulosPrec_Prov.AsString:=EdPreProv.Text;
    TbArticulosUnidad.AsString:=CoBoUnidad.Text;
    //TbArticulosCapacidad.AsString:=EdCapacidad.Text;
    TbArticulosExistencia.AsFloat:= {TbArticulosExistencia.AsFloat +
}StrToFloat(EdExistencia.Text);
    TbArticulosStock.AsString:=EdStock.Text;
    TbArticulosCve_Prov.AsString:=EdCveProv.Text;
    if EdCostooz.Text<>" then
      TbArticulosCTO_EX.AsFloat:=StrToFloat(EdCostooz.text)
    else
      TbArticulosCTO_EX.AsFloat:=0;
    if EdCapacidad.Enabled then
      TbArticulosCapacidad.AsString:=EdCapacidad.Text;
    {else
      TbArticulosCapacidad.AsString:='1';}

//TbArticulosTOT_CAP.AsFloat:=TbArticulosTOT_CAP.AsFloat+(strtofloat(EdExistencia.Text)*TbArticulosCAPACIDAD.AsFloat);

aux:=0;
aux_2:=0;

aux:=trunc(TbArticulosTOT_CAP.AsFloat/TbArticulosCAPACIDAD.AsFloat);
aux_2:=(TbArticulosTOT_CAP.AsFloat/TbArticulosCAPACIDAD.AsFloat)-aux;
aux_2:=aux_2*TbArticulosCAPACIDAD.AsFloat;
if TbArticulosUNIDAD.AsString='oz' then
  edExistAct.Text:=inttostr(aux)+' botellas '+
    (floattostr(aux_2))+' onzas'
else
  if TbArticulosUNIDAD.AsString='g' then
    edExistAct.Text:=inttostr(aux)+' Kilos '+
      (floattostr(aux_2))+' gramos'
else
  edExistAct.Text:=inttostr(aux)+' Piezas ';

//edExistAct.Text:=TbArticulosTOT_CAP.AsString;
//TbArticulosEXISTENCIA.AsFloat:=TbArticulosEXISTENCIA.AsFloat+aux1;
```

```
TbArticulos.Post;

if TbArticulosUNIDAD.AsString='pza' then
  Label13.Caption:='Stock mínimo en piezas'
else if TbArticulosUNIDAD.AsString='oz' then
  Label13.Caption:='Stock mínimo en onzas'
else
  Label13.Caption:='Stock mínimo en gramos';
TbArticulos.Refresh;
Showmessage('El Registro se modifico');
Ti1.Enabled:=TRUE;

EdDescCArt.Clear;
EdDescLArt.Clear;
CoBoUnidad.Text:='Elija una Opcion';
EdCapacidad.Clear;
EdExistencia.Clear;
EdStock.Clear;
EdCveProv.Clear;
EdPreProv.Clear;
LbNomProv.Caption:="";
//EdDescCArt.SetFocus;
edExistAct.Clear;
EdCostooz.Clear;

EdDescCArt.Enabled:=false;
EdDescLArt.Enabled:=false;
CoBoUnidad.Enabled:=false;
EdCapacidad.Enabled:=false;
EdExistencia.Enabled:=false;
EdStock.Enabled:=false;
EdCveProv.Enabled:=false;
EdPreProv.Enabled:=false;
EdCostooz.Enabled:=false;
SpEdBConvertir.Enabled:=false;
BtMultiplicar.Enabled:=false;
SpeedButton1.Enabled:=false;
Maximo;
end;
end;

procedure TFmArticulos.TiGuardarTimer(Sender: TObject);
begin
  //Reviso si todos los campos tiene contenido,
  if(Trim(EdDescCArt.Text)<>"")and
    (Trim( EdDescLArt.Text)<>")and
```

