



INSTITUTO POLITÉCNICO NACIONAL



ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y
ELÉCTRICA
UNIDAD CULHUACAN

UNIDAD PROFESIONAL INTERDISCIPLINARIA
DE INGENIERÍA Y CIENCIAS SOCIALES Y ADMINISTRATIVAS

SEMINARIO DE TITULACIÓN
ADMINISTRACIÓN DE PROYECTOS
FNS5062005/15/2008

“SOFTWARE DE CÓDIGO DE BARRAS”

T E S I N A

QUE PARA OBTENER EL TÍTULO DE:

OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACION

PRESENTA

NELSON ROGELIO TRUJILLO ZAVALA

INGENIERO EN COMUNICACIONES Y ELECTRÓNICA

PRESENTA

ARIEL BRIZUELA ALQUICIRA

PEDRO ISIDRO ZAMORA

INGENIERO MECÁNICO

PRESENTA

ANDRÉS DE JESÚS MARTÍNEZ CHÁVEZ

A S E S O R E S

M. EM C. EDNA CARLA VASCO

ING. AMPARO BAÑUELOS DURAN

LAE. DALILA VIVIANA HERNANDEZ VASCO

México D.F. noviembre de 2008

INSTITUTO POLITECNICO NACIONAL
ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA UNIDAD
CULHUACAN
UNIDAD PROFESIONAL INTERDISCIPLINARIA DE INGENIERÍA Y CIENCIAS
SOCIALES Y ADMINISTRATIVAS

T E S I N A

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

PRESENTA

NELSON R. TRUJILLO ZAVALA

INGENIERO MECÁNICO

PRESENTA

ANDRÉS DE JESÚS MARTÍNEZ
CHÁVEZ

**INGENIERO EN COMUNICACIONES
Y ELECTRÓNICA**

PRESENTA

ARIEL BRIZUELA ALQUICIRA
PEDRO ISIDRO ZAMORA

SEMINARIO DE TITULACIÓN
ADMINISTRACIÓN DE PROYECTOS
FNS5062005/15/2008

“SOFTWARE DE CÓDIGO DE BARRAS”

C A P I T U L A D O

1. MARCO DE REFERENCIA
2. ESTUDIO DE MERCADO
3. PLANEACION DEL PROYECTO
4. EJECUCIÓN Y CONTROL DEL PROYECTO
5. EVALUACIÓN DE RESULTADOS

MÉXICO, D.F. NOVIEMBRE DE 2008

M. EM. C. EDNA CARLA VASCO
MENDEZ
DIRECTORA DEL SEMINARIO

ING. AMPARO BAÑUELOS DURAN
ASESORA

LAE. DALILA VIVIANA
HERNANDEZ VASCO
ASESORA

ING. MAGDALENO VASQUEZ
RODRÍGUEZ
JEFE DE INGENIERIA MECÁNICA

Índice

Introducción	8
Detección de Necesidades	9
Planteamiento del Problema.....	9
Justificación.....	9
Objetivo General.....	10
Objetivos Específicos.	10
Alcance.	10
Metas.....	10
Misión.....	11
Visión.....	11
1. Marco de Referencia.....	12
1.1. Tiendas de Conveniencia.....	12
1.1.1. Historia y evolución de las Tiendas de Conveniencia.	12
1.1.2. GENERALIDADES DE UNA TIENDA DE CONVENIENCIA.....	14
1.2. Código de Barras.....	15
1.2.1. Historia del código de barras.....	15
1.2.2. Elementos del código.....	17
1.2.3. ID de información.....	17
1.2.4. Sistema Dual o Binario	18
1.2.5. Simbología.....	18
1.2.6. Nomenclatura Básica.....	18
1.2.7. Tipos, Características y Aplicaciones.....	19
1.2.7.1. Códigos Lineales o Una Dimensión 1-D.....	19
1.2.7.1.2. Código 3 de 9 o Código 39.....	20
1.2.7.1.3. Código Codabar	21
1.2.7.1.4. Código 128.....	22
1.2.7.1.5. Código EAN-13	23
1.2.7.1.6. Código UPC-A.....	24
1.2.7.2. Códigos Bidimensionales o Dos Dimensión 2-D.....	25
1.2.7.2.1. Código 49.....	25

1.2.7.2.2.	Código 16K	26
1.2.7.2.3.	Código PDF417.....	26
1.2.7.2.4.	Código DataMatrix	27
1.2.7.2.5.	Código MaxiCode	27
1.2.7.2.6.	Código Aztec	28
1.2.7.2.7.	Código QR	29
1.3.	AMECE.....	30
1.3.1.	Historia	31
1.4.	Marco Legal.....	32
2.	Estudio de Mercado	33
2.1.	Objetivo	33
2.2.	Metodología de Investigación.....	33
2.2.1.	Puntos de Investigación	33
2.2.1.1.	Atención a Clientes.	33
2.2.1.2.	Impacto Económico.....	34
2.3.	Cuestionario	34
2.4.	Graficas de Resultados	35
2.5.	Análisis y Recomendaciones.....	44
3.	Planeación.....	45
3.1.	Tabla de Actividades.....	47
3.2.	Diagrama de Gantt.....	49
4.	Ejecución y Control del Proyecto.....	50
4.1.	Análisis y requisitos del software	50
4.1.1.	Realizar análisis de necesidades.....	50
4.1.2.	Borrador de las especificaciones preliminares del software.....	51
4.1.3.	Desarrollar presupuesto preliminar.....	51
4.1.4.	Incorporar los comentarios a las especificaciones del software	51
4.1.5.	Calcular los tiempos y fechas de entrega.....	51
4.2.	Diseñar.....	52
4.2.1.	Especificaciones preliminares del software.....	52
4.2.2.	Desarrollar especificaciones de funcionamiento	53
4.2.3.	Desarrollar prototipo basado en las especificaciones de funcionamiento.....	54

4.2.4.	Revisar especificaciones de funcionamiento	54
4.2.5.	Incorporar comentarios a las especificaciones de funcionamiento	54
4.2.6.	Obtener aprobación para continuar	55
4.3.	Desarrollo.....	55
4.3.1.	Revisar especificaciones de funcionamiento	55
4.3.2.	Identificar parámetros de diseño modular y de componentes separados	55
4.3.3.	Desarrollar el código	56
4.3.3.1.	Diseño de la Base de Datos	56
4.3.3.2.	Variables y Constantes de Entorno	57
4.3.3.3.	Modulo Familias de Producto.	58
4.3.3.4.	Centro de control.	69
4.3.3.5.	Modulo de Usuarios	73
4.3.3.6.	Pantalla Principal	75
4.3.3.7.	Función de Importación	84
4.4.	Pruebas.....	96
4.4.1.	Desarrollar planes de pruebas de unidades usando las especificaciones del producto	96
4.4.2.	Desarrollar planes de pruebas de integración usando las especificaciones del producto.....	96
4.4.3.	Pruebas de unidades	96
4.4.3.1.	Revisar el código modular.....	96
4.4.3.2.	Probar si los módulos de los componentes se ajustan a las especificaciones del producto	96
4.4.3.3.	Identificar anomalías y anotarlas en las especificaciones del producto	97
4.4.3.4.	Modificar código.....	97
4.4.3.5.	Volver a probar el código modificado	97
4.4.4.	Documentación	97
5.	Evaluación de Resultados.....	98
5.1.	Resultados.....	98
5.2.	Conclusiones.....	99

Introducción

La existencia de varios tipos de códigos de barras se debe a que las simbologías están diseñadas para resolver problemas a las tiendas de conveniencias. De acuerdo al tipo de necesidad de identificación interna del negocio, se debe optar por el sistema de codificación más adecuado. Por tal motivo se realizará el proyecto para construir o presentar la puesta en marcha del sistema en códigos de barras, facilitando a las tiendas de conveniencia en sus actividades en el POS en un tiempo menor y así darle un mejor servicio al cliente.

De igual manera nace este proyecto con la necesidad de resolver algunos de los problemas más comunes en las tiendas de conveniencias, y diseñar un software el cual proporcione datos más certeros conforme al desplazamiento tanto entradas y salidas de los artículos y tener un mejor control de los SKU, UPC o PLU de productos, con esto se tendrá un mejores días de inventario.

No importa la industria que sea, la tecnología de captura de datos por código de barras puede ayudarle a vencer los retos más difíciles con los que se enfrenta.

Detección de Necesidades.

Algunas tiendas de conveniencias no cuentan con una base de datos de productos confiable debido a la duplicidad de códigos.

- No cuentan con un sistema de administración de productos - códigos
- No cuenta con un procedimiento de levantamiento de órdenes de compra.
- No cuenta con un procedimiento de estandarización de productos.
- No utiliza la infraestructura que tiene para la generación de etiquetas.
- No cuentan con un sistema que realice una segmentación de productos.

Planteamiento del Problema.

Debido a la necesidad de reducir tiempos de espera y quitando de raíz el sobre precio, se debe por tener más de dos código de barras en un mismo producto en las tiendas de conveniencia, nos vemos en la necesidad de desarrollar un software que optimice y administre los productos en el sistema del cualquier negocio, los tiempo de mano de obra serán más eficientes en otras actividades en cada uno de los puntos de venta al tener este sistema; los gastos serán menores, la utilidad va a ser mayor hasta un 10 % en menos de 6 meses después de la implementación, ya que con esto se tendrá una mayor visibilidad para negociar con los proveedores en el área comercial, dando como resultado credibilidad en los precios y un mejor servicio a los clientes.

Justificación.

En la actualidad se presenta pérdidas económicas de debido a que no cuentan con sistema que reporte productos con uno o más códigos de barras, lo que conlleva a errores en inventarios, problemas con el consumidor, además que esto lleva al cierre de sucursales por la PROFECO¹. Aunado a esto la falta de una estandarización de métodos y proceso ocasiona que estos tomen más tiempo de lo debido.

¹ Procuraduría Federal de Consumidor

Objetivo General.

Diseñar un software que de mayores ventajas y suministre herramientas básicas sobre los sistemas de códigos de barras y así poder reducir costos, brindar un mejor servicio e incrementar las utilidades de las tiendas de conveniencia.

Objetivos Específicos.

- Optimización de gastos.
- Reducción de cargas de datos manuales por error humanos.
- Inventarios reales (lo que se traduce en cero pérdidas de productos y una contabilidad certera del producto que sale y que está en el depósito).
- Crear sus propias tarjetas de código de barras para los artículos que no cuentan con uno en los productos.
- Mantener actualizados los puntos de venta de las tiendas al 100%.
- Eliminar la duplicidad de códigos y documentar y mejorar los procesos de control de inventarios.
- Aumentar la autonomía entre empleados de punto de venta y de piso.
- Obtener un control por categorías

Alcance.

Ofrecer un sistema que permita el registro y control de inventarios, control de altas de nuevos productos, así como entradas y salidas de productos en almacenes y bodegas, en una tienda de conveniencia dentro del área metropolitana.

Metas.

Poner o implementar el software dentro de una tienda de conveniencia en un lapso de tiempo, iniciando con este proyecto en febrero del 2009.

Misión.

Analizar y desarrollar software de código de barras, adaptado a cada una de estas empresa, de aceptable calidad y con un moderado costo de desarrollo, entregando confiabilidad y certeza en el levantamiento de inventarios físicos reduciendo las perdidas en pesos y la optimización de días de inventarios, así brindaran un mejor servicio al cliente.

Teniendo como valores fundamentales:

- Honestidad.
- Servicio.
- Amabilidad y respeto.

Brindando a nuestros clientes información veraz para la toma de decisiones

Visión.

Ser la mejor opción del mercado en el desarrollo y consultoría especializada en el análisis de administración de productos e inventarios, brindando apoyos y soluciones para la mejora en la estructura, infraestructura y tecnología de tiendas de conveniencias.

1. Marco de Referencia.

1.1. Tiendas de Conveniencia.

Existen tiendas de conveniencia de todos tamaños, las hay desde las más pequeñas como las de tipo Waldo's Mart, existen cadenas de tiendas de autoservicio manejados por los grandes corporaciones por ejemplo, Wal-Mart, Home Depot, Comercial Mexicana, Soriana, Chedraui, Carrefour, Makro y Auchan, entre otros.

1.1.1. Historia y evolución de las Tiendas de Conveniencia.

La misma historia que se dio en Estados Unidos se está repitiendo ahora en México, pero lo que se tardó allí 25 años aquí la curva de crecimiento va a ser mucho más corta.

La historia y evolución de los años setenta a los ochenta, fue una época de búsqueda de la consistencia, los autoservicios tenían que crear una presentación de mercancías consistente, tienda por tienda, con mercancía de alta calidad. En los años setenta cada tienda hacía lo que quería, y no se reconocía qué tipo de tienda era porque todos se veían muy distintos.

Otro énfasis que se dio fue en el concepto de plano gramas, aquí la idea, era consolidar la sección completa de artículos con un proveedor o pocos proveedores, concentrar mercancía relacionada en una sección de la tienda, porque se investigó que al consumidor le molestaba mucho ir buscando en la tienda por todas partes los artículos.

Una solución fue con la utilización de empaques, en el cual se intentó guiar al consumidor con empaques claros, llamativos e informativos. Más carteles, más exhibidores y elementos para informar y guiar al consumidor en su compra, el énfasis fue también la idea de que el consumidor llegue a la tienda y que encuentre todo lo que requiera bajo un techo y una sección clave.

Esto fue lo que realmente dio vida al crecimiento de las **Tiendas de Conveniencia**. En la evolución de todo esto se puede ver el cambio y quien tiene el poder real. En los años cincuenta y sesenta el poder estaba con los fabricantes, con los proveedores, ellos inventaban un producto, lo tiraban al mercado y a ver que caía, y los consumidores no tenían opciones, lo que hay, hay.

Luego en los setenta, inicio de los ochenta, se vio un cambio en el poder porque las tiendas de conveniencia, los detallistas ya tenían más lugares, más tiendas, la distribución era el control del poder, quien tenía el alcance al consumidor finalmente.

En los ochenta y los noventa se dieron hechos concretos:

Los autoservicios empezaron a golpear para ver quien llegaba primero y quien colocaba primero su tienda en un pueblo o ciudad. Lo que resultó en demasiadas tiendas compitiendo para lo mismo. Este fue el resultado lógico, vemos que ya empezaron a llegar tiendas del tipo Wall Mart, Soriana, Office Depot, Oxxo, etc.

Finalmente se vio que en apertura de tiendas se empezó a aflojar, representó un porcentaje menor de crecimiento total pero podemos decir que la suma de todos los autoservicios en cualquier producto no representaba más de un 20, 25% del mercado total, hoy se está al revés en autoservicios se vende más del 70% de cualquier producto en el mercado total, y los independientes representan un 20, 25%.

Hoy en México se está como estaba E.U. hace 25 años, todavía el mercado de mayorista, de menudeo es mucho más grande de lo que es la combinación de autoservicios, pero va a cambiar y va a cambiar muy rápido. Y finalmente se vio que el enfoque de ventas en las tiendas era hacia **racionalizar mercancías**, presentaciones, y la **consolidación de proveedores**.

1.1.2. GENERALIDADES DE UNA TIENDA DE CONVENIENCIA

El manejo de una tienda de conveniencia está compuesto por varios elementos o áreas sin los cuales la misma no podría funcionar correctamente.

- Recibo
- Bodegas de abarrotes
- Bodegas de líneas generales
- Perecederos
- Piso de ventas
- Área de caja
- Atención al cliente
- Gerencia
- Recursos Humanos
- Sistemas, etc.

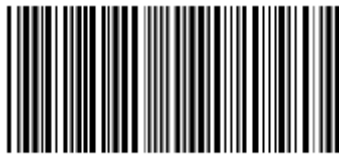
Los distribuidores de mercancías para este tipo de establecimientos, han conseguido un código de barras para cada uno de sus productos a través de una asociación con la organización nacional miembro de la Global Standards One (GS1).

Esta marca nace a partir de la alianza de principalmente organismos internacionales, UCC (Uniform Code Council), EAN International (European Article Number), quienes son los que a nivel mundial pueden establecer los códigos de barras que distinguirán dichas mercancías alrededor del mundo si fuese necesario.

1.2. Código de Barras

El código de barras es la representación de una determinada información mediante un conjunto de líneas paralelas verticales de diferente grosor y espaciado. La correspondencia o mapeo entre los mensajes que representan y el código de barras se denomina simbología. Las simbologías pueden ser clasificadas en dos grupos:

Los lineales (1-D) como los que se usan en productos y permiten incluir mensajes cortos.



Los de dos dimensiones (2-D) que han empezado a usarse en documentos para controlar su envío o en seguros médicos y, en general, en documentos que requieren la inserción de mensajes más grandes (de hasta 2 725 dígitos) como un expediente clínico completo.



1.2.1. Historia del código de barras

Todo empezó en Estados Unidos, en 1970. En principio se pensó para la industria farmacéutica por sus especiales características de gran cantidad de artículos y necesidad de rotación de stocks.

En 1973 se publicaron las primeras normas o estándares bajo la denominación de Universal Product Coding (Codificación Universal de Productos), quizá más conocidas por sus iniciales UPC. El sistema suponía la creación de un “banco de números” en el que se registraba el número asignado a cada fabricante. Después, el fabricante daba el número del producto, el cual se ponía en la segunda mitad del código, detrás del número de fabricante.

Aunque al principio se acogió el sistema con cierto entusiasmo, enseguida empezaron a aparecer algunos problemas de tipo laboral al creer los sindicatos que su implantación podría afectar al empleo. Por su parte, los minoristas se mostraban poco dispuestos a realizar la inversión necesaria en la compra de equipos de lectura, incluso los consumidores se manifestaron partidarios de “ver” los precios individuales en cada artículo.

A pesar de ello, algunos fabricantes se mostraron muy decididos a seguir adelante y marcaron sus productos con el código de barras UPC. Las críticas negativas desaparecieron y enseguida empezaron a coexistir los sistemas tradicional y nuevo en los grandes establecimientos.

El hecho de que el código de barras permitía una facturación más rápida a la salida de los supermercados y que el recibo o “ticket” daba una información muy detallada de la compra, convenció a los consumidores en el sentido de preferir los establecimientos que trabajaban con el nuevo método. Tanto fue así que los minoristas que lo adoptaron vieron crecer sus ventas notablemente por esta causa.

Cuando los resultados obtenidos fueron publicados y los aparatos de lectura se ofrecieron a precios más accesibles, la mayoría de cadenas de establecimientos se decidieron a utilizar el sistema, popularizándose a nivel general.

En 1977, representantes de la industria y del comercio de doce países europeos, tomaron la decisión de formar una organización a la cual llamaron "European Article Numbering Association" (EAN). Al unírsele al poco tiempo países no europeos, el nombre fue cambiado por el de "International Article Numbering Association" (EAN INTERNACIONAL).

El objetivo de esta organización es el de difundir y administrar el Código del producto, así como promover el uso de estándares de identificación y del Intercambio Electrónico de Datos "EDI".

El primer sistema patentado de verificación de códigos de barras por medio de láser aparece en el mercado en 1978.

En 1981, La tecnología de CCD (Charge Coupled Device) es aplicada en un escáner. En la actualidad este tipo de tecnología tiene bastante difusión en el mercado asiático, mientras que el láser domina en el mundo occidental. En ese año también aparece el código 128, de tipo alfanumérico.

Posteriormente, aparece la norma ANSI MH10.8M que especifica las características técnicas de los códigos 39, Codabar, e ITF (Interleaved Two of Five). En 1990 se publica la especificación ANS X3.182, que regula la calidad de impresión de códigos de barras lineales.

En ese mismo año, Symbol Technologies presenta el código bidimensional PDF417.

1.2.2. Elementos del código

Los códigos de barras se distinguen principalmente por dos componentes

Código.- Es un número asignado a la mercancía correspondiente, que es única e irrepetible con respecto a las demás mercancías de la tienda, las combinaciones de números son prácticamente infinitas y las probabilidades de que se repitan dos o más códigos son muy remotas (de hecho, hay software's especiales que se encargan de evitar que esto suceda).

Símbolo.- Es la representación gráfica de la combinación numérica (en ocasiones alfanumérica) de dicha mercancías. Es la serie de líneas paralelas verticales, gruesas y delgadas que podemos percibir con nuestra vista cuando nos fijamos cuidadosamente en el empaque del producto que vamos a adquirir, sin él, sería imposible que, tanto el sistema SIMA de cajas, como el lector láser de la symbol, pudieran leer, capturar y traducir la información que en ella se presenta.

1.2.3. ID de información

Cada código de barras, como ya se ha mencionado, es único e irrepetible, esto es gracias a que contiene una ID que lo identifica del resto de los códigos de barras. El código de barras, es prácticamente un compendio de información en un espacio reducido, el cual es práctico y funcional para los usuarios, los cuales son leídos con facilidad por la symbol.

1.2.4. Sistema Dual o Binario

A menudo, el sistema de código de barras, se ha comparado en el sistema binario de "1" y "0" que son el lenguaje primario de todo ordenador, a menudo de igual forma se han comparado con el sistema de clave morse, haciendo la analogía entre las barras anchas (que serían el equivalente a las rayas), y las barras angostas (que serían el equivalente a los puntos), esto por supuesto sin olvidar sus respectivos espacios o "pausas".

1.2.5. Simbología

Se trata de un lenguaje utilizado entre el sistema SIMA (o el correspondiente) asociado a la symbol, y al código gráfico del producto que contiene la información. La simbología sirve para que el lector láser de la terminal (escáner) y el código de barras de la correcta comunicación entre ellas. El sistema entiende y traduce a la perfección la información que viene implícita en el código de barras.

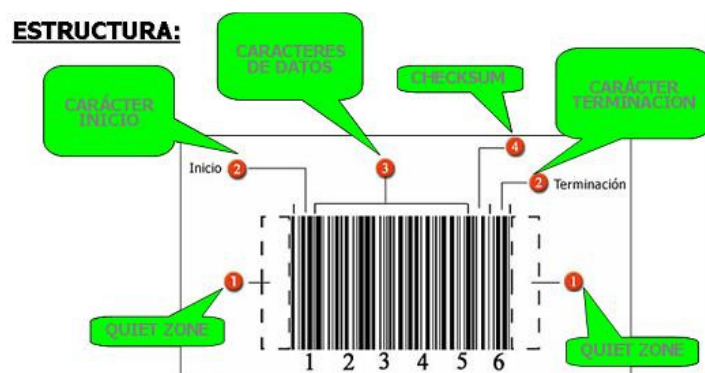
1.2.6. Nomenclatura Básica

Módulo: Es la unidad mínima o básica de un código. Las barras y espacios están formados por un conjunto de módulos.

Barra: El elemento (oscuro) dentro del código. Se hace corresponder con el valor binario 1.

Espacio: El elemento (claro) dentro del código. Se hace corresponder con el valor binario 0.

