

25  
20j

A DIOS

POR DARME LA FUERZA DE  
VOLUNTAD Y GUIAR MIS PASOS  
EN EL CAMINO DE LA VIDA.

" DIOS NOS DA Y NOSOTROS TOMAMOS LO QUE QUEREMOS "

TESIS CON  
FALLA DE ORIGEN



## **UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso**

### **DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

**A MIS PADRES**

**JOSEFINA DEL TORO ZEPEDA  
ALFONSO MENDEZ BRAMBILA  
POR SER MI FURZA,  
MI INSPIRACION Y  
MI ORGULLO.**

**A TODOS**

**A TODOS AQUELLOS QUE PARTICIPARON  
EN EL DESARROLLO DE ESTA TESIS,  
MAESTROS, COMPAÑEROS Y AMIGOS  
ASI COMO UN RECONOCIMIENTO  
ESPECIAL A JOSE MANUEL SANCHEZ.**

# INDICE

	PAG.
- AGRADECIMIENTOS.....	I
- INDICE.....	1
- INTRODUCCION.....	3
- QUE ES EL CODIGO DE BARRAS ?.....	5
- HISTORIA DEL CODIGO DE BARRAS.....	6
<b>CAPITULO I ( tipos de código de barras )</b>	
- CARACTERISTICAS DE LA SIMBOLOGIA.....	11
- TIPOS DE CODIGO DE BARRAS.....	12
- UPC.....	12
- EAN.....	16
- INTERLEAVED 2 DE 5.....	18
- CODEBAR.....	20
- CODE 39.....	23
- CODE 128.....	23
- CODE 93.....	28
- CODE 49.....	30
- OTROS.....	34
<b>CAPITULO II ( impresión del código de barras )</b>	
- IMPRESION DEL CODIGO DE BARRAS.....	38
- IMPRESORAS PARA EL CODIGO DE BARRAS.....	39
- SELECCION DEL TIPO IMPRESORA DE CODIGO DE BARRAS.....	40
- SELECCION DEL TIPO DE PAPEL PARA EL CODIGO DE BARRAS..	41
- SELECCION DEL TAMAÑO DE LA IMPRESION DEL CODIGO DE .. BARRAS	41
<b>CAPITULO III ( decodificadores )</b>	
- QUE ES EL DECODIFICADOR.....	43
- QUE ES EL DECODIFICADOR DE CODIGO DE BARRAS.....	43
- DECODIFICADORES UNDECODED.....	44
- DECODIFICADORES DECODED.....	45

	PAG.
<b>CAPITULO IV ( lectores de código de barras )</b>	
- QUE ES UN LECTOR DE CODIGO DE BARRAS.....	48
- TIPOS DE LECTORES.....	50
- LECTORES WAND.....	52
- LECTORES CCD.....	53
- LECTORES VLD.....	55
- LECTORES A MANOS LIBRES.....	57
<b>CAPITULO V ( conexiones del código de barras)</b>	
- ESQUEMA DE CONECCION DE CODIGO DE BARRAS.....	61
- ESQUEMA DE CONECCION UNDECODED.....	61
- ESQUEMA DE CONECCION DECODED.....	62
<b>CAPITULO VI ( aplicaciones )</b>	
- QUE ES AUTOMATIZACION .....	65
- ELEMENTOS NECESARIOS PARA LA AUTOMATIZACION UTILIZANDO EQUIPO DE TECNOLOGIA DE CODIGO DE BARRAS.....	65
- APLICACION DEL CODIGO DE BARRAS EN LA AUTOMATIZACION EN CASOS PRACTICOS.....	65
- AUTOMATIZACION DE UN PUNTO DE VENTA.....	76
- DIAGRAMAS DE FLUJO.....	80
- SELECCION DEL TIPO DE LENGUAJE.....	87
- PROGRAMACION.....	87
<b>CAPITULO VII</b>	
- CONCLUSIONES.....	130
<b>BIBLIOGRAFIA.....</b>	<b>132</b>

## **INTRODUCCION**

El código de barras en el proceso de la automatización juega un papel importante debido a sus características de ser un código, el cual puede ser leído con facilidad y sin errores, es decir, permite poder ahorrar tiempo en horas hombre y ahorrar tiempo máquina, nos permite el ahorro de recursos que son de vital importancia para cualquier empresa que desee obtener mayor eficiencia en cualquier proceso.

El código de barras nos permite poder establecer un seguimiento de algún producto cualquiera que se desee en algún proceso industrial cualesquiera que sea; una toma de inventarios de manera rápida y sin errores; automatizar algún punto de venta; implantar el control de acceso a nuestra empresa imprimiendo el código de barras en credenciales para poder tener cualquier tipo de control que se desee del personal; podemos como en los grandes supermercados tener todos nuestros productos marcados cada uno con un código de barras específico para cada producto y automatizar la venta de nuestro productos para poder vender de manera más eficiente; en un negocio de préstamo y renta de equipo como por ejemplo un videoclub podemos marcar nuestros productos y por medio del código de barras controlar la renta de los mismos y también la identificación de cada videousuario.

El código de barras asignado para poder identificar nuestros productos, depende de la aplicación que se trate; en el caso del videoclub, este código asignado será para uso interno y nos servirá para identificar un video en especial; si se trata de alguna aplicación industrial el producto que se quiera marcar con el código de barras se comercializará para la venta en cualquier lugar comercial, es decir, donde podemos encontrar una gran variedad de productos marcados con código de barras, en este caso nuestro código deberá ser asignado por alguna asociación la cual tenga la tarea de asignar códigos de barras, esto es con el fin de que no puedan existir 2 productos diferentes y un mismo código de barras para los dos y esto pueda ocasionar un conflicto en los lugares de venta de estos productos; en México la asociación que

se dedica a esto es la AMECOP ( ASOCIACION MEXICANA DEL CODIGO DEL PRODUCTO ), esta asociación se dedica a otorgar los códigos de barras para cada producto específico .

El objetivo principal de esta tesis es el tener los conocimientos necesarios para poder aplicar el código de barras en casos específicos de automatización cualesquiera que sean estos.

Los objetivos particulares son dar un enfoque preciso de lo que es el código de barras, los tipos de código de barras existentes, las características de cada uno, las especificaciones necesarias para cada uno de ellos, los métodos de impresión del código de barras, los métodos de lectura del código de barras existentes y que son los codificadores del código de barras.

Actualmente existe una gran variedad de códigos de barras de los cuales los más usados en general son los siguientes: UPC, EAN, INTERLEAVED 2 OF 5, CODE 39, CODE 128, CODE 93, CODE 49, CODE 16.

Siendo estos códigos los más comunes en la industria, debido a que cada uno de ellos reúne características especiales. Una característica especial es la de que algunos son *Alfanuméricos* y otros simplemente son *Númericos*.

Una característica muy especial que diferencia a un código de barras de otro es la cantidad de caracteres que puedan contener en un espacio determinado .

## QUE ES EL CODIGO DE BARRAS ?

El código de barras es la impresión de una serie de líneas y espacios en blanco en forma prefijada los cuales significan algo para un código de barras específico que se desea utilizar.

Esto es, dependiendo del tipo de código de barras aunque el mensaje sea el mismo el orden de las barras es diferente.

Un ejemplo claro del código de barras es la figura que se muestra a continuación:



El código de barras está formado por una serie de continua de líneas y espacios continuos dependiendo del código de barras .

Este es un ejemplo de lo que es el código de barras muy general, la mayoría de estos códigos son parecidos en cuanto a su arquitectura, solo varían en el tamaño de las barras y de los espacios en blanco, siendo esta la característica principal entre los códigos de barras existentes.

## HISTORIA DEL CODIGO DE BARRAS

Al hablar del código de barras *no se conoce precisamente cual es su inicio*, pero se conoce bien que este surge como una necesidad de automatizar procesos en los cuales la captura de datos es siempre la misma y es repetitiva con muy pocas variables de una captura a otra.

En el año de 1932 Wallace Flint, desarrolló como proyecto de tesis un sistema de automatización, para controlar en un supermercado el pago y salida de productos, esto fué en la Universidad de Harvard, él utilizó un sistema de figuras, las cuales representan a cada producto específico, pudiendo tener una automatización del proceso de pago y salida de los productos marcados.

En 1940, Joe Woodland y Bery Silver, realizaron una investigación técnica aprovechando la gran variedad de productos y diferentes precios de un supermercado para checar automáticamente el número de producto y su precio respectivo culminando su investigación en 1949, patentando este sistema. Las investigaciones de Woodland y Bery los llevaron a patentar un código circular, hablando técnicamente, el código circular y el código de barras es el mismo con algunas pequeñas diferencias, 20 años después el Ing. Woodland y el grupo de ingenieros de la compañía IBM forman un grupo para el desarrollo del código UPC.

El código inventado por estos científicos era parecido al siguiente ejemplo:



Código circular

En 1950 y parte de los años 60's muchos ingenieros y científicos trataron de construir un estilizado sistema de lectura de código de barras para la lectura automática de estos, un destacado científico en esta investigación fue Giraldo Feissel, el cual en 1960 patenta un sistema que podía leer códigos del 0 al 9, los cuales se formaban con un arreglo de 7 barras y 7 espacios, pero este nunca pudo llevarlo a la comercialización por tener demasiados problemas en la lectura de los mismos.

En 1970 es abierto en una tienda de autoservicio, en Cincinnati EEUU, un concurso para presentar el código de barras más adecuado para este tipo de tienda y seleccionar un código estándar para este tipo de tienda de autoservicio, la persona encargada del proyecto es R. Bert Gookin presidente de la misma fue el líder del proyecto.

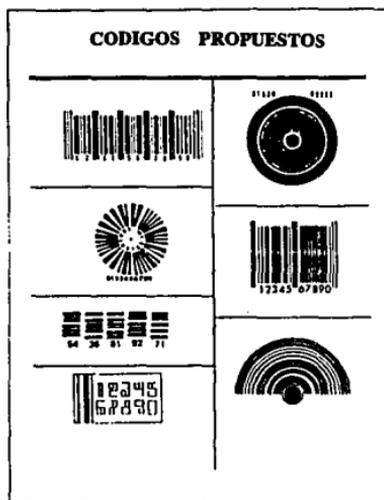
De el comité antes mencionado, se forma un nuevo comité para seleccionar y escoger el código más adecuado para todo tipo de industria en EEUU y así formar un estándar en los códigos de barras, aquí se propusieron siete diferentes tipos de código de barras y símbolos especiales para el posible control por computadora de el código de barras y estos fueron los que se muestran en la pag. siguiente.

Con esta primera evaluación se tuvieron a participantes como los laboratorios Battelle Memorial Institute, Graphic Arts Technical Foundation, IBM, etc. los cuales participaron en la elaboración de los códigos de barras, resultando elegido el código denominado como UPC el día 3 de abril de 1973, este fue propuesto por la empresa IBM. Desde este momento los mercados Canadienses y Estadounidenses se unificaron para el código al UPC. También se trató de unificar el mercado Europeo, siendo al final escogido para el mercado Europeo el código EAN (European Article Numbering) en diciembre de 1976.

En un principio muchas compañías trataron de implementar un código, como en 1971 la compañía Plessey Company, diseñó un sistema para el préstamo de libros en una biblioteca. En 1972 nace el código de barras CODEBAR el cual fue creado por Monarch

Marking Systems, este código actualmente se utiliza en algunas bibliotecas de los EEUU. Antes de 1974 el código de barras solo incluía números, en 1974 el Dr. David C. Allais de la compañía Interneec Developed inventó el código 39 y actualmente es uno de los más usados.

En la época de los 80's nacen varios tipos de código por ejemplo en 1981 es introducido el código 128 en 1982 es introducido el código 93, los cuales son una derivación de lo que es el código 39. En 1987 el Dr. David C. Allias introduce un nuevo código denominado código 49 el cual presenta características superiores a los mencionados. En 1989 después de un año de la invención del código 49 nace el código 16K. En 1990 la compañía Symbol Technologies anuncia un tipo de código bidimensional llamado PDF417.



Códigos propuestos por el nuevo comité

## ***CAPITULO I***

**CARACTERISTICAS DE LA SIMBOLOGIA**, existen diferentes parámetros para comprender el código de barras, el primero es el " CHARACTER SET ", el cual es el que describe el rango de los caracteres de los datos que pueden ser decodificados en cada simbología de código, algunos códigos solo codifican números y algunos otros son alfanuméricos.

" TIPO DE SIMBOLOGIA ", las simbologías las podríamos describir dentro de dos grupos discretos y continuos. En los códigos discretos cada caracter puede ser codificado independientemente de otros caracteres, y en los códigos continuos el código es intermezclado, un caracter ocupa un pequeño espacio del anterior o posterior a este, esto con el fin de poder tener una codificación más pequeña en espacio.

"ANCHURA DE ELEMENTOS", en la simbología del código de barras, para escribir el ancho de las barras y de los espacios en blanco, existen principalmente dos métodos los cuales son los siguientes: En el primero tener solo dos anchos ( estrecho y ancho ), y en el segundo utilizar múltiples anchos.

En múltiples anchos de la etiqueta, esto quiere decir que se puede escribir en muy diferentes anchos, la mayoría de las simbologías son modulares, así, cuando tenemos diferentes anchos, podemos decrecer modularmente para poder al final llegar a identificar encontrar lo que se encuentra decodificado en barras.

"LA LONGITUD VARIABLE O FUA" , Algunos códigos por tener una estructura solo pueden contener mensajes de un ancho fijo, otros códigos debido a no contar con una estructura predefinida pueden tener grandes variantes en el ancho.

"LA DENSIDAD", En los códigos de barras difieren unos de otros debido a la cantidad de datos que puedan ser codificados en cierto espacio.

"DIGITO VERIFICADOR " (SELF-CHECKING), muchos códigos cuentan con este, sirve para poder detectar algún tipo de error en la decodificación, aunque para otros sea un defecto debido a que tiene que ocupar un espacio en la codificación.

"CODIGO DE INICIO Y PARADA" , estas son las que permiten indicar al

escaner el inicio o fin de la lectura y posterior decodificación de los datos, también indican el sentido de los mismos y la forma en que se tiene que leer.

"**BIDIRECCIONAL**", este término es empleado en la mayoría de los códigos y quiere decir que el código de barras puede ser leído de derecha a izquierda y de izquierda a derecha.

**TIPOS DE CODIGO DE BARRA**, Actualmente existen diferentes códigos de barras en el mercado, siendo los más comunes el código 39, código UPC, código EAN, código INTERLEAVED 2 OF 5, código 93, código 49, entre los más comunes.

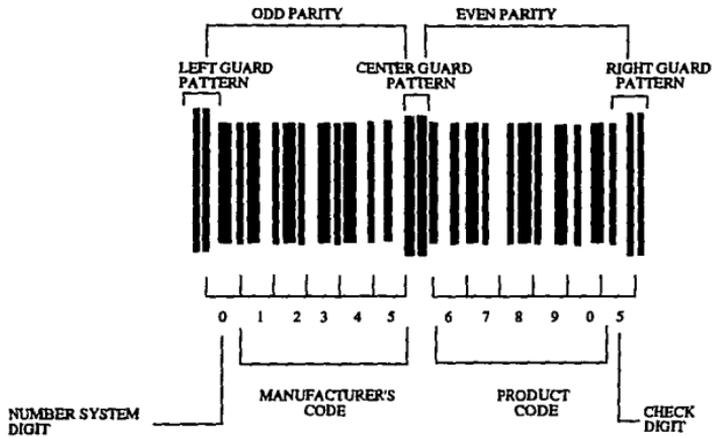
Estos códigos son los más difundidos siendo los más comunes a nivel comercial el UPC, EAN y código 39.

Estos códigos son los más comunes debido a su gran facilidad de manejo y su fácil reconocimiento por cualquier dispositivo de lectura sea la marca y modelo que fuese.

**UPC**, Este tipo de código fué el primero utilizado en la industria para automatizar procesos utilizando el código de barras como el elemento principal y código utilizado. El código UPC esta compuesto principalmente de diez caracteres o dígitos, de los cuales los primeros cinco son utilizados para representar el **ITEM** del producto, los cinco siguientes son utilizados para identificar al producto, estos diez dígitos están precedidos por un dígito denominado "number system" ( número de sistema) y después de estos diez dígitos está otro denominado "check digit" ( dígito verificador), con el cual se asegura que lo que se esté leyendo sea correcto ( ver figura 1.1.).

El código UPC es un código por entero numérico, es decir, no se puede utilizar caracteres de tipo alfabético; el código tiene diferentes versiones, como son la versión A que puede contener 12 dígitos y la versión E, la cual tiene capacidad solo para seis dígitos.

El primer dígito representa el número del sistema e indica qué tipo de producto es dependiendo del símbolo utilizado, como se ve en la tabla 1.1



CODIGO UPC VERSION A

figura 1.1.

<b>SIGNIFICADO DEL NUMERO DE SISTEMA</b>	
<b>DIGITO</b>	<b>APLICACION</b>
0	PRODUCTOS MANUFACTURADOS
1	RESERVADO
2	PARA EMPAQUE DE PRODUCTOS
3	PRODUCTOS FARMACEUTICOS
4	USO INTERNO EN SUPERMERCADOS
5	CUPONES EN UPC
6	IDENTIFICACION DE 100,000 PRODUCTOS MANUFACTURADOS
7	IDENTIFICACION DE 100,000 PRODUCTOS MANUFACTURADOS
8	RESERVADO
9	RESERVADO

**TABLA 1.1**

## **TESIS EL CODIGO DE BARRAS Y LA AUTOMATIZACION**

El dígito final o dígito chegador es un valor que se obtiene matemáticamente, el cual está basado en los números utilizados en el código de barras. Este dígito verificador sirve para evitar errores y tener el porcentaje de error mucho menor en el proceso de lectura de los datos. El método se describe a continuación.

Primero se identificarán las zonas del código denominadas "EVEN y ODD PARIDAD".

Segundo, sumar los valores numéricos de la posición denominada ADD.

Tercero, sumar los valores numéricos de la posición denominada y el total multiplicarlo por tres.

Cuarto, sumar el total obtenido de las zonas ADD y EVEN de los pasos segundo y tercero.

Quinto, obtener la suma de los números del resultado obtenido en el paso número cuarto este tiene que ser menor de 10 y esté es el dígito verificador.

En la figura 1.1. se ve un código UPC este lo tomaremos como ejemplo para demostrar el método.

Al identificar las partes ODD y EVEN, en las cuales la primera está formada por los dígitos 0,1,2,3,4,5 y la segunda por los dígitos 6,7,8,9,0.