```
((CoBoUnidad.Text='pza'))or((CoBoUnidad.Text<>'pza')and(Trim(EdCapacidad.Text)<>"')
))and
(Trim(EdExistencia.Text)<>")and
(Trim(EdStock.Text)<>")and
(((CoBoUnidad.Text<>'oz')and(not
EdCostooz.Enabled))or((CoBoUnidad.Text='oz')and(Trim(EdCostooz.Text)<>"))))and
//(Trim(EdCveProv.Text)<>")and
(Trim(EdPreProv.Text)<>) then

//si es asi activo el SpEdBGuardar,eliminar y modificar
begin
SpEdBGuardar.Enabled:=TRUE;

end
else
Begin
//sino lo desactivo para q no los puedan usar
SpEdBGuardar.Enabled:=FALSE;
end;

end;
```

COBRAR

```
procedure TFmCobrar.EdEntregadoKeyPress(Sender: TObject; var Key: Char);
var i: Integer;
begin
if (StrScan('0123456789.'+ Chr(7)+ Chr(8),Key)= Nil) then
Key := #0;

{CONTROLA LA ENTRADA DE SOLO UN PUNTO DECIMAL}
For i:= 1 to Length(EdEntregado.Text) do
if (Copy(EdEntregado.Text,i,1)= '.') and not (StrScan('.',Key)= Nil) then
Key := #0;
end;

procedure TFmCobrar.SpEdBCancelarClick(Sender: TObject);
begin
Close;
end;
```

COMPRAR

```
procedure TFmcomprar.SpeedButton1Click(Sender: TObject);
begin
  TbArticulos.Open;
  Tbcompras.Open;
  Tbcompras.Append;
  TbcomprasFECHA.AsDateTime:=now;
  TbcomprasCVE_ART.AsInteger:=FmArticulos.cve_art;
  TbcomprasCANTIDAD.AsFloat:=StrToFloat(EdCapacidad.Text) *
  StrToFloat(EdExistencia.Text);
  TbcomprasCVE_PROV.AsInteger:=FmArticulos.cve_prov;
  TbcomprasIMPORT_TOT.AsFloat:=StrToFloat(EdPreProv.Text);
  Tbcompras.Post;
  Tbcompras.Close;

  TbArticulos.Locate('Cve_Art',varArrayOf([FmArticulos.cve_art]),[]);

  aux1:=TbArticulosEXISTENCIA.AsFloat;
  TbArticulos.Edit;
  TbArticulosPrec_ProvAsString:=EdPreProv.Text;
  TbArticulosCve_ProvAsInteger:=FmArticulos.cve_prov;
  TbArticulosCapacidadAsString:=EdCapacidad.Text;
  TbArticulosFECH_ULT_C.AsDateTime:=now;
```

CONSULTAS

```
procedure TFmConsultaArticulos.BBAgregarArtClick(Sender: TObject);
begin
  if FmProductos.Edit = '1' then
  begin
    FmProductos.EdIng1.Text:= TbArticulosCVE_ART.DisplayText;
    FmProductos.LbNom1.Caption:= TbArticulosDES_ART_C.DisplayText;
    FmProductos.LbCant1.Caption:=TbArticulosUNIDAD.DisplayText;
  end;

  if FmProductos.Edit = '2' then
  begin
    FmProductos.EdIng2.Text:= TbArticulosCVE_ART.DisplayText;
    FmProductos.LbNom2.Caption:= TbArticulosDES_ART_C.DisplayText;
    FmProductos.LbCant2.Caption:=TbArticulosUNIDAD.DisplayText;
  end;
```

```
procedure TFmConsultaArticulos.EdNomArtChange(Sender: TObject);
begin
  TbArticulos.SetKey;
  TbArticulos['DES_ART_C]:= Trim(EdNomArt.Text);
  TbArticulos.GotoNearest;
end;

procedure TFmConsultaArticulos.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
  fmPuntoVta.Art:=TbArticulosCVE_ART.AsString;
  FmPuntoVta.desc:=TbArticulosDES_ART_C.AsString;
  TbArticulos.Close;
```

CONTRASEÑA