Carácter: Formado por barras y espacios. Normalmente se corresponde con un carácter alfanumérico.



1.2.7. Tipos, Características y Aplicaciones.

Así como en el mundo existen muchos idiomas y alfabetos, también hay varias simbologías de código de barras. Todos ellos fueron desarrollados con propósitos distintos; a primera vista se parecen, sin embargo, tienen sus diferencias, dependiendo de la aplicación para la que fueron creados.

1.2.7.1. Códigos Lineales o Una Dimensión 1-D

1.2.7.1.1. Código Entrelazado 2 de 5 (Interleaved 2 of 5 ITF)

Es un código numérico que utiliza dos grosores, y como su nombre indica, se basa en la técnica de intercalar caracteres. El primer carácter se representa en barras, y el segundo por los espacios que se intercalan en las barras del primero. Esto hace al Intercalado 2 de 5 (I 2/5) un código muy denso, con una cantidad par de dígitos. La posibilidad de una lectura parcial es más alta de lo deseable, especialmente si se utiliza un lector láser. Por lo tanto, generalmente se toman ciertas medidas de seguridad, como codificar un carácter de verificación al final del símbolo, programar al lector para que espere leer una cantidad fija de caracteres y colocar barras gruesas en los extremos superior e inferior del código.

Características

- Codifica SOLAMENTE números
- Usa poco espacio
- Por sus características, puede leerse parcialmente de modo accidental, por lo que se debe configurar el lector de código de barras para lea cierta longitud de caracteres y evitar que se pierda información
- Solo codifica número de caracteres pares, por lo que añade ceros al inicio cuando se requiere

Aplicaciones

- Se usa en Estados de Cuenta, Recibos de Servicios (agua, luz, teléfono, etc.)
- Se usa para etiquetar productos y hacer Inventarios (codifica solo números)
- Se usa para etiquetar cajas de empaque (cajas, embalajes, etc.)

COMPROBANTE CAJERO



1.2.7.1.2. Código 3 de 9 o Código 39

Este tipo es actualmente el más utilizado para aplicaciones industriales y comerciales. Permite la codificación de caracteres numéricos, así como letras mayúsculas y algunos símbolos como -, \$, /, +, % y "espacio". Un símbolo puede codificar uno o muchos caracteres, con una posibilidad de error de sustitución muy baja. Se utilizan sólo dos grosores tanto para barras como para espacios.

Aunque es muy fiable, el Código 39 requiere muchas barras y espacios para representar un solo carácter. Esto rara vez representa un problema, pero ocasionalmente puede derivar en la creación de códigos de barras muy largos que superen el tamaño de la etiqueta.

Características

- Codifica TODOS los caracteres ASCII (números, letras, símbolos)
- Longitud variable
- Ofrece gran seguridad a la lectura

Aplicaciones

- Es uno de los códigos de barras más populares. Se utiliza en muchos y muy variados ambientes
- Estados de cuenta, recibos de servicios (agua, luz, teléfono, etc.)
- Números de Serie, Inventarios, gafetes y credenciales para control de acceso, etiquetas para identificar productos y tomar inventarios, etc.



1.2.7.1.3. Código Codabar

Es uno de los códigos de barras más antiguos y aún se utiliza con frecuencia en tres actividades: bibliotecas, bancos de sangre y seguimiento de paquetes. Es una simbología de longitud variable que codifica únicamente números. Sus especificaciones permiten una impresión con un alto grado de tolerancia, lo que popularizó su uso en empresas de mensajería (las impresoras de impacto utilizadas para crear los albaranes de envío no producen una impresión muy buena para código de barras). Este código utiliza dos tipos de grosores para barras y espacios y su densidad es similar a la del Código 39. Debido a esto, muchos usuarios prefieren la mayor flexibilidad que brinda este último al codificar una mayor cantidad de caracteres.

Características

- Codifica SOLO números y los símbolos: - : \$ / +
- Longitud Variable

Aplicaciones

- Librerías para control de libros
- Bancos de sangre
- Laboratorios fotográficos
- Se utiliza en la industria médica para etiquetar muestras y reactivos
- Empresas de mensajería

1.2.7.1.4. Código 128

El Código 128 utiliza 4 grosores diferentes para las barras y espacios, con una densidad muy alta, ocupando en promedio sólo el 60% del espacio requerido para codificar información similar con el Código 39. Tiene una baja probabilidad de error de sustitución y como su nombre indica, puede codificar los 128 caracteres ASCII.

El método que utiliza el código 128 para codificar tantos caracteres y mantener su alta densidad es el uso de tres juegos de caracteres, conocidos como Conjuntos A, B y C. Es decir, un patrón de barras y espacios puede representar tres caracteres diferentes, y su interpretación depende del juego en contexto, que se indica mediante un carácter especial dentro del código. Debido a su flexibilidad y alta densidad, es un tipo muy versátil.

Características

- Codifica todos los caracteres ASCII (números, letras, símbolos)
- Longitud variable
- Es un código muy seguro, que utiliza poco espacio para su impresión

Aplicaciones

- Industria detallista (EAN 128)
- Vales de despensa
- Mensajería y Paquetería
- Estados de Cuenta (tarjetas de crédito), recibos de servicios, gafetes y credenciales de control de acceso, etc.



1.2.7.1.5. Código EAN-13

EAN (European Article Numbering o Sistema de Numeración Europeo)

Es la versión europea del Código UPC. El EAN-13 codifica 13 dígitos, uno más que el UPC. Este dígito extra se combina con uno o dos de los otros dígitos para representar un código de país, indicando el origen de la mercancía. El Código EAN-8 es similar al UPC-E, codificando menos caracteres y aumentando la confusión. Excluyendo a Estados Unidos y Canadá, el Código EAN-13 se utiliza virtualmente en todo el mundo en la venta de productos de consumo. Además (y ahora sí universalmente) se emplea en la codificación de libros como un subconjunto del ISBN, que es un sistema de numeración único para todos los libros publicados en el mundo.

Características

- Codifica solo números
- 12 caracteres y un dígito verificador
- Representa un sistema de decodificación mundial
- Puede tener adendum de 2 y 5 dígitos

Aplicaciones

- Industria detallista
- Los 2 ó 3 primeros dígitos representan el código del país de origen: para México es 750



1.2.7.1.6. Código UPC-A

UPC (Código Universal de Producto)

Hemos dejado para el final esta simbología (así como la EAN) porque es definitivamente la más confusa de todas. UPC es el código de barras utilizado en prácticamente todos los productos de consumo de EEUU y Canadá. Este código es el resultado del esfuerzo de luchas agrupaciones trabajando unidas no sólo para acordar un estándar, sino virtualmente para inventar una simbología de código de barras. Por lo tanto, para explicar la simbología UPC, es necesario explicar el estándar.

El estándar UPC (denominado UPC-A) es un número de 12 dígitos (sólo se pueden codificar números en UPC). El primero es llamado "número del sistema". La mayoría de los productos tienen un "1" o un "7" en esta posición. Esto indica que el producto tiene un tamaño y peso determinado, y no un peso variable. Los dígitos del segundo al sexto representan el número del fabricante. Esta clave de 5 dígitos (adicionalmente al "número del sistema") es única para cada fabricante, y la asigna un organismo rector que administra la numeración para evitar duplicidades.

Los caracteres del séptimo al onceavo son un código que el fabricante asigna a cada uno de sus productos, denominado "número del producto". El doceavo carácter es el "dígito verificador", resultando de un algoritmo que involucra a los 11 números previos.

Para hacer las cosas aún más complicadas, se inventó el Código UPC-E para ayudar a los fabricantes que no podían acomodar un código de 12 dígitos. Hubo una cantidad muy limitada de estos códigos, el último de los cuales se asignó hace ya varios años.

Características

- Codifica solo números
- Codifica 11 caracteres y un dígito verificador
- Representa un sistema de decodificación mundial
- Puede tener adendum (código más pequeño) de 2 y 5 dígitos

Aplicaciones

- Industria detallista
- Los 2 ó 3 primeros dígitos representan el código del país de origen



1.2.7.2. Códigos Bidimensionales o Dos Dimensión 2-D

Los códigos de barras de dos dimensiones, 2D, son códigos que su principal característica es que pueden “guardar” más información que la que contienen los códigos de una dimensión 1D. Dependiendo del tipo de código, se pueden “guardar” hasta casi 7,000 caracteres.

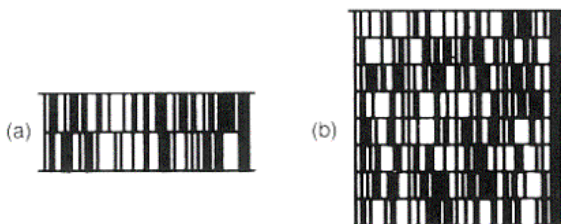
1.2.7.2.1. Código 49

Características

- Codifica todos los caracteres ASCII
- 49 alfanuméricos ó 81 numéricos
- Puede tener desde 2 hasta 8 renglones de alto

Aplicaciones

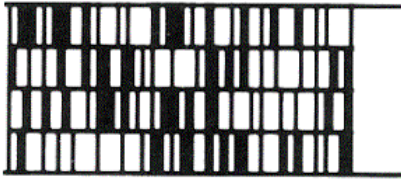
- Creado por Intermec para codificar objetos pequeños



1.2.7.2.2. Código 16K

Características.

- Codifica todos los caracteres ASCII
- 77 alfanuméricos ó 154 numéricos
- Puede tener desde 2 hasta 16 renglones de alto
- Aplicaciones
- Creado para codificar objetos pequeños



1.2.7.2.3. Código PDF417

Características

- Codifica todos los caracteres ASCII
- 1,850 texto ó 2,710 dígitos ó 1,108 bytes
- Puede tener desde 3 hasta 90 renglones
- Tiene 9 niveles de seguridad (para corrección de errores)
- A mayor nivel de seguridad, más grande el código resultante

Aplicaciones

- Creado por Symbol Technologies para codificar mayor información en menor espacio
- Pedimentos aduanales, gafetes, licencias de manejo, etc.



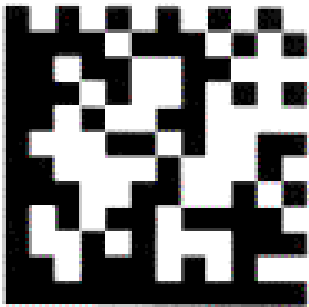
1.2.7.2.4. Código DataMatrix

Características

- Codifica todos los caracteres ASCII
- De 1 a 2,000 caracteres
- Es Omnidireccional y se lee con lectores imager
- El tamaño depende del número de datos

Aplicaciones

- Se utiliza en la industria detallista para codificar lote y fecha de fabricación.
- Se utiliza en el Servicio Postal Mexicano para automatizar el proceso de selección.



1.2.7.2.5. Código MaxiCode

Características

- Codifica todos los caracteres ASCII
- 93 caracteres
- Siempre es del mismo tamaño:1.1"x1"
- Puede ser leído sobre bandas transportadoras de alta velocidad

Aplicaciones

- Creado por United Parcel Systems UPS Mensajería y Paquetería



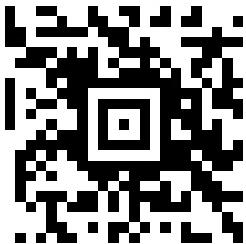
1.2.7.2.6. Código Aztec

Características

- Codifica todos los caracteres ASCII además de información binaria (imágenes)
- Codifica de 12 a 3,800 caracteres
- Es Omnidireccional y se lee con lectores imagen
- El tamaño depende del número de datos

Aplicaciones

- Se utiliza en ambientes de control de acceso y seguridad



1.2.7.2.7. Código QR

Características

- Su nombre se basa en la frase "Quick Response (Respuesta Rápida)" ya que se diseñó para ser decodificado a alta velocidad.
- Es el código de dos dimensiones más popular en Japón
- Codifica todos los caracteres ASCII además de información binaria (imágenes)
- Codifica hasta 7,089 caracteres
- Es Omnidireccional y se lee con lectores imager
- El tamaño depende del número de datos

Aplicaciones

- Originalmente se diseñó para el marcado de partes automotrices
- En Japón, los teléfonos celulares con cámara, pueden capturar el código de barras QR y guardar la información en su directorio. Actualmente en las tarjetas de presentación se imprime este código para ser capturado por los celulares.



1.3. AMECE

La Asociación Mexicana de Estándares para el Comercio Electrónico, AMECE, es un organismo de la iniciativa privada que tiene el objetivo de incrementar la productividad y competitividad de las empresas a través del uso de herramientas tecnológicas basadas en estándares internacionales.

AMECE agremia alrededor de 24,000 empresas en todo el país, cuyas ventas anuales se ubican entre 100 mil pesos y 850 millones de pesos. Las empresas agrupadas operan principalmente en los sectores de Manufactura, Comercio, Banca y Servicios.

La fuerza empresarial que AMECE representa la ha convertido en un participante activo en los foros nacionales e internacionales donde se discuten y acuerdan leyes y normas que tienen que ver con la automatización de la actividad comercial. Sus asociados están representados en esos foros y al tanto de sus avances.

Como miembro de GS1, AMECE se ha encargado desde hace 22 años de difundir y administrar los estándares globales en nuestro país.

GS1 es un organismo mundial líder con más de treinta años de existencia, dedicado al diseño e implementación de estándares y soluciones para mejorar la eficiencia y la visibilidad de las cadenas de distribución y demanda en distintos sectores con una meta muy clara: simplificar el comercio global, conectando el flujo de información con el flujo de productos.

El sistema de estándares desarrollado por GS1 es el más utilizado para administrar la cadena de valor en todo el mundo; de ahí que su cobertura abarque más de 100 países y que al día se realicen millones de operaciones basadas en los estándares: desde la lectura de un código de barras para la identificación de materia prima hasta el paso de un producto en el punto de venta, atravesando por la entrada y salida de mercancía en Centros de Distribución, el intercambio de documentos vía electrónica (pedidos, avisos de embarque, facturas, etc.), la catalogación de productos, etc.

Esto lo entienden cada vez más las empresas en México, que, independientemente de su tamaño y giro, buscan mejorar sus procesos para aumentar sus niveles de eficiencia y productividad, y por ende ser más competitivos, ya no sólo localmente, sino en el mercado mundial.

Entre otras actividades, es la entidad responsable de la implantación del sistema de código de barras en México y, a la fecha, el sistema que AMECE administra permite la identificación del 95.0% de los productos que se distribuyen a través de las cadenas de autoservicio.

1.3.1. Historia

1986: 18 empresarios, bajo la iniciativa de Henry Davis, entonces Director General del Grupo Aurrerá, se reúnen con la finalidad de echar a andar el código de barras en México. Así nace AMECOP, como organismo empresarial sin fines de lucro, donde participan la industria y el comercio privado.

1991: Ya hay casi unas 600 empresas que tienen puesto en un buen número de sus productos el código de barras.

1992: Unas 3000 empresas implementan los códigos y así el sistema empieza a crecer.

A esto se suma el crecimiento de las transacciones electrónicas entre clientes y proveedores en todo el mundo...

1995: Un primer grupo de trabajo empieza a analizar como traer la tecnología EDI a México. Se publican los primeros estándares para transacciones electrónicas: orden de compra, aviso de inventario, reporte de ventas.

1997-1998: Ya hay una implementación masiva de la orden de compra y de los reportes de inventarios y ventas electrónicos. AMECOP se convierte en AMECE y se comienza a analizar la implementación de la factura electrónica.

1999: Comienza la optimización y reordenamiento del uso del código de barras y de las transacciones electrónicas a través de la creación de un catálogo en línea.

2001: Lanzamiento del primer catálogo electrónico en México. Creación del primer estándar de repositorio electrónico. 2004. Puesta a punto de la factura electrónica y diversificación del alcance de AMECE.

2006: Reforma integral a la Asociación, nueva estructura organizacional, nuevo consejo, visión estratégica para los próximos 5 años que hará que AMECE se vuelva mucho mayor.

1.4. Marco Legal

Para el diseño de software se deben de tomar en cuenta las siguientes leyes:

Ley Federal de Derechos de Autor (LFDA) y su reglamento Primeros artículos y el capítulo IV del título IV

LFDA Artículo 5: "La protección de desarrollo de algún software se concede a las obras desde el momento en que hayan sido fijadas en un soporte material... El reconocimiento de los derechos de autor y de los derechos conexos no requiere registro ni documento de ninguna especie..."

Ley de Propiedad Industrial (LPI) y su reglamento

La LPI, que regula el otorgamiento de patentes en México a las invenciones de productos o de procesos, menciona en su artículo 19 sección IV que los programas de cómputo no son considerados invenciones.

Un programa de cómputo podría ser parte de un invento, sea un producto o un proceso, como un elemento para realizar alguna de las funciones del invento.

Protege elementos que pueden acompañar a un programa de cómputo, como son: marcas, dibujos o íconos distintivos.

Código Penal Federal y Código Federal de Procedimientos Penales

Sancionan la producción masiva de copias no autorizadas de programas de cómputo o su venta.

Para su distribución se debe de tomar en cuenta:

Puedes transmitir tu derecho patrimonial sobre el mismo a otra persona o una empresa, mediante un contrato de cesión de derechos y a cambio de una retribución económica negociada. Dejarás de ser el titular del derecho de autor del programa.

Puedes licenciar el uso de tu programa a otros, redactando una licencia o usando una ya existente, que indica qué sí se puede hacer y qué no. Tú sigues siendo el titular del derecho de autor del programa.

2. Estudio de Mercado

2.1. Objetivo

El objetivo de este estudio de mercado es obtener y analizar la información relacionada con los procesos de asignación de códigos de barras y sus repercusiones en tiempo, en los costos en las tiendas de conveniencia, en su interés por establecer una estrategia de no tener pérdidas, que le permita corregir de una manera eficiente sus inventarios

La información recabada en el estudio proveerá opciones a las tiendas de conveniencia que podrá considerar para desarrollar sus políticas de control.

2.2. Metodología de Investigación

La metodología utilizada para la realización de este estudio de mercado se efectuó en una fase.

- Investigación primaria, consistió de entrevistas con gerentes de tiendas, empleados y clientes de tiendas de conveniencia (7Eleven, Oxxo, CCK), los cuales se encuentran ubicados en la ciudad de México.

2.2.1. Puntos de Investigación

El Estudio de mercado se enfocó en dos puntos críticos que son:

- Atención a Clientes
- Impacto Económico

2.2.1.1. *Atención a Clientes.*

El problema que se presenta debido a la duplicidad de códigos, es que a un cliente un producto le cuesta un precio y a otro, lo que hace que el cliente se moleste y en ocasiones abandone la tienda sin adquirir dicho producto.

2.2.1.2. Impacto Económico.

Las pérdidas que presentan las tiendas de conveniencia debido al problema antes mencionado ascienden a \$12, 000,000.00² anuales por sucursal.

2.3. Cuestionario

El cuestionario aplicado fue:

¿Hay productos que cuentan con más de un código de barras?

Si () No ()

¿El producto escaneado coincide con su código?

Si () No ()

¿El precio coincide con lo marcado en las etiquetas?

Si () No ()

¿Es eficiente el sistema utilizado para el marcaje del producto?

Si () No ()

¿Atenido problemas con el escaneo de sus productos?

Si () No ()

¿Cómo cuales?

R=

Consideras que la impresión del Código de Barras del producto tuviera que coincidir con el inventario de la tienda

Si () No ()

¿Has dejado de comprar por algún error en precio?

Si () No ()

El control con código de barras te permite conocer con datos confiables las referencias de artículos vendidos

² Boletín Interno Oxxo Diciembre 2007

Si () No ()

¿Con que frecuencia actualizan inventarios en la tienda?

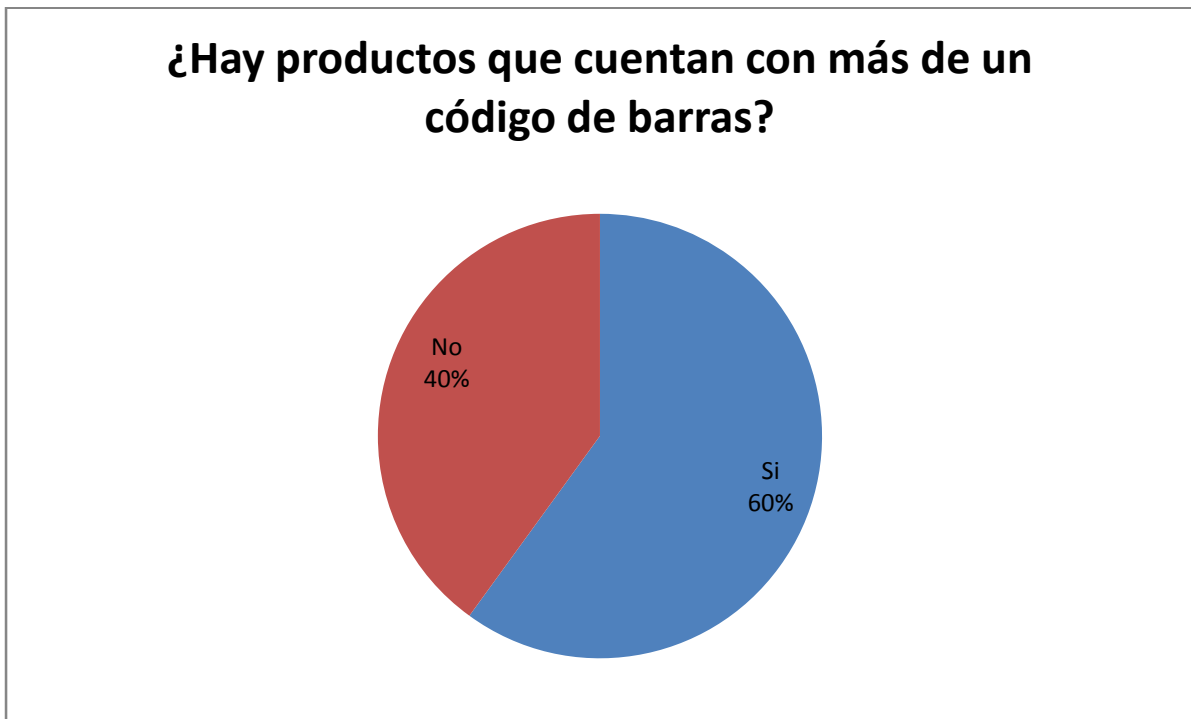
R=

¿Con que frecuencia actualizan las bases de datos que se ligan con el funcionamiento de la lectora de los códigos de barras?