Aplicando los pasos segundo y tercero obtenemos en el segundo un resultado de 14 y en el paso tercero un valor de 90.

Sumamos los resultados obtenidos y tenemos un total de 104 y al realizar la suma de los dígitos obtenemos el número 5 que es el dígito verificador.

EL código UPC en su versión A los primeros seis dígitos y los segundos seis dígitos están separados por dos barras centrales, los caracteres están encerrados por dos barras a la derecha y dos a la izquierda, estas sirven para indicar el inicio y fin de código al tipo de lector utilizado, los actuales datos están decodificados por dos barras y dos espacios.

Los números no serán decodificados de igual manera en las partes ODD y EVEN,

es decir, la codificación de un mismo número será diferente en la parte ODD y EVEN como se muestra en la figura 1.2. EAN, (European Article Numbering), como su nombre lo indica este código nació en Europa y es empleado en cualquier artículo Europeo. Existen actualmente dos versiones de este código que son EAN-13 y EAN-8, las cuales codifican 13 y 8 dígitos respectivamente, el EAN 13 contiene el mismo número de barras del código UPC; el código más utilizado actualmente es el EAN-8, debido a lo similar del UPC con el EAN-13, la forma del código EAN-8 es como la que se muestra en la figura 1.2.1 como se puede ver en la figura el código EAN-8 está constituido por cuatro dígitos en la parte ODD y cuenta con el GUARD PATTERN y otros cuatro dígitos en la parte ADD, así como un par de barras a cada lado del código, las cuales sirven como bit de inicio y parada. Este código codifica lo siguiente: dos dígitos que sirven como banderas, cinco dígitos los cuales son los datos y un dígito checodeador o verificador. La forma en la que se codifican los símbolos en código de barras es idéntica a la mostrada en la figura 1.2, la forma de las barras es igual a la del código UPC.

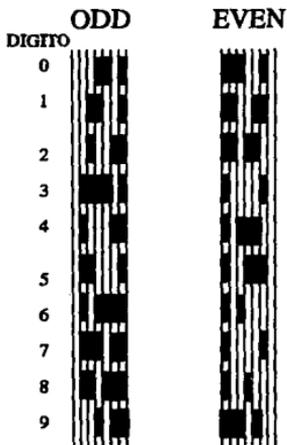


FIGURA 1.2

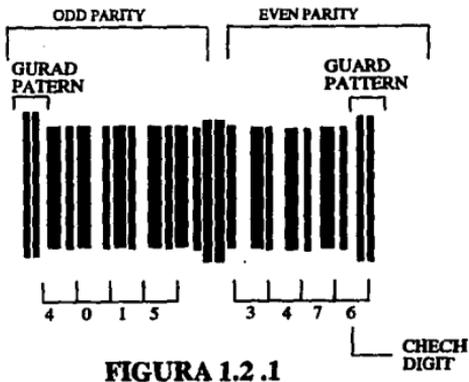


FIGURA 1.2.1

**INTERLEAVED 2 OF 5**, este código cuenta es un código de alta densidad , se autochecha él mismo y utiliza la simbología numérica continua. Este es muy usado en industria. Cada caracter **INTERLEAVED 2 OF 5** codifica dos dígitos, uno en las barras y otro en los espacios en blanco; son cinco barras, de estas dos son anchas y tres son estrechas y la forma de codificarlo es como se muestra en la figura 1.3. En esta se ve cómo podemos codificar cualquiera de los símbolos numéricos del 0-9, la ventaja de este es que, como lo dice su nombre 2 of 5, podemos en estos poner dos dígitos diferentes uno con espacios en blanco y otro con rallas negras, como se ve en la figura 1.4

El código completo consiste en el caracter de inicio, el cual esta constituido por dos rallas negras delgadas, los caracteres codificados y el caracter de parada, el cual está constituido por una ralla gruesa y una delgada, como se muestra en la figura 1.5.

El mínimo valor aceptado por este código para una barra o un espacio en blanco es de 7.5 milímetros.

### DIGITOS EN 2OF 5

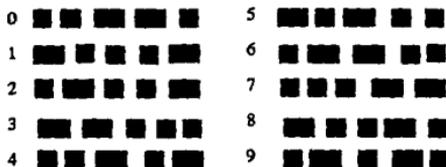


FIGURA 1.3.

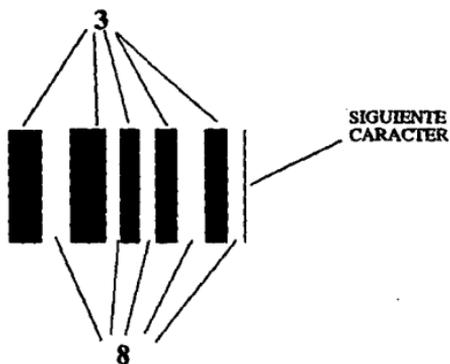


FIGURA 1.4.

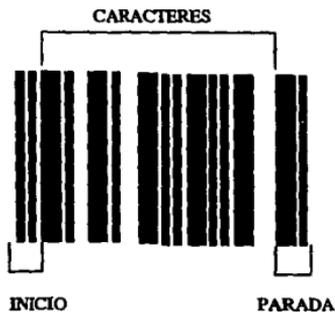


FIGURA 1.5

**CODEBAR**, inventado en el año de 1972 el CODEBAR es todavía usado en librerías, bancos de sangre y en otras aplicaciones. Este código cuenta con la característica de poder codificar 16 caracteres diferentes, los cuales son 0-9, \$, :, A, B, C, D, y cuenta también





**CODE 39**, este código fué el primer código alfanumérico existente, utilizado en un principio en fábricas que necesitaban caracteres alfabéticos combinados con numéricos para descripción de sus productos. En la figura 1.9 se muestra la codificación en código 39.

Como se puede ver en la figura 1.9 cada caracter en código 39 consta de 5 barras y 4 espacios en blanco en un total de 9 elementos. Las barras y los espacios representan 0 y 1 como se muestra en la figura 1.10

El código 39 cuenta con la capacidad de poder codificar 44 diferentes caracteres, entre los cuales se cuenta con el \* el cual es considerado como un caracter reservado debido a que este se utiliza como inicio y final del código ( STAR, STOP CODE ).

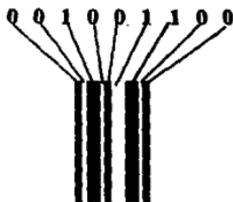
**CODE 128**, Este código fué introducido en 1981, es un código de muy alta densidad, es decir, que en menor espacio puede codificar mayor número de caracteres. Este es de longitud variable, es un código continuo. Cada caracter codificado en código 128 cuenta con 11 módulos, los cuales pueden ser barras o espacios en blanco, cada caracter cuenta con tres barras y tres espacios, como se muestra en la figura 1.11.

El código 128 puede codificar 106 diferentes caracteres, cada caracter impreso puede tener de uno a tres diferentes significados dependiendo del caracter empleado, esto depende del caracter de inicio (STAR) usado, como se muestra en la figura 1.12.

1	■ ■ ■ ■ ■	M	■ ■ ■ ■ ■ ■ ■
2	■ ■ ■ ■ ■ ■ ■	N	■ ■ ■ ■ ■ ■ ■
3	■ ■ ■ ■ ■ ■ ■	O	■ ■ ■ ■ ■ ■ ■
4	■ ■ ■ ■ ■ ■ ■	P	■ ■ ■ ■ ■ ■ ■
5	■ ■ ■ ■ ■ ■ ■	Q	■ ■ ■ ■ ■ ■ ■
6	■ ■ ■ ■ ■ ■ ■	R	■ ■ ■ ■ ■ ■ ■
7	■ ■ ■ ■ ■ ■ ■	S	■ ■ ■ ■ ■ ■ ■
8	■ ■ ■ ■ ■ ■ ■	T	■ ■ ■ ■ ■ ■ ■
9	■ ■ ■ ■ ■ ■ ■	U	■ ■ ■ ■ ■ ■ ■
0	■ ■ ■ ■ ■ ■ ■	V	■ ■ ■ ■ ■ ■ ■
A	■ ■ ■ ■ ■ ■ ■	W	■ ■ ■ ■ ■ ■ ■
B	■ ■ ■ ■ ■ ■ ■	X	■ ■ ■ ■ ■ ■ ■
C	■ ■ ■ ■ ■ ■ ■	Y	■ ■ ■ ■ ■ ■ ■
D	■ ■ ■ ■ ■ ■ ■	Z	■ ■ ■ ■ ■ ■ ■
E	■ ■ ■ ■ ■ ■ ■	-	■ ■ ■ ■ ■ ■ ■
F	■ ■ ■ ■ ■ ■ ■	.	■ ■ ■ ■ ■ ■ ■
G	■ ■ ■ ■ ■ ■ ■	SPACE	■ ■ ■ ■ ■ ■ ■
H	■ ■ ■ ■ ■ ■ ■	*	■ ■ ■ ■ ■ ■ ■
I	■ ■ ■ ■ ■ ■ ■	\$	■ ■ ■ ■ ■ ■ ■
J	■ ■ ■ ■ ■ ■ ■	/	■ ■ ■ ■ ■ ■ ■
K	■ ■ ■ ■ ■ ■ ■	+	■ ■ ■ ■ ■ ■ ■
L	■ ■ ■ ■ ■ ■ ■	%	■ ■ ■ ■ ■ ■ ■

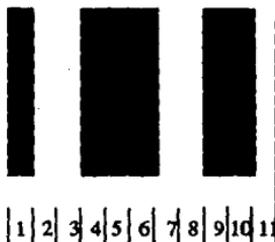
codificación en código 39

figura 1.9



codificación letra I en código 39

figura 1.10



Cada caracter del código 128 contiene tres barras y tres espacios divididos en 11 módulos como se muestra

figura 1.11

En la figura 1.12 se muestra claramente como se puede ir de un código a otro y poder tener una gran variedad de codificación según se necesite.



En esta implementación de código 128 se muestra cómo se realiza el cambio de código para el código 128, cómo se demuestra en esta, va del código C, al B y por último al A.

FIGURA 1.12

En la figura 1.13 que se muestra a continuación podemos ver todos los caracteres y símbolos especiales con los que cuenta el código 128, este consta de 106 caracteres diferentes y cada uno de ellos puede tener tres significados diferentes, como se explicó anteriormente esto depende del carácter de código de que esté precedido.

TESIS EL CODIGO DE BARRAS Y LA AUTOMATIZACION

CODE VALOR			CODE VALOR		
A	B	C	A	B	C
Space	Space	00	V	V	54
		01	W	W	55
		02	X	X	56
#	#	03	Y	Y	57
\$	\$	04	Z	Z	58
%	%	05	[	[	59
&	&	06	\	\	60
'	'	07	]	]	61
(	(	08			62
)	)	09			63
*	*	10	NUL	r	64
+	+	11	SOH	a	65
,	,	12	STX	b	66
-	-	13	ETX	c	67
.	.	14	EOT	d	68
/	/	15	ENQ	e	69
0	0	16	ACK	f	70
1	1	17	BEL	g	71
2	2	18	BS	h	72
3	3	19	HT	i	73
4	4	20	LF	j	74
5	5	21	VT	k	75
6	6	22	FF	l	76
7	7	23	CR	m	77
8	8	24	SO	n	78
9	9	25	SI	o	79
:	:	26	DLE	p	80
;	;	27	DC1	q	81
<	<	28	DC2	r	82
=	=	29	DC3	s	83
>	>	30	DC4	t	84
?	?	31	NAK	u	85
		32	SYN	v	86
A	A	33	ETB	w	87
B	B	34	CAN	x	88
C	C	35	EM	y	89
D	D	36	SUB	z	90
E	E	37	ESC		91
F	F	38	FS		92
G	G	39	GS		93
H	H	40	RS	ñ	94
I	I	41	US	DEL	95
J	J	42	FNC3	FNC3	96
K	K	43	FNC2	FNC2	97
L	L	44	Shift	Shift	98
M	M	45	code C	Code C	99
N	N	46	code B	FNC4	Code B
O	O	47	fn4	Code A	Code A
P	P	48	fn1	FNC1	FNC1
Q	Q	49	start	(Code A)	102
R	R	50	start	(Code B)	104
S	S	51	start	(Code C)	105
T	T	52	stop	stop	stop
U	U	53			

En esta figura podemos ver como se utilizan los 3 tipos de código A,B, y C y el significado que adquiere el código de barras así como su valor numérico.

figura 1.13

**CODIGO 93**, este código fué introducido en 1982 y fué diseñado como un complemento del código 39, en este código su longitud es variable, es un código continuo. Cada caracter codificado en este código está compuesto por nueve módulos, los cuales pueden estar marcados con elementos en negro o espacios en blanco, cada caracter cuenta con tres barras y tres espacios dentro de los nueve módulos mencionados, como se muestra en la figura 1.14.



código 93 compuesto siempre por tres barras y tres espacios en blanco.

figura 1.14

Este código cuenta también con símbolos especiales para el inicio y final del código de barras, estos son representados por un cuadrado ■, también se cuenta con los únicos cuatro elementos en círculo ●●●●, estos son utilizados para distinguir al código ASCII; al poner antes de un caracter podemos obtener cualquiera de los 128 caracteres ASCII, en la figura 1.15 se puede observar la forma en la cual el código 93 puede codificar diferentes caracteres.

Al ver esta tabla se podrán dar cuenta cómo el código 93 es más reducido que los códigos anteriores y con la ventaja de poder utilizar cualquiera de los caracteres ASCII.

## TESIS EL CODIGO DE BARRAS Y LA AUTOMATIZACION

Al momento de realizar nuestra codificación tendremos varias cosas que tomar en consideración; para poder ejemplificar tomaremos la figura 1.16 como punto de referencia.



FIGURA 1.16

Como podemos observar en la figura 1.16, la codificación inicia y termina con los símbolos de inicio y final los cuales habíamos denominado como un cuadrado en la figura

1.15, estos caracteres de inicio y final se tendrían que hallar en todo código 93, al final de este encontraremos siempre una barra y esta indica el final de la codificación e indica al lector de código de barras el fin de las barras y que el lector pueda comenzar su decodificación. El código 93 no es auto-chequeable por lo que utiliza dos elementos de seguridad, en este caso denominados como "C" y "K", estos elementos solamente son impresos, pero al leer el lector estos caracteres lo interpreta como un elemento de control y no lo transfiere como parte de los datos. Estos elementos de control se deducen a partir de los datos que se encuentran en el código de barras, analizando el ejemplo de la figura 1.16, el dato codificado fué "CODE 93 ", si observamos en la figura 1.15, cada letra tiene o cuenta con un valor numérico, en este caso la letra C=12, O=24, D=13, E=14, space= 38, 9=9, 3=3, para poder obtener el valor de " C " y " k " se tiene que realizar una serie sencilla de operaciones, las cuales se realizan y detallan una por una en la figura m-1.

Code 49, este código fué introducido en el mercado en el año de 1987, es una simbología especial para pequeñas aplicaciones, para publicar en pequeños objetos; este código en especial puede tener desde 2 hasta 8 módulos diagonales, los cuales se encuentran separados unos de otros por una simple línea separadora, que tiene la función de separar un área de otra del código de barras para así poder dividir en segmentos el código de barras y obtener mucho mayor información condensada en menor espacio y estas se muestran en la figura 1.17.

El número de caracteres que puede contener el código 49 por renglones se muestra en la tabla 1.3.

### Capacidad del código 49

número de renglones	número de caracteres alfanuméricos	número de caracteres numéricos
2	9	15
3	16	26
4	23	38
5	30	50
6	37	61
7	42	70
8	49	81

Cada renglón del código 49 contiene 18 barras en 70 módulos o espacios de distribución y, como en cualquier código de barras, la primera y última barra son los caracteres de inicio y final.

El número de renglones es determinado automáticamente por el código y este depende del número de caracteres que se estén codificando, cada renglón contendrá cuatro palabras (word), a este método de codificación le llamaremos (16, 4). Basado en los caracteres que se estén codificando, cada palabra puede estar en el rango de 0 y 2400 como máximo, el valor de esta palabra es determinado por la suma de todos los valores en la mitad derecha de la palabra y en la mitad izquierda el valor obtenido pero multiplicado por 49 veces este valor. Esto porque cada palabra contiene dos caracteres y en cada renglón hay 8 caracteres, cada palabra es decodificada en (16,4), en la estructura se encuentra definida la paridad y esta permite identificar de cual renglón se trata. El último caracter es el checkador y este se encuentra en todos los renglones, en el código 128 pueden ser utilizados todos los 128 caracteres ascii.

	VALOR		VALOR
0	0	0	24
1	1	P	25
2	2	Q	26
3	3	R	27
4	4	S	28
5	5	T	29
6	6	U	30
7	7	V	31
8	8	W	32
9	9	X	33
A	10	Y	34
B	11	Z	35
C	12	-	36
D	13	.	37
E	14	SPACE	38
F	15	\$	39
G	16	/	40
H	17	+	41
I	18	%	42
J	19	●	43
K	20	●	44
L	21	●	45
M	22	●	46
N	23	□	

codificación en código 93

figura 1.15

Datos	C	O	D	E	SP	9	3	"C"	"k"
Valor numérico	12	24	13	14	38	9	3		
Peso en "C"	7	6	5	4	3	2	1		
Peso en "K"	8	7	6	5	4	3	2	1	

Para poder obtener "C", obtenemos la suma de los siguientes productos:

$$(1 \times 3) + (2 \times 9) + (3 \times 38) + (4 \times 14) + (5 \times 13) + (6 \times 24) + (7 \times 12) = 484$$

El número obtenido lo dividimos entre 47

$$484 / 47 = 10.29$$

multiplicamos la parte entera por 47

$$10 \times 47 = 470$$

restamos este al resultado de la suma de productos

$$484 - 470 = 14$$

este número obtenido corresponde al valor de "C", según los valores dados en la tabla 1.15

Para poder obtener "K", realizamos algo similar a lo anterior

$$(1 \times 14) + (2 \times 3) + (3 \times 9) + (4 \times 38) + (5 \times 14) + (6 \times 13) + (7 \times 24) + (8 \times 12) = 611$$

El número 14 es el obtenido en el cálculo anterior

El número obtenido lo dividimos entre 47

$$611 / 47 = 13$$

multiplicamos la parte entera por 47

$$13 \times 47 = 611$$

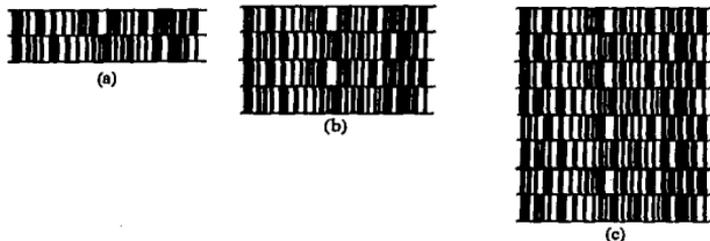
restamos este resultado de la suma de productos

$$611 - 611 = 0$$

este número corresponde al valor de "K", según los valores de la tabla 1.5

**figura m-1**

**OTROS**, Existen actualmente una gran cantidad de simbologías en que utilizan el código de barras para representar caracteres y después poder interpretarlo como datos en una lectura, de estos la gran mayoría son derivaciones o son el mismo código anterior solo con algunas mejoras, un caso muy específico podríamos citar en el mercado al código **16K**, este código es el mismo que el código **49**, la diferencia entre uno y otro radica simplemente que el código **49** tiene solo ocho renglones como máximo y el código **16K** tiene un máximo de 16 y las barras están representadas por espacios en blanco y los espacios en blanco están representados por barras es decir utiliza el llamado "video inverso". Debido a lo anterior, solo mencionaremos el nombre y daremos una breve descripción de estos códigos.