```
if not registro.finalizado then begin
  Reset(archivo);
  read(archivo,registro);
  TbUsuario.Locate('login_usu',vararrayof([registro.usuario]),[]);
  if (TbUsuarioNOM_USU.AsString=EdNombreU.Text)and
    (TbUsuarioCONTRS_USU.AsString=EdContUsu.Text) then begin
    FmPuntoVta.TipoU:= TbUsuarioTIPO_USU.DisplayText;
    FmPuntoVta.usuario:=trunc(TbUsuarioLOGIN_USU.Asfloat);
    FmPuntoVta.contrasena:=TbUsuarioCONTRS_USU.AsString;
    FmPuntoVta.Nombre:=TbUsuarioNOM_USU.AsString;
    if TbUsuarioTIPO_USU.AsString='Administrador' then
      if MessageDlg('Desea iniciar su sesión como vendedor?',
        ,mtConfirmation, [mbYes, mbNo], 0) = mrYes then
        FmPuntoVta.vend:=true;
    closefile(archivo);
    FmPuntoVta.Sesion:=true;
    TbUsuario.close;
    close;
  end else begin
    if (TbUsuario.Locate('Nom_Usu',vararrayof([EdNombreU.Text]),[]))and
      (TbUsuarioCONTRS_USU.AsString=EdContUsu.Text)then begin
      if TbUsuarioTIPO_USU.AsString='Administrador' then begin
        ShowMessage('Sólo puedes iniciar tu sesión como administrador');
      //   FmPuntoVta.vend:=false;
      close;
    end else
      ShowMessage('Imposible iniciar como vendedor');
    end else
  end
```

```
ShowMessage('El usuario no existe o la contraseña es incorrecta');
end;
end;
```

CORTE FINAL

```
procedure TFmCortefinal.FormActivate(Sender: TObject);
var total,t: real;
i:integer;
fecha,fecha_2,formato:string;
begin
SetLength(gastos,0);
Memo1.Clear;
AssignFile(archivo,FmPuntoVta.directorio+'log.Dat');
Reset(archivo);
Read(archivo,registro);
total:=0;
TbComandas.Open;
TbDetalleComanda.Open;
TbProductos.Open;
TbUsuarios.Open;
TbCorte.Open;
TbExtras.Open;
TbArticulos.Open;
qrycortefinal.Active:=false;
qrycortefinal.Active:=true;
TbCorte.Locate('INDICE',vararrayof([qrycortefinal.fields[0].Asinteger]),[]); 
total:=TbCorteFONDO.AsFloat;

EdFecha.Text:=DateToStr(now);

formato:=ShortDateFormat;
ShortDateFormat := 'mm/dd/yyyy';
fecha:=DateToStr(registro.fechini);
fecha_2:=DateToStr(now);

TbCorte.Last;
TbCorte.Edit;
EdFondofinal.Text:=TbCorteFONDOAsString;
```

GASTOS

```
procedure TFmGastos.BtAceptarClick(Sender: TObject);
begin
if (EdCantPagar.Text<>"")and(MemoGastos.Text<>") then begin
  TbGastos.Open;
  TbGastos.Append;
  qryMaxGasto.Active:=false;
  qryMaxGasto.Active:=true;
  TbGastosTIPO.AsBoolean:=tipo;

  TbGastosCVE_GASTO.AsInteger:=qryMaxGasto.Fields[0].AsInteger+1;
  TbGastosIMPORT_GTO.AsFloat:=strtofloat(EdCantPagar.Text);
  TbGastosCONCEPTOAsString:=MemoGastos.Text;
  TbGastosFECHA.AsDateTime:=now;
  TbGastosHORAAsString:=TimeToStr(now);
  TbGastosLOGIN_USU.AsInteger:=FmPuntoVta.usuario;
  TbGastos.Post;
  TbGastos.Close;