R=

2.4. Graficas de Resultados

Pregunta 1



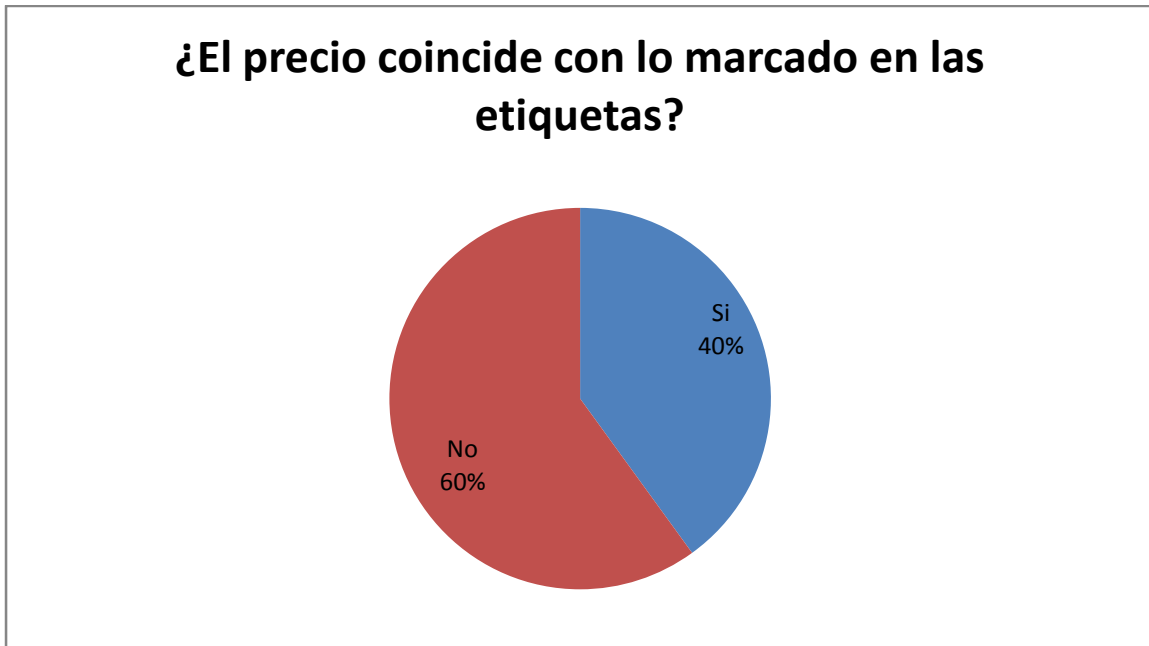
El 60% de los encuestados menciona haber encontrado por lo menos 1 producto con un código de barras repetido, mientras que el 40% restante no ha tenido este contratiempo.

Pregunta 2



El 68% de los encuestando indica que el código de barras del producto no coincide con el código del sistema, es decir, se escanea un refresco de un sabor y el personal de la tienda marca otro en el sistema POS son arroga el precio similar lo que llegaba a que marcan uno por otro

Pregunta 3



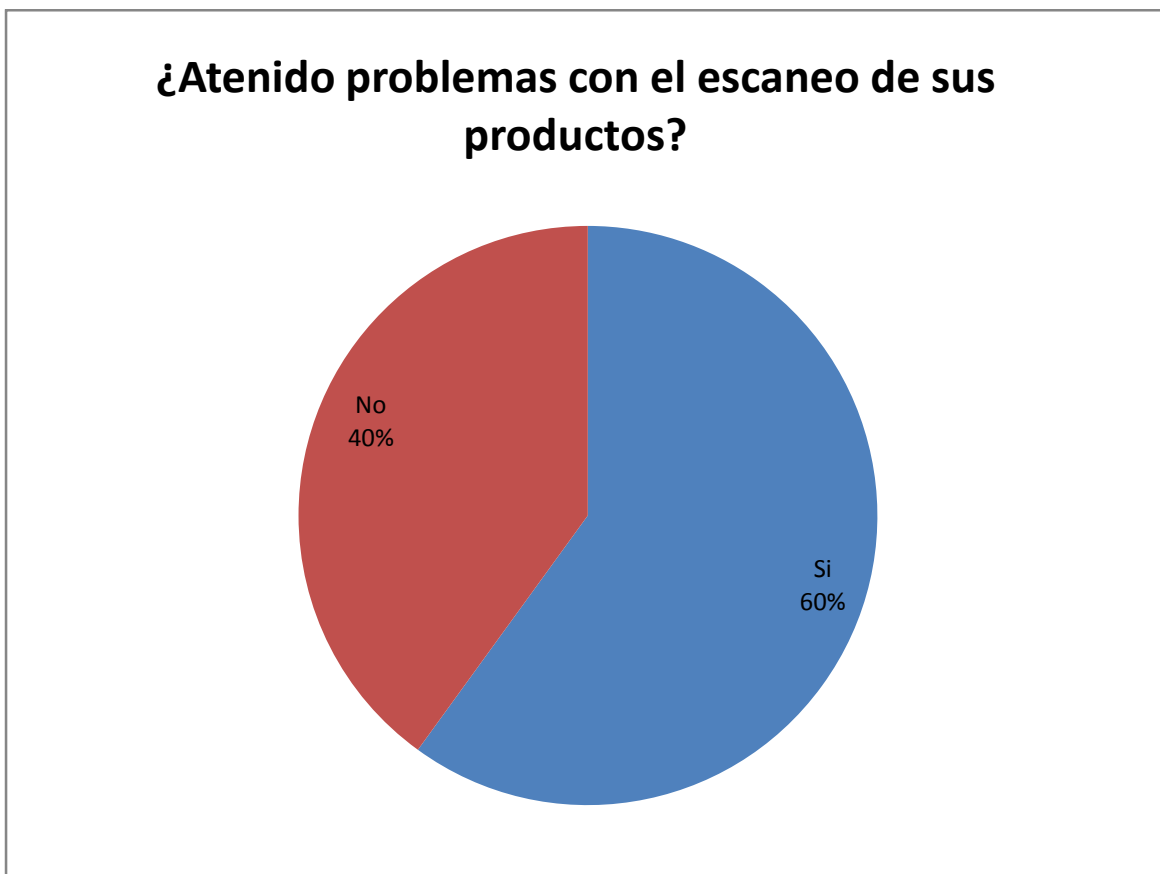
El 60% de los encuestados nos indica que el precio que está en la etiqueta del producto, no coincide con el precio que se encuentra vigente en el POS

Pregunta 4



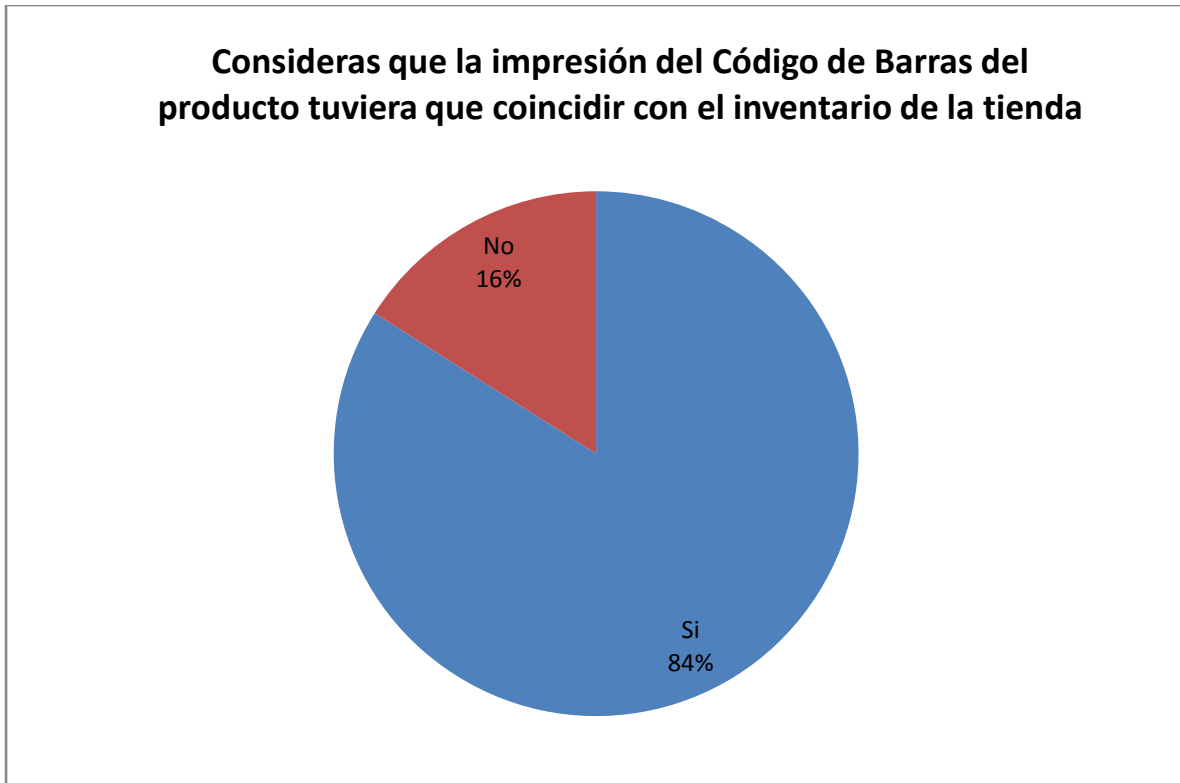
Únicamente el 24% de los encuestados comenta contar un sistema confiable y seguro de POS, el 60% restante comenta que el sistema es ineficiente debido a que este tiene mal asignados los precios

Pregunta 5



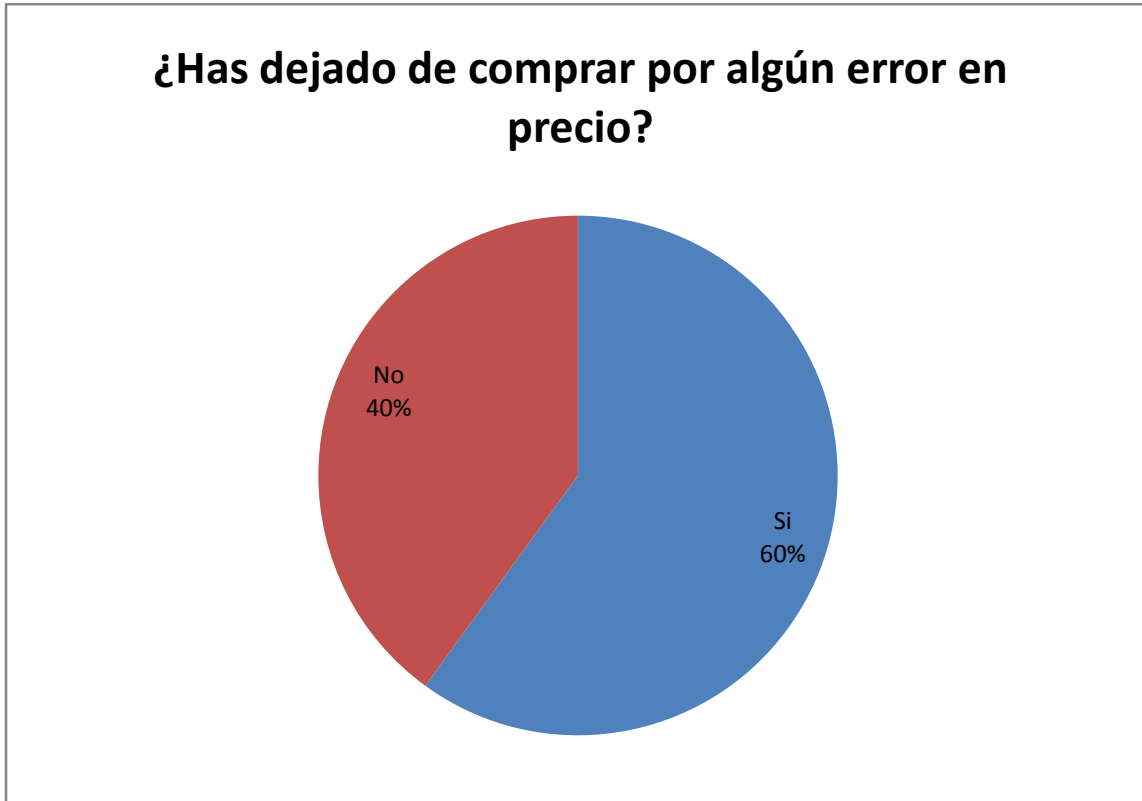
El 60% de los encuestados comenta haber tenido por lo menos 1 vez algún problema con lo que respecta al código de barras del producto.

Pregunta 6



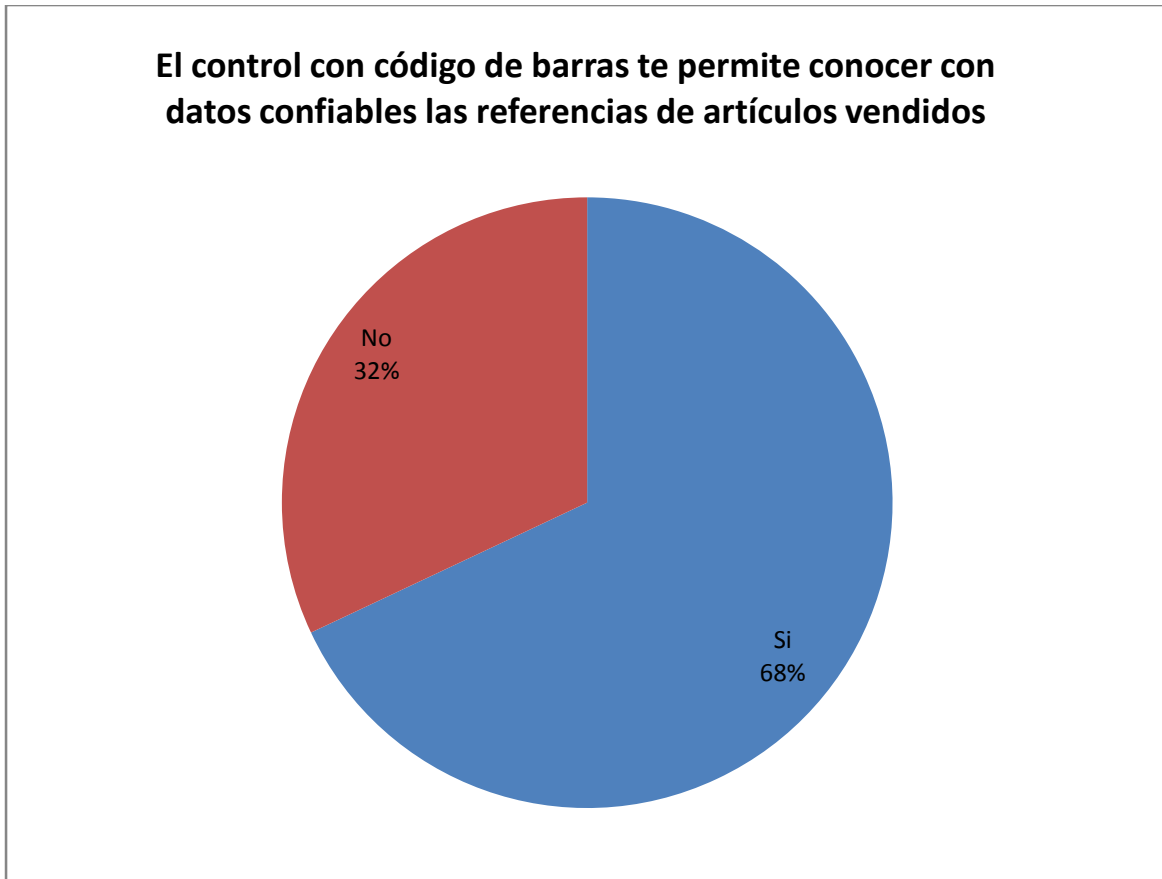
El 84% de los encuestados coincide que el código de barras con el que viene etiquetado el producto debe de ser el código con el que ese dado de alta en el sistema

Pregunta 7



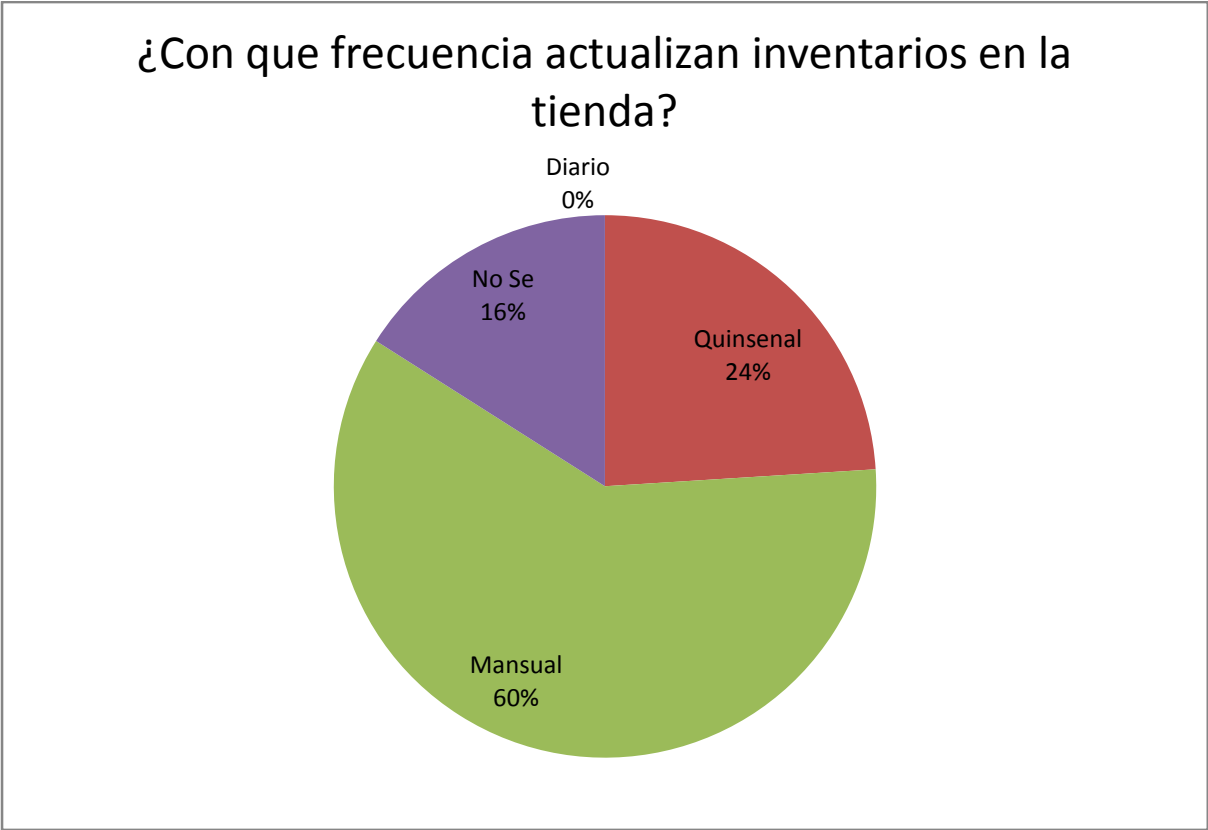
El 60% de los encuestados nos comenta que por lo menos en 1 ocasión el consumidor ha dejado de comprar debido a que el precio o producto no coinciden con el producto que representa ese código en el sistema POS

Pregunta 8



El 68% de los encuestados nos indica que si los datos estuvieran cargados correctamente en el sistema POS, el escaneo de los códigos de barras nos brindaría una fuente confiable de información

Pregunta 9



2.5. Análisis y Recomendaciones

El estudio de mercado arrojó que el 76% de los encuestados menciona que el sistema con el que actualmente cuentan es ineficiente para la operación del día a día, la mayoría de los sistemas son hechos especialmente a las necesidades de cada compañía.

La mayoría de los productos ofertados tienen problemas con su estructura de código de barras – sistema POS, ya que los productos presentan errores en el precio del sistema en comparación con los ofertados en las etiquetas, lo que repercute directamente en las ganancias ya que los usuarios al ver esto no compran o no regresan argumentando un mal servicio.

En conclusión es necesaria una nueva política de emisión, inserción de códigos en las bases de datos de los sistemas POS que administre las altas de nuevos productos, dándole una visibilidad no mayor a 3 meses en la rotación con un buen análisis de rotación o de 80-20, así como una mejora en el manejo de Inventarios en más de dos códigos de barras en un mismo producto.

3. Planeación

El proceso de desarrollo del software se compone de varios ciclos de desarrollo. Cada ciclo está compuesto de las siguientes fases:

- **Inicio.** Revisión del plan de desarrollo por los miembros del Equipo de Trabajo para lograr un entendimiento común del proyecto y para obtener el compromiso de su realización.
- **Requerimientos.** Conjunto de actividades cuya finalidad es obtener la documentación de la Especificación de Requerimientos y Plan de Pruebas de Sistema, para conseguir un entendimiento común entre el cliente y el proyecto.
- **Análisis y diseño.** Conjunto de actividades en las cuales se analizan los requerimientos específicos para producir una descripción de la estructura de los componentes de software, la cual servirá de base para la construcción. Como resultado se obtiene la documentación de análisis y diseño y plan de pruebas de Integración.
- **Construcción.** Conjunto de Actividades para producir componente(s) de software que correspondan al análisis y diseño, así como la realización de pruebas unitarias. Como resultado se obtienen el (los) componente(s) de software probados.
- **Integración y Pruebas.** Conjunto de actividades para integrar y probar los componentes de software, basados en los planes de pruebas de Integración y se sistema, con la finalidad de obtener el software que satisfaga los requerimientos especificados. Se genera la versión final del Manual de Usuario, Manual de Operación y Manual de Mantenimiento. Como resultado se obtiene el producto de software probado y documentado.

- **Cierre.** Integración final de software generada en las fases para su entrega. Identificación y documentación de las Lecciones Aprendidas. Generación de Reportes de Mediciones y Sugerencias de Mejora.

Para generar los productos de cada una de estas fases se realizaran las siguientes actividades:

- Distribución de tareas, se asignaran responsabilidades de cada miembro del Equipo de trabajo de acuerdo al Plan de Desarrollo.
- Producción, verificación, validación o prueba de los productos así como su corrección correspondiente.
- Generación de Reporte Horas.

Objetivos:

- Lograr que el sistema de Código de Barras sea consistente con los productos de entrada en cada fase de un ciclo de desarrollo mediante las actividades de verificación, validación y pruebas.³
- Sustentar la realización de ciclos posteriores o proyectos de mantenimiento futuro mediante la integración de la configuración del software del ciclo actual.⁴
- Llevar a cabo las actividades de las fases mediante el cumplimiento del plan de desarrollo.⁵

Indicadores.

- (O₁) En cada fase de un ciclo se efectúan todas las actividades de verificación validación o pruebas, así como las correcciones correspondientes.⁶
- (O₂) La configuración de software está integrada por los productos generados en el ciclo⁷
- (O₃) Las Actividades planificadas en cada fase de un ciclo se realizaran a lo establecido en el plan de desarrollo.⁸

³ En lo posterior se denominara O₁

⁴ En lo posterior se denominara O₂

⁵ En lo posterior se denominara O₃

⁶ En lo posterior se denominara I₁

⁷ En lo posterior se denominara I₂

⁸ En lo posterior se denominara I₃

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
0	Desarrollo de software	128 días	lun 07/04/08	mié 01/10/08		
1	Análisis y requisitos del software	13 días	lun 07/04/08	mié 23/04/08		
2	Realizar análisis de necesidades	5 días	lun 07/04/08	vie 11/04/08		Analista
3	Borrador de las especificaciones preliminares del software	3 días	lun 14/04/08	mié 16/04/08	2	Analista
4	Desarrollar presupuesto preliminar	2 días	jue 17/04/08	vie 18/04/08	3	Jefe de proyecto
5	Incorporar los comentarios a las especificaciones del software	1 día	lun 21/04/08	lun 21/04/08	4	Analista
6	Calcular los tiempos y fechas de entrega	2 días	mar 22/04/08	mié 23/04/08	5	Jefe de proyecto
7	Análisis completado	0 días	mié 23/04/08	mié 23/04/08	6	
8	Diseñar	14.5 días	jue 24/04/08	mié 14/05/08		
9	Especificaciones preliminares del software	2 días	jue 24/04/08	vie 25/04/08	7	Analista
10	Desarrollar especificaciones de funcionamiento	5 días	lun 28/04/08	vie 02/05/08	9	Analista
11	Desarrollar prototipo basado en las especificaciones de funcionamiento	4 días	lun 05/05/08	jue 08/05/08	10	Analista
12	Revisar especificaciones de funcionamiento	2 días	vie 09/05/08	lun 12/05/08	11	Administración
13	Incorporar comentarios a las especificaciones de funcionamiento	1 día	mar 13/05/08	mar 13/05/08	12	Administración
14	Obtener aprobación para continuar	4 horas	mié 14/05/08	mié 14/05/08	13	Administración, Jefe de pro
15	Diseño completado	0 días	mié 14/05/08	mié 14/05/08	14	
16	Desarrollo	64 días	mié 14/05/08	mar 12/08/08		
17	Revisar especificaciones de funcionamiento	6 días	mié 14/05/08	jue 22/05/08	15	Programador
18	Identificar parámetros de diseño modular y de componentes separados	7 días	jue 22/05/08	lun 02/06/08	17	Programador
19	Asignar el personal de desarrollo	1 día	lun 02/06/08	mar 03/06/08	18	Programador
20	Desarrollar el código	50 días	mar 03/06/08	mar 12/08/08	19	Programador
21	Pruebas de los programadores (depuración primaria)	20 días	vie 20/06/08	jue 17/07/08	20FC-75%	Programador
22	Desarrollo completado	0 días	jue 17/07/08	jue 17/07/08	21	
23	Pruebas	73.5 días	mié 14/05/08	lun 25/08/08		
24	Desarrollar planes de pruebas de unidades usando las especificaciones	4 días	mié 14/05/08	mar 20/05/08	15	Personal de pruebas
25	Desarrollar planes de pruebas de integración usando las especificaciones	4 días	mié 14/05/08	mar 20/05/08	15	Personal de pruebas
26	Pruebas de unidades	15 días	vie 18/07/08	jue 07/08/08		
27	Revisar el código modular	5 días	vie 18/07/08	jue 24/07/08	24,22	Personal de pruebas
28	Probar si los módulos de los componentes se ajustan a las especificaciones	2 días	vie 25/07/08	lun 28/07/08	22,27	Personal de pruebas
29	Identificar anomalías y anotarlas en las especificaciones del producto	3 días	mar 29/07/08	jue 31/07/08	28	Personal de pruebas
30	Modificar código	3 días	vie 01/08/08	mar 05/08/08	29	Personal de pruebas
31	Volver a probar el código modificado	2 días	mié 06/08/08	jue 07/08/08	30	Personal de pruebas
32	Pruebas de unidades completadas	0 días	jue 07/08/08	jue 07/08/08	31	
33	Pruebas de integración	12 días	vie 08/08/08	lun 25/08/08		
34	Probar la integración de los módulos	5 días	vie 08/08/08	jue 14/08/08	32	Personal de pruebas
35	Identificar anomalías y anotarlas en las especificaciones	2 días	vie 15/08/08	lun 18/08/08	34	Personal de pruebas
36	Modificar código	3 días	mar 19/08/08	jue 21/08/08	35	Personal de pruebas
37	Volver a probar el código modificado	2 días	vie 22/08/08	lun 25/08/08	36	Personal de pruebas
38	Pruebas de integración completadas	0 días	lun 25/08/08	lun 25/08/08	37	
39	Curso	70.5 días	mié 14/05/08	mié 20/08/08		
40	Desarrollar las especificaciones del curso para los usuarios finales	3 días	mié 14/05/08	lun 19/05/08	15	Formadores
41	Desarrollar las especificaciones del curso para el personal de soporte al	3 días	mié 14/05/08	lun 19/05/08	15	Formadores
42	Identificar la metodología de los cursos (utilizando PC, aulas, etc.)	2 días	mié 14/05/08	vie 16/05/08	15	Formadores
43	Desarrollar materiales de formación	3 sem.	vie 18/07/08	jue 07/08/08	40,22,41,42	Formadores
44	Realizar estudio de posibilidades de uso de los cursos	4 días	vie 08/08/08	mié 13/08/08	43	Formadores
45	Finalizar los materiales de los cursos	3 días	jue 14/08/08	lun 18/08/08	44	Formadores
46	Desarrollar mecanismo para impartir los cursos	2 días	mar 19/08/08	mié 20/08/08	45	Formadores
47	Materiales de los cursos completados	0 días	mié 20/08/08	mié 20/08/08	46	
48	Documentación	59 días	mié 14/05/08	mar 05/08/08		
58	Fase piloto	95 días	jue 24/04/08	mié 03/09/08		
65	Implantación	11 días	jue 04/09/08	jue 18/09/08		
72	Revisión posterior a la implementación	9 días	vie 19/09/08	mié 01/10/08		
77	Plantilla de desarrollo de software completada	0 días	mié 01/10/08	mié 01/10/08	76	

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
48	<input type="checkbox"/> Documentación	59 días	mié 14/05/08	mar 05/08/08		
49	Desarrollar especificación de ayuda	1 día	mié 14/05/08	jue 15/05/08	15	Escritores técnicos
50	Desarrollar el sistema de ayuda	3 sem.	mar 08/07/08	mar 29/07/08	49,20FC-50%	Escritores técnicos
51	Revisar la documentación de la ayuda	3 días	mar 29/07/08	vie 01/08/08	50	Escritores técnicos
52	Incorporar los comentarios a la documentación de la ayuda	2 días	vie 01/08/08	mar 05/08/08	51	Escritores técnicos
53	Desarrollar las especificaciones de los manuales de usuario	2 días	mié 14/05/08	vie 16/05/08	15	Escritores técnicos
54	Desarrollar los manuales de usuario	3 sem.	mar 08/07/08	mar 29/07/08	53,20FC-50%	Escritores técnicos
55	Revisar toda la documentación para el usuario	2 días	mar 29/07/08	jue 31/07/08	54	Escritores técnicos
56	Incorporar comentarios a la documentación de usuario	2 días	jue 31/07/08	lun 04/08/08	55	Escritores técnicos
57	Documentación completada	0 días	mar 05/08/08	mar 05/08/08	56,52	
58	<input type="checkbox"/> Fase piloto	95 días	jue 24/04/08	mié 03/09/08		
59	Designar grupo de pruebas	1 día	jue 24/04/08	jue 24/04/08	7	Jefe de proyecto
60	Desarrollar el mecanismo de entrega del software	1 día	vie 25/04/08	vie 25/04/08	59	
61	Instalar y distribuir el software	1 día	mar 26/08/08	mar 26/08/08	38,60,57,47	Grupo de implantación
62	Obtener comentarios de los usuarios	1 sem	mié 27/08/08	mar 02/09/08	61	Grupo de implantación
63	Evaluar la información de las pruebas	1 día	mié 03/09/08	mié 03/09/08	62	Grupo de implantación
64	Fase piloto completada	0 días	mié 03/09/08	mié 03/09/08	63	
65	<input type="checkbox"/> Implantación	11 días	jue 04/09/08	jue 18/09/08		
66	Determinar la estrategia final de distribución	2 días	jue 04/09/08	vie 05/09/08	64	Grupo de implantación
67	Desarrollar la metodología de distribución	3 días	lun 08/09/08	mié 10/09/08	66	Grupo de implantación
68	Afianzar recursos de implantación	2 días	jue 11/09/08	vie 12/09/08	67	Grupo de implantación
69	Dar cursos al personal de soporte	3 días	lun 15/09/08	mié 17/09/08	68	Grupo de implantación
70	Distribuir el software	1 día	jue 18/09/08	jue 18/09/08	69	Grupo de implantación
71	Implantación completada	0 días	jue 18/09/08	jue 18/09/08	70	
72	<input type="checkbox"/> Revisión posterior a la implementación	9 días	vie 19/09/08	mié 01/10/08		
73	Documentar la experiencia adquirida	3 días	vie 19/09/08	mar 23/09/08	71	Jefe de proyecto
74	Distribuir a los integrantes del equipo	3 días	mié 24/09/08	vie 26/09/08	73	Jefe de proyecto
75	Crear equipo de mantenimiento del software	3 días	lun 29/09/08	mié 01/10/08	74	Jefe de proyecto
76	Revisión posterior a la implementación completada	0 días	mié 01/10/08	mié 01/10/08	75	
77	Plantilla de desarrollo de software completada	0 días	mié 01/10/08	mié 01/10/08	76	

4. Ejecución y Control del Proyecto.

En este capítulo se tratan a detalles las actividades mencionadas en el Capítulo Anterior.

4.1. Análisis y requisitos del software

En esta fase se revisa toda la documentación preliminar, se documentan alcances, y roles del grupo de trabajo. Se realizan cuestionarios para la obtención de las necesidades reales que va a cubrir el software.

4.1.1. Realizar análisis de necesidades

La Ingeniería de Requerimientos cumple un papel primordial en el proceso de producción de software, ya que enfoca un área fundamental: la definición de lo que se desea producir.

Su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, el comportamiento del sistema; de esta manera, se pretende minimizar los problemas relacionados al desarrollo de sistemas.

Los casos de uso se han convertido en la técnica más utilizada a nivel mundial para el levantamiento y la comunicación clara y eficiente de los requisitos (mejor conocidos como “requerimientos”) para el desarrollo de sistemas.

4.1.2. Borrador de las especificaciones preliminares del software

Para este software se plantearon varios casos de uso con vario posibles clientes (Oxxo, Circulo K), se utilizo como apoyo el cuestionario de levantamiento de especificaciones de Microsoft⁹.

Esta actividad nos dio como resultado un documento denominado “Especificación de Requerimientos Ver. 1” donde se plasman todos los procesos que debe llevar el software, los datos que lo deben de integrar.

4.1.3. Desarrollar presupuesto preliminar

Se desarrollo una propuesta de diseño de software entre los integrantes del grupo de trabajo, donde se definen los alcances, los módulos que va a contener el sistema, los roles de cada miembro del equipo de trabajo.

4.1.4. Incorporar los comentarios a las especificaciones del software

Se realiza una revisión al documento de “Especificación de Requerimientos Ver. 1” donde los miembros de grupo de trabajo, hacen correcciones y cometarios a este documento.

4.1.5. Calcular los tiempos y fechas de entrega

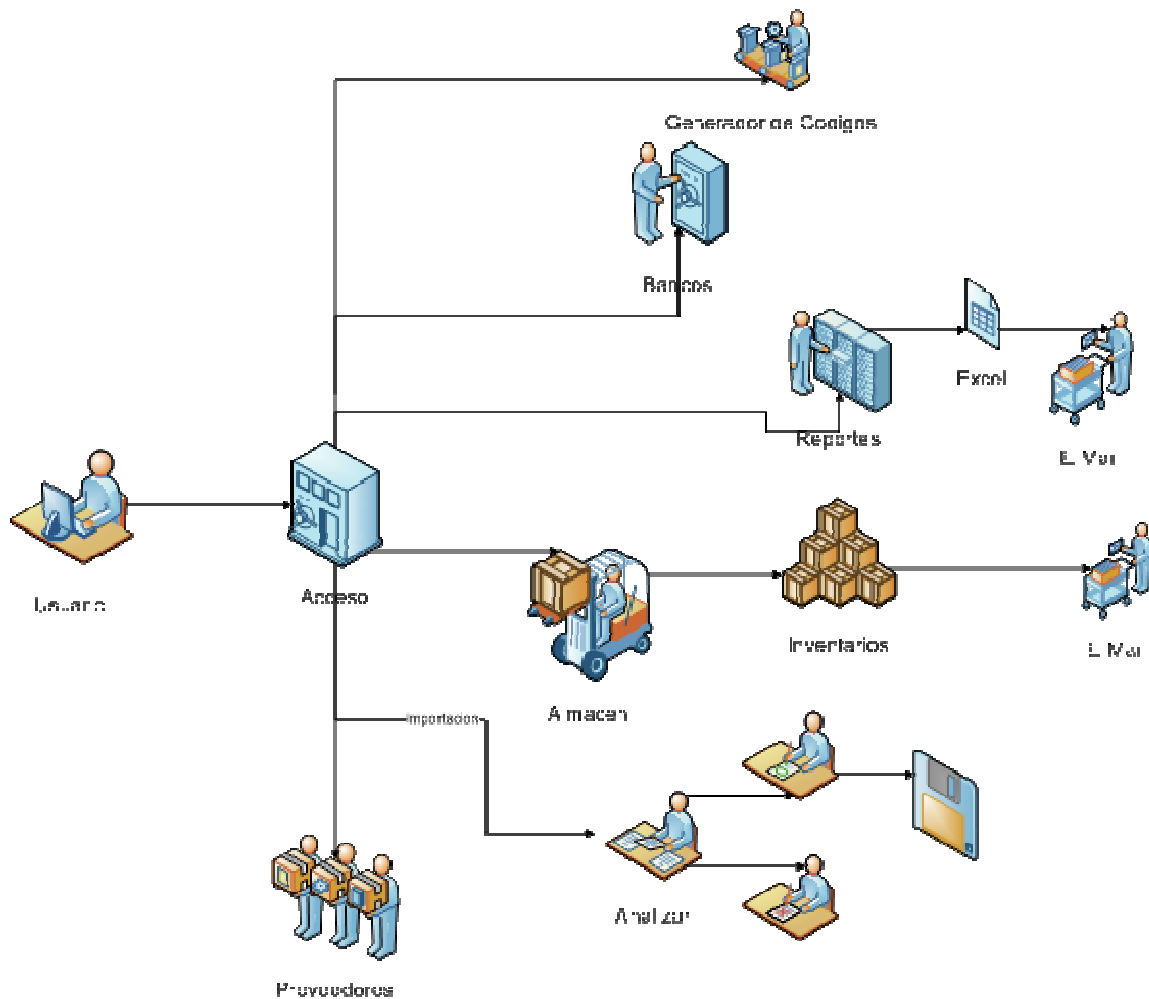
El equipo de trabajo delimita y acuerda tiempos para cada una de las actividades realizadas, se generan las tablas de actividades y diagrama de Gantt.

⁹ Modelo de Procesos para la Industria del Software

4.2. Diseñar

4.2.1. Especificaciones preliminares del software

En esta fase se retoma el documento de las “Especificación de Requerimientos Ver. 1” se realiza un bosquejo en papel de las pantallas del software, las validaciones que tendrá, los roles de usuario, y un diagrama del software.



4.2.2. Desarrollar especificaciones de funcionamiento

Se desarrolla un documento donde se especifican todas las funciones que tendrá el software, los roles que contendrá el sistema.

El software contendrá los siguientes Módulos o Acciones:

Acciones

- Alta de Productos. Modulo en el que se darán de alta al sistema nuevos productos, contara con los siguientes campos:
 - Nombre de Producto.
 - Código de Barras del Producto. Código de barras que el fabricante le asigna a su producto
 - Código Interno. Código que el administrador le asigna al producto¹⁰
 - Descripción. Breve descripción del producto.
 - Familia. Familia a la que pertenece el producto.
 - Imagen y tipo de código
 - Imagen del Producto.

- Centro de Control. En este modulo se pueden realizar actualizaciones a los productos dados de alta, imprimir etiquetas de los productos
- Generador de Códigos. Este modulo permitirá generar códigos de barras
- Familia de Productos. Este modulo nos permite crear o actualizar las familias de producto
- Salir. Cerrara el Sistema

Utilidades

- Compactar Base de Datos. Compactara la base de datos, para reducir el espacio en disco de esta

- Importación. Esta utilidad nos permitirá hacer la carga masiva de o actualización de :
 - Productos
 - Familias

¹⁰ Se recomienda que sea el mismo código de la etiqueta

- Exportación. Nos permitirá exportar los siguientes datos :
 - Productos
 - Familias
- Comparación. Esta utilidad revisara la basa de datos de productos o el archivo a importar para detectar los siguientes errores:
 - Duplicidad en códigos de Producto
 - Duplicidad de códigos internos
 - Duplicidad en nombres de producto

4.2.3. Desarrollar prototipo basado en las especificaciones de funcionamiento

Se desarrolla un prototipo conceptual del software, es decir a través de un diagrama de bloques se genera el software colocando la funcionalidad de cada bloque, se hacen correcciones a la base de datos.

4.2.4. Revisar especificaciones de funcionamiento

Todo el equipo de trabajo revisa el prototipo de software en base al documento de “Especificación de Requerimientos Ver. 1”, se genera una lluvia de ideas para realizar correcciones al prototipo.

4.2.5. Incorporar comentarios a las especificaciones de funcionamiento

Se incorporan los comentarios y acuerdos llegados en la revisión de especificaciones de funcionamiento al archivo “Especificación de Requerimientos Ver. 1” y se genera la versión final del archivo. Como resultado se obtiene el documento “Especificación de -Requerimientos”.

4.2.6. Obtener aprobación para continuar

En esta fase se llegan a acuerdos entre todos los miembros del equipo de trabajo de la plataforma en la que se va a desarrollar, el motor de base de datos que se va utilizar, para poder dar el banderazo de salida a la fase de desarrollo (Comenzar a escribir el código Fuente)

En este proyecto, el equipo de trabajo decidió desarrollar en Microsoft Visual Basic 6 SP 6, con una base de Datos en Access.

4.3. Desarrollo

En esta fase del proyecto es donde se lleva a cabo el desarrollo del software, se genera el código fuente de este y se realizan las primeras pruebas por el equipo de desarrolladores designado.

4.3.1. Revisar especificaciones de funcionamiento

El equipo de desarrolladores revisa nuevamente el documento de especificación de requerimientos, hace las anotaciones pertinentes para el proceso de desarrollo.

4.3.2. Identificar parámetros de diseño modular y de componentes separados

El sistema cuenta con los siguientes componentes:

- Base de Datos
- Conector de Base de Datos
- Interfaz de Usuario
- Pantallas
- Variables de Sesión
- Variables Globales

4.3.3. Desarrollar el código

En esta parte de la fase de Desarrollo se desarrollo el código fuente del software, se comienza con el manual de usuario y de administrador.

4.3.3.1. Diseño de la Base de Datos

En base a el documento de Especificación de Requerimientos Ver. Final, el equipo de desarrolladores genero la siguiente base de Datos:

Nombre de la Base de Datos: DBCodigos

Estructura de las Tablas.

tblProductos	
Nombre de Columna	Tipo
IdProducto	Autonumérico
Producto	Texto (50)
Código Interno	Texto (50)
CodigoBarras	Texto (50)
Codificación	Texto (50)
IdCodificación	Número
Descripción	Texto (50)
Imagen	Objeto OLE
IdFamilia	Número

tblFamilias

Nombre de Columna	Tipo
IdFamilia	Autonumérico
Familia	Texto (50)

4.3.3.2. Variables y Constantes de Entorno

En el sistema se utilizan las siguientes variables:

```
Module Variables
```

```
'Variables de Conexion
```

```
Public oDataReader As OleDb.OleDbDataReader ' crear DataReader
```

```
Public oDataReader1 As OleDb.OleDbDataReader
```

```
Public oConexion As New OleDb.OleDbConnection()
```

```
Public Const CadenaConexion = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=|DataDirectory|\CB\DBCodigo.mdb;Persist Security Info=True"
```

```
'Variables de Sesión
```

```
Public LoginUsuario As String
```

```
Public NombreUsuario As String
```

```
Public Password As String
```

```
Public InicioSesion As Boolean
```

```
'Variables de Encriptacion
```

```
Public Encriptado As String
```

```
Public Desencriptado As String
```

```
'Datos de Inventario
```

```
Public Numero As Integer
```

```
Public Titulo As String
```

```
Public Texto As String
```

```
'Permisos
```

```

Public Permisos As Boolean = False
'Pedido
Public ArticuloPedido As String
Public EmpaqPedido As String
Public CantidadArtPedido As Integer
Public CtoArtPedido As Single
Public CveArtPedido As String
Public NumPedido As String
Public Cancelar As Boolean
End Module

```

4.3.3.3. *Modulo Familias de Producto.*

La función de este modulo es crear, editar o eliminar las categorías, los datos que se reciben de este modulo son:

Categoría, este campo es el nombre de la Familia (Lácteos, abarrotes, etc.), su correspondencia en la base de datos es el campo **Familia** de la tabla tblFamilia.

Las funciones que se generan en este modulo son:

Función Carga_Familias, esta función carga todas las familias que actualmente están dadas de alta en la base de datos.

CF:

```

Private Sub CargaCategorias()
    Dim SQL As String
    Dim Key As String
    Dim Aux As Integer
    IFamilia.Items.Clear()
    Try
        Aux = 0
        oConexion.ConnectionString = CadenaConexion
    End Try
End Sub

```

```

SQL = "Select * from Familias order by Descrip"
Dim oComando As New OleDb.OleDbCommand(SQL, oConexion)
oConexion.Open()
oDataReader = oComando.ExecuteReader()
While oDataReader.Read()
    Aux = Aux + 1
    Key = "F" + Format(Aux, "000")
    IFamilia.Items.Add(Key, "", 0)
    IFamilia.Items(Key).SubItems.Add(oDataReader("Descrip"))
End While
oDataReader.Close()
oConexion.Close()
Catch oExcep As OleDb.OleDbException
    MessageBox.Show("Error al conectar con datos" & _
ControlChars.CrLf & _
oExcep.Message & ControlChars.CrLf & _
oExcep.Source())
End Try
End Sub

```

Función Agregar_Familia, esta función se encarga de validar la familia que se está intentando dar de alta, si la validación es correcta, agrega la familia a la base de datos

CF:

```

Private Sub Categoria_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles Categoria.KeyPress

    If e.KeyChar = ""c Then

        e.Handled = True

        Exit Sub

    End If

```

```

If e.KeyChar = Convert.ToChar(Keys.Return) Then
    If Categoria.Text = "" Then
        MsgBox("Ingrese la categoria", MsgBoxStyle.Critical, "Error al
Agregar")
    Exit Sub
Else
    Cuenta = TablaFamilias.CuentaFam(Categoria.Text)
    If Cuenta = 0 Then
        TablaFamilias.Agregar(Categoria.Text)
        CargaCategorias()
        Categoria.Text = ""
    Else
        MsgBox("La Categoria ya fue ingresada", MsgBoxStyle.Critical,
"Error al Ingresar Categoria")
    End If
End If
End If
End Sub

```

Función Editar_Familia, esta función permite editar una familia que ya se encuentra dada de alta.

Función Eliminar, esta función valida que se desee eliminar la familia a través de un mensaje, si la respuesta es afirmativa, elimina la familia.

CF:

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

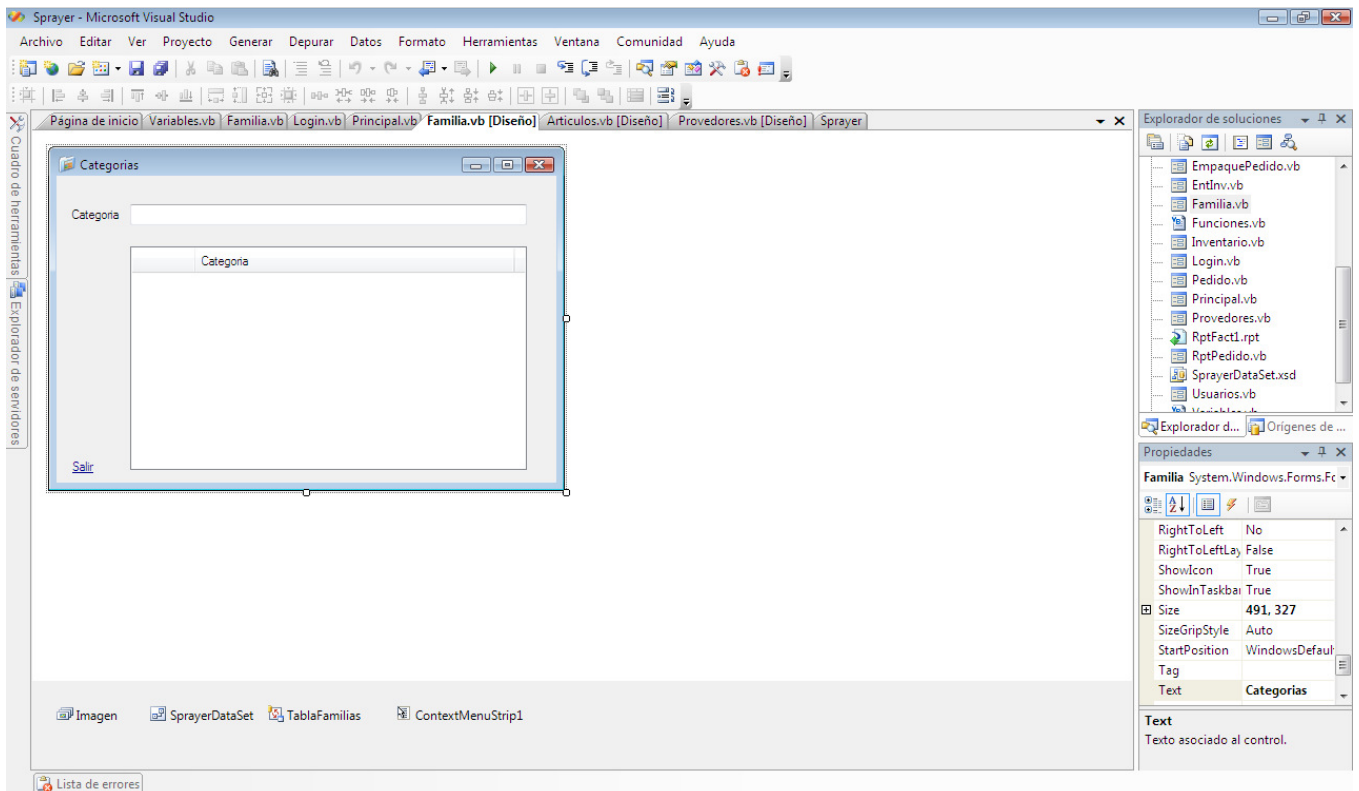
```
    Dim Cat As String
```

```
    Cat = IFamilia.SelectedItems.Item(0).SubItems.Item(1).Text
```

```
    TablaFamilias.Eliminar(Cat)
```

```
    CargaCategorias()
```

```
End Sub
```



Modulo de Productos

En este modulo se pueden desarrollar las siguientes acciones:

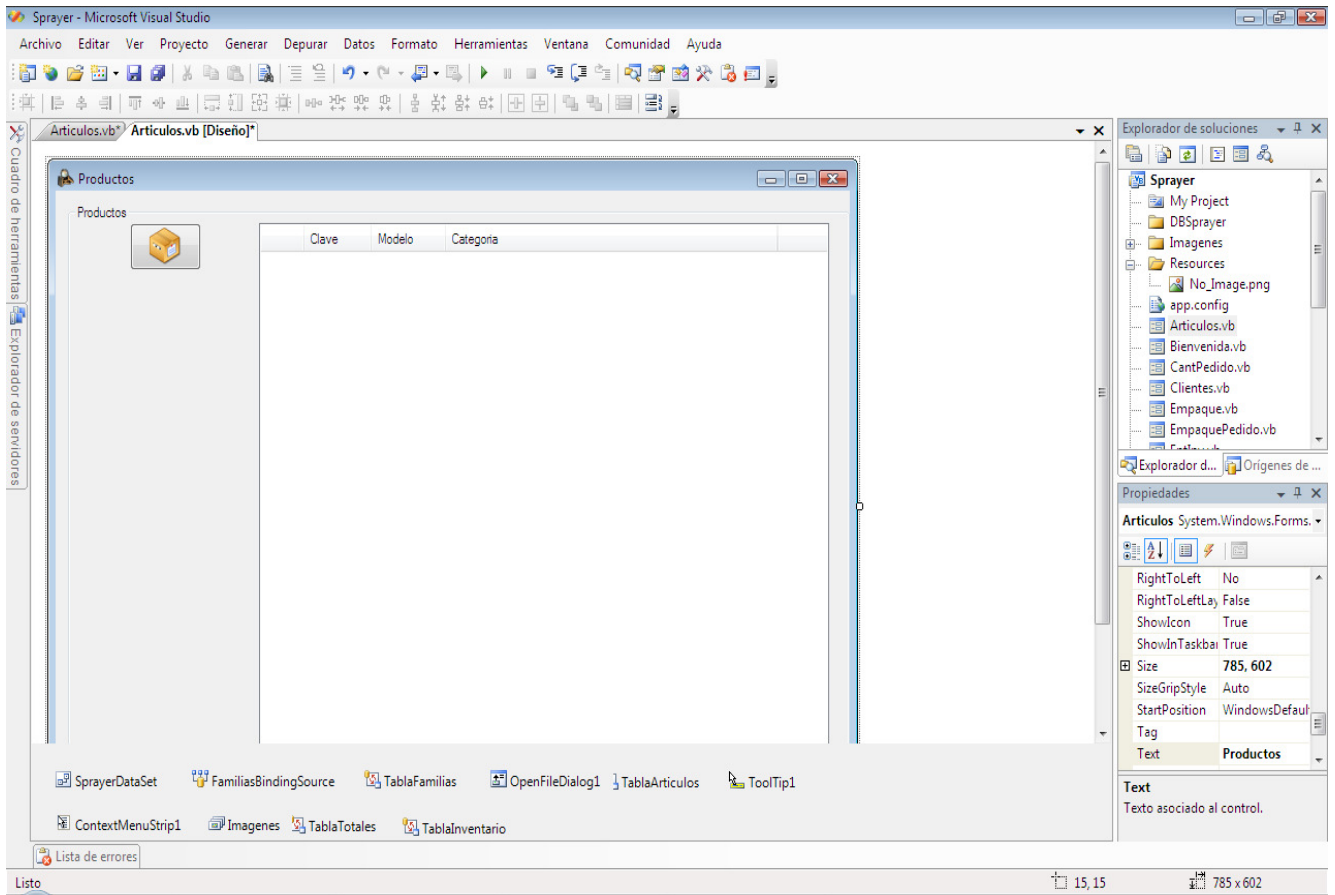
Ver todos los productos que se encuentran dados de alta en la base de datos.

CF:

```

Private Sub CargaArticulos()
    Dim SQL As String
    Dim Key As String
    Dim Aux As Integer
    IArticulos.Items.Clear()
    Try
        Aux = 0
        oConexion.ConnectionString = CadenaConexion
        SQL = "Select Clave,Modelo,Categoria from Articulos order by
Clave"
        Dim oComando As New OleDb.OleDbCommand(SQL, oConexion)
        oConexion.Open()
        oDataReader = oComando.ExecuteReader()
        While oDataReader.Read()
            Aux = Aux + 1
            Key = "P" + Format(Aux, "000")
            IArticulos.Items.Add(Key, "", 0)
            IArticulos.Items(Key).SubItems.Add(oDataReader("Clave"))
            IArticulos.Items(Key).SubItems.Add(oDataReader("Modelo"))
            IArticulos.Items(Key).SubItems.Add(oDataReader("Categoria"))
        End While
        oDataReader.Close()
        oConexion.Close()
    Catch oExcep As OleDb.OleDbException
        ' si se produce algún error,
        ' lo capturamos mediante el objeto
        ' de excepciones particular para
        ' el proveedor de OLEDB
        MessageBox.Show("Error al conectar con datos" & _
ControlChars.CrLf & _
oExcep.Message & ControlChars.CrLf & _
oExcep.Source())
    End Try
End Sub

```



Dar de Alta un nuevo producto, esta acción se ejecuta en 2 partes, la primera es verificar que el código que se está ingresando no se encuentre en la base de datos, y la otra es guardar el producto en la base de datos.

CF:

```

Private Sub Guardar_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Guardar.Click

    Dim C1 As Integer
    Dim C2 As Integer
    Dim C3 As Integer
    Dim C4 As Integer
    Dim N As Integer
  
```

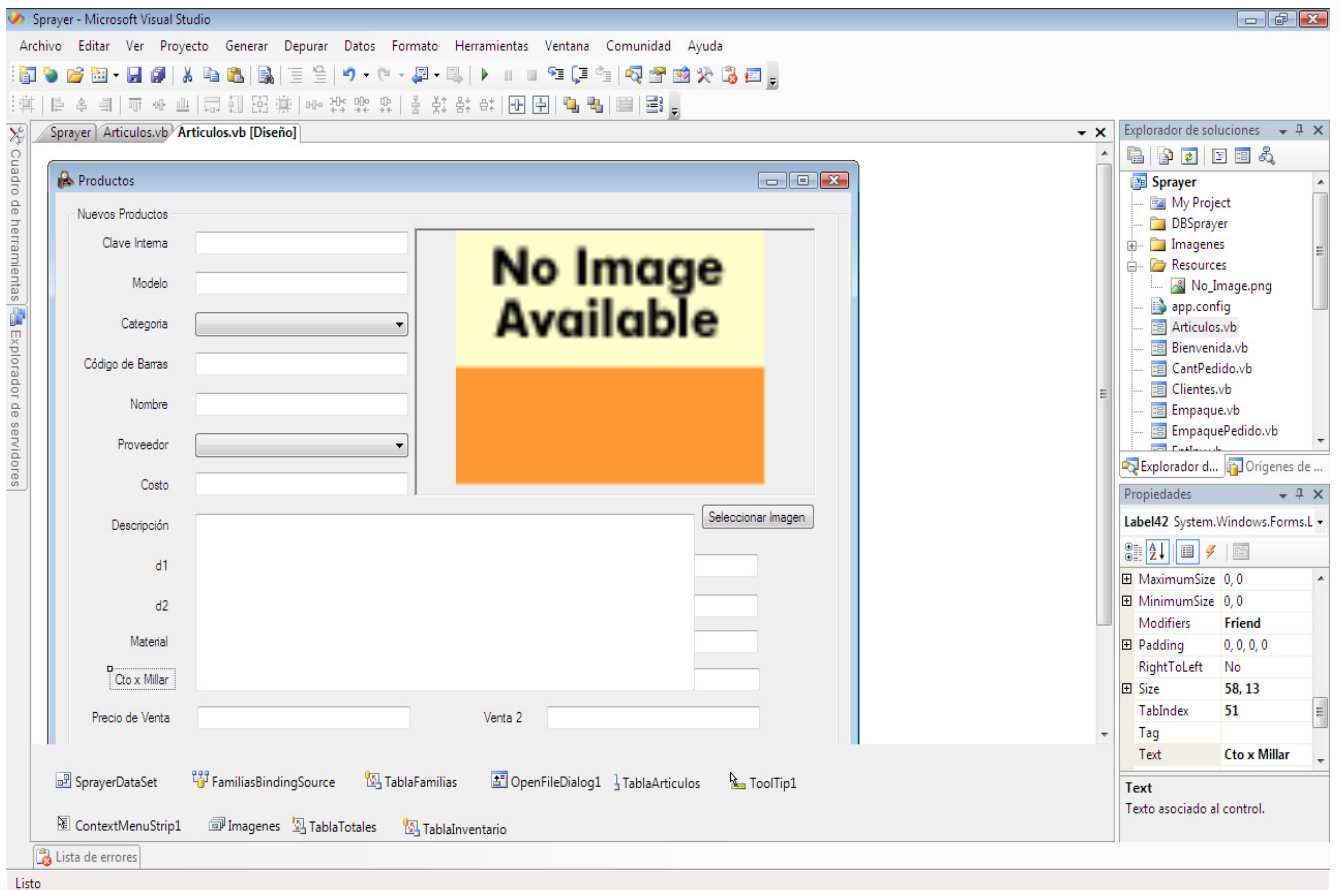
```

If Clave.Text = "" Then
    MsgBox("Es necesario Ingresar la clave del producto",
MsgBoxStyle.Critical, "Error al Agregar")
    Exit Sub
End If
N = TablaArticulos.CuentaClave(Clave.Text)
If N <> 0 Then
    MsgBox("La Clave ya fue dada de Alta")
    Clave.Text = ""
    Clave.Focus()
    Exit Sub
End If
If Modelo.Text = "" Then
    MsgBox("Es necesario Ingresar el Modelo del producto",
MsgBoxStyle.Critical, "Error al Agregar")
    Exit Sub
End If
PhatImagen = PBIimagen.ImageLocation
If Costo1.Text = "" Then
    C1 = 0
Else
    C1 = Costo1.Text
End If

If Costo2.Text = "" Then
    C2 = 0
Else
    C2 = Costo2.Text
End If
If Costo3.Text = "" Then

```

```
        C3 = 0
    Else
        C3 = Costo3.Text
    End If
    If Costo4.Text = "" Then
        C4 = 0
    Else
        C4 = Costo4.Text
    End If
    TablaArticulos.Agregar(Clave.Text, Modelo.Text, C1, C2, C3 _
        , C4, d1.Text, d2.Text, Capacidad.Text, Categoria.Text _
        , Color.Text, Fi.Text, h1.Text, h2.Text, Material.Text,
Orif1.Text _
        , Orif2.Text, PhatImagen, TD.Text, CtoMillar.Text,
CtoPza.Text)
    TablaTotales.Agregar(Clave.Text, 0)
    TablaInventario.AgregaEmpaque(Clave.Text)
    Limpia()
End Sub
```



Actualizar un producto, este modulo permite actualizar los datos de los productos excepto el código Interno, y se necesitan privilegios de administrador para poder cambiar el Código de Barras del Producto.

CF:

```

Private Sub Actua_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Actua.Click

    Dim C1 As Integer
    Dim C2 As Integer
    Dim C3 As Integer
    Dim C4 As Integer
    Dim PathActualizar As String
    If ActClave.Text = "" Then
        MsgBox("Es necesario Ingresar la clave del producto",

```

```
MsgBoxStyle.Critical, "Error al Agregar")
    Exit Sub
End If
If ActModelo.Text = "" Then
    MsgBox("Es necesario Ingresar el Modelo del producto",
MsgBoxStyle.Critical, "Error al Agregar")
    Exit Sub
End If
PhatImagen = PBIImagen.ImageLocation
If ActCosto1.Text = "" Then
    C1 = 0
Else
    C1 = ActCosto1.Text
End If

If ActCosto2.Text = "" Then
    C2 = 0
Else
    C2 = ActCosto2.Text
End If

If ActCosto3.Text = "" Then
    C3 = 0
Else
    C3 = ActCosto3.Text
End If

If ActCosto4.Text = "" Then
    C4 = 0
```

```

Else
    C4 = ActCosto4.Text
End If

PathActualizar = ActPB.ImageLocation
TablaArticulos.Actualizar(ActModelo.Text, C1, C2, C3 _
    , C4, ActD1.Text, ActD2.Text, ActCapacidad.Text,
ActCat.Text _
    , ActColor.Text, ActFi.Text, ActH1.Text, ActH2.Text,
ActMaterial.Text, ActOrif1.Text _
    , ActOrif2.Text, PhatActualizar, ActTD.Text,
ActCtoMillar.Text _
    , ActCtoPza.Text, ActClave.Text)

GBArticulos.Visible = False
GBProductos.Visible = True
GBActualizar.Visible = False
CargaArticulos()
Me.AcceptButton = Me.NuevoProducto

End Sub

```

Eliminar un producto, esta función elimina productos de la base de datos.

CF:

```

Private Sub Eliminar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Eliminar.Click

    Dim Producto As String

    Dim n As Integer

    If IArticulos.SelectedItems.Count = 0 Then

        Exit Sub
    End If

```

```

End If

Producto = IArticulos.SelectedItems.Item(0).SubItems.Item(1).Text
n = TablaArticulos.Eliminar(Producto)

CargaArticulos()

End Sub

```

4.3.3.4. Centro de control.

Este modulo es el encargado de mostrar las categorías y los productos que pertenecen a esta, realizar el cambio de categoría e imprimir la etiqueta de código de barras.

En este modulo se pueden realizar las siguientes acciones:

Ver Todos los productos y la categoría a la que pertenecen

CF:

```

Private Sub CargaInicial()
    Dim Sql As String
    Sql = "SELECT Nombre, Barras, SKU, Categoria " + _
        "FROM Categorias INNER JOIN Productos ON "
+ _
        "(Categorias.Id_Cat = Productos.Id_Cat) order
by Categoria"
    DgClientes.Rows.Clear()
    Try

        oConexion.ConnectionString = CadenaConexion
        Dim oComando As New
        MySql.Data.MySqlClient.MySqlCommand(Sql, oConexion)
        oConexion.Open()
        oDataReader = oComando.ExecuteReader()
    End Try
End Sub

```

```
While oDataReader.Read()  
    Dim Nombre As String  
    Dim RFC As String  
    Dim Ejecutivo As String  
    Dim Cve As String  
    Dim n As String  
    Nombre = oDataReader("Nombre")  
    Codigo = oDataReader("Barras")  
    Categoria = oDataReader("Categoria")  
    SKU = oDataReader("SKU")  
    DgClientes.Rows.Add()  
    n = DgClientes.RowCount - 1  
    DgClientes.Rows(n).Cells(0).Value = SKU  
    DgClientes.Rows(n).Cells(1).Value = Nombre  
    DgClientes.Rows(n).Cells(2).Value = Categoria  
    DgClientes.Rows(n).Cells(3).Value = Codigo
```

End While

```
oDataReader.Close()
```

```
oConexion.Close()
```

```
Catch oExcep As OleDb.OleDbException
```

```
' si se produce algún error,
```

```
' lo capturamos mediante el objeto
```

```
' de excepciones particular para
```

```
' el proveedor de OLEDB
```

```
MessageBox.Show("Error al conectar con datos" & _
```

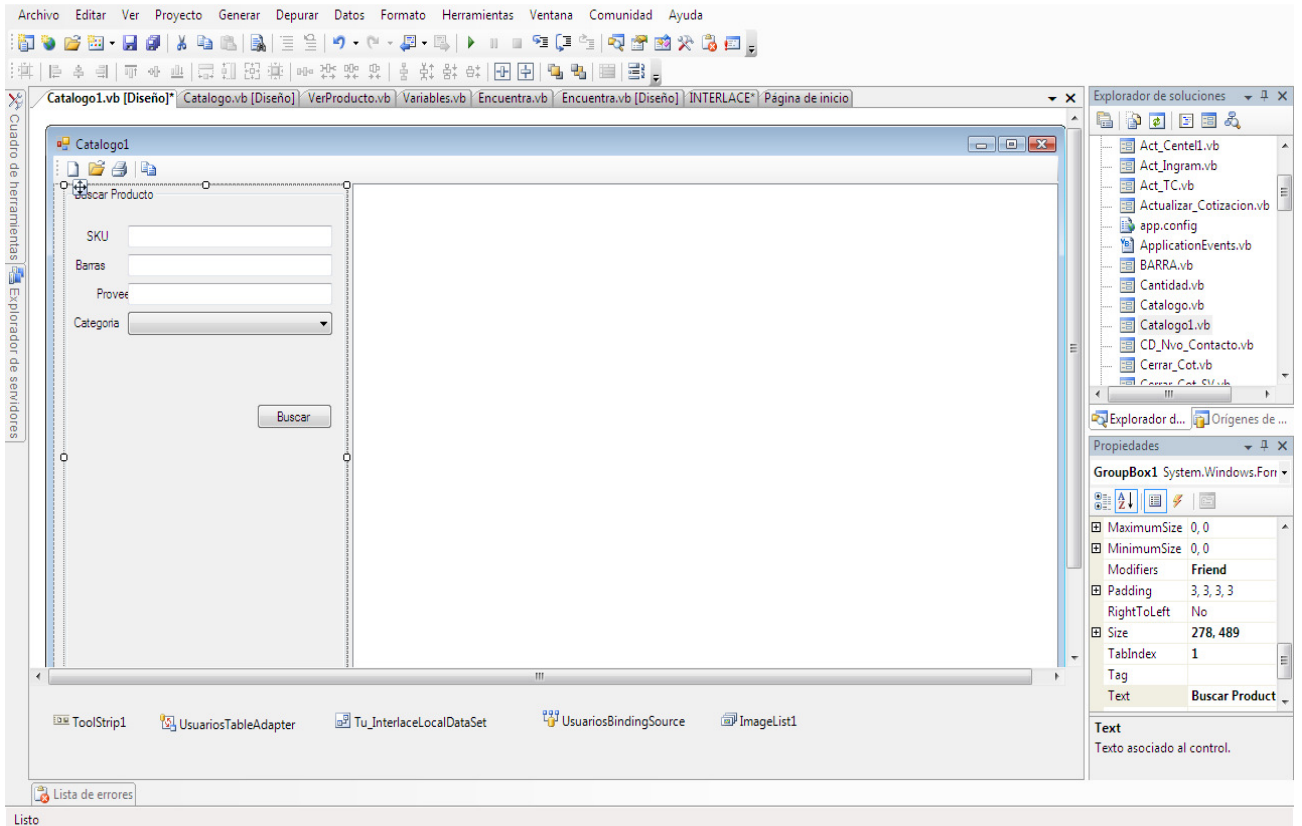
```
ControlChars.CrLf & _
```

```
oExcep.Message & ControlChars.CrLf & _
```

```
oExcep.Source())
```

```
oConexion.Close()
```

```
End Try
End Sub
```



Imprimir etiqueta de código de barras.

CF:

```
Private Sub cEtiqueta_Click()  
    Dim Descripcion As String  
    If AbrePuertoSerial = True Then  
        IMessage.Caption = "Generando Etiqueta..."  
        IMessage.Refresh()  
    End If  
End Sub
```

```

If strDescripcion = "2" Then ' Descripcion larga
    If Len(tDescripcionActivo.Text) > MAXDescrpEtiqueta Then
        Descripcion = Mid(Replace(tDescripcionActivo.Text, Chr(34), "
"), 1, MAXDescrpEtiqueta)
    Else
        Descripcion = Replace(tDescripcionActivo.Text, Chr(34), " ")
    End If
Else 'Descripcion Corta
    Descripcion = Replace(tDescripcionCorta.Text, Chr(34), " ")
End If

ImprimeEtiqueta(TituloEtiqueta, tClaveActivo.Text, Descripcion)
IMensaje.Caption = ""
IMensaje.Refresh()
CierraPuertoSerial()

End If
End Sub

```



4.3.3.5. Modulo de Usuarios

En este modulo se pueden realizar las siguientes acciones:

Crear y actualizar Usuarios.

CF:

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

    Dim Log As String
    Dim n, m As Integer
    Dim Contra As String = ""
    Dim Per As String = CreaPermisos()
    Log = TxtLogin.Text
    n = TablaUsuarios.Checa_Login(Log)
    If n <> 0 Then
        m = ChecaErrores()
        If m <> 0 Then
            Exit Sub
        End If

        TablaUsuarios.Act_Datos(TxtNombre.Text, 0, Per, TxtPuesto.Text,
Log)
        If PBFirma.ImageLocation.ToString <> "" Then

            End If
            n = TablaDatos.Checa_Login(Log)
            If n = 0 Then
                TablaDatos.Insert(TxtLogin.Text, TxtTel.Text, TxtCelular.Text,
TxtMail.Text, TxtCalle.Text, _
                    TxtExterior.Text, TxtInterior.Text, TxtColonia.Text,
TxtDel.Text, TxtEdo.Text, TxtCP.Text)
```

```

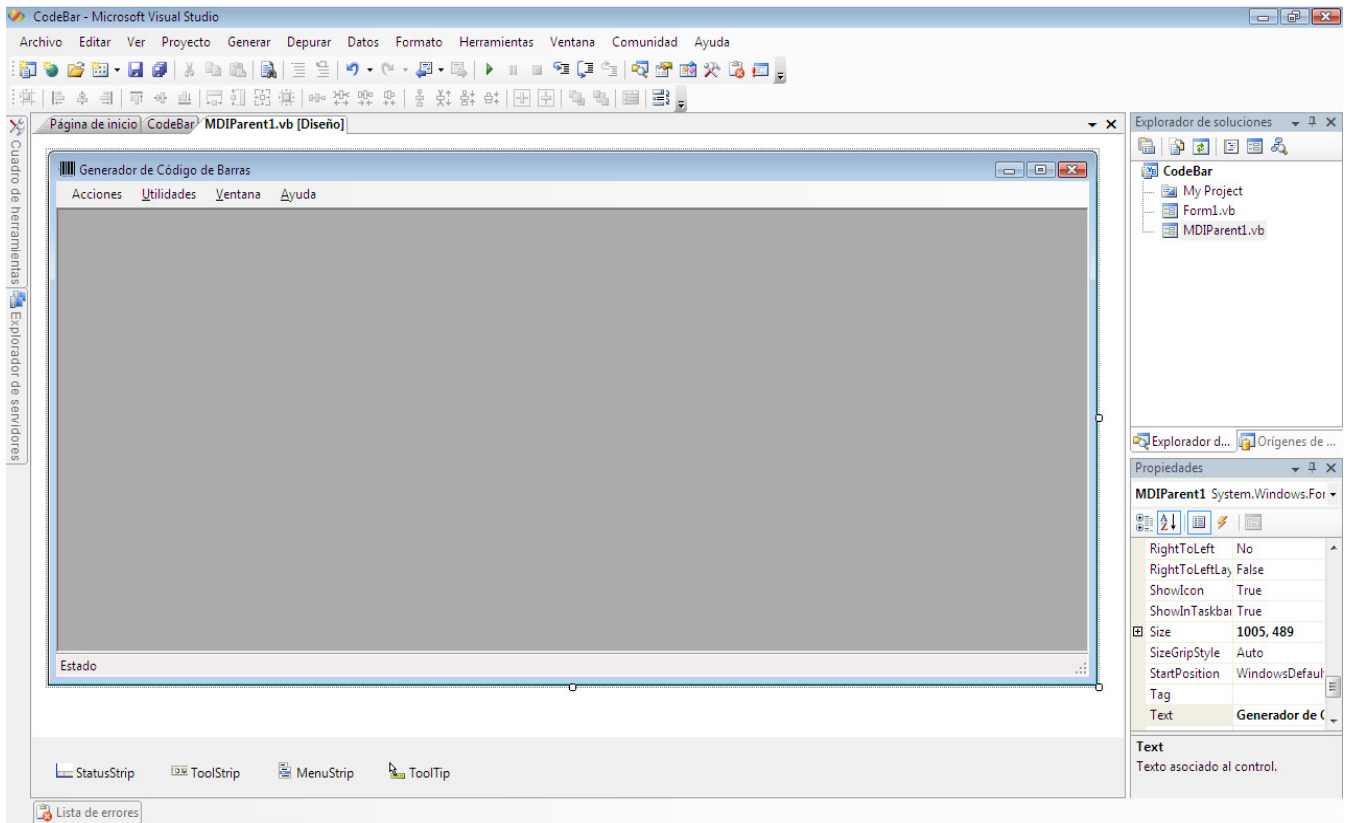
Else
    TablaDatos.Act_Datos(TxtTel.Text, TxtCelular.Text, TxtMail.Text,
TxtCalle.Text, TxtExterior.Text, _
        TxtInterior.Text,    TxtColonia.Text,    TxtDel.Text,
TxtEdo.Text, TxtCP.Text, Log)
End If

Else
    m = ChecaErrores()
    If m <> 0 Then
        Exit Sub
    End If
    While Contra = ""
        Contra = InputBox("Ingrese la contraseña al Sistema",
"Contraseña")
    End While
    Encripta(Contra, 1)
    TablaUsuarios.Insert(TxtLogin.Text, TxtNombre.Text, Encriptado, 0,
Per,                                TxtPuesto.Text,
ImageToByte(Image.FromFile(PBFirma.ImageLocation.ToString)),    0,
TxtMail.Text)
    TablaDatos.Insert(TxtLogin.Text,    TxtTel.Text,    TxtCelular.Text,
TxtMail.Text, TxtCalle.Text, _
        TxtExterior.Text,    TxtInterior.Text,    TxtColonia.Text,
TxtDel.Text, TxtEdo.Text, TxtCP.Text)
    Cargalni()
    MsgBox("Alta de Usuario Correcta", MsgBoxStyle.Information, "Alta
de Usuario")
    DGVUsuarios.Enabled = True
End If
End Sub

```

4.3.3.6. Pantalla Principal

La pantalla principal contendrá una barra de menús donde se encontraran los módulos con los que cuenta el sistema, así como las funciones que puede realizar. En la pantalla principal se cargarán las sub-pantallas de cada proceso



CF:

```
Public Class Principal
```

```
Private Sub AbreUsuarios(ByVal sender As Object, ByVal e As EventArgs)  
Handles MAUsuarios.Click, BotonUsuarios.Click
```

```
    Usuarios.MdiParent = Me
```

```
    Usuarios.Show()
```

End Sub

Private Sub AbreClientes(ByVal sender As Object, ByVal e As EventArgs)
Handles MAClientes.Click, BotonClientes.Click

Clientes.MdiParent = Me

Clientes.Show()

End Sub

Private Sub AbreArticulos(ByVal sender As Object, ByVal e As EventArgs)
Handles MAArticulos.Click, BotonArticulos.Click

Articulos.MdiParent = Me

Articulos.Show()

End Sub

Private Sub AbrePedidos(ByVal sender As Object, ByVal e As EventArgs)
Handles BotonPedido.Click, MAArticulos.Click

Pedido.MdiParent = Me

Pedido.Show()

End Sub

Private Sub AbreEmpaque(ByVal sender As Object, ByVal e As EventArgs)
Handles MAEmpaque.Click, BotonEmpaque.Click

Empaque.MdiParent = Me

Empaque.Show()

End Sub

Private Sub AbreCategorias(ByVal sender As Object, ByVal e As EventArgs)
Handles MACategorias.Click, BotonCategorias.Click

```
Familia.MdiParent = Me
```

```
Familia.Show()
```

```
End Sub
```

```
Private Sub Abreinventario(ByVal sender As Object, ByVal e As EventArgs)  
Handles MAInventario.Click, BotonInventario.Click
```

```
    Inventario.MdiParent = Me
```

```
    Inventario.Show()
```

```
End Sub
```

```
Private Sub ExitToolsStripMenuItem_Click(ByVal sender As Object, ByVal e As  
EventArgs) Handles ExitToolStripMenuItem.Click
```

```
    Desconecta()
```

```
    Global.System.Windows.Forms.Application.Exit()
```

```
End Sub
```

```
Private Sub Conecta()
```

```
    Login.MdiParent = Me
```

```
    Login.Show()
```

```
End Sub
```

```
Private Sub Desconecta()
```

```
    Dim Tipo As String
```

```
    EditMenu.Enabled = False
```

```
    ViewMenu.Enabled = False
```

```
    ToolsMenu.Enabled = False
```

```
    WindowsMenu.Enabled = False
```

```
HelpMenu.Enabled = False
MAUsuarios.Enabled = False
MAClientes.Enabled = False
MAArticulos.Enabled = False
MAEmpaque.Enabled = False
MACategorias.Enabled = False
MAInventario.Enabled = False
MACerrarSesion.Enabled = False
BotonUsuarios.Enabled = False
BotonClientes.Enabled = False
BotonArticulos.Enabled = False
BotonEmpaque.Enabled = False
BotonCategorias.Enabled = False
BotonInventario.Enabled = False
BotonCerrarSesion.Enabled = False
Tipo = LoginUsuario + "CoNec" + Password + "o0"
Encripta(Tipo, 1)
Tipo = Encriptado
If Len(LoginUsuario) <> 0 Then
    TablaUsuarios.ActualizarEdo(Tipo, False, LoginUsuario)
End If
LoginUsuario = ""
End Sub
```

```
Private Sub InicioSesion(ByVal sender As Object, ByVal e As EventArgs)
Handles MAInicioSesion.Click, BotonIniciarSesion.Click
    Conecta()
```

End Sub

Private Sub CierraSesion(ByVal sender As Object, ByVal e As EventArgs) Handles MACerrarSesion.Click, BotonCerrarSesion.Click

Desconecta()

Me.BotonIniciarSesion.Enabled = True

Me.MAInicioSesion.Enabled = True

End Sub

Private Sub CutToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs) Handles CutToolStripMenuItem.Click

' Utilice My.Computer.Clipboard para insertar el texto o las imágenes seleccionadas en el Portapapeles

End Sub

Private Sub CopyToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs) Handles CopyToolStripMenuItem.Click

' Utilice My.Computer.Clipboard para insertar el texto o las imágenes seleccionadas en el Portapapeles

End Sub

Private Sub PasteToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs) Handles PasteToolStripMenuItem.Click

'Utilice My.Computer.Clipboard.GetText() o My.Computer.Clipboard.GetData para recuperar la información del Portapapeles.

End Sub

Private Sub ToolBarToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs) Handles ToolBarToolStripMenuItem.Click

```
Me.ToolStrip.Visible = Me.ToolBarToolStripMenuItem.Checked
```

```
End Sub
```

```
Private Sub StatusBarToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs) Handles StatusBarToolStripMenuItem.Click
```

```
Me.StatusStrip.Visible = Me.StatusBarToolStripMenuItem.Checked
```

```
End Sub
```

```
Private Sub CascadeToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs) Handles CascadeToolStripMenuItem.Click
```

```
Me.LayoutMdi(MdiLayout.Cascade)
```

```
End Sub
```

```
Private Sub TileVerticalToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs) Handles TileVerticalToolStripMenuItem.Click
```

```
Me.LayoutMdi(MdiLayout.TileVertical)
```

```
End Sub
```

```
Private Sub TileHorizontalToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs) Handles TileHorizontalToolStripMenuItem.Click
```

```
Me.LayoutMdi(MdiLayout.TileHorizontal)
```

```
End Sub
```

```
Private Sub ArrangelconsToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs) Handles ArrangelconsToolStripMenuItem.Click
```

```
Me.LayoutMdi(MdiLayout.Arrangelcons)
```

```
End Sub
```

```
Private Sub CloseAllToolStripMenuItem_Click(ByVal sender As Object, ByVal e
```

```

As EventArgs) Handles CloseAllToolStripMenuItem.Click
    ' Cierre todos los formularios secundarios del primario.
    For Each ChildForm As Form In Me.MdiChildren
        ChildForm.Close()
    Next
End Sub

Private Sub Principal_EnabledChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.EnabledChanged
    If Permisos = True Then
        'ChecaPermisos()
    End If
End Sub

Private Sub Principal_FormClosing(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing
    Desconecta()
End Sub

Private Sub Principal_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
    Dim Aux As Integer = 0
    Dim Ubicacion As String
    Dim Contra As String
    Dim Tipo As String
    Aux = TablaUsuarios.ChecaLogin("Admin")
    If Aux = 0 Then
        Ubicacion = "Sprayer S.A. de C.V."
        Encripta("admin", 1)
    End If
End Sub

```

```

Contra = Encriptado
Tipo = "Admin" + "CoNec" + "admin" + "o0"
Encripta(Tipo, 1)
Tipo = Encriptado
TablaUsuarios.Agregar("Admin", "Administrador", Contra, Tipo,
Date.Today, _
Ubicacion, False)
End If
Desconecta()
Conecta()
End Sub
Private Sub ChecaPermisos()
Dim SQL As String
Dim SQL2 As String
Dim Ubicacion As String
Dim Crear As Boolean
Dim Cerrar As Boolean
Dim Area As Boolean
Dim Observaciones As Boolean
If LoginUsuario = "Admin" Or LoginUsuario = "admin" Then
Exit Sub
End If

BotonUsuarios.Enabled = False
MAUsuarios.Enabled = False

Try

```

```

oConexion.ConnectionString = CadenaConexion

SQL = "Select Gpo_Descrip from Usuarios where Login=" +
Campo(LoginUsuario)

Dim oComando As New OleDb.OleDbCommand(SQL, oConexion)
oConexion.Open()
oDataReader = oComando.ExecuteReader()
While oDataReader.Read()
    Ubicacion = oDataReader("Gpo_Descrip")
End While
oDataReader.Close()

SQL2 = "Select * from Grupos where Gpo_Descrip=" + Campo(Ubicacion)
Dim oComando1 As New OleDb.OleDbCommand(SQL2, oConexion)
Dim oDataReader1 As OleDb.OleDbDataReader
oDataReader1 = oComando1.ExecuteReader()
While oDataReader1.Read()
    Crear = oDataReader1("Crear")
    Cerrar = oDataReader1("Cerrar")
    Observaciones = oDataReader1("Observaciones")
    Area = oDataReader1("Eliminar")
End While
oDataReader1.Close()
oConexion.Close()

Catch oExcep As OleDb.OleDbException
    ' si se produce algún error,
    ' lo capturamos mediante el objeto
    ' de excepciones particular para

```

```

' el proveedor de OLEDB
MessageBox.Show("Error al conectar con datos" & _
ControlChars.CrLf & _
oExcep.Message & ControlChars.CrLf & _
oExcep.Source())

End Try

End Sub

End Class

```

4.3.3.7. Función de Importación

Esta función se encarga de importar datos ya sea desde un archivo de Excel (*.xls,*.xlsx), un archivo de texto (*.txt) o un archivo (*.csv), en este proceso además se valida que no se agregue al sistema algún otro código de barras con un SKU diferente, al terminar el proceso el sistema emite un reporte con el total de productos importados, las nuevas categorías agregadas, y que códigos no se ingresaron debido a que generarían una duplicidad de códigos.

CF:

```

Public Sub Actualizar_Ingram_Nvo(ByVal Path As String)
'Despublica()
Dim registro As DatImportacion
Dim Producto As DatProductos
Dim TC As Double = 10.9
Dim coma As Integer
Dim Reg As String
Dim Descrip As String
Dim n, Medidas As Integer
Dim TablaCategorias As New
tu_InterlaceLocalDataSetTableAdapters.CategoriasTableAdapter
Dim TablaProductos As New
tu_InterlaceLocalDataSetTableAdapters.productosTableAdapter
Dim Imagen As String = "Y:\No_Image.png"

```

```

Dim Precio As String
Dim CostoMXP As String
Dim Marca As String
Dim Factor As Double
Dim Actual As Double
Dim SQL As String
Dim Existe As Integer
Dim Special As Double
Dim PE, SP As String
Dim Publicado As Integer

If InStr(Path, "eCatalog") = 0 Then
    MsgBox("El archivo es incorrecto", MsgBoxStyle.Information, "Error")
    Exit Sub
End If
Dim sr As New System.IO.StreamReader(Path)
sr.ReadLine()
TablaIngram.Limpia_Encon(0)
Thread.Sleep(500)

BARRA.Show()
BARRA.ProgressBar1.Maximum = 1000000
BARRA.Label2.Text = 1
n = 2

'Cominzan a leer los Productos

While sr.EndOfStream = False
    BARRA.Label2.Text = n - 1
    BARRA.Refresh()
    Reg = sr.ReadLine.Replace(Chr(34), "")
    coma = InStr(Reg, ",")
    registro.Sku = Mid(Reg, 1, coma - 1).TrimEnd
    'MsgBox(registro.Sku)
    Reg = Mid(Reg, coma + 1)

    coma = InStr(Reg, ",")
    registro.CvePto = Mid(Reg, 1, coma - 1).TrimEnd
    Reg = Mid(Reg, coma + 1)

```

```
coma = InStr(Reg, ",")
registro.UPC = Mid(Reg, 1, coma - 1).TrimEnd
Reg = Mid(Reg, coma + 1)
```

```
coma = InStr(Reg, ",")
registro.Proveedor = Mid(Reg, 1, coma - 1).TrimEnd
Reg = Mid(Reg, coma + 1)
```

```
coma = InStr(Reg, ",")
registro.Cve_Proveedor = Mid(Reg, 1, coma - 1).TrimEnd
Reg = Mid(Reg, coma + 1)
```

```
coma = InStr(Reg, ",")
registro.Descrip1 = Mid(Reg, 1, coma - 1).TrimEnd
Reg = Mid(Reg, coma + 1)
```

```
coma = InStr(Reg, ",")
registro.Descrip2 = Mid(Reg, 1, coma - 1).TrimEnd
Reg = Mid(Reg, coma + 1)
```

```
coma = InStr(Reg, ",")
registro.NewCategory = Mid(Reg, 1, coma - 1).TrimEnd
Reg = Mid(Reg, coma + 1)
```

```
coma = InStr(Reg, ",")
registro.NewSubCategory = Mid(Reg, 1, coma - 1).TrimEnd
Reg = Mid(Reg, coma + 1)
```

```
coma = InStr(Reg, ",")
registro.NewCPUCode = Mid(Reg, 1, coma - 1).TrimEnd
Reg = Mid(Reg, coma + 1)
```

```
coma = InStr(Reg, ",")
registro.Tipo = Mid(Reg, 1, coma - 1).TrimEnd
Reg = Mid(Reg, coma + 1)
```

```
coma = InStr(Reg, ",")
registro.Rotacion = Mid(Reg, 1, coma - 1).TrimEnd
Reg = Mid(Reg, coma + 1)
```

```
coma = InStr(Reg, ",")
registro.ProductType = Mid(Reg, 1, coma - 1).TrimEnd
Reg = Mid(Reg, coma + 1)

coma = InStr(Reg, ",")
registro.ProductStatus = Mid(Reg, 1, coma - 1).TrimEnd
Reg = Mid(Reg, coma + 1)

coma = InStr(Reg, ",")
registro.Weight = Mid(Reg, 1, coma - 1).TrimEnd
Reg = Mid(Reg, coma + 1)

coma = InStr(Reg, ",")
registro.Height = Mid(Reg, 1, coma - 1).TrimEnd
Reg = Mid(Reg, coma + 1)

coma = InStr(Reg, ",")
registro.Width = Mid(Reg, 1, coma - 1).TrimEnd
Reg = Mid(Reg, coma + 1)

coma = InStr(Reg, ",")
registro.Costo = Mid(Reg, 1, coma - 1).TrimEnd
Reg = Mid(Reg, coma + 1)

coma = InStr(Reg, ",")
registro.Lista = Mid(Reg, 1, coma - 1).TrimEnd
Reg = Mid(Reg, coma + 1)

coma = InStr(Reg, ",")
registro.CostoEsp = Mid(Reg, 1, coma - 1).TrimEnd
Reg = Mid(Reg, coma + 1)

coma = InStr(Reg, ",")
registro.PEF = Mid(Reg, 1, coma - 1).TrimEnd
Reg = Mid(Reg, coma + 1)

coma = InStr(Reg, ",")
registro.ItemTypeInvisibleCode = Mid(Reg, 1, coma - 1).TrimEnd
Reg = Mid(Reg, coma + 1)
```

```

coma = InStr(Reg, ",")
registro.CurrencyCode = Mid(Reg, 1, coma - 1).TrimEnd
Reg = Mid(Reg, coma + 1)

coma = InStr(Reg, ",")
registro.CurrencyCalculationCode = Mid(Reg, 1, coma - 1).TrimEnd
Reg = Mid(Reg, coma + 1)

'coma = InStr(Reg, ",")
registro.Stock = Reg.TrimEnd
'Reg = Mid(Reg, coma + 1)
Descrip = registro.Descrip1 & " " & registro.Descrip2
'If registro.CvePto = "62105601" Then
'  MsgBox("hOLA")
'End If
If oConexion.State = ConnectionState.Open Then
  oConexion.Close()
End If
Try
  'Cambiar OR por And Cuando Factores Listos
  SQL = "SELECT vendors.Vendor_Name, factores.Factor FROM " + _
        "vendors LEFT OUTER JOIN factores ON (vendors.Vendor_Nbr =
factores.Vendor_Nbr) WHERE " + _
        "factores.`Type`='" + registro.Tipo + "' OR vendors.Vendor_Nbr='" +
registro.Cve_Proveedor + "'"
  oConexion.ConnectionString = CadenaConexion
  Dim oComando As New MySql.Data.MySqlClient.MySqlCommand(SQL,
oConexion)
  oConexion.Open()
  oDataReader = oComando.ExecuteReader()
  While oDataReader.Read()
    If oDataReader("Vendor_Name") IsNot DBNull.Value Then
      Marca = oDataReader("Vendor_Name")
      Marca = Marca.ToUpper
    Else
      Marca = "Generico"
    End If
    If oDataReader("Factor") IsNot DBNull.Value Then
      Factor = oDataReader("Factor")
    Else

```

```

        Factor = 0.229263864
    End If
End While
oDataReader.Close()
oConexion.Close()
Catch oExcep As MySql.Data.MySqlClient.MySqlException
    MessageBox.Show("Error al conectar con datos" & _
        ControlChars.CrLf & _
        oExcep.Message & ControlChars.CrLf & _
        oExcep.Source())
oConexion.Close()
End Try

If registro.CurrencyCode = "USD" Then
    registro.CurrencyCode = "US"
    CostoMXP = Val(registro.Costo) * TC
    Precio = Val(CostoMXP) * (Factor + 1)
    If registro.PEF = "" Then
        Special = 0
        PE = 0
    Else
        Special = Val(registro.CostoEsp) * TC
        Precio = Val(Special) * (Factor + 1)
        PE = 1
    End If
Else
    registro.CurrencyCode = "MN"
    CostoMXP = Val(registro.Costo)
    Precio = Val(CostoMXP) * (Factor + 1)
    If registro.PEF = "" Then
        Special = 0
        PE = 0
    Else
        Special = Val(registro.CostoEsp) * TC
        Precio = Val(Special) * (Factor + 1)
        PE = 1
    End If
End If

If registro.Rotacion = "S" Then

```

```

    SP = 1
    Publicado = 0
    Elself registro.Rotacion = "A" Or registro.Rotacion = "B" Then
        SP = 0
        Publicado = 1
    Else
        SP = 0
        Publicado = 0
    End If

```

'Guardando en Tablas

```

Try
    Existe = TablaIngram.Checa_Alta(registro.Sku)
    If Existe = 0 Then
        Existe = TablaProductos.Checa_Alta(registro.CvePto)
        If Existe <> 0 Then
            TablaIngram.Insert(registro.Sku, _
                registro.CvePto, _
                registro.Lista, _
                registro.Costo, _
                registro.Stock, _
                registro.CurrencyCode, 1, _
                CostoMXP, _
                PE, _
                SP, _
                Special)
            Medidas = TablaMedidas.Checa(registro.CvePto)
            If Medidas <> 0 Then
                TablaMedidas.Insert(registro.CvePto, _
                    registro.Height, _
                    registro.Width, _
                    0, _
                    registro.Weight)
            End If
        'Actualizar Productos
    Try
        'Cambiar OR por And Cuando Factores Listos
        SQL = "SELECT productos.PTO_CATEGORIA,

```

```

productos.PTO_PUBLICADO, " + _
        "productos.PTO_PRECIO,          productos.PTO_MARCA,
grupos.GRP_FACTOR FROM productos " + _
        "LEFT OUTER JOIN      grupos      ON
(productos.PTO_CATEGORIA = " + _
        "grupos.Grp_Phato)where Pto_Cve=" + registro.CvePto + ""
oConexion.ConnectionString = CadenaConexion
Dim          oComando          As          New
MySQL.Data.MySqlCommand(MySqlCommand(SQL, oConexion)
oConexion.Open()
oDataReader = oComando.ExecuteReader()
While oDataReader.Read()
    Producto.Categoria = oDataReader("Pto_Categoria")
    Producto.Factor = oDataReader("Grp_Factor")
    Producto.Precio = oDataReader("Pto_Precio")
    Producto.Publicado = oDataReader("Pto_Publicado")
    Producto.Marca = oDataReader("Maraca")
End While
oDataReader.Close()
oConexion.Close()
Catch oExcep As MySQL.Data.MySqlCommand.MySqlCommandException
    MessageBox.Show("Error al conectar con datos" & _
        ControlChars.CrLf & _
        oExcep.Message & ControlChars.CrLf & _
        oExcep.Source())
oConexion.Close()
End Try
Actual = Val(Producto.Precio) / Val(Precio)
If Actual > 1.5 Then
    TablaIngram.Actualiza_Precios(registro.Lista, registro.Costo,
registro.Stock, _
        "US", 1, registro.Sku)
    TablaIngram.CambioMxp(Val(registro.Costo) * TC, registro.Sku)
    Precio = Val(Precio) * TC
    'TablaProductos.Actaliza_Precio_Publico(Precio,
registro.CvePto)
End If
If registro.Rotacion = "A" Or registro.Rotacion = "B" Then
    Producto.Publicado = 1
End If

```

```

If Producto.Marca = "Generico" Then
    Producto.Marca = Marca.ToUpper
End If
TablaProductos.Importacion(Producto.Publicado, Precio,
Producto.Marca.ToUpper, 0, _
PE, SP, registro.Rotacion, registro.CvePto)

'Fin Actualiza Ptos
Thread.Sleep(10)
Else
TablaProductos.Insert(registro.CvePto, _
    Descrip, _
    Descrip, _
    "Sin Clasificar", _
    Imagen, _
    "X", _
    Imagen, _
    Publicado, _
    Precio, _
    Marca.ToUpper, _
    ImageToByte(Image.FromFile(Imagen)), _
    1, 0, PE, SP, registro.Rotacion)

TablaIngram.Insert(registro.Sku, _
    registro.CvePto, _
    registro.Lista, _
    registro.Costo, _
    registro.Stock, _
    registro.CurrencyCode, 1, _
    CostoMXP, _
    PE, _
    SP, _
    Special)
TablaMenor.Insert(registro.CvePto, "Ingram", CostoMXP)
TablaMedidas.Insert(registro.CvePto, _
    registro.Height, _
    registro.Width, _
    0, _
    registro.Weight)
Thread.Sleep(10)

```

```

End If
Else
'BB = BB + 1
TablaIngram.Importa(registro.Lista, _
                    registro.Costo, _
                    registro.Stock, _
                    registro.CurrencyCode, _
                    PE, SP, _
                    CostoMXP, 1, _
                    Special, _
                    registro.Sku)
Medidas = TablaMedidas.Checa(registro.CvePto)
If Medidas = 0 Then
    TablaMedidas.Insert(registro.CvePto, _
                        registro.Height, _
                        registro.Width, _
                        0, _
                        registro.Weight)

Else

    TablaMedidas.Actualizar(registro.Height, _
                            registro.Width, _
                            0, _
                            registro.Weight, _
                            registro.CvePto)

End If
'Actualizar Productos
Try
'Cambiar OR por And Cuando Factores Listos
SQL = "SELECT      productos.PTO_CATEGORIA,
productos.PTO_PUBLICADO, " + _
      "productos.PTO_PRECIO,          productos.PTO_MARCA,
grupos.GRP_FACTOR FROM productos " + _
      "LEFT OUTER JOIN grupos ON (productos.PTO_CATEGORIA
= " + _
      "grupos.Grp_Phato)where Pto_Cve=" + registro.CvePto + ""
oConexion.ConnectionString = CadenaConexion

```

```

Dim oComando As New
MySql.Data.MySqlClient.MySqlCommand(SQL, oConexion)
oConexion.Open()
oDataReader = oComando.ExecuteReader()
While oDataReader.Read()
    Producto.Categoria = oDataReader("Pto_Categoria")
    Producto.Factor = oDataReader("Grp_Factor")
    Producto.Precio = oDataReader("Pto_Precio")
    Producto.Publicado = oDataReader("Pto_Publicado")
    Producto.Marca = oDataReader("pto_Marca")
End While
oDataReader.Close()
oConexion.Close()
Catch oExcep As MySql.Data.MySqlClient.MySqlException
    MessageBox.Show("Error al conectar con datos" & _
        ControlChars.CrLf & _
        oExcep.Message & ControlChars.CrLf & _
        oExcep.Source())
oConexion.Close()
End Try
Actual = Val(Producto.Precio) / Val(Precio)
If Actual > 1.5 Then
    TablaIngram.Actualiza_Precios(registro.Lista, registro.Costo,
registro.Stock, _
        "US", 1, registro.Sku)
    TablaIngram.CambioMxp(Val(registro.Costo) * TC, registro.Sku)
    If registro.CostoEsp <> "0" Then
        registro.Costo = registro.CostoEsp
    End If
    Precio = (Val(Producto.Factor) + 1) * TC * Val(registro.Costo)
    'TablaProductos.Actaliza_Precio_Publico(Precio, registro.CvePto)
End If
If registro.Rotacion = "A" Or registro.Rotacion = "B" Then
    Producto.Publicado = 1
End If
If Producto.Marca = "Generico" Then
    Producto.Marca = Marca.ToUpper
End If
TablaProductos.Importacion(Producto.Publicado, Precio,
Producto.Marca, 0, _

```

PE, SP, registro.Rotacion, registro.CvePto)

```
'Fin Actualiza Ptos
Thread.Sleep(10)
End If
n = n + 1
Catch
n = n + 1
'NN = NN + 1
End Try
End While
'MXPIngram()
BARRA.Label2.Text = ""
BARRA.Label1.Text = "Actualizando"
Publica()
Despublica()
sr.Close()
'CargaOfertas()
BARRA.Close()
End Sub
```

4.4. Pruebas

4.4.1. Desarrollar planes de pruebas de unidades usando las especificaciones del producto

El equipo de trabajo se reúne y genera un documento en el cual se plasman las pruebas que se realizarán al sistema, los tiempos y los encargados en realizar dichas pruebas. Este plan de pruebas contempla pruebas a nivel de módulos individuales, es decir no contemplan la interacción con otros módulos, estas pruebas están basadas en lo especificado en el documento de “Especificación de Requerimientos.”.

4.4.2. Desarrollar planes de pruebas de integración usando las especificaciones del producto

El equipo de trabajo se reúne y genera un documento en el cual se plasman las pruebas que se realizarán al sistema, los tiempos y los encargados en realizar dichas pruebas. Este plan de pruebas contempla pruebas a nivel de todo el sistema, es decir son pruebas llevadas a cabo con todos los módulos interactuando uno con otra.

4.4.3. Pruebas de unidades

4.4.3.1. Revisar el código modular

Se revisa el código fuente estructura por estructura, para detectar errores o vulnerabilidades en cada uno de los procesos que se efectúan en cada módulo

4.4.3.2. Probar si los módulos de los componentes se ajustan a las especificaciones del producto

Se genera una matriz de pruebas basada en el documento “Especificación de Requerimientos”, y se revisa cada módulo chequeando que cumpla con todos los procesos indicados en la matriz de pruebas.

4.4.3.3. Identificar anomalías y anotarlas en las especificaciones del producto

Se genera un documento en el cual se plasman los errores y las vulnerabilidades encontradas en la revisión.

4.4.3.4. Modificar código

Se procede a realizar las optimizaciones, correcciones y adaptaciones al código fuente del sistema.

4.4.3.5. Volver a probar el código modificado

Se realiza nuevamente las pruebas al sistema basadas en la matriz de pruebas.

4.4.4. Documentación

El equipo de Escritores realiza la documentación para la entrega del software, la cual incluye los siguientes Documentos:

- Manual de Instalación
- Manueta del Operador.
- Manual del Administrador.
- Manual de Fallas y Contratiempos.
- Documentación de Limitantes.
- Documentación de Mejoras.

5. Evaluación de Resultados

5.1. Resultados

Este software se implemento en un minisupers, donde no se tenía un control de la mercancía que se surtía, ni se llevaba un inventario periódico. Al implementar este software se comenzó con realizar una carga inicial de los productos dados de alta en su sistema POS.

Esta primera carga nos entrego que el 23% de los artículos dados de alta no contaban con un código de barras, del 78% restante que si contaba con código de barras se encontró que el 8% de los productos estaba dado de alta más de una vez, el 5% de los productos presentaban el mismo código aunque el producto era diferente.

Se procedió a realizar el etiquetado del 23% de los productos a través del modulo de generador de códigos del software, se dieron de baja los productos duplicados, y se generaron nuevas etiquetas para los productos diferentes que tenían el mismo código de barras.

Al terminar los procesos iniciales, el software redujo el tiempo de toma de inventarios en un 60%, el tiempo de espera de un cliente en una compra se redujo un 30%.

5.2. Conclusiones

Logramos terminar el presente proyecto que se veía casi inalcanzable, por el grado de exigencia que presentó, todo se hizo basado en la realidad y sometándose a las leyes y políticas que nos enmarcan, después de varias sesiones de trabajo ideamos un método realizando un software de buena calidad a un bajo costo, y así satisfacer las necesidades del cliente.

Podemos concluir que al terminar el proyecto, aprendimos realmente lo que las tiendas de conveniencias requieren, y lo que significa estar dentro de ellas, se ejecutaron las innumerables e interminables funciones que en ella hay, nos dimos cuenta del grado de responsabilidad que significa el estar dirigiendo a una organización tan compleja como esta. En el transcurso del estudio también se observó que no existe una cultura y procesos relacionados a este grupo, por lo tanto los empresarios no tienen una idea clara del valor real al que asciende un Sistema Informático.

Finalmente esperamos que el estudio realizado dé frutos, así mismo sirva de consulta y de soluciones a los interesados en estos temas, siguiendo un orden lógico y secuencial en lo que respecta a su proceso de selección, tomando en cuenta las necesidades actuales del medio en que nos desenvolvemos, el desarrollar un software dedicado al Análisis y desarrollo de pequeñas y medianas empresas tiene valor.

BIBLIOGRAFIA Y MANUALES

A. C. Nielsen ,
El Progreso del Código de Barras y Scanner durante 1983, Investigación y Marketing
(1984), 15, julio, Pág. 33

ADAMS, Andy ,
Integrating Promotional Evaluation into the Business Cycle,
Enero 1997

AECOC
Record en el Crecimiento de los Sistemas de Captura Automática de Datos,
Código 84, marzo 1997

BARON Larrarte, Luis
Investigación de Mercados
Ediciones ICAI, Madrid.
1998

BELLO, Laurentino y Vazquez, Rodolfo
Investigación de Mercados y Estrategias de Marketing,
Editorial Civitas, Madrid 1993

BRAÑA Teresa, Varela, Jesús y Rial, Antonio
Cómo Medir la Eficacia de las Promociones de Ventas,

CAPPS jr., Oral (1989),
Utilizing Scanner Data to Estimate Retail Demand Functions for Meat Products
American Agricultural Economics Association,
1998.

CASARES Ripol, Javier y Rebollo Arévalo, Alfonso
Distribución Comercial,
Editorial Civitas, Madrid. 1996










E. Amable, Edgardo (1996),
Historia no Oficial de los Códigos de Barras", *página Web Internet Identificación Automática - Código de Barras (<http://www.ent.ohiou.edu/~amable/>)*, noviembre.

"Estado Actual de las Tecnologías de Identificación Automática",
página Web Internet Identificación Automática - Código de Barras (<http://www.ent.ohiou.edu/~amable/>), junio.1997










ESTELLER, David (1995),
"El Sistema EAN, un Estándar Necesario", *Código 84*, marzo.

ANEXOS










Id	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
0	Desarrollo de software	95.75 días	lun 07/04/08	lun 18/08/08		
1	Ámbito	3.5 días	lun 07/04/08	jue 10/04/08		
2	Determinar el ámbito del proyecto	4 horas	lun 07/04/08	lun 07/04/08		Administración
3	Asegurarse patrocinadores para el proyecto	1 día	lun 07/04/08	mar 08/04/08	2	Administración
4	Definir recursos preliminares	1 día	mar 08/04/08	mié 09/04/08	3	Jefe de proyecto
5	Afianzar recursos principales	1 día	mié 09/04/08	jue 10/04/08	4	Jefe de proyecto
6	Ámbito completado	0 días	jue 10/04/08	jue 10/04/08	5	
7	Análisis y requisitos del software	14 días	jue 10/04/08	mié 30/04/08		
8	Realizar análisis de necesidades	5 días	jue 10/04/08	jue 17/04/08	6	Analista
9	Borrador de las especificaciones preliminares del software	3 días	jue 17/04/08	mar 22/04/08	8	Analista
10	Desarrollar presupuesto preliminar	2 días	mar 22/04/08	jue 24/04/08	9	Jefe de proyecto
11	Revisar las especificaciones del software y el presupuesto con el cliente	4 horas	jue 24/04/08	jue 24/04/08	10	Jefe de proyecto,Analista
12	Incorporar los comentarios a las especificaciones del software	1 día	vie 25/04/08	vie 25/04/08	11	Analista
13	Calcular los tiempos y fechas de entrega	1 día	lun 28/04/08	lun 28/04/08	12	Jefe de proyecto
14	Obtener aprobaciones para continuar (concepto, fechas, presupuesto)	4 horas	mar 29/04/08	mar 29/04/08	13	Administración,Jefe de proyecto
15	Conseguir los recursos necesarios	1 día	mar 29/04/08	mié 30/04/08	14	Jefe de proyecto
16	Análisis completado	0 días	mié 30/04/08	mié 30/04/08	15	
17	Diseñar	14.5 días	mié 30/04/08	mar 20/05/08		
18	Especificaciones preliminares del software	2 días	mié 30/04/08	vie 02/05/08	16	Analista
19	Desarrollar especificaciones de funcionamiento	5 días	vie 02/05/08	vie 09/05/08	18	Analista
20	Desarrollar prototipo basado en las especificaciones de funcionamiento	4 días	vie 09/05/08	jue 15/05/08	19	Analista
21	Revisar especificaciones de funcionamiento	2 días	jue 15/05/08	lun 19/05/08	20	Administración
22	Incorporar comentarios a las especificaciones de funcionamiento	1 día	lun 19/05/08	mar 20/05/08	21	Administración
23	Obtener aprobación para continuar	4 horas	mar 20/05/08	mar 20/05/08	22	Administración,Jefe de proyecto
24	Diseño completado	0 días	mar 20/05/08	mar 20/05/08	23	
25	Desarrollo	21.75 días	mié 21/05/08	jue 19/06/08		
26	Revisar especificaciones de funcionamiento	1 día	mié 21/05/08	mié 21/05/08	24	Programador
27	Identificar parámetros de diseño modular y de componentes separados	1 día	jue 22/05/08	jue 22/05/08	26	Programador
28	Asignar el personal de desarrollo	1 día	vie 23/05/08	vie 23/05/08	27	Programador
29	Desarrollar el código	15 días	lun 26/05/08	vie 13/06/08	28	Programador
30	Pruebas de los programadores (depuración primaria)	15 días	jue 29/05/08	jue 19/06/08	29FC-75%	Programador
31	Desarrollo completado	0 días	jue 19/06/08	jue 19/06/08	30	
32	Pruebas	48.75 días	mié 21/05/08	lun 28/07/08		
33	Desarrollar planes de pruebas de unidades usando las especificaciones	4 días	mié 21/05/08	lun 26/05/08	24	Personal de pruebas
34	Desarrollar planes de pruebas de integración usando las especificaciones	4 días	mié 21/05/08	lun 26/05/08	24	Personal de pruebas
35	Pruebas de unidades	15 días	jue 19/06/08	jue 10/07/08		
36	Revisar el código modular	5 días	jue 19/06/08	jue 26/06/08	33,31	Personal de pruebas
37	Probar si los módulos de los componentes se ajustan a las especificaciones	2 días	jue 26/06/08	lun 30/06/08	31,36	Personal de pruebas
38	Identificar anomalías y anotarlas en las especificaciones del proyecto	3 días	lun 30/06/08	jue 03/07/08	37	Personal de pruebas

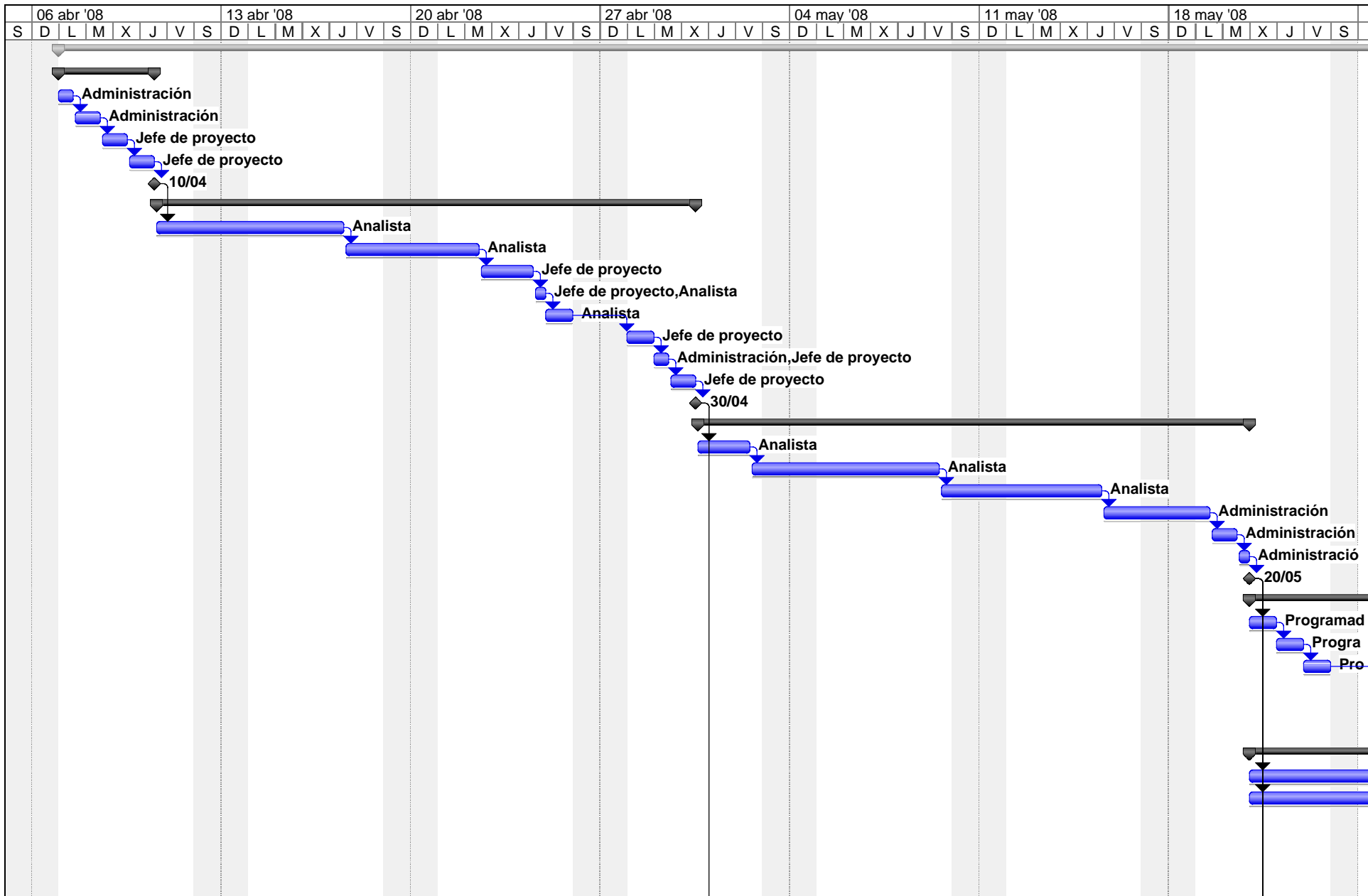
Proyecto: Desarrollo de software Fecha: jue 02/10/08	Tarea		Hito		Tareas externas	
	División		Resumen		Hito externo	
	Progreso		Resumen del proyecto		Fecha límite	

Id	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
39	Modificar código	3 días	jue 03/07/08	mar 08/07/08	38	Personal de pruebas
40	Volver a probar el código modificado	2 días	mar 08/07/08	jue 10/07/08	39	Personal de pruebas
41	Pruebas de unidades completadas	0 días	jue 10/07/08	jue 10/07/08	40	
42	Pruebas de integración	12 días	jue 10/07/08	lun 28/07/08		
43	Probar la integración de los módulos	5 días	jue 10/07/08	jue 17/07/08	41	Personal de pruebas
44	Identificar anomalías y anotarlas en las especificaciones	2 días	jue 17/07/08	lun 21/07/08	43	Personal de pruebas
45	Modificar código	3 días	lun 21/07/08	jue 24/07/08	44	Personal de pruebas
46	Volver a probar el código modificado	2 días	jue 24/07/08	lun 28/07/08	45	Personal de pruebas
47	Pruebas de integración completadas	0 días	lun 28/07/08	lun 28/07/08	46	
48	Curso	45.75 días	mié 21/05/08	mié 23/07/08		
49	Desarrollar las especificaciones del curso para los usuarios finales	3 días	mié 21/05/08	vie 23/05/08	24	Formadores
50	Desarrollar las especificaciones del curso para el personal de soporte	3 días	mié 21/05/08	vie 23/05/08	24	Formadores
51	Identificar la metodología de los cursos (utilizando PC, aulas, etc.)	2 días	mié 21/05/08	jue 22/05/08	24	Formadores
52	Desarrollar materiales de formación	3 sem.	jue 19/06/08	jue 10/07/08	49,31,50,51	Formadores
53	Realizar estudio de posibilidades de uso de los cursos	4 días	jue 10/07/08	mié 16/07/08	52	Formadores
54	Finalizar los materiales de los cursos	3 días	mié 16/07/08	lun 21/07/08	53	Formadores
55	Desarrollar mecanismo para impartir los cursos	2 días	lun 21/07/08	mié 23/07/08	54	Formadores
56	Materiales de los cursos completados	0 días	mié 23/07/08	mié 23/07/08	55	
57	Documentación	30.5 días	mié 21/05/08	mié 02/07/08		
58	Desarrollar especificación de ayuda	1 día	mié 21/05/08	mié 21/05/08	24	Escritores técnicos
59	Desarrollar el sistema de ayuda	3 sem.	mié 04/06/08	mié 25/06/08	58,29FC-50%	Escritores técnicos
60	Revisar la documentación de la ayuda	3 días	mié 25/06/08	lun 30/06/08	59	Escritores técnicos
61	Incorporar los comentarios a la documentación de la ayuda	2 días	lun 30/06/08	mié 02/07/08	60	Escritores técnicos
62	Desarrollar las especificaciones de los manuales de usuario	2 días	mié 21/05/08	jue 22/05/08	24	Escritores técnicos
63	Desarrollar los manuales de usuario	3 sem.	mié 04/06/08	mié 25/06/08	62,29FC-50%	Escritores técnicos
64	Revisar toda la documentación para el usuario	2 días	mié 25/06/08	vie 27/06/08	63	Escritores técnicos
65	Incorporar comentarios a la documentación de usuario	2 días	vie 27/06/08	mar 01/07/08	64	Escritores técnicos
66	Documentación completada	0 días	mié 02/07/08	mié 02/07/08	65,61	
67	Fase piloto	70.25 días	mié 30/04/08	mié 06/08/08		
68	Designar grupo de pruebas	1 día	mié 30/04/08	jue 01/05/08	16	Jefe de proyecto
69	Desarrollar el mecanismo de entrega del software	1 día	jue 01/05/08	vie 02/05/08	68	
70	Instalar y distribuir el software	1 día	lun 28/07/08	mar 29/07/08	47,69,66,56	Grupo de implantación
71	Obtener comentarios de los usuarios	1 sem	mar 29/07/08	mar 05/08/08	70	Grupo de implantación
72	Evaluar la información de las pruebas	1 día	mar 05/08/08	mié 06/08/08	71	Grupo de implantación
73	Fase piloto completada	0 días	mié 06/08/08	mié 06/08/08	72	
74	Implantación	5 días	mié 06/08/08	mié 13/08/08		
75	Determinar la estrategia final de distribución	1 día	mié 06/08/08	jue 07/08/08	73	Grupo de implantación
76	Desarrollar la metodología de distribución	1 día	jue 07/08/08	vie 08/08/08	75	Grupo de implantación
77	Afianzar recursos de implantación	1 día	vie 08/08/08	lun 11/08/08	76	Grupo de implantación

Proyecto: Desarrollo de software Fecha: jue 02/10/08	Tarea		Hito		Tareas externas	
	División		Resumen		Hito externo	
	Progreso		Resumen del proyecto		Fecha límite	

Id	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
78	Dar cursos al personal de soporte	1 día	lun 11/08/08	mar 12/08/08	77	Grupo de implantación
79	Distribuir el software	1 día	mar 12/08/08	mié 13/08/08	78	Grupo de implantación
80	Implantación completada	0 días	mié 13/08/08	mié 13/08/08	79	
81	Revisión posterior a la implementación	3 días	mié 13/08/08	lun 18/08/08		
82	Documentar la experiencia adquirida	1 día	mié 13/08/08	jue 14/08/08	80	Jefe de proyecto
83	Distribuir a los integrantes del equipo	1 día	jue 14/08/08	vie 15/08/08	82	Jefe de proyecto
84	Crear equipo de mantenimiento del software	1 día	vie 15/08/08	lun 18/08/08	83	Jefe de proyecto
85	Revisión posterior a la implementación completada	0 días	lun 18/08/08	lun 18/08/08	84	
86	Plantilla de desarrollo de software completada	0 días	lun 18/08/08	lun 18/08/08	85	

Proyecto: Desarrollo de software Fecha: jue 02/10/08	Tarea		Hito		Tareas externas	
	División		Resumen		Hito externo	
	Progreso		Resumen del proyecto		Fecha límite	



Proyecto: Desarrollo de software
 Fecha: jue 02/10/08

Tarea



Hito



Tareas externas



División



Resumen



Hito externo



Progreso

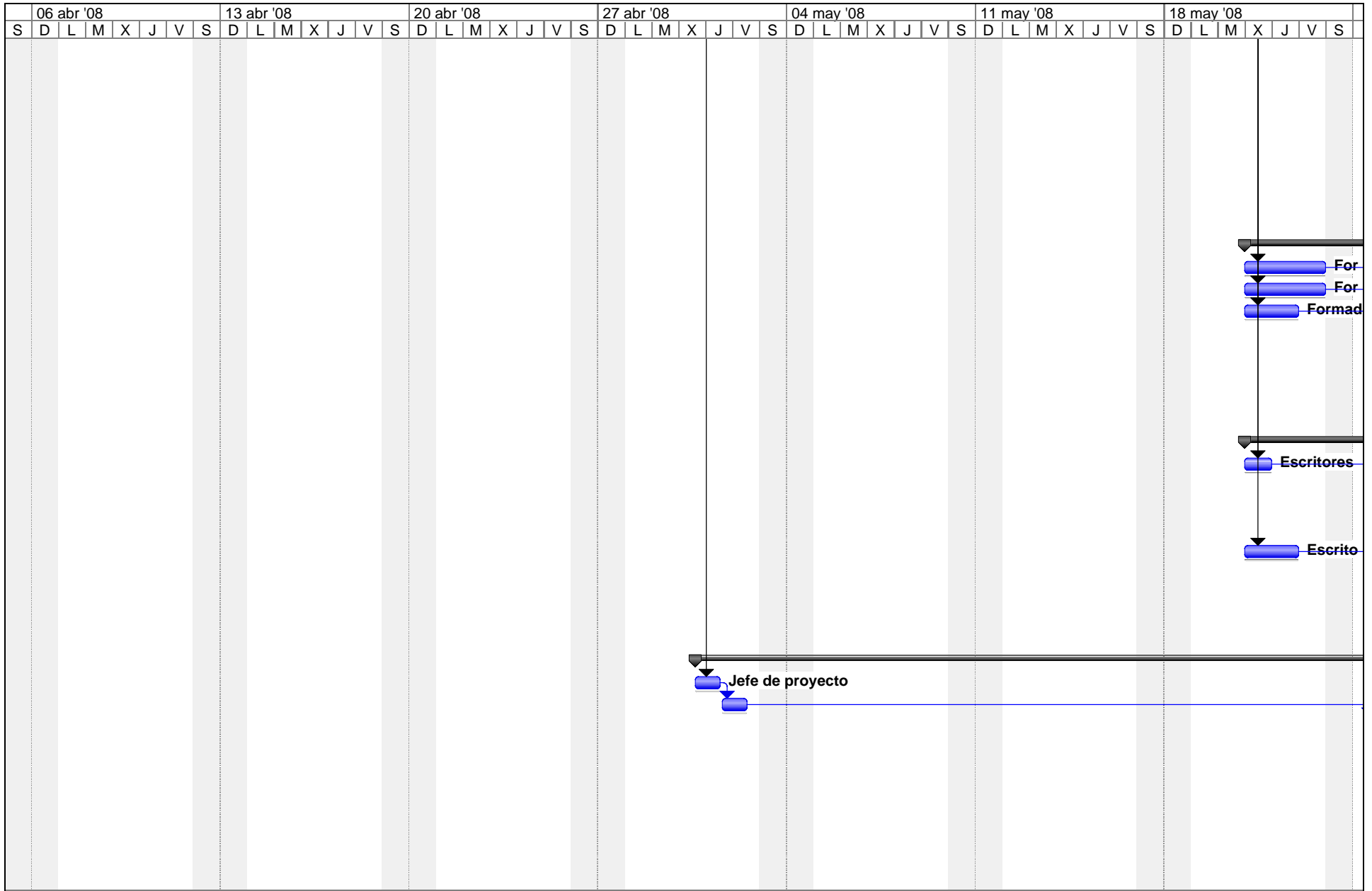


Resumen del proyecto



Fecha límite














Proyecto: Desarrollo de software Fecha: jue 02/10/08	Tarea		Hito		Tareas externas	
	División		Resumen		Hito externo	
	Progreso		Resumen del proyecto		Fecha límite	

06 abr '08							13 abr '08							20 abr '08							27 abr '08							04 may '08							11 may '08							18 may '08							
S	D	L	M	X	J	V	S	D	L	M	X	J	V	S	D	L	M	X	J	V	S	D	L	M	X	J	V	S	D	L	M	X	J	V	S	D	L	M	X	J	V	S	D	L	M	X	J	V	S



Proyecto: Desarrollo de software Fecha: jue 02/10/08	Tarea		Hito		Tareas externas	
	División		Resumen		Hito externo	
	Progreso		Resumen del proyecto		Fecha límite	

25 may '08							01 jun '08							08 jun '08							15 jun '08							22 jun '08							29 jun '08							06 jul '08							13 i							
D	L	M	X	J	V	S	D	L	M	X	J	V	S	D	L	M	X	J	V	S	D	L	M	X	J	V	S	D	L	M	X	J	V	S	D	L	M	X	J	V	S	D	L	M	X	J	V	S	D	L	M	X	J	V	S	D

ón, Jefe de proyecto

Programador
Programador

Programador

Programador

19/06

Personal de pruebas
Personal de pruebas

Personal de pruebas

Personal de pruebas

Personal de pruebas

Proyecto: Desarrollo de software
Fecha: jue 02/10/08

Tarea



Hito



Tareas externas



División



Resumen



Hito externo



Progreso

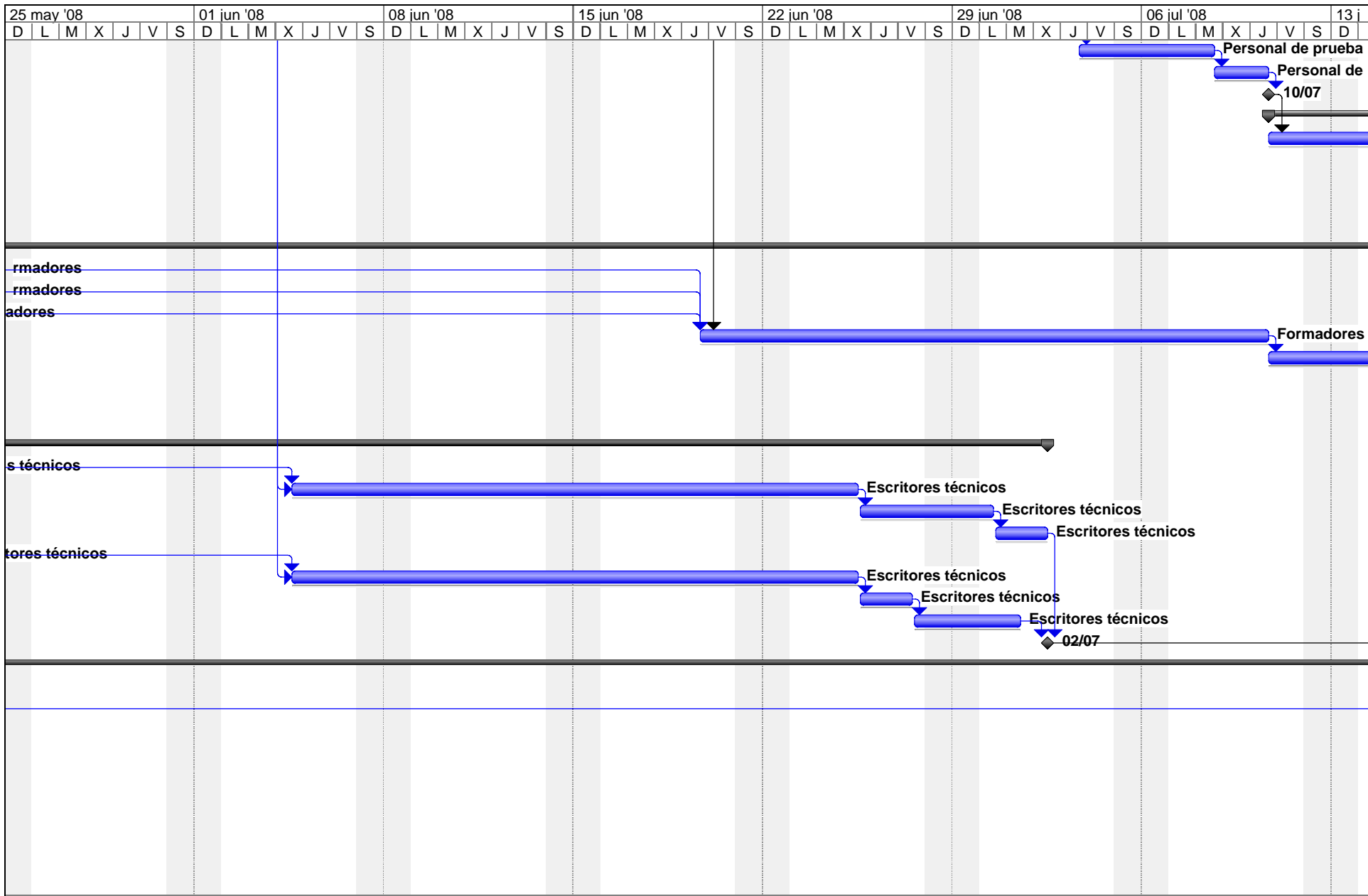


Resumen del proyecto



Fecha límite












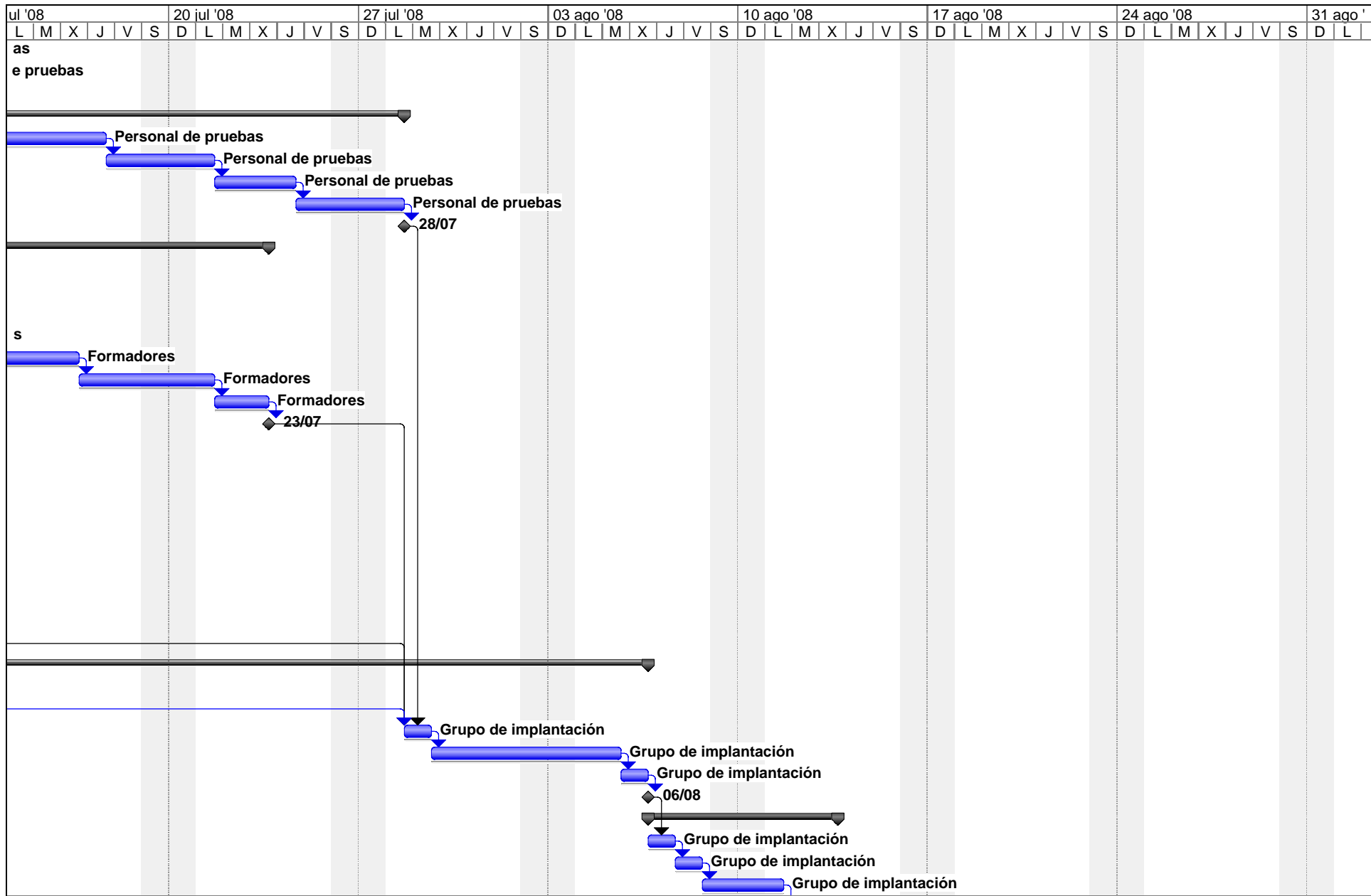


Proyecto: Desarrollo de software Fecha: jue 02/10/08	Tarea		Hito		Tareas externas	
	División		Resumen		Hito externo	
	Progreso		Resumen del proyecto		Fecha límite	

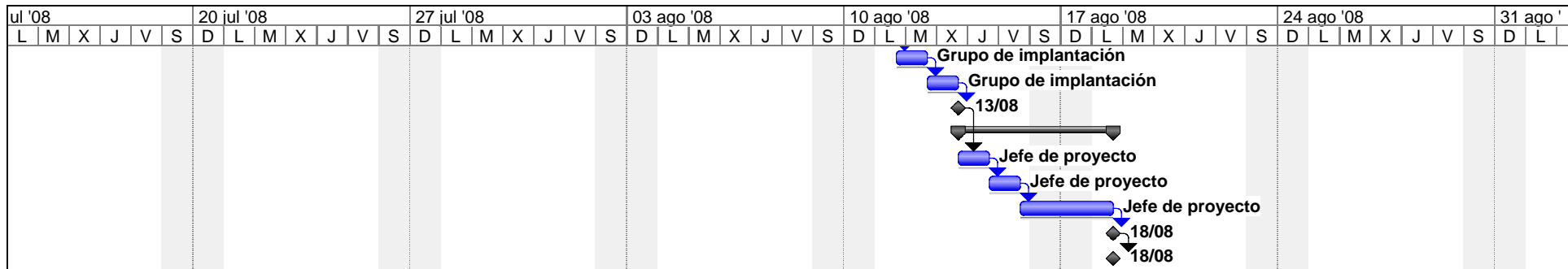
25 may '08							01 jun '08							08 jun '08							15 jun '08							22 jun '08							29 jun '08							06 jul '08							13 i
D	L	M	X	J	V	S	D	L	M	X	J	V	S	D	L	M	X	J	V	S	D	L	M	X	J	V	S	D	L	M	X	J	V	S	D	L	M	X	J	V	S	D	L	M	X	J	V	S	D



Proyecto: Desarrollo de software Fecha: jue 02/10/08	Tarea		Hito		Tareas externas	
	División		Resumen		Hito externo	
	Progreso		Resumen del proyecto		Fecha límite	



Proyecto: Desarrollo de software Fecha: jue 02/10/08	Tarea		Hito		Tareas externas	
	División		Resumen		Hito externo	
	Progreso		Resumen del proyecto		Fecha límite	



Proyecto: Desarrollo de software Fecha: jue 02/10/08	Tarea		Hito		Tareas externas	
	División		Resumen		Hito externo	
	Progreso		Resumen del proyecto		Fecha límite	