El código 49 puede tener desde 2 hasta 8 renglones en el caso (a) estamos utilizando el mínimo en el caso (b) estamos utilizando una etapa intermedia y en el caso (c) se encuentra el máximo.

**figura 1.17**

Código 11, fué descubierto en 1977 y es utilizado en aquellos casos donde se necesite una gran intensidad de datos y la densidad de los mismos tenga que ser muy alta, este se emplea mucho en aquellos productos que necesiten una etiqueta con código de barras pero no tengan mucho espacio disponible, este código se utiliza en dispositivos de comunicaciones y

equipos electrónicos. Debido a la gran intensidad de datos éste necesita dos dígitos verificadores para poder asegurar su lectura.

Matrix 2 of 5, este es una variación del código 11 y fué introducido por la compañía Nief en Holanda, solamente puede contener 10 dígitos y cuenta también con los dígitos de inicio y parada.

Delta Distance A, este código fué introducido por IBM en 1971, es un código discreto y numérico y en este el papel principal lo constituye la distancia entre barra y barra y no el tamaño o grosor de estas.

Postnet, este código nace con la necesidad para las estampillas postales de automatizar el proceso del envío de las mismas, éste utiliza 5 barras y cuatro espacios en blanco para representar a cada caracter y en cada caracter el largo de las barras juega un papel importante debido a que siempre como mínimo dos deben ser de la mitad del tamaño de las tres restantes.

PDF417, En el año de 1990 la compañía Simbol technologies, y siguiendo los pasos del código 49, inventa este código con la característica especial de poder codificar un gran número de datos en un área muy pequeña y este código cuenta con una limitante; para poder leer este tipo de código, se necesita un tipo especial de lector y un tipo especial de codificador, los cuales son propiedad de la empresa Simbol y no es posible encontrarlos muy fácilmente en el mercado del código de barras, actualmentete este código es empleado en las formas de agencias aduanales de la república Mexicana, debido a la gran cantidad de información que pueden contener.

## ***CAPITULO II***

**IMPRESION DEL CODIGO DE BARRAS**, la impresión del código de barras es de vital importancia ya que, si la impresión es de mala calidad, la lectura del código de barras puede llegar a tener errores, aunque los dispositivos lectores suelen tener técnicas de corrección de errores para así poder llegar a la lectura correcta.

La impresión de las barras para el código de barras seleccionado se lleva a cabo con los estándares mencionados en el capítulo 1, para poder tener una mejor idea y ser más precisos en la selección del código de barras y el grosor y tamaño de las barras se puede acudir a AMECOP, la cual tiene y cuenta con toda la información necesaria para los estándares de cada código de barras, así como especificaciones técnicas; cabe mencionar que en el mercado del software existen diferentes paquetes, los cuales imprimen directamente el código de barras con todos los estándares necesarios, por mencionar algunos se puede contar con **Matrix Label**, **Barcom**, **Print Bar**.

Cabe aclarar que el código de barras puede ser impreso en cualquier tipo de material o superficie, llámese plástico, tela, cartón, etiquetas, acero, en pocas palabras cualquier superficie que tenga la capacidad de reflejar la luz por mínima que esta sea, la única recomendación es que no se emplee un fondo de color rojo por tener cualidades que pueden afectar a nuestra lectura.

Existen impresoras especiales de código de barras, las cuales son llamadas impresoras de transferencia térmica o impresoras de transferencia directa, aunque, como se mencionó anteriormente, no son necesarias para imprimir un código de barras, simplemente que estas impresoras son las que reúnen las mejores características para imprimir código de barras.

Estas impresoras necesitan papel especial, el cual se mencionará posteriormente así como en qué caso se podría utilizar cada una.

**IMPRESORAS DE CODIGO DE BARRAS**, como se había mencionado, la impresión del código de barras puede ser en cualquier tipo de impresora, pero la calidad de impresión es muy variada; podemos hacer una clasificación de las impresoras y el orden de importancia que tendrían en la impresión del código de barras.

### **Impresoras de código de barras**

- 1.- Impresoras de Transferencia Térmica
- 2.- Impresoras de Transferencia Directa
- 3.- Impresoras Láser
- 4.- Impresoras Matriz de Puntos 15 agujas
- 5.- Impresoras Matriz de Puntos 9 agujas

Estas impresoras están incorporadas en orden de calidad y servicio para imprimir código de barras, no quiere decir que una impresora sea mala o no, simplemente que para imprimir código de barras no son lo más recomendable.

De estos cinco tipos de impresora 2 son especiales para código de barras y 3 son de propósito general, aunque las impresoras de tipo láser nos dan muy buena calidad de impresión de código de barras, las impresoras especiales para código de barras son las dos primens.

En la actualidad existen un tipo de impresoras especiales para el código de barras, estas impresoras son llamadas de transferencia o directas, pueden trabajar cualquiera de los dos métodos de impresión de código de barras, estas impresoras pueden ser programadas para que trabajen ya sea del modo de transferencia o del modo directo, de estas impresoras existen en el mercado muchas y de muy diferentes marcas como lo son **SATO, COGNITIVE SOLUTION, ZEBRA**, etc.

Existe un método más para imprimir el código de barras y este será el llamado serigráfico; utilizando métodos serigráficos podemos conseguir muy buena calidad en los códigos de barras, esta técnica se emplea en aquellos productos a los cuales se les quiera incorporar el código de barras en el propio recipiente del producto, este es un método muy difundido en la actualidad.

**SELECCION DEL TIPO DE IMPRESORA**, de ahora en adelante nos enfocaremos a dos tipos de impresoras, las cuales serán las impresoras de transferencia y las directas, para la selección del tipo de impresora se tendrán que tomar diversos factores entre los cuales mencionaremos los siguientes:

1.- Tiempo de uso de la etiqueta, esto es dependiendo del tiempo que esté la etiqueta en uso, durante cuánto tiempo la vamos a utilizar; por ejemplo, si nuestra etiqueta estará por un largo período de tiempo en uso se recomendará la impresora de transferencia térmica, si por el contrario ésta solo se utiliza una sola vez, como por ejemplo en las terminales aéreas donde etiquetan el equipaje se recomendaría la impresora de transferencia directa.

2.- Uso frecuente, cuántas veces se tendrá que utilizar esta etiqueta y con qué frecuencia; si la utilizamos muy frecuentemente o tenemos contacto directo con la etiqueta del código de barras se recomendaría la impresora de transferencia térmica, si por el contrario solo la utilizamos una, dos o tres veces se recomendaría la impresora de transferencia directa.

3.- El código de barras tiene contacto directo con el agua, en este caso se tendrá que utilizar exclusivamente las etiquetas de transferencia térmica, debido a que estas cuentan con una capa de plástico, la cual protege la impresión del código de barras.

En general, la selección del tipo de impresora depende del uso que se le quiera dar y qué tan rudo sea el trato a darse a esta etiqueta así como el número de veces que sea escaneada.

Otro punto muy importante sería ver qué tipo de lector de código de barras se

utilizaría para leer los códigos debido a que hay lectores que son de contacto físico y hay otros que son de tipo láser y no existe ningún contacto físico con el código de barras.

**SELECCION DEL TIPO DE PAPEL PARA EL CODIGO DE BARRAS,** esta selección estará basada exclusivamente en el tipo de impresora que hallamos seleccionado y la aplicación que le daremos a nuestra etiqueta, si tenemos la impresora directa tendremos que conseguir papel de transferencia directa, este papel tiene la cualidad de que, al ejercer cierta presión sobre este papel, se libera un tipo especial de tinta en el área mencionada y el código de barras queda automáticamente marcado. En este tipo de papel hay de diferentes estilos como por ejemplo las de tipo etiqueta autoadheribles o no, las etiquetas de cartón, de poliéster, etc. La selección de esta será, como se mencionó anteriormente, dependiendo del uso a darle.

En el caso de las impresoras de transferencia térmica se podrá utilizar cualquier tipo de papel debido a que, al utilizar este tipo de impresora, contaremos con una impresión autoadherible del código de barras; esta técnica en principio es parecida a las máquinas de escribir electrónicas en las cuales la letra no queda impresa, sino autoadherida, la única desventaja de esto es la de tener que comprar constantemente esta cinta para poder llevar a cabo la impresión de nuestras etiquetas.

**SELECCION DEL TAMAÑO DEL CODIGO DE BARRAS,** es de vital importancia tener en cuenta el tamaño de impresión del código de barras debido a que, al adquirir una impresora de transferencia térmica o una impresora directa, la característica principal será el tamaño máximo de impresión a lo ancho, ya que, si se desea imprimir en un ancho mucho mayor, no será posible debido a que la cabeza de impresión no tendrá la capacidad de imprimir.

***CAPITULO III***

**QUE ES EL DECODIFICADOR ?** es un dispositivo que se encarga de transformar una señal de un tipo a un código, en este caso estamos hablando de señales eléctricas de forma cuadradas a señales en un código especial ASCII.

**QUE ES EL DECODIFICADOR DE CODIGO DE BARRAS?** el decodificador de código de barras es un dispositivo que transforma una señal de tipo cuadrada obtenida de un lector de código de barras, sea del tipo que fuere (WAND, CCD, LASER, MANOS LIBRES), en datos en código ASCII, este procedimiento se lleva a cabo de la siguiente manera.

Primero, el decodificador debe ser programado, poner los parámetros adecuados para una correcta lectura y evitar alguna pérdida de tiempo, esto en la mayoría de los decodificadores se lleva a cabo a través de dip-switch, con los cuales se logra, por ejemplo, si se desea que este transmita los dígitos verificadores en el caso de códigos de barras que los tengan, se puede habilitar un solo tipo de código de barras y así ahorrar tiempo de reconocimiento de código de barras, o por el contrario se puede habilitar todos si se desea.

Segundo, el decodificador recibe la señal proveniente del lector de código de barras.

Tercero, el decodificador pule la señal proveniente del lector de código de barras, esto es, le quita cualquier tipo de señal que sea ajena a la señal original así como ruido de cualquier naturaleza.

Cuarto, si el decodificador fué programado para un solo tipo de código de barras, simplemente decodifica, si no es así, éste primero tiene que identificar de qué tipo de código se trata y posteriormente decodificar.

En general este es el procedimiento que se lleva a cabo en el decodificador teniendo por supuesto algunas otras especificaciones.

La mayoría de los decodificadores de código de barras cuentan con dip-switch, con los cuales se programa al decodificador, en algunos otros casos esto se logra a través de códigos de barras en secuencias preestablecidas, las cuales le indican al decodificador que lo están programando.

Los decodificadores suelen dividirse en grupos para poder ver sus aplicaciones específicas, cada grupo tiene una función específica como, por ejemplo, la de conectarse a alguna computadora especial a algún otro grupo que sirva para terminales de red en las cuales queramos utilizar el código de barras, un caso muy especial es el de las máquinas APLEE MACHINTOCH, las cuales necesitan conectores especiales así como decodificadores especiales, otro caso muy común son las máquinas IBM, las cuales necesitan un par de convertidores para sus máquinas de punto de venta.

Los decodificadores pueden ser de muy diferentes marcas y modelos, los más comunes son los de las marcas UNITEC y PERCON, estos son los más difundidos en el ámbito comercial.

**DECODIFICADORES UNDECODÉD**, estos decodificadores se denominan así porque se encuentran separados del lector de código de barras, son un módulo aparte en el proceso de lectura de los códigos, la figura 3.1 nos muestra modularmente en donde se debe colocar este lector de código Undecoded de barras.

Estos decodificadores también son denominados en ocasiones de tipo teclado, ya que al conectarlos a la máquina estos se conectan en serie con el teclado y los datos son introducidos por la vía del teclado, utilizando el puerto del teclado.

Estos decodificadores, de manera general, son programados a través de dip-switch y en algunas raras ocasiones a través de código de barras.

**DECODIFICADORES DECODED**, estos decodificadores son especiales en cuanto a su construcción debido a que estos están dentro del dispositivo lector de código de barras, llámese CCD, LASER o MANOS LIBRES, en el interior del cuerpo del lector es habilitado el decodificador.

En la figura 3.2 se muestra modularmente como debe ir conectado el decodificador Decoded a la computadora.

En la figura 3.2 podemos ver como el lector y el decodificador de código de barras se encuentran en un solo dispositivo, aunque ahorramos hardware, surge la necesidad de crear o adquirir un software para introducir los datos a la computadora a través del puerto serial com1, com2, etc., este software nos habilita el puerto serie para poder recibir datos a través de él.



En esta figura se ve cómo el decodificador Undecoded debe de quedar entre la computadora y el lector de código de barras

figura 3.1



En la figura se muestra cómo se conecta el decodificador decodificado, el cual es conectado a la computadora via puerto serial.

figura 3.2

## ***CAPITULO IV***

**QUE ES UN LECTOR DE CODIGO DE BARRAS ?** el lector de código de barras es un dispositivo que tiene como función principal la de poder leer una serie de caracteres impresos ( barras y espacios), para transmitirlos a un dispositivo almacenador de datos o algún tipo de computadora. El proceso de la lectura se lleva a cabo a través de dispositivos de convertidores analógicos digitales, primero emitiendo un haz de luz , después de reflejarse en una superficie esta regresa al lector donde el reflejo es interpretado y convertido en una serie de impulsos digitales, los cuales son codificados para convertirlos a datos que puedan ser interpretados por una computadora o algún dispositivo especial.

Cualquier tipo de lector óptico emplea técnicas foto-electrónicas para leer los diferentes tipos de códigos de barras, la forma como ocurre este proceso de lectura se muestra en la figura 4.1.

En la figura observamos cómo al emitir un haz de luz esta es dirigida al código de barras y, al llegar a su objetivo, es reflejada hacia donde se encuentra el foto-detector, el cual recibe esta señal que contiene la información completa de lo que está escrito en código de barras. Al momento de que el foto-detector detecta barras y espacios emite señales eléctricas, esto se ve en la figura 4.2, esta señal es parecida a la que obtendríamos si conectáramos un osciloscopio al extremo del foto-detector. Originalmente esta señal obtenida sería de forma senoidal y, después de pasar por un dispositivo denominado WAVESHAPER, la señal es convertida en una señal de forma cuadrada como la que se muestra en la figura 4.2.

El diagrama completo de la entrada de los datos se ilustra en la figura 4.3., en este se ve cómo la señal es emitida, después leída por un foto-lector, la señal del foto-lector es amplificada y después es convertida en una señal de forma cuadrada, la señal que se obtiene del foto-lector es una señal aproximadamente del tipo senoidal.



figura 4.1

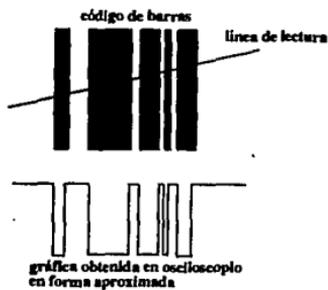


figura 4.2

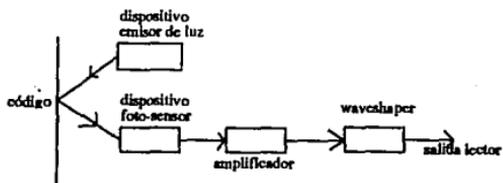
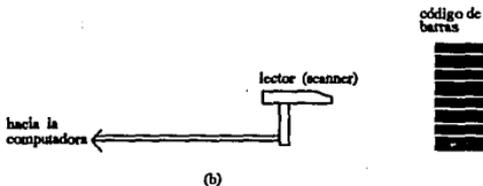
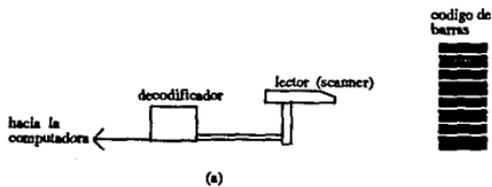


figura 4.3

**TIPOS DE LECTORES DE CODIGO DE BARRAS**, Los lectores de códigos de barras se suelen dividir en varios tipos, como lo son el CCD, VLD, etc, pero cada uno de estos puede tener dos versiones y estas son UNDECODED y DECODED, esto es la manera como están contruidos intetnamente y de esto depende la manera como se deben de conectar. En la figura 4.4 se muestra cómo se tienen que conectar los lectores UNDECODED y DECODED, los primeros son llamados Undecoded debido a que no cuentan con un decodificador integrado dentro del dispositivo lector y necesitan un decodificador externo denominado DECODIFICADOR, para poder introducir los datos a la computadora. Los segundos o llamados DECODED son dispositivos que cuentan junto con el dispositivo lector, con un decodificador interno y por lo tanto, sus datos son accésados directamente a la computadora.

Estos tipos UNDECODED y DECODED son también llamados dispositivos vía teclado o dispositivos vía puerto serial respectivamente.

La utilización de algún tipo de estas pistolas depende de factores internos a la computadora así como externos.



En la figura (a) se muestra como debe de ser la conexión de los lectores UNDECODED, los cuales necesitan un decodificador para poder introducir los datos a la computadora.

En la figura (b) se muestra como se debe de ser la conexión de los lectores DECODED, los cuales no necesitan un decodificador para conectarse a la computadora.

figura 4.4

**LECTORES WAND**, estos lectores son los más difundidos actualmente ya que su costo es muy inferior a cualquier otro dispositivo lector. Comúnmente conocido como lápiz lector este lector de tipo wand fué el primer lector de código de barras, únicamente lo encontraremos del tipo UNDECODED debido a su tamaño, aproximadamente es del tamaño de una pluma fuente, este siempre debe de ir acompañado de un decodificador externo.

Existen diferentes marcas de lápices lectores, entre las más comunes tenemos las marcas HP, Spectra Physics, Percon, ETC.

La utilización de este tipo de escaner depende de factores externos como lo son: qué tan rápido se desée realizar la lectura del código de barras, qué tan continuamente utilizaremos el lector y, el factor económico, ya que este escaner es el más barato dentro de los lectores de código de barras.

La forma como se lleva a cabo el escaneo en este dispositivo es pasando por encima el wand y correrlo a lo largo del código de barras de derecha a izquierda o de izquierda a derecha esto debido a que es omnidireccional, es muy importante que al realizar el escaneo el wand tenga un contacto físico con el código de barras, el wand debe estar a un ángulo de inclinación de aproximadamente 45 grados para poder llevar a cabo la lectura del código de barras.

En la figura 4.5. se muestra una fotografía de un lector tipo wand, en esta podemos notar la similitud con una pluma fuente así como un cable conector el cual es el dispositivo por el que se conecta este lector, la única diferencia entre este lápiz y otros solamente será la marca y el tipo de construcción del lápiz, por lo general todos estos son muy parecidos.



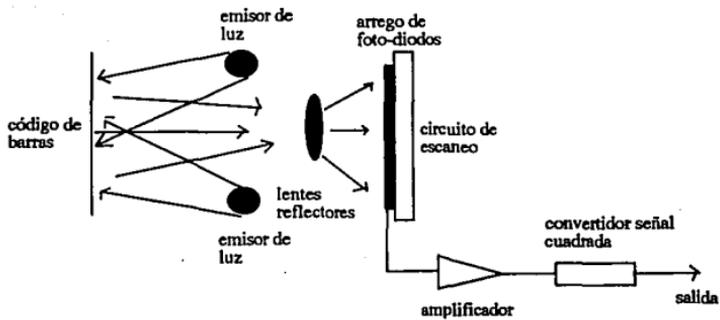
Este es un tipo común de lápiz óptico todas, las marcas son muy parecidas

Figura 4.5

**LECTORES CCD**, estos lectores de código de barras se denominan lectores de rastrillo o de contacto, esto se debe a que, para poder leer el código de barras, es necesario que el escaner tenga un contacto físico directo con el código de barras, estos cuentan con las características de leer un largo específico de código, este no puede leer etiquetas mucho mayores al ancho de su ventana o zona de escaneo, estos físicamente se parecen a una rasuradora eléctrica y en la parte superior se encuentra la ventana de escaneo, siendo aquí donde se lleva a cabo la lectura del código de barras, cuando hablamos de un escaner de contacto decimos que tiene un tamaño máximo, es de gran importancia al adquirir un escaner de este tipo ver el ancho de ventana debido a que esta medida nos dice qué tan larga puede ser nuestra etiqueta de código de barras, si esta es más grande que el ancho de ventana la lectura del código de barras no podrá llevarse a cabo porque la ventana no puede abarcar el total de la etiqueta de código de barras, estos lectores poseen la característica de contar con un gatillo o botón, con el cual podemos comenzar el escaneo de estos códigos, en ocasiones se les llaman de "contacto con gatillo".

La forma como se lleva a cabo la lectura del código de barras se muestra en la figura 4.6 . En esta se ve cómo está compuesta la ventana de lectura, el lector emite luz por medio de diodos emisores de luz, los cuales son reflejados en la superficie del código de barras y esta luz rebotada pasa a través de lentes, los que distribuyen el reflejo por medio de una serie de foto-diodos que se encargan de convertir este reflejo en una señal eléctrica mediante una serie de circuitos de escaneo para posteriormente pasar esta señal a una etapa de potencia y enseguida al convertidor a señales cuadradas, terminando en la salida para decodificar. En esta figura se ve por qué el tamaño de la ventana es importante, ya que la luz emitida abarca solo un largo específico y, si el código de barras es más grande a este tamaño, no podrá ser leído y, por consiguiente no podrá ser decodificado.

La forma básica de este tipo de escaner o lector de código de barras se muestra en la figura 4.7, esta puede sufrir algunas modificaciones en diferentes tipos de marcas pero siempre tendrán un parecido con la figura 4.7.



En esta figura se ve como se lleva acabo la lectura del código de barras en un lector CCD

figura 4.6



En esta imagen se ve la forma básica de un lector del tipo CCD y podemos observar como para poder leer hay que realizar un contacto físico con la etiqueta para que pueda ser leída

figura 4.7

## **TESIS EL CODIGO DE BARRAS Y LA AUTOMATIZACION**

Los lectores CCD son muy utilizados en aquellas ocasiones en que se necesite un lector rápido y de no muy alto costo, es un lector confiable debido a que la lectura la lleva no por una línea de escaneo como, lo es el wand, sino la lleva por toda una área de escaneo, es decir, escanea toda el área de la ventana corrigiendo en algunas ocasiones errores de impresión de las etiquetas; es muy común encontrarlos en tiendas de autoservicio medianas.

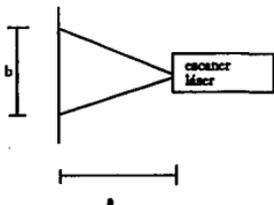
Estos lectores pueden ser undecoded, que necesitan un decodificador externo o decoded, que se puedan conectar al puerto serial de la computadora; los más comunes en el mercado son de la marca SPECTRA PHISICS en los modelos 210 y 210R, estos son undecoded y decoded respectivamente; los hay en otras marcas como lo son METROLOGIC y SIMBOL.

**LECTORES VLD**, estos son lectores a distancia o también son llamados del tipo láser, esto porque cuentan con un láser que permite realizar lecturas de código de barras a través de un rayo láser y este puede ser a una distancia considerable sin existir contacto físico con el código de barras.

Estos escaners internamente cuentan con un mecanismo rotomotor en movimiento circular y dentro de una zona poligonal en la cual se encuentran colocados una serie de espejos que tienen la función de reflectar la luz emitida por el láser y dirigirla hacia el código de barras para leerlo. La velocidad de escaneo de este tipo de lectores es de 40 - 2000 escaneos por segundo, el ojo humano ve esta serie de líneas como una sola sin alcanzar a distinguir las líneas emitidas por el láser y reflectadas por los espejos hacia el código de barras.

Estos lectores necesitan una fuente externa de alimentación en el caso de ser del tipo DECODED, en el caso de los UNDECODED no necesitan este tipo de aditamento, pero sí de un decodificador externo para poder introducir los datos a la máquina vía teclado.

En este tipo de lector, por ser láser y tener cualidades ópticas, la determinación del tamaño de máximo de lectura del código de barras dependerá de la distancia del láser al código de barras, teniendo la siguiente regla: entre más grande sea la distancia entre el código de barras y el lector láser mayor será la abertura del láser y mayor será el tamaño de lectura máxima, y entre más pequeña sea la distancia menor será el tamaño de posible lectura del código de barras. En la figura 4.8 se muestra este efecto y el por qué del tamaño de lectura.



Entre mayor sea (a) mayor será (b) y si (a) decrece también decrece (b) es decir son proporcionales.

figura 4.8

La forma básica de un lector de código de barras tipo láser es como la que se muestra en la figura 4.9.



**Lector tipo laser SP400 marca Spectra Phisics**

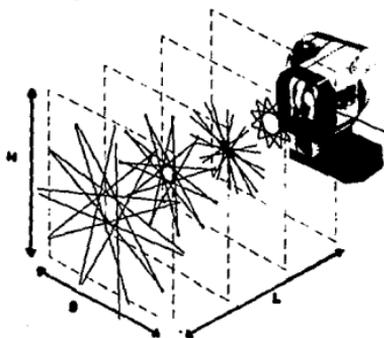
**Figura 4.9**

Cabe mencionar que en la actualidad se encuentran este tipo de lectores tipo láser en una gran variedad de modelos y rangos, los rangos actuales son: el láser puede leer códigos de barras a una distancia máxima de 10 mts. y un mínimo de 5 cm. los modelos son desde una pistola que pueda leer un código muy denso, esto es, la separación entre barras y espacios sea muy chica, hasta los que no son muy densos y aquellos que pueden leer a través de cristales o ventanas.

Estos lectores pueden ser de dos tipos undecoded o decoded y de esto depende la forma en la cual se conectarán a la computadora solamente, existen de diferentes marcas como lo son Spectra Phisics, Metrologic, Aedex, etc.

**LECTORES MANOS LIBRES**, este tipo de lector es muy frecuente utilizarlo en supermercados o tiendas de autoservicio, en las cuales el proceso de captura de datos sea muy

grande y los tiempos para la lectura de estos sea mínimo o se tenga que optimizar lo más posible, los lectores a manos libres son los más adecuados para este tipo de trabajo y esto se debe a diferentes factores entre los cuales tenemos que, la velocidad de escaneo de estos lectores es de 2000 escaneos por segundo lo que lo convierte en un lector muy rápido, estos no son ni de contacto ni son tipo láser, son una derivación de estos últimos ya que cuentan con un láser el cual a través de una serie de espejos, proporcionan de 20 a 30 líneas de lectura dependiendo del tipo empleado y estas en lugar de dominar solo una sola línea dominan toda un área es decir son omnidireccionales y esto quiere decir que al pasar el lector el área dominada por el código de barras será leído y por tener un alcance determinado se dice que pueden leer en cualquier parte del volumen establecido. Esto se puede ver en la figura 4.10.



En esta figura se puede observar cómo las líneas de lectura dominan una área y por la distancia a la que pueden leer se dice que pueden leer en todo este volumen; para sacar el volumen simplemente realizamos

una operación matemática  $V = H \cdot B \cdot L$

Figura 6.10

Este tipo de lector, como se puede ver, cuenta con la capacidad de tener o contar con toda un área de escaneo, la cual brinda una mayor velocidad de lectura y un menor tiempo de escaneo; este tipo de lectores en México son muy utilizados en tiendas como lo son Aurrerá,

Videocentro y algún otro tipo de centro comercial.

Estos lectores pueden ser empotrados en paredes o mesas para poder pasar el código de barras por encima de ellos y así poder tener una posición más cómoda para efectuar la lectura del código de barras. También pueden ser montados con una base especial, la cual nos posiciona este lector a la altura deseada.

Existen dos tipos de estos lectores los DECODED y los UNDECODED en el primer caso tendremos que dotar al lector de una fuente de poder para poder hacer que funcione este lector y también tener en la memoria de la computadora un software que nos habilite el puerto serial para la entrada de los datos, en el segundo caso tendremos también que conseguir una fuente de poder y un decodificador para poder introducir los datos vía teclado.

***CAPITULO V***

**ESQUEMA DE CONECCION DEL CODIGO DE BARRAS**, se cuenta principalmente con dos tipos de conexiones para código de barras y estas conexiones son para la forma decoded y en forma undecoded.

Estas varían en cuanto a grado de complejidad, la más sencilla es la forma undecoded, en la cual solo hay que conectar una serie de cables de manera adecuada y esto es suficiente para leer el código de barras; en el método decoded es necesario conectar únicamente al puerto serial y realizar un software para introducir los datos por el puerto serial, este software puede ser realizado por un programador o adquirido a una compañía que cuente con este.

**ESQUEMA DE CONECCION UNDECODED**, en la figura 5.1. se puede ver como se conecta un decodificador undecoded y un lector de código de barras a una computadora personal o PC.

En esta se ve cómo simplemente es necesario conectar un conjunto de cables y así podemos introducir los datos vía teclado.

En esta figura el teclado se conecta al decodificador a través de un conector ding o miniding, el decodificador se conecta a la entrada del teclado en la computadora por medio de un conector ding o miniding y el escaner se conecta a un puerto adicional que tiene el decodificador y esta es la forma esencial del tipo undecoded.

Cabe mencionar que la forma de conexión no cambiará si se utiliza un escaner CCD, LASER o MANOS LIBRES, en cualquiera de estos casos se tendrá la misma conexión.

En el caso de querer conectar un decodificador a una red es necesario solamente cambiar los tipos de cables, conectores para introducir los datos al sistema, sea esta la terminal que fuere.

**EESQUEMA DE CONECCION DECODED**, en la figura 5.2 se puede ver la manera como se conecta un lector decoded y como se puede observar es muy sencilla, ya que se conecta simplemente al com1 o puerto serial, no es necesario que el puerto sea com1, puede ser cualquiera de los puertos seriales existentes en la computadora.

Cabe aclarar que en esta conexión es necesario instalar un software para poder habilitar el puerto serial seleccionado y así poder introducir los datos a la computadora a través del puerto serial, el software tiene la función de emular teclado y así no provocar ningún tipo de conflicto con algún otro software instalado en la memoria de la computadora.

Un software muy conocido en el medio es el ofrecido por la marca Percom llamado BARCOM, este programa se encarga de todos los parámetros necesarios para introducir los datos a la computadora.

También se puede observar como el escaner necesita una fuente de poder la cual proporcionará la energía para activar este lector de código de barras, la mayoría de las lectoras necesitan una polarización de 9V.

Hay que mencionar que no existe diferencia alguna entre conectar un decoded en un escaner CCD, LASER, o MANOS LIBRES siempre las conexiones serán las mismas .

Si este tipo de lector se desea conectar a la terminal de una red es necesario ver que las terminales cuenten con un puerto serial libre para poder conectar el lector, después al disco de arranque adicionar el software para poder habilitar el lector de código de barras que se desee, otra forma sería instalando el archivo habilitador de puerto serial en el servidor y habilitar desde el server los puertos seriales de la terminal que se desee.

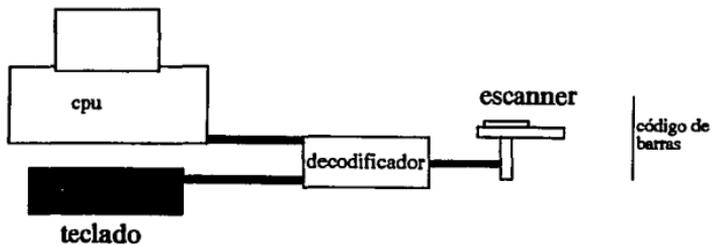
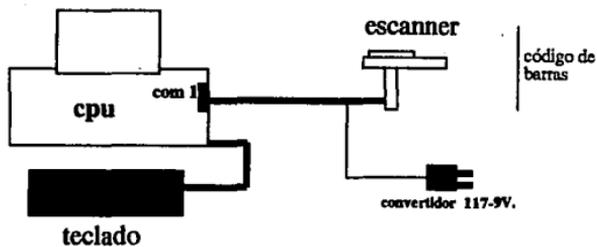


figura 5.1



En esta conexión necesitamos un software de comunicaciones para poder introducir los datos a la computadora via puerto serial.

figura 5.2

***CAPITULO VI***

## **TESIS EL CODIGO DE BARRAS Y LA AUTOMATIZACION**

**QUE ES AUTOMATIZACION,** automatización significa el poder llevar a cabo cualquier proceso sin necesidad de la intervención o supervisión humana siendo a través de una serie de mecanismos eléctricos, mecánicos o algún otro dispositivo, los cuales lleven a cabo su propósito sin necesidad de tener supervisión, la automatización puede emplearse en cualquier área del conocimiento humano.

**ELEMENTOS NECESARIOS PARA LOGRAR LA AUTOMATIZACION UTILIZANDO EQUIPO DE TECNOLOGIA DE CODIGO DE BARRAS,** No todos los procesos pueden ser automatizados utilizando el código de barras, pero una gran cantidad de ellos sí como lo son inventarios, nómina, captura de información, entradas y salidas de productos, puntos de venta y muchos otros.

Los elementos necesarios para poder automatizar procesos utilizando código de barras son simplemente 3, el equipo que imprima el código de barras es el punto de partida; el equipo lector de código de barras y el equipo donde se procese la información que se obtiene de los segundos, a través de un sistema establecido que puede ser cualquiera.

En el caso de los primeros puede ser alguno de los equipos mencionados en el capítulo II, mediante serigrafía o equipos de impresión especiales para el código de barras; el equipo lector de código de barras será alguno de los mencionados en el capítulo IV, esto depende de la aplicación que se tenga; por último, el equipo donde se procese la información puede ser cualquier equipo de cómputo, desde un PC hasta una poderosa HP9000, el único requerimiento es el de contar con software para procesar los datos obtenidos en los equipos de código de barras.

**APLICACION DEL CODIGO DE BARRAS EN LA AUTOMATIZACION EN CASOS PRACTICOS** , tomaremos tres de los principales casos prácticos que se presentan para la automatización de procesos utilizando el código de barras:

1.- PROBLEMA :

Recolección de inventarios.

La recolección de inventarios físicos es un trabajo necesario en cualquier tienda de autoservicio o cualquier otro tipo de establecimiento que maneje productos o que tenga inmuebles en su establecimiento. Para poder llevar una toma de inventarios, el método tradicional es el de ir al lugar donde se encuentran los materiales deseados para la toma de su inventario; posteriormente ir producto a producto, leer el nombre del producto, reconocer su clave y anotarla junto con su existencia para posteriormente estos datos pasarios a un capturista y que este los introduzca a un dispositivo computacional y procesarlos; este resulta un problema tedioso y complicado sobre todo en lugares donde se tenga una gran variedad de productos y tomando en cuenta que hay establecimientos que toman inventarios 2 veces por mes termina siendo muy pesado.

SOLUCION:

Analizando el problema se da uno cuenta que existe en primera instancia una duplicidad de trabajos en el sentido de la recolección de los datos, primero por la persona que toma físicamente el inventario y posteriormente el capturista que introduce los datos a la computadora para procesarlos, en un segundo análisis se ve que el rango de error en la captura de los datos es alto debido a que intervienen en este proceso 2 personas, una en la toma de los datos y otra en la introducción de los datos a la computadora. El tercer punto sería el tiempo de toma de los inventarios ya que es demasiado tardado.

Debido a estos tres puntos mencionados nos damos cuenta de la necesidad de automatizar este proceso con el menor rango de error posible y disminuir el tiempo de la toma de los inventarios.

Partiremos pensando que el lugar donde se requiere de esta toma de inventarios es una tienda de productos comestibles como lo pueden ser tiendas de autoservicio (AURRERA,

COMERCIAL MEXICANA, GIGANTE, ETC.) y tomando en cuenta que todos sus productos actualmente cuentan con el código de barras impreso ya contamos con un avance notable, lo único que necesitamos es un dispositivo que lea estos códigos y los almacene para poder tomar el inventario.

Existen diferentes y muy variados productos los cuales podemos ocupar para la toma de estos inventarios, específicamente existen productos como los denominados **HAND HELD**, o recolectoras de datos, las cuales son dispositivos con memoria e incluyen dispositivos que nos permiten conectar equipos lectores de códigos de barras o existen algunos híbridos los formados por la combinación de los dos, son dispositivos recolectores de datos y cuentan con un lector de código de barras. En este caso el equipo sugerido es el lector de la marca **NIPPON DENSO (Toyota)**, el cual es una combinación de recolector y lector de código de barras. Este equipo se muestra en la figura 6.1, en este se ve el dispositivo lector.

Teniendo resuelto el problema del tipo de equipo a utilizar, la cuestión se reduce a realizar un programa que se encargue de almacenar la clave del producto, en este caso el código de barras y el respectivo número de unidades de cada una y que el mismo programa se encargue de la transmisión de la información al equipo donde se vaya a transmitir la información para procesarla.

El equipo seleccionado esta vez se necesita programar en un lenguaje denominado **BASIC** compiler el cual es muy parecido al lenguaje de programación **BASIC** común y corriente y el programa realizado se detalla a continuación.

Este programa funciona de la siguiente manera, una primera pantalla pregunta:

#### **INVENTARIO**

**1.- COLECTAR DATOS**

**2.- TRANSMITIR INFORMACION**

Al oprimir el número uno es cuando hay que darle la lectura del código de barras y la cantidad de los mismos y, cuando se ha terminado con la captura de los mismos, con la tecla **F1** del equipo regresamos al primer menú.

**COLECTAR DATOS**

**CLAVE:**

**CANTIDAD:**

Cuando ya terminamos la captura de nuestros inventarios y estamos dispuestos a transmitir la información a nuestro equipo de procesamiento, el equipo lector lo conectamos a nuestra computadora vía puerto serial y oprimimos el número 2 para proseguir con la transmisión de la información en forma de un archivo ASCII el cual posteriormente lo procesaremos para obtener el inventario físico real.

**CONCLUSIONES:**

Con este programa logramos los objetivos planteados, suprimir la duplicidad de captura de información al reducirla a una sola y disminuir el rango de errores a un dato en un millón de lecturas, lo cual es el estándar en los equipos de códigos de barras y, por último, reducir el tiempo de la toma de inventarios a un mínimo del tiempo utilizado con el método tradicional.

TESIS EL CODIGO DE BARRAS Y LA AUTOMATIZACION

INIT:

```

DT.LEN%=13
DIM ITEMS(13)
ENTS=CHR$(&h0D)
BS=CHR$(&h08)
CS=CHR$(&h18)
F1$="A"
SPACES=" "
MENU1$="Colectar datos"
MENU2$="Trans. datos"
DT.NAMES="INVENT00.TXT"
GOSUB CAL.RCD
    
```

MENU:

```

CLS
CURSOR OFF
PRINT "INVENTARIO"
PRINT "1: ";MENU1$
PRINT "2: ";MENU2$
    
```

MENU.01:

```

WAIT 0, &h01
K$=INKEY$
IF K$="1" THEN
    GOSUB CLCT.DT
    GOTO MENU
ENDIF
IF K$="2" THEN
    GOSUB TRMT.FL
    GOTO MENU
ELSE
    MENU.01
ENDIF
    
```

CLCT.DT

```

B.OPEN%=0
DT.CNT%=0
ERCODE%=0
CLS
PRINT MENU1$
LOCATE 12,4
PRINT RIGHT$("00000"+MID$(STR$(RD.CNT%), 2), 5);
LOCATE 1, 2
PRINT ">";
CURSOR ON
    
```

CL.DT.01:

```

IF ( B.OPEN%=0 AND DT.CNT%=0 ) THEN
    OPEN "BAR:" AS #1 CODE "A", "I", "M"
    B.OPEN%=1
ENDIF
    
```

TESIS EL CODIGO DE BARRAS Y LA AUTOMATIZACION

CL.DT.02:

```

WAIT 0, &h03
IF ( B.OPEN%=1 AND DT.CNT%=0 ) THEN
    IF EOF(1)=0 THEN
        INPUT #1, ITEMS$
        GOSUB DT.SAVE
        GOTO CL.DT.03
    ENDIF
ENDIF
K$=INKEY$
SELECT K$
CASE ENT$
    IF DT.CNT%>0 THEN
        GOSUB DT.SAVE
    ENDIF
CASE C$
    IF DT.CNT%>0 THEN
        ITEMS$=""
        DT.CNT%=0
        GOSUB DSP.DAT
    ENDIF
CASE BS$
    IF DT.CNT%>0 THEN
        ITEMS$=LEFT$(ITEMS$,
            DT.CNT%-DT.CNT%-1
        )
        GOSUB DSP.DAT
    ENDIF
CASE ELSE
    K%=ASC(K$)
    IF ( K%>=&h30 AND K%<=&h39 ) THEN
        IF DT.CNT%<DT.LEN% THEN
            ITEMS$=ITEMS$+K$
            DT.CNT%=DT.CNT
            %+1
        ENDIF
        CLOSE #1
        B.OPEN%=0
        GOSUB DSP.DAT
    ENDIF
ENDIF
END SELECT

```

CL.DT.03

```

IF ( K$=F1$ OR ERCODE%<>0 ) THEN
    CLOSE
    B.OPEN%=0
    RETURN
ELSE
    GOTO CL.DT.01
ENDIF

```

TRMT.FL

TESIS EL CODIGO DE BARRAS Y LA AUTOMATIZACION

```
GOSUB CAL.RCD
CLS
PRINT MENU$
LOCATE 1,3
PRINT " 1:Yes 2:No"
```

TR.FL.01:

```
WAIT 0, &h01
K$=INKEY$
IF K$="1" THEN
    CLS
    PRINT MENU$
    LOCATE 1,3
    PRINT " 00000/"
    PRINT RIGHT$("00000"+MID$(STR$(RD.CNT%),2),5)
    LOCATE 3,3
    OPEN "COM:19200,N,8,1" AS #3
    ON ERROR GOTO COMERR
    XFILE DT.NAMES, "SPM"
    ON ERROR GOTO 0
    CLOSE
    BEEP 10
    TIMEA=10
    KILL DT.NAMES
    WAIT 0, &h10
    GOSUB CLR.KEY
    RETURN
ENDIF
IF (K$="2" OR K$=F1$) THEN
    RETURN
ELSE
    TR.FL.01
ENDIF
```

OPEN.DT:

```
OPEN DT.NAMES AS #2 RECORD 10000
FIELD #2, DT.LEN% AS FLD1$, 8 AS FLD2$
RETURN
```

CAL.RCD:

```
GOSUB OPEN.DT
RD.CNT%=LOF(2)
CLOSE #2
RETURN
```

DT.SAVE:

```
ERCODE%=0
DT.CNT%=0
ON ERROR GOTO SAVERR
GOSUB OPEN.DT
RD.PT%=SEARCH(#2, FLD1$, ITEM$)
IF RD.PT%=0 THEN
    FLD1$=ITEM$
    PRINT FLD1$
```

TESIS EL CODIGO DE BARRAS Y LA AUTOMATIZACION

```

LOCATE 1,4 : PRINT " ";
BEEP 1
LOCATE 7,3 : PRINT " ";
LOCATE 1,3 : PRINT LEFTS(">Cant:",12);
INPUT QTY$
FLD2$=QTY$
FLD2$=STR$(VAL(QTY$))
RD.CNT%=RD.CNT%+1
PUT #2, RD.CNT%

ENDIF

IF RD.PT%>0 THEN
GET #2, RD.PT%
FLD1$=ITEM$
PRINT FLD1$
LOCATE 1,4 : PRINT " ";
LOCATE 1,4 : PRINT "A";FLD2$;
BEEP 1
LOCATE 7,3 : PRINT " ";
LOCATE 1,3 : PRINT LEFTS(">Cant:",12);
INPUT QTY$
FLD3$=QTY$
FLD2$=STR$(VAL(FLD2$)+VAL(FLD3$))
PUT #2, RD.PT%

ENDIF

BEEP 1
GOSUB DSP.DAT
GOSUB DT.TSMT

DT.SV.01:
CLOSE #2
ON ERROR GOTO 0
ITEM$=""
RETURN

DSP.DAT:
CURSOR OFF
LOCATE 2,2
PRINT LEFTS$(ITEM$+SPACE$, DT.LEN%);
LOCATE 1,3

PRINT LEFTS(">Cant:"+QTY$+SPACE$, 12);
LOCATE 12,4
PRINT RIGHTS$("00000"+MID$(STR$(RD.CNT%), 2), 5);
LOCATE DT.CNT%+2, 2
CURSOR ON
RETURN

CLR.KEY:
K$=INKEY$
IF K$<>" " THEN
CLR.KEY

```

```

ELSE
    RETURN
ENDIF

SAVERR:
    BEEP 1, 1, 5
    CURSOR OFF
    CLS
    PRINT MENU$
    LOCATE 1,3
    IF (ERR=7 OR ERR=59 OR ERR=62) THEN
        PRINT "Memory full !!"
    ELSE
        PRINT "ERR = "; HEX$(ERR)
    ENDIF
    ERCODE%=ERR
    TIMEA=20
    WAIT 0, &h10
    GOSUB CLR.KEY
    RESUME DT.SV.01

DT.TSMT:
    OPEN "COM:" AS #4
    TM$=CHR$(13)+CHR$(10)
    ON ERROR GOTO COMERR
    PRINT #4, ITEM$
    CLOSE #4
    RETURN

COMERR:
    BEEP 1,1,5
    CLOSE
    LOCATE 1,2
    PRINT " Error ! "
    PRINT " Retry ? "
    PRINT " 1:Yes 2:No";
    RESUME TR.FL.01

```

2.- PROBLEMA :

Control de nómina.

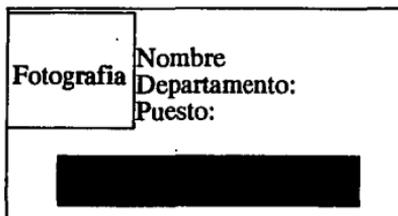
El control de nómina es un problema muy frecuente en cualquier industria como controlar el pago de los empleados.

SOLUCION:

Existen diferentes soluciones a este problema, daremos la mas sencilla la cual radica en:

Una primera etapa sería credencializar a todos nuestros trabajadores con una credencial que tuviera los siguientes datos: nombre, departamento, puesto y los datos generales de la compañía, así como la inserción de un código de barras especial para cada trabajador, esto es dar una clave en código de barra especial a cada trabajador, se sugiere que este contenga la primera letra de cada uno de sus nombres con una terminación numérica y esta será la clave única del trabajador; por ejemplo Juan Marty Fernández tendrá el código de barras siguiente "LMP01", en caso de que se repitieran las primeras letras con la combinación de números posteriores a estas podemos tener una gran cantidad de combinaciones distintas, se recomienda que la credencial tenga la forma que se ve en la figura 6.2; por otra parte, para evitar que los trabajadores dupliquen las etiquetas de código de barras, se recubrirá este con una cinta negra, la cual evitará que el código de barras sea duplicado por los mismos trabajadores y tener fraudes en la información.

El equipo que necesitaremos para leer el código de barras en este caso será un equipo especial denominado equipo lector infrarojo, este equipo cuenta con la capacidad de leer códigos de barras ocultos, este será del tipo ranura especial para leer códigos en documentos como credenciales o recibos, este se conectará a una terminal o una PC, la cual, al momento de leer un código, inmediatamente se encargue de adicionar a este la fecha y hora de entrada, esta se dará por medio de programación y el esquema de esto será como el que esta en la figura 6.3



**Credencial para cada empleado**

**figura 6.2**

El código recomendado para ser utilizado en este caso es el código 39, el cual cuenta con la capacidad de poder codificar caracteres numéricos y alfanuméricos.

**CONCLUSION:**

Con la inclusión del código de barras en la nómina podemos aumentar tanto la eficiencia en la captura de los datos, como la automatización de la entrada de los empleados sin tener que checar y evitar así posibles errores en este, también se logra con esto

## **TESIS EL CODIGO DE BARRAS Y LA AUTOMATIZACION**

que nuestros pagos de nómina estén al día debido a que la captura de información es automática y los procesos de pago se automatizan mediante programación utilizando los datos que se obtienen del código de barras.

### **3.- PROBLEMA :**

Automatización de un punto de venta:

El punto de venta se define como aquel sitio donde se realiza una transacción de un bien por un servicio y que este procedimiento se efectúe a través de una computadora, la cual sirva como elemento de enlace entre el comprador y el vendedor.

En tiendas de autoservicio como Aurrerá, es donde se ve esta necesidad ya que, debido al flujo de clientela y la gran cantidad de datos que se necesitan procesar, resulta casi indispensable implementar sistemas mucho más rápidos que los tradicionales y mucho más efectivos para evitar errores en la captura.

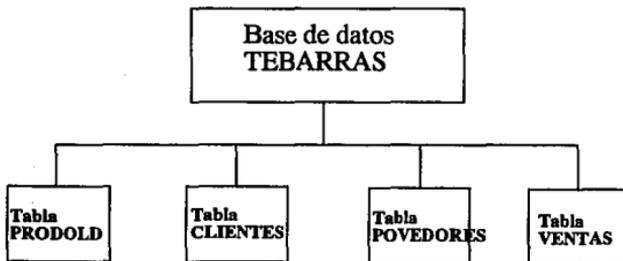
### **SOLUCION:**

La solución de este problema se explica ampliamente en el siguiente punto donde se automatiza completamente la captura de información a través del código de barras y emitiendo un documento o factura al final del procedimiento esto es necesario en cualquier punto de venta.

**AUTOMATIZACION DE UN PUNTO DE VENTA**, para poder solucionar este problema necesitamos de un sistema que maneje todos los elementos necesarios en un punto de venta como lo son altas, bajas, cambios, consultas, reports y en el punto de venta todos estos se necesitan englobar en un solo sistema para poder tener un buen control de todas las variables que ocurren en este tipo de sistemas.

**Planteamiento de la base de datos**, La base de datos contará con la siguiente

estructura y la llamaremos "TEBARRAS", con las siguientes tablas "PRODOLD" en esta tabla se almacenan todos los productos de nuestro sistema, "CLIENTES" en esta se almacenan todos los datos de nuestros clientes, "PROVEEDORES" en esta se almacenan todos los datos de nuestros proveedores, "VENTAS" en esta llevamos el control de todas la ventas efectuadas en nuestro punto de venta; estas se especifican en la figura 6.4.



**DIAGRAMA DE LA BASE DE DATOS**

**FIGURA 6.4**

Cada tabla consta de sus propios campos y cada uno de ellos cuenta con las

siguientes características.

**TABLA DE PRODUCTOS "PRODOLD"**

NOMBRE CAMPO	TIPO	LONGITUD
CLAVE	CARACTER	20
NOMBRE	CARACTER	20
DESCRIPCION	CARACTER	30
CODIGO DE BARRAS	CARACTER	20
MAXIMO	ENTERO	
MINIMO	ENTERO	
PRECIO 1	DECIMAL	16,2
PRECIO 2	DECIMAL	16,2
PRECIO 3	DECIMAL	16,2
PROVEEDOR 1	CARACTER	20
PROVEEDOR 2	CARACTER	20
STOCK	ENTERO	

**TABLA DE CLIENTES "CLIENTES"**

NOMBRE CAMPO	TIPO	LONGITUD
CLAVE	CARACTER	15
NOMBRE	CARACTER	40
DIRECCION	CARACTER	40
TELEFONO	CARACTER	20
RFC	CARACTER	20
CODIGO POSTAL	ENTERO	
CD Y ESTADO	CARACTER	30
CONTACTO	CARACTER	2

**TABLA DE PROVEEDORES "PROVEEDORES"**

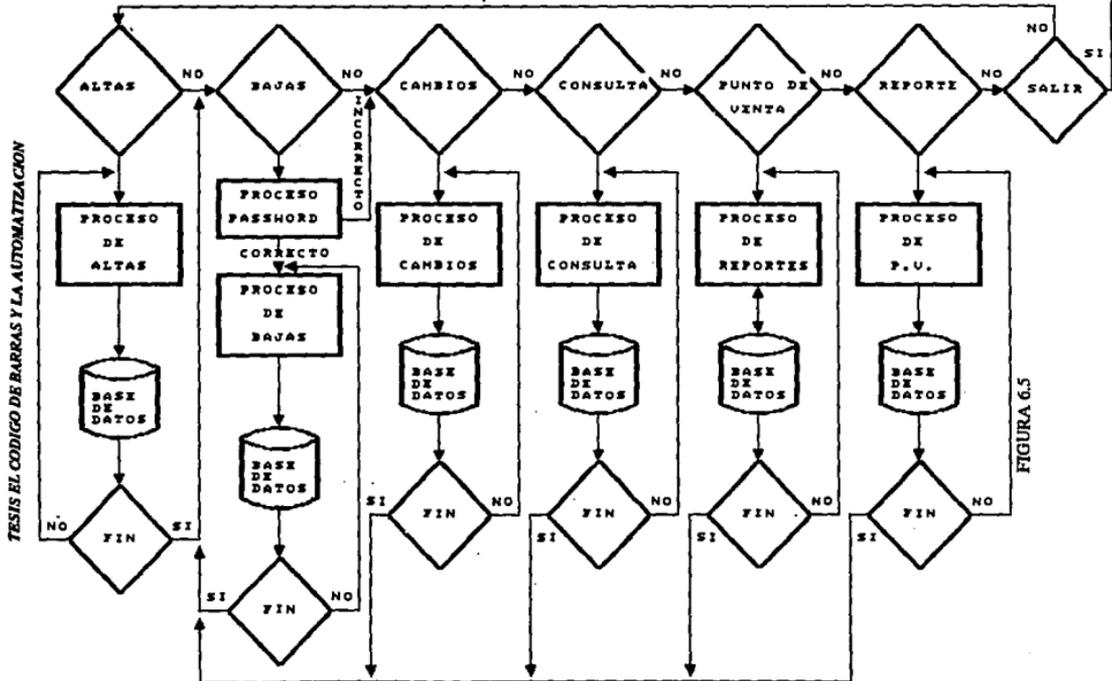
NOMBRE CAMPO	TIPO	LONGITUD
CLAVE	CARACTER	20
NOMBRE	CARACTER	20
DIRECCION	CARACTER	30
TELEFONO	CARACTER	20
RFC	CARACTER	20
CD Y ESTADO	CARACTER	30
CONTACTO	CARACTER	2

**TABLA DE VENTAS "VENTAS"**

<b>NOMBRE CAMPO</b>	<b>TIPO</b>	<b>LONGITUD</b>
CLAVE DEL PRODUCTO	CARACTER	20
NUMERO DE UNIDADES	CARACTER	20
FECHA	DATE	
TOTAL DE LA VENTA	DECIMAL	16,2
CLAVE DEL CLIENTE	CARACTER	20
PRECIO UNITARIO	DECIMAL	8,2
FACTURA	ENTERO	5

El diagrama de flujo del problema es el que se muestra enseguida en la figura 6.5 este representa al programa principal, la figura 6.6 representa al proceso de altas del programa principal, la figura 6.7 representa el proceso de password y bajas del programa principal, la figura 6.8 representa el proceso de cambios del programa principal, la figura 6.9 representa el proceso de consultas del programa principal, la figura 6.10 representa el proceso de reportes del programa principal, la figura 6.11 representa el proceso del punto de venta del programa principal.

PROGRAMA PRINCIPAL



TESIS EL CODIGO DE BARRAS Y LA AUTOMATIZACION

FIGURA 6.5

PROCESO DE ALTAS

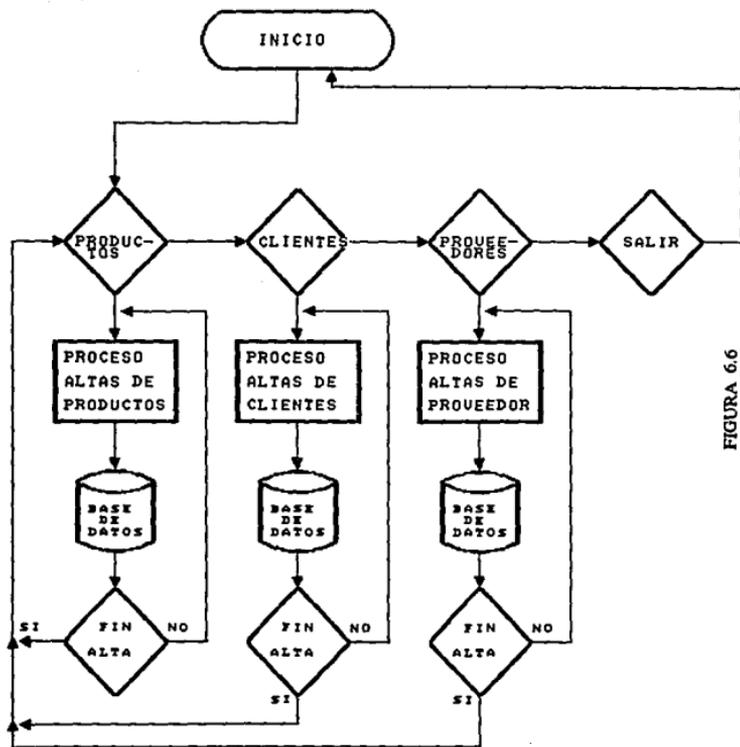


FIGURA 6.6

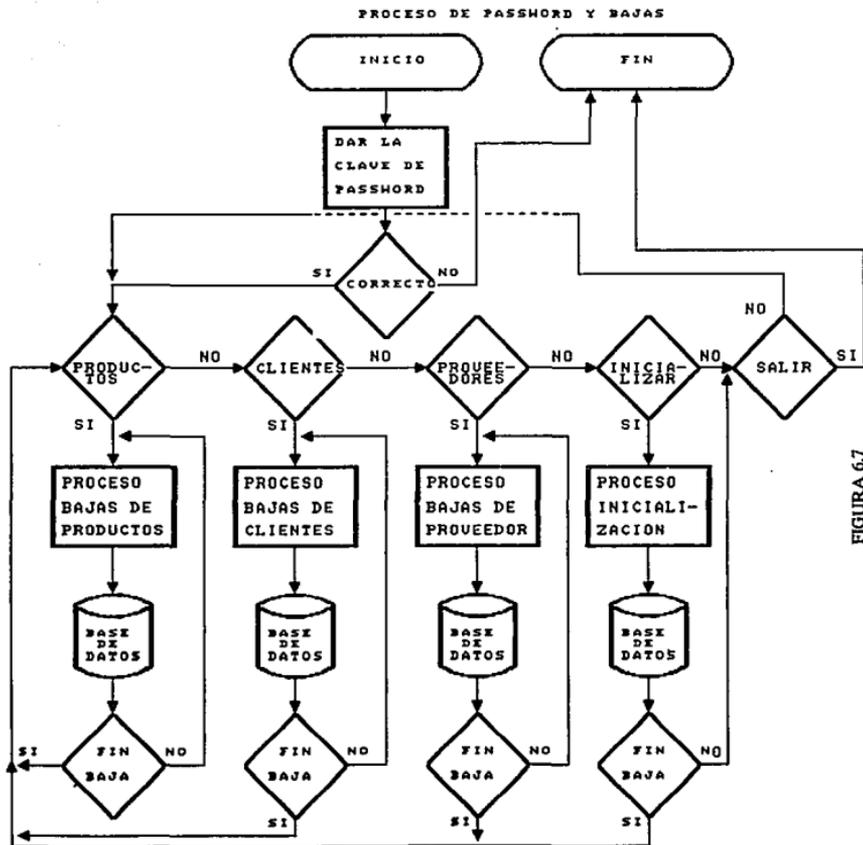


FIGURA 6.7

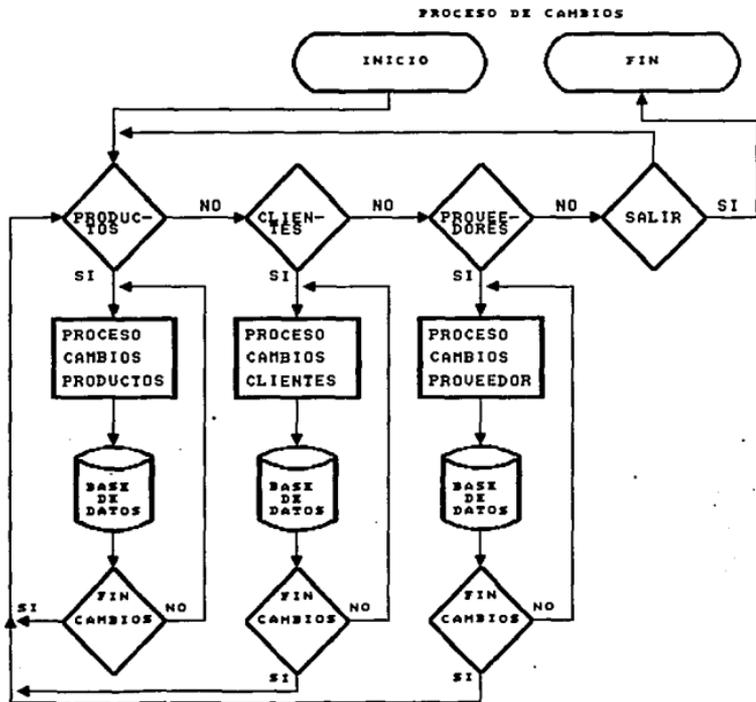


FIGURA 6.8

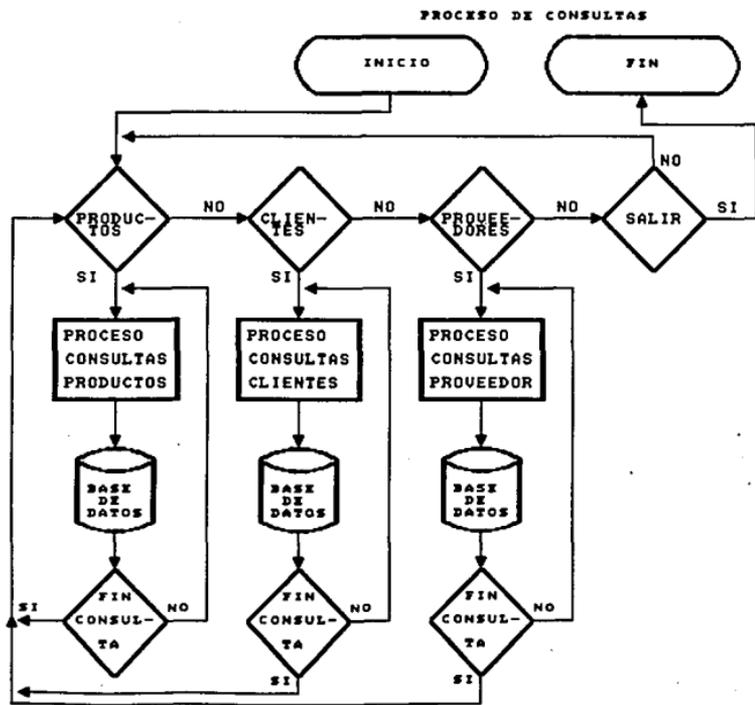


FIGURA 6.9

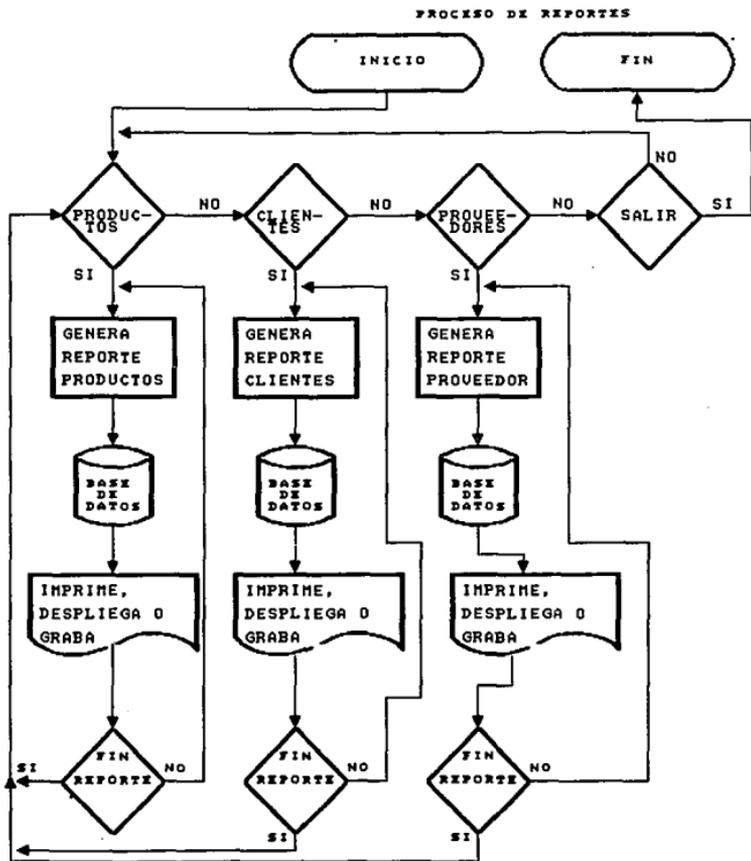


FIGURA 6.10

PROCESO PUNTO DE VENTA

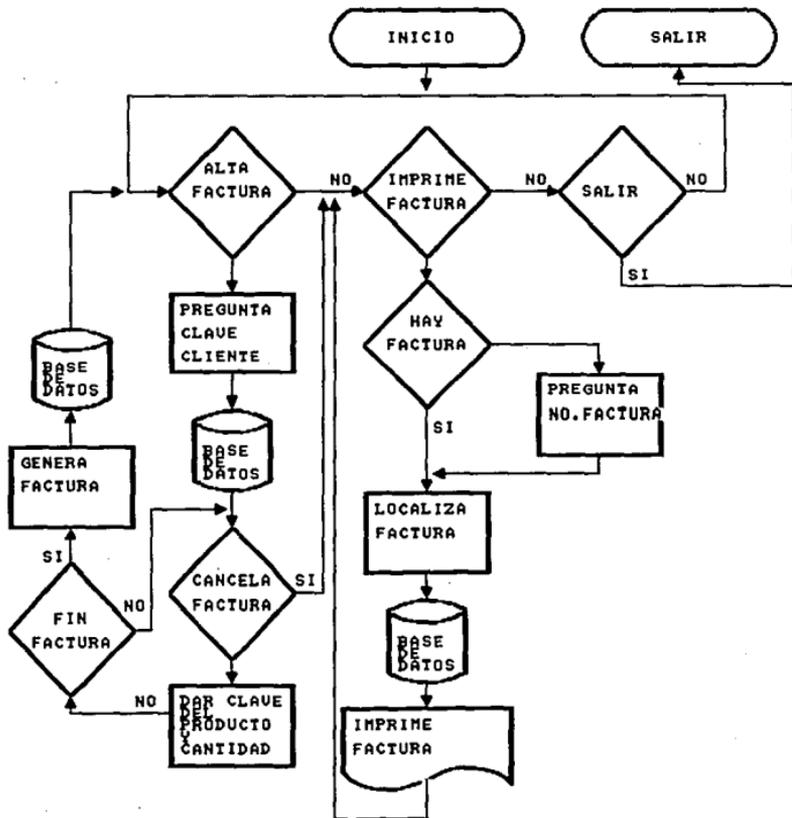


FIGURA 6.11

**SELECCION DEL LENGUAJE DE PROGRAMACION**, el lenguaje utilizado en este programa es INFORMIX este fué seleccionado por ser un lenguaje especialmente diseñado para el manejo de bases de datos y por ser un lenguaje de cuarta generación, las bases de datos y las tablas fueron generadas con SQL este lenguaje también es de cuarta generación y es especial para manejar bases de datos también.

### PROGRAMACION

Este programa es la implementación de los diagramas de flujo mencionado anteriormente.

database tebarras

globals

```
define
  con_cli record like clientes.*,
  con_pro record like prodold.*,
  con_prov record like provedores.*,
  altas_c record like clientes.*,
  altas_p record like prodold.*,
  altas_pro record like provedores.*,
  resp char(1),
  paso char(4),
  claclient like clientes.clave,
  z_flag smallint,
  vfactura like ventas.factura,
  iva float,tot float,sub float,variable float,
  a_factura array[50] of record
  aclave like prodold.clave, adescpcion like prodold.descripcion,
  apv like prodold.precio1,
  unidades smallint,
  aunjt float
end record,
vfecha date,
clavepro char(20),
clavecli char(20),
claveprov char(20),
numero integer,
r integer,
vban smallint,
```

```

    ayuda char(1)
end globals

main
defer interrupt
let paso = "TORO"
call menu_principal()
clear screen
end main

function menu_principal()
open window menu1 at 2,2 with 3 rows,76 columns attributes (yellow)
menu "menu principal"
command "ALTAS" "ALTAS PRODUCTOS, CLIENTES Y PROVEEDORES"
    call altas()
command "BAJAS" "BAJAS PRODUCTOS, CLIENTES Y PROVEEDORES"
    call password()
    if r = true then
        call bajas()
    end if
command key (M) "CAMBIOS" "CAMBIOS PRODUCTOS, CLIENTES Y PROVEEDORES"
    call cambios()
command "CONSULTAS" "CONSULTAS PRODUCTOS, CLIENTES Y PROVEEDORES"
    call consulta()
command "PUNTO DE VENTA" "ENTRAR A PUNTO DE VENTA"
    call punto_de_venta()
command "REPÓRTEES" "ENTRAR A SECCION DE REPORTEES"
    call reporte_global()
command "SALIR" "SALIR DE PROGRAMA PRICIPAL"
exit menu
end menu
close window menu1
end function

function altas()
menu "menu de altas"
command "PRODUCTOS" "PROCESO ALTAS PRODUCTOS"
    call altas_productos("S")
command "CLIENTES" "PROCESO ALTAS CLIENTES"
    call altas_clientes("S")
command key (R) "PROVEEDORES" "PROCESO ALTAS PROVEEDORES"
    call altas_proveedores("S")
command "SALIR" "REGRESAR A MENU PRICIPAL"
exit menu
end menu
end function

function bajas()
menu "menu de bajas"
command "PRODUCTOS" "PROCESO BAJAS PRODUCTOS"
    call bajas_productos()
command "CLIENTES" "PROCESO BAJAS CLIENTES"
    call bajas_clientes()
command key (R) "PROVEEDORES" "PROCESO BAJAS PROVEEDORES"

```

```

    call bajas_proveedores()
    command "INIALIZAR" "INICIALIZA TODA LA BASE DE DATOS"
    call inicializar()
    command "sALIR" "REGRESAR A MENU PRINCIPAL"
    exit menu
end menu
end function

function cambios()
    menu "menu de cambios"
    command "pRODUCTOS" "PROCESO CAMBIOS PRODUCTOS"
    call cambios_productos()
    command "cLIENTES" "PROCESO CAMBIOS CLIENTES"
    call cambios_clientes()
    command key (R) "PrOVEEDORES" "PROCESO CAMBIOS PROVEEDORES"
    call cambios_proveedores()
    command "sALIR" "REGRESAR A MENU PRINCIPAL"
    exit menu
end menu
end function

function consulta()
    menu "menu de consultas"
    command "pRODUCTOS" "PROCESO CONSULTAS PRODUCTOS"
    call consultas_productos()
    command "cLIENTES" "PROCESO CONSULTAS CLIENTES"
    call consultas_clientes()
    command key (R) "PrOVEEDORES" "PROCESO CONSULTAS PROVEEDORES"
    call consultas_proveedores()
    command "sALIR" "REGRESAR A MENU PRINCIPAL"
    exit menu
end menu
end function

function punto_de_venta()
    menu "menu de punto de venta"
    command "pV" "ENTRAR A PUNTO DE VENTA"
    call pv()
    command "sALIR" "REGRESAR A MENU PRINCIPAL"
    exit menu
end menu
end function .

function reporte_global()
    menu "menu de reportes"
    command "cLIENTES" "REPORTE DE VENTAS A CLIENTES "
    call reporte_clientes()
    command "pRODUCTOS" "REPORTE DE PRODUCTOS "
    call reporte_productos()
    command key (R) "PrOVEEDORES" "REPORTE DE PROVEEDORES"
    call reporte_proveedores()
    command "sALIR" "REGRESAR A MENU PRINCIPAL"
    exit menu
end menu

```

```

end function

function reporte_clientes()
menu "Reporte Clientes"
    command "aRCHIVO" "ESCRIBE EL REPORTE EN UN ARCHIVO
LLAMADO'clientes.lis"
    start report reporte to "clientes.lis"
    call reportes()
finish report reporte
command "pANTALLA" "DESPLIEGA EL REPORTE EN PANTALLA"
    start report reporte
    call reportes()
finish report reporte
command "iMPRESORA" "IMPRIME EL REPORTE, EN PUERTO PARARELO "
    start report reporte to pipe "lpt1"
    call reportes()
finish report reporte
command "sALIR"
exit menu
end menu
end function

function reportes()
declare wcursor cursor for select * from clientes
foreach wcursor into altas_c.*
    output to report reporte(altas_c.*)
end foreach
end function

report reporte(vreg)
define vreg record like clientes.*
format
on every row
print "NOMBRE      :", vreg.nombre
print "CLAVE       :", vreg.clave
print "DIRECCION    :", vreg.direccion
print "TELEFONO     :", vreg.telefono
print "RFC          :", vreg.rfc
print "CODIGO POSTAL:", vreg.codigo_postal
print "CD. Y ESTADO :", vreg.cd_y_estado
print "CONTACTO    :", vreg.contacto
print "
print "-----"
end report

function reporte_productos()
menu "Reporte Productos"
    command "aRCHIVO" "ESCRIBE EL REPORTE EN UN ARCHIVO
LLAMADO'producto.lis"
    start report reporte1 to "producto.lis"
    call reportes1()
finish report reporte1
command "pANTALLA" "DESPLIEGA EL REPORTE EN PANTALLA"
    start report reporte1

```

```

    call reportes1()
finish report reporte1
command "IMPRESORA" "IMPRIME EL REPORTE, EN PUERTO PARARELO "
start report reporte1 to pipe "lpt1"
call reportes1()
finish report reporte1
command "sALIR"
exit menu
end menu
end function

function reportes1()
declare xcursor cursor for select * from prodold
foreach xcursor into altas_p.*
    output to report reporte1(altas_p.*)
end foreach
end function

report reporte1(vreg)
define vreg record like prodold.*
format
on every row
print "NOMBRE      :", vreg.nombre
print "CLAVE       :", vreg.clave
print "CODIGO DE BARRAS :", vreg.codigo_de_barras
print "DESCRIPCION  :", vreg.descripcion
print "MAXIMO      :", vreg.maximo
print "MINIMO      :", vreg.minimo
print "PRECIO1     :", vreg.precio1
print "PRECIO2     :", vreg.precio2
print "PRECIO3     :", vreg.precio3
print "DESCUENTO MAXIMO :", vreg.descuento_maximo
print "PROVEDOR1    :", vreg.proveedor1
print "PROVEDOR2    :", vreg.proveedor2
print "STOCK       :", vreg.stock
print "
"
print "-----"
end report

function reporte_provedores()
menu "Reporte Provedores"
command "aARCHIVO" "ESCRIBE EL REPORTE EN UN ARCHIVO
LLAMADO proveedor.lis"
start report reporte2 to "proveedor.lis"
call reportes2()
finish report reporte2
command "pPANTALLA" "DESPLIEGA EL REPORTE EN PANTALLA"
start report reporte2
call reportes2()
finish report reporte2
command "IMPRESORA" "IMPRIME EL REPORTE, EN PUERTO PARARELO "
start report reporte2 to pipe "lpt1"
call reportes2()
finish report reporte2

```

```

command "sALIR"
exit menu
end menu
end function

function reportes2()
declare yeursor cursor for select * from proveedores
foreach yeursor into altas_pro.*
output to report reporte2(altas_pro.*)
end foreach
end function

report reporte2(vreg)
define vreg record like proveedores.*
format
on every row
print "NOMBRE      :", vreg.nombre
print "CLAVE       :", vreg.clave
print "DIRECCION    :", vreg.direccion
print "CD. Y ESTADO  :", vreg.cd_y_estado
print "RFC          :", vreg.rfc
print "CONTACTO     :", vreg.contacto
print "TELEFONO      :", vreg.telefono
print "              "
print "-----"
end report

function altas_productos(ban)
define ban char(1)
if ban = "S" then
initialize altas_p.* to null
end if
open window prod at 2,3 with form "auxilia" attributes (border,yellow)
input altas_p.clave,
      altas_p.codigo_de_barras,
      altas_p.nombre,
      altas_p.descripcion,
      altas_p.maximo,
      altas_p.minimo,
      altas_p.precio1,
      altas_p.precio2,
      altas_p.precio3,
      altas_p.descuento_maximo,
      altas_p.proveedor1,
      altas_p.proveedor2,
      altas_p.stock
without defaults
from clave,
      codigo_de_barras,
      nombre,
      descripcion,
      maximo,
      minimo,
      precio1,

```

```

    precio2,
    precio3,
    descuento_maximo,
    proveedor1,
    proveedor2,
    stock
after field prodold.clave
if look_clave2(altas_p.clave) then
error "Error dato duplicado"
next field clave
end if
after field prodold.codigo_de_barras
if look_clave3(altas_p.codigo_de_barras) then
error "Error dato duplicado"
next field codigo_de_barras
end if
end input
if int_flag = true then
let int_flag = false
display " Regreso a menú Altas" at 2,2
sleep 2
close window prod
else
open window presp at 23,44 with 2 rows, 35 columns
attributes(border,green)
prompt "Son correctos los datos (S/N) : ? " for resp
close window presp
let resp=upshift(resp)
if resp = "S" then
message "La captura se ha realizado"
insert into prodold values (altas_p.*)
clear form
close window prod
call altas_productos("S")
else
close window prod
call altas_productos("N")
end if
end if
end function

function altas_clientes(ban)
define ban char(1)
if ban = "S" then
initialize altas_c.* to null
end if
open window vprod at 2,3 with form "alt_for" attributes (border,yellow)
input altas_c.nombre,
altas_c.clave,
altas_c.direccion,
altas_c.telefono,
altas_c.rfc,
altas_c.codigo_postal,

```

```

    altas_c.cd_y_estado,
    altas_c.contacto
without defaults
from nombre,
    clave,
    direccion,
    telefono,
    rfc,
    codigo_postal,
    cd_y_estado,
    contacto
after field clientes.clave
if look_clave(altas_c.clave) then
    error "Error dato duplicado"
    next field clave
end if
end input
if int_flag = true then
    let int_flag = false
    display " Regreso menú Altas" at 2,2
    sleep 2
    close window vprod
else
    open window vresp at 23,44 with 2 rows, 35 columns
    attributes(border,green)
    prompt "Son correctos los datos (S/N) : ? " for resp
    close window vresp
    let resp=upshift(resp)
    if resp = "S" then
        message "La captura se ha realizado"
        insert into clientes values (altas_c.*)
        clear form
        close window vprod
        call altas_clientes("S")
    else
        close window vprod
        call altas_clientes("N")
    end if
end if
end function

function altas_provedores(ban)
define ban char(1)
if ban = "S" then
    initialize altas_pro.* to null
end if
open window prove at 2,3 with form "alt_prov" attributes (border,yellow)
input altas_pro.clave,
    altas_pro.nombre,
    altas_pro.direccion,
    altas_pro.cd_y_estado,
    altas_pro.rfc,
    altas_pro.contacto,
    altas_pro.telefono

```

```

without defaults
from clave,
  nombre,
  direccion,
  cd_y_estado,
  rfc,
  contacto,
  telefono
after field clientes.clave
if look_clave1(altas_pro.clave) then
error "Error dato duplicado"
next field clave
end if
end input
if int_flag = true then
let int_flag = false
display " Regreso menú Altas" at 2,2
sleep 2
close window prove
else
open window ppresp at 23,44 with 2 rows, 35 columns
attributes(border,green)
prompt "Son correctos los datos (S/N) : ? " for resp
close window ppresp
let resp=upshift(resp)
if resp = "S" then
message "la captura se ha realizado"
insert into proveedores values (altas_pro.*)
clear form
close window prove
call altas_provedores("S")
else
close window prove
call altas_provedores("N")
end if
end if
end function

function look_clave (vclave)
define
vclave like clientes.clave,
valcla smallint
select count(*) into valcla from clientes
where clave = vclave
if valcla >= 1 then
return true
else
return false
end if
end function

function look_clave1 (vclave)
define

```

```

vclave like proveedores.clave,
valcla smallint
select count(*) into valcla from proveedores
where clave = vclave
if valcla >= 1 then
    return true
else
    return false
end if
end function

function look_clave2 (vclave)
define
vclave like prodold.clave,
valcla smallint
select count(*) into valcla from prodold
where clave = vclave
if valcla >= 1 then
    return true
else
    return false
end if
end function

function look_clave3 (vclave)
define
vclave like prodold.codigo_de_barras,
valcla smallint
select count(*) into valcla from prodold
where codigo_de_barras = vclave
if valcla >= 1 then
    return true
else
    return false
end if
end function

function consultas_productos()
open window prod at 2,3 with form "auxicp" attributes (border,yellow)
open window clave_ven at 5,10 with 2 rows, 60 columns
attributes(border,green)
prompt "Dar la clave del Producto a consultar: ? " for clavepro
close window clave_ven
initialize con_pro.* to null
let clavepro=upshift(clavepro)
select * into con_pro.*
from prodold where prodold.clave =clavepro or prodold.codigo_de_barras = clavepro
if status = notfound then
    error "clave incorrecta"
    sleep 2
end if
display
con_pro.clave,
con_pro.codigo_de_barras,

```

```

con_pro.nombre,
con_pro.descripcion,
con_pro.maximo,
con_pro.minimo,
con_pro.precio1,
con_pro.precio2,
con_pro.precio3,
con_pro.descuento_maximo,
con_pro.proveedor1,
con_pro.proveedor2,
con_pro.stock
to
prodold.clave,
prodold.codigo_de_barras,
prodold.nombre,
prodold.descripcion,
prodold.maximo,
prodold.minimo,
prodold.precio1,
prodold.precio2,
prodold.precio3,
prodold.descuento_maximo,
prodold.proveedor1,
prodold.proveedor2,
prodold.stock
sleep 2
open window ppresp at 23,44 with 2 rows, 35 columns
attributes(border,green)
prompt "deseas otra consulta (S/N) : ? " for resp
close window ppresp
let resp=upshift(resp)
if resp = "S" then
    close window prod
    call consultas_productos()
else
    close window prod
end if
end function

function consultas_clientes()
open window cli at 2,3 with form "alt_fore" attributes (border,yellow)
open window clave_ven at 5,10 with 2 rows, 50 columns
attributes(border,green)
prompt "Dar la clave del Cliente a consultar: ? " for clavecli
close window clave_ven
initialize con_cli.* to null
let clavecli=upshift(clavecli)
select * into con_cli.*
from clientes where clientes.clave = clavecli
if status = notfound then
    error "clave incorrecta"
    sleep 2
end if
display

```

```

con_cli.nombre,
con_cli.clave,
con_cli.direccion,
con_cli.telefono,
con_cli.rfc,
con_cli.codigo_postal,
con_cli.cd_y_estado,
con_cli.contacto
to
clientes.nombre,
clientes.clave,
clientes.direccion,
clientes.telefono,
clientes.rfc,
clientes.codigo_postal,
clientes.cd_y_estado,
clientes.contacto
sleep 2
open window ppresp at 23,44 with 2 rows, 35 columns
attributes(border,green)
prompt "deseas otra consulta (S/N) : ? " for resp
close window ppresp
let resp=upshift(resp)
if resp = "S" then
close window clie
call consultas_clientes()
else
close window clie
end if
end function

function consultas_provedores()
open window prov at 2,3 with form "alt_prep" attributes (border,yellow)
open window clave_ven at 5,10 with 2 rows, 50 columns
attributes(border,green)
prompt "Dar la clave del Proveedor a consultar: ? " for claveprov
close window clave_ven
initialize con_prov.* to null
let claveprov=upshift(claveprov)
select * into con_prov.*
from provedores where provedores.clave = claveprov
if status = notfound then
error "clave incorrecta"
sleep 2
end if
display
con_prov.clave,
con_prov.nombre,
con_prov.direccion,
con_prov.cd_y_estado,
con_prov.rfc,
con_prov.contacto,
con_prov.telefono
to

```

```

    proveedores.clave,
    proveedores.nombre,
    proveedores.direccion,
    proveedores.cd_y_estado,
    proveedores.rfc,
    proveedores.contacto,
    proveedores.telefono
sleep 2
open window ppresp at 23,44 with 2 rows, 35 columns
attributes(border,green)
prompt "deseas otra consulta (S/N) : ? " for resp
close window ppresp
let resp=upshift(resp)
if resp = "S" then
    close window prov
    call consultas_provedores()
else
    close window prov
end if
end function

function bajas_productos()
open window prod at 2,3 with form "auxilia1" attributes (border,yellow)
open window clave_ven at 5,10 with 2 rows, 60 columns
attributes(border,green)
prompt "Dar la clave del Producto a Borrar: ? " for clavepro
close window clave_ven
let clavepro=upshift(clavepro)
select * into con_pro.*
from prodold where prodold.clave =clavepro or prodold.codigo_de_barras = clavepro
if status = notfound then
    error "clave incorrecta"
sleep 2
open window ppresp at 23,44 with 2 rows, 35 columns
attributes(border,green)
prompt "Continuar con bajas(S/N): ? " for resp
close window ppresp
clear form
close window prod
let resp=upshift(resp)
if resp = "S" then
    call bajas_productos()
else
end if
else
display
con_pro.clave,
con_pro.codigo_de_barras,
con_pro.nombre,
con_pro.descripcion,
con_pro.maximo,
con_pro.minimo,
con_pro.precio1,
con_pro.precio2,

```

```

con_pro.precio3,
con_pro.descuento_maximo,
con_pro.proveedor1,
con_pro.proveedor2,
con_pro.stock

to
prodold.clave,
prodold.codigo_de_barras,
prodold.nombre,
prodold.descripcion,
prodold.maximo,
prodold.minimo,
prodold.precio1,
prodold.precio2,
prodold.precio3,
prodold.descuento_maximo,
prodold.proveedor1,
prodold.proveedor2,
prodold.stock

sleep 2
let resp = "N"
if status = notfound then
clear form
else
end if

open window ppresp at 23,44 with 2 rows, 35 columns
attributes(border,green)
prompt "Quiere dar de baja (S/N): ? " for resp
close window ppresp
let resp=upshift(resp)
let clavepro = con_pro.clave
if resp = "S" then
clear form
delete from prodold where prodold.clave = clavepro
error "Dando de baja "
sleep 2
let resp = "N"
open window pppresp at 23,44 with 2 rows, 35 columns
attributes(border,green)
prompt "Quiere baja de otro (S/N) : ? " for resp
close window pppresp
let resp=upshift(resp)
if resp = "S" then
close window prod
call bajas_productos()
else
open window sal at 23,44 with 2 rows, 25 columns
attributes(border,green)
prompt "Quiere salir (S/N) : ? " for resp
close window sal
let resp = upshift(resp)
if resp = "N" then
close window prod
call bajas_productos()

```

```

        else
            close window prod
        end if
    end if
else
    open window sal at 23,44 with 2 rows, 25 columns
    attributes(border,green)
    prompt "Quiere salir (S/N) : ? " for resp
    close window sal
    let resp = upshift(resp)
    if resp = "N" then
        clear form
        close window prod
        call bajas_productos()
    else
        #clear form#
        close window prod
    end if
end if
end if
end function

function bajas_clientes()
    open window prod at 2,3 with form "alt_for1" attributes (border,yellow)
    open window clave_ven at 5,10 with 2 rows, 60 columns
    attributes(border,green)
    prompt "Dar la clave del Cliente a Borrar: ? " for clavepro
    close window clave_ven
    let clavepro=upshift(clavepro)
    select * into con_cli.*
    from clientes where clientes.clave = clavepro
    if status = notfound then
        error "clave incorrecta"
        sleep 2
    open window ppresp at 23,44 with 2 rows, 35 columns
    attributes(border,green)
    prompt "Continuar con bajas(S/N): ? " for resp
    close window ppresp
    clear form
    close window prod
    let resp=upshift(resp)
    if resp ="S" then
        call bajas_clientes()
    else
    end if
else
display
    con_cli.nombre,
    con_cli.clave,
    con_cli.direccion,
    con_cli.telefono,
    con_cli.rfc,
    con_cli.codigo_postal,
    con_cli.cd_y_estado,

```

```

con_cli.contacto
to
  clientes.nombre,
  clientes.clave,
  clientes.direccion,
  clientes.telefono,
  clientes.rfc,
  clientes.codigo_postal,
  clientes.cd_y_estado,
  clientes.contacto
sleep 2
  let resp = "N"
  if status = notfound then
    clear form
  else
    end if
open window ppresp at 23,44 with 2 rows, 35 columns
  attributes(border,green)
  prompt "Quiere dar de baja (S/N): ? " for resp
close window ppresp
  let resp=upshift(resp)
  let clavecli = con_cli.clave
  if resp = "S" then
    clear form
    delete from clientes where clientes.clave = clavecli
    error "Procediendo con la baja "
    sleep 2
    let resp = "N"
    open window pppresp at 23,44 with 2 rows, 35 columns
    attributes(border,green)
    prompt "Quiere baja de otro (S/N) : ? " for resp
    close window pppresp
    let resp=upshift(resp)
    if resp = "S" then
      close window prod
      call bajas_clientes()
    else
      open window sal at 23,44 with 2 rows, 25 columns
      attributes(border,green)
      prompt "Quiere salir (S/N) : ? " for resp
      close window sal
      let resp = upshift(resp)
      if resp = "N" then
        close window prod
        call bajas_clientes()
      else
        close window prod
      end if
    end if
  else
    open window sal at 23,44 with 2 rows, 25 columns
    attributes(border,green)
    prompt "Quiere salir (S/N) : ? " for resp
    close window sal

```

```

    let resp = upshift(resp)
    if resp = "N" then
        clear form
        close window prod
        call bajas_clientes()
    else
        close window prod
    end if
end if
end if
end function

function bajas_provedores()
open window prod at 2,3 with form "altprov1" attributes (border,yellow)
open window clave_ven at 5,10 with 2 rows, 60 columns
attributes(border,green)
prompt "Dar la clave del Cliente a Borrar: ? " for claveprov
close window clave_ven
let claveprov=upshift(claveprov)
select * into con_prov,*
from provedores where provedores.clave = claveprov
if status = notfound then
    error "clave incorrecta"
    sleep 2
open window pppres at 23,44 with 2 rows, 35 columns
attributes(border,green)
prompt "Continuar con bajas(S/N): ? " for resp
close window pppres
clear form
close window prod
let resp=upshift(resp)
if resp = "S" then
    call bajas_provedores()
else
end if
else
display
con_prov.clave,
con_prov.nombre,
con_prov.direccion,
con_prov.cd_y_estado,
con_prov.rfc,
con_prov.contacto,
con_prov.telefono
to
provedores.clave,
provedores.nombre,
provedores.direccion,
provedores.cd_y_estado,
provedores.rfc,
provedores.contacto,
provedores.telefono
sleep 2
let resp = "N"

```

TESIS EL CODIGO DE BARRAS Y LA AUTOMATIZACION

```

if status = notfound then
  clear form
else
  end if
open window pprresp at 23,44 with 2 rows, 35 columns
attributes(border,green)
prompt "Quiere dar de baja (S/N): ? " for resp
close window pprresp
let resp=upshift(resp)
let clavecli = con_prov.clave
if resp = "S" then
  clear form
  delete from proveedores where proveedores.clave = clavecli
  error "Procediendo con la baja "
  let resp = "N"
  open window pppresp at 23,44 with 2 rows, 35 columns
  attributes(border,green)
  prompt "Quiere baja de otro (S/N) : ? " for resp
  close window pppresp
  let resp=upshift(resp)
  if resp = "S" then
    close window prod
    call bajas_provedores()
  else
    open window sal at 23,44 with 2 rows, 25 columns
    attributes(border,green)
    prompt "Quiere salir (S/N) : ? " for resp
    close window sal
    let resp = upshift(resp)
    if resp = "N" then
      close window prod
      call bajas_provedores()
    else
      close window prod
    end if
  end if
else
  open window sal at 23,44 with 2 rows, 25 columns
  attributes(border,green)
  prompt "Quiere salir (S/N) : ? " for resp
  close window sal
  let resp = upshift(resp)
  if resp = "N" then
    clear form
    close window prod
    call bajas_provedores()
  else
    close window prod
  end if
end if
end if
end function

function password()

```

```

define passwd char(4)
define i,j integer
define c char(1)
options
  prompt line 2
let r=true
if r and paso is not null then
open window wclave at 11,11 with 2 rows,58 columns attribute (border,yellow)
display "Teclee su clave de acceso: " at 1,1
for i=1 to 4
  prompt "" for char c
  let c=upshift(c)
  if c is null then
    exit for
  end if
  let j = i+28
  display "#" at 1,j attribute(blink)
  let passwd[i,i] = c
end for
if passwd <> paso or passwd is null then
  display "clave incorrecta" at 2,1 attribute (reverse,blink)
  sleep 1
  let r = false
end if
close window wclave
end if
end function

function inicializar()
call password()
if r= true then
open window clave_ven at 5,5 with 2 rows, 60 columns
attributes(border,green)
prompt "Quiere inicilizar todos los datos (S/N) : ? " for resp
close window clave_ven
let resp = upshift(resp)
if resp = "S" then
  delete from clientes
  delete from prodold
  delete from proveedores
end if
end if
end function

function cambios_productos()
open window prod at 2,3 with form "auxicp1" attributes (border,yellow)
open window clave_ven at 5,10 with 2 rows, 60 columns
attributes(border,green)
prompt "Dar la clave del Producto a cambiar: ? " for clavepro
close window clave_ven
let clavepro=upshift(clavepro)
initialize altas_p.* to null
select * into altas_p.*
from prodold where prodold.clave =clavepro or prodold.codigo_de_barras = clavepro

```

```

if status = notfound then
error "clave incorrecta"
sleep 2
clear form
open window presp at 10,20 with 2 rows, 35 columns
attributes (border,green)
prompt " Desea dar otro cambio (S/N): ? " for resp
close window presp
let resp= upshift(resp)
if resp = "S" then
close window prod
call cambios_productos()
else
clear form
close window prod
call cambios()
end if
else
display "relice sus cambios" at 2,1
sleep 1
input alias_p.clave,
      alias_p.codigo_de_barras,
      alias_p.nombre,
      alias_p.descripcion,
      alias_p.maximo,
      alias_p.minimo,
      alias_p.precio1,
      alias_p.precio2,
      alias_p.precio3,
      alias_p.descuento_maximo,
      alias_p.proveedor1,
      alias_p.proveedor2,
      alias_p.stock
without defaults
from clave,
      codigo_de_barras,
      nombre,
      descripcion,
      maximo,
      minimo,
      precio1,
      precio2,
      precio3,
      descuento_maximo,
      proveedor1,
      proveedor2,
      stock
before field clave
next field codigo_de_barras
end input
if int_flag = true then
let int_flag = false
display " funcion cambios abortada" at 2,2
sleep 2

```

```

close window prod
call cambios()
else
open window presp at 23,44 with 2 rows, 35 columns
attributes(border,green)
prompt "Son correctos los datos (S/N) : ? " for resp
close window presp
let resp = upshift(resp)
if resp = "S" then
    message "Los cambios se han realizado"
    update prodold set prodold.* = altas_p.* where clave= altas_p.clave
    clear form
end if
open window presp at 15,20 with 2 rows, 40 columns
attributes(border,green)
prompt "Desea realizar otro cambio (S/N) : ? " for resp
close window presp
let resp = upshift(resp)
if resp = "S" then
    close window prod
    call cambios_productos()
else
clear form
close window prod
end if
end if
end if
end function

function cambios_clientes()
open window cli at 2,3 with form "alt_feli" attributes (border,yellow)
open window clave_ven at 5,10 with 2 rows, 50 columns
attributes(border,green)
prompt "Dar la clave del Cliente a Cambiar: ? " for clavecli
close window clave_ven
let clavecli=upshift(clavecli)
initialize altas_c.* to null
select * into altas_c.*
from clientes where clientes.clave = clavecli
if status = notfound then
    error "clave incorrecta"
    sleep 2 .
    clear form
open window presp at 10,20 with 2 rows, 35 columns
attributes (border,green)
prompt " Desea dar otro cambio (S/N): ? " for resp
close window presp
let resp= upshift(resp)
if resp = "S" then
close window cli
call cambios_clientes()
else
clear form
close window cli

```

```

    call cambios()
  end if
else
  display "Relice sus cambios" at 2,1
  sleep 1
input altas_c.nombre,
      altas_c.clave,
      altas_c.direccion,
      altas_c.telefono,
      altas_c.rfc,
      altas_c.codigo_postal,
      altas_c.cd_y_estado,
      altas_c.contacto
without defaults
from nombre,
    clave,
    direccion,
    telefono,
    rfc,
    codigo_postal,
    cd_y_estado,
    contacto
before field clave
next field direccion
end input
if int_flag = true then
  let int_flag = false
  display "funcion cambios abortada" at 2,2
  sleep 2
  close window cli
  call cambios()
else
  open window presp at 23,44 with 2 rows, 35 columns
  attributes(border,green)
  prompt "Son correctos los datos (S/N) : ?" for resp
  close window presp
  let resp = upshift(resp)
  if resp = "S" then
    message "Los cambios se han realizado"
    update clientes set clientes.* = altas_c.* where clave = altas_c.clave
    clear form
  end if
  open window presp at 15,20 with 2 rows, 40 columns
  attributes(border,green)
  prompt "Desea realizar otro cambio (S/N) : ?" for resp
  close window presp
  let resp = upshift(resp)
  if resp = "S" then
    close window cli
    call cambios_clientes()
  else
    clear form
    close window cli
  end if
end if

```

```

end if
end if

end function

function cambios_provedores()
open window prov at 2,3 with form "alt_ppro" attributes (border,yellow)
open window clave_ven at 5,10 with 2 rows, 50 columns
attributes(border,green)
prompt "Dar la clave del Proveedor a Cambiar: ? " for claveprov
close window clave_ven
let claveprov=upshift(claveprov)
initialize altas_pro.* to null
select * into altas_pro.*
from provedores where provedores.clave = claveprov
if status = notfound then
error "clave incorrecta"
sleep 2
clear form
open window presp at 10,20 with 2 rows, 35 columns
attributes (border,green)
prompt " Desea dar otro cambio (S/N): ? " for resp
close window presp
let resp= upshift(resp)
if resp = "S" then
close window prov
call cambios_provedores()
else
clear form
close window prov
call cambios()
end if
else
display "Relice sus cambios" at 2,1
sleep 1
input altas_pro.clave,
      altas_pro.nombre,
      altas_pro.direccion,
      altas_pro.cd_y_estado,
      altas_pro.rfc,
      altas_pro.contacto,
      altas_pro.telefono
without defaults
from clave,
      nombre,
      direccion,
      cd_y_estado,
      rfc,
      contacto,
      telefono
before field clave
next field nombre
end input
if int_flag = true then

```

TESIS EL CODIGO DE BARRAS Y LA AUTOMATIZACION

```

let int_flag = false
display "funcion cambios abortada" at 2,2
sleep 2
close window prov
call cambios()
else
open window presp at 23,44 with 2 rows, 35 columns
attributes(border,green)
prompt "Son correctos los datos (S/N) : ? " for resp
close window presp
let resp = upshift(resp)
if resp = "S" then
message "Los cambios se han realizado"
update proveedores set proveedores.* = altas_pro.* where clave= altas_pro.clave
clear form
end if
open window presp at 15,20 with 2 rows, 40 columns
attributes(border,green)
prompt "Desea realizar otro cambio (S/N) : ? " for resp
close window presp
let resp = upshift(resp)
if resp= "S" then
close window prov
call cambios_provedores()
else
clear form
close window prov
end if
end if
end if
end function

function pv()
let sub = 0
let iva = 0
let tot = 0
initialize vfactura to null
open window cuadro at 2,2 with form "fpercep" attributes (border)
open window captura1 at 2,2 with 2 rows, 76 columns
current window is captura1
menu "Punto Venta"
command "Alta " "DAR ALTA UNA FACTURA"
current window is cuadro
display " Altas: [ESC] Guarda, [DEL] Cancela" at 1,2
display " [CTRL] [T] Comenzar" at 2,1
call alta_factura()
display " " at 1,2
command "Facturar" "Factura la alta"
current window is cuadro
display " Para cambiar [ESC] para Facturar [INS]" at 1,2
call facturar(vfactura)
display " " at 1,2
command "Salir " "Salir punto de venta"
exit menu

```

```

end menu
close window cuadro
close window captura
end function

function alta_factura()
clear form
input claclient without defaults from clientes.clave
before field clave
let z_flag= null
let vfecha = today
display vfecha to formonly.vfe1
after field clave
if claclient is null then
error "Necesita dar la clave del cliente"
next field clientes.clave
end if
if checar (claclient) then
error "El cliente no está dado de alta"
next field clientes.clave
end if
if (claclient is not null) and checar (claclient)= false then
call dar_factura(claclient)
end if
end input
end function

function checar (clacliche)
define
clacliche like clientes.clave,
nombr like clientes.nombre
let nombr = null
select max(factura) into vfactura
from ventas
if status = notfound or vfactura is null then
let vfactura = 1
else
let vfactura = vfactura+1
end if
select nombre into nombr from clientes
where clave=clacliche
if status = notfound then
display nombr to formonly.nom1
return true
else
display nombr to formonly.nom1
display vfactura to formonly.fact1
return false
end if
end function

function facturar (vfact)
define
vfact like ventas.factura,

```

**TESIS EL CODIGO DE BARRAS Y LA AUTOMATIZACION**

```

vrf record like ventas.* ,
vdesc like prodold.clave,
vcont smallint

while vfact is null
  prompt " Dame el número de factura " for vfact
  select count(*) into vcont
  from ventas
  where factura = vfact
  if status = notfound or vcont =0 then
    let vfact = null
  end if
end while
declare wcf cursor for select a.*,b.descripcion
  from ventas a,prodold b
  where factura=vfact and
  (clave_del_producto=b.clave or
  clave_del_producto=b.codigo_de_barras)
start report repfact to pipe "lpt1"
foreach wcf into vrf.*,vdesc
  output to report repfact(vrf.*,vdesc)
end foreach
finish report repfact
end function

report repfact(vrf,vdesc)
define
vrf record like ventas.* ,
vcli record like clientes.* ,
psub,piva,total float,
vdesc like prodold.clave
format
page header
select * into vcli.* from clientes where clave = vrf.clave_del_cliente
print column 20, " Alfonso Méndez del Toro"
print column 20, " NOMBRE DE LA EMPRESA"
print column 50, "PATRICIO SANZ 501 "
print column 50, " COL. DEL VALLE "
print column 50, " MEXICO D.F. "
print column 50, "TEL: 682-90-21"
print column 50, "RFC: META 690729D01 "
print column 1 ,vcli.nombre,column 50, "Factura No. ", vrf.factura
print column 1 ,vcli.direccion
print column 1 ,vcli.rfc
print column 1 ,today
skip 3 lines
print column 1, "CLAVE PRODUCTO",column 20, "DESCREPCION",
column 40, "UNIDADES", column 55, "P.U.", column 65, "IMPORTE"
skip 2 lines
on every row
print column 1, vrf.clave_del_producto,
column 20,vdesc,
column 40,vrf.numero_de_unidades
using "----",

```

```

column 50,vrf.precio_unitario
using "--,--.&&".
column 60,vrf.precio_unitario*vrf.numero_de_unidades
using "--,--.&&".
on last row
skip 2 lines
print column 45, " Subtotal", column 65,
sum(vrf.precio_unitario*vrf.numero_de_unidades)
using "--,--.&&".
print column 45, " Iva",column 65,
(1*sum(vrf.precio_unitario*vrf.numero_de_unidades))
using "--,--.&&".
print column 65, "-----"
print column 45, " Total N",column 65,
(1.1*sum(vrf.precio_unitario*vrf.numero_de_unidades))
using "$--,--.&&".
page trailer
pause "presiona cualquier tecla"
end report

function dar_factura(ciaclient)
define pa_curr smallint,
sp_curr smallint,
i smallint,
ajuste integer,
aju like prodold.stock,
ciaclient like ventas.clave_del_cliente
let pa_curr = 0
let vfecha = today
display " [Ctrl] [C] para abortar, [Ctrl-f] facturar " at 2,1
input array a_factura
from s_perce.*
after field clave
let sp_curr = scr_line()
let pa_curr= arr_curr()
if checkarprod (a_factura[pa_curr].aclave,pa_curr,sp_curr) then
error "La clave es incorrecta"
next field clave
end if
before field desc1
next field unit1
after field unit1
let pa_curr= arr_curr()
if a_factura[pa_curr].unidades = 0 or a_factura[pa_curr].unidades is null
then
error "Este campo no puede ser nulo"
next field desc1
end if
if a_factura[pa_curr].unidades <= -1 then
call password()
if r=true then
next field cost1
else
next field desc1

```

```

end if
end if
after field cost1
let ajuste = (aju - a_factura[pa_curr].unidades)
let sp_curr= scr_line()
let pa_curr= arr_curr()
let a_factura[pa_curr].aunit =
a_factura[pa_curr].apv * a_factura[pa_curr].unidades
display a_factura[pa_curr].aunit to s_perce[sp_curr].cost1
let sub = (sub + a_factura[pa_curr].aunit )
let iva = ( sub * .1)
let tot = (sub + iva)
call calculo()
on key (control-f)
exit input
end input
if int_flag = true then
let int_flag = false
error " Captura Cancelada"
else
let pa_curr=arr_curr()
if pa_curr = 1 and a_factura[i].aclave is null
then
error "captura cancelada"
else
for i = 1 to pa_curr
if a_factura[i].aclave is not null then
insert into ventas
(clave_del_producto,numero_de_unidades,fecha,total_de_la_venta,
clave_del_cliente,precio_unitario,factura)
values
(a_factura[i].aclave,a_factura[i].unidades,vfecha,tot,clacient,
a_factura[i].apv,vfactura)
update prodold set stock=stock-a_factura[i].unidades
where clave=a_factura[i].aclave or
codigo_de_barras = a_factura[i].aclave
end if
end for
error "Factura dada de alta"
end if
end if
end function

function calculo()
display sub to formonly.sub
display iva to formonly.iva
display tot to formonly.tot
end function

function checarprod(aclave,pcur,scur)
define
aclave like prodold.clave,
pcur smallint,

```

```

scur smallint,
v_descri like prodold.descripcion,
vpv like prodold.precio1
let a_factura[pcur].descripcion = ""
if aclave is null then
    return true
else
    select descripcion,precio1 into v_descri,vpv
    from prodold
    where clave = aclave or codigo_de_barras =aclave
    if status = notfound or v_descri is null then
        return true
    else
        let a_factura[pcur].descripcion = v_descri
        let a_factura[pcur].apv = vpv
        display a_factura[pcur].descripcion to s_perce[scur].desc1
        display a_factura[pcur].apv to s_perce[scur].pv1
        display a_factura[pcur].aunit to s_perce[scur].unit1
        return false
    end if
end if
end function

```

#Esta forma sirve para generar la pantalla de bajas de proveedores#

```

database TEBARRAS
screen
(
      BAJAS   PROVEEDORES
-----
clave      [f000      ]
nombre     [f001      ]
direccion  [f002      ]
cd_y_estado [f003      ]
rfc        [f004      ]
contacto   [f005      ]
telefono   [f006      ]
)
end
tables
proveedores
attributes
f000 = proveedores.clave,upshift,reverse;
f001 = proveedores.nombre,upshift,reverse;
f002 = proveedores.direccion,upshift,reverse;
f003 = proveedores.cd_y_estado,upshift,reverse;
f004 = proveedores.rfc,upshift,reverse;
f005 = proveedores.contacto,upshift,reverse;
f006 = proveedores.telefono,upshift,reverse;
end
instructions
delimiters " "
    
```

#En el sistema la llamamos con el nombre de Altprov1#

#Esta forma sirve para generar cambios en clientes#

database TEBARRAS

screen

{ Ctrl\_C para abortar

CAMBIOS CLIENTES

-----

nombre [f000 ]

clave [f001 ]

direccion [f002 ]

telefono [f003 ]

rfc [f004 ]

codigo\_postal [f005 ]

cd\_y\_estado [f006 ]

contacto [f7]

}  
end

tables

clientes

attributes

f000 = clientes.nombre,comments= " Dame el nombre ",reverse,upshift;

f001 = clientes.clave,reverse,upshift;

f002 = clientes.direccion,upshift,reverse,comments= "Dame la direccion";

f003 = clientes.telefono,upshift,reverse,picture = "###-###-##-##";

f004 = clientes.rfc,reverse,upshift;

f005 = clientes.codigo\_postal,reverse,upshift;

f006 = clientes.cd\