End;
End;
```

MERMAS

```
procedure TFmMermas.tigTimer(Sender: TObject);
begin
if (Trim(EdNA.Text)<>"")and
  (Trim(EdExistencia.Text)<>")and
  (Trim(EdExiRel.Text)<>") then
  SpEdBrebajar.Enabled:=true
else
  SpEdBrebajar.Enabled:=false;
end;

procedure TFmMermas.EdExiRelKeyPress(Sender: TObject; var Key: Char);
var i: Integer;
begin

if (StrScan('0123456789.'+ Chr(7)+ Chr(8),Key)= Nil) then
  Key := #0;

{CONTROLA LA ENTRADA DE SOLO UN PUNTO DECIMAL}
```

```
For i:= 1 to Length(EdExiRel.Text) do
  if (Copy(EdExiRel.Text,i,1)= '.') and not (StrScan('.',Key)= Nil) then
    Key := #0;
  end;
```

MULTIPLICAR

```
procedure TFmMult.BtMultClick(Sender: TObject);
begin
  if EdPrecio.Text <> " then begin
    EdRtdo.Text:= FloatToStr((StrToFloat(EdPrecio.Text)* StrToFloat(EdCantidad.Text )));
    FmArticulos.EdPreProv.Text:= EdRtdo.Text;
    FmMult.Close;
  end;
end;

procedure TFmMult.BtCancelarClick(Sender: TObject);
begin
  FmArticulos.EdPreProv.Text:= "";
  FmMult.Close;
end;

procedure TFmMult.FormActivate(Sender: TObject);
begin
  EdPrecio.Clear;
  EdCantidad.Clear;
  EdRtdo.Clear;
end;
```

PERSONAS

```
procedure TFmPersonas.BtAceptarClick(Sender: TObject);
begin
  fmPuntoVta.Personas:=StrToInt(EdPersonas.Text);
  FmPuntoVta.Mesero:=StrToInt(EdMesero.Text);
  FmPuntoVta.res:=mrOk;
  close;
end;

procedure TFmPersonas.EdPersonasKeyPress(Sender: TObject; var Key: Char);
begin
  if (not(key in['0'..'9','#8'])) then
    key:="#0";
```

```
end;

procedure TFmPersonas.EdMeseroKeyPress(Sender: TObject; var Key: Char);
begin
  if (not(key in['0'..'9','#8])) then
    key:=#0;
end;

procedure TFmPersonas.Button1Click(Sender: TObject);
begin
  FmPuntoVta.res:=mrNone;
  close;
end;
END;
```

REPORTES

```
procedure TFmReportesTodos.SpeedButton15Click(Sender: TObject);
begin
  Fmfechpe.showmodal;
end;

procedure TFmReportesTodos.SpeedButton14Click(Sender: TObject);
begin
  FmFechcp.Showmodal;
end;

procedure TFmReportesTodos.SpeedButton13Click(Sender: TObject);
begin
  FmFechct.ShowModal;
end;

procedure TFmReportesTodos.SpeedButton12Click(Sender: TObject);
begin
  FmFechca.Showmodal;
end;

procedure TFmReportesTodos.SpeedButton11Click(Sender: TObject);
begin
  FmRsm.TbArticulos.Open;
  FmRsm.QRsm.PreviewModal;
  FmRsm.TbArticulos.Close;
end;
```

```
procedure TFmReportesTodos.SpeedButton5Click(Sender: TObject);
begin
  FmFechvd.Showmodal;
end;

procedure TFmReportesTodos.SpeedButton6Click(Sender: TObject);
begin
  FmFechvp.Showmodal;
end;

procedure TFmReportesTodos.SpeedButton7Click(Sender: TObject);
begin
  FmFechvpp.Showmodal;
end;

procedure TFmReportesTodos.SpeedButton8Click(Sender: TObject);
begin
  FmFechvm.Showmodal;
end;

procedure TFmReportesTodos.SpeedButton9Click(Sender: TObject);
begin
  FmFechbu.Showmodal;
end;

procedure TFmReportesTodos.SpeedButton10Click(Sender: TObject);
begin
  FmFechgi.Showmodal;
end;

procedure TFmReportesTodos.SpeedButton3Click(Sender: TObject);
begin
  FmFecher.Showmodal;
end;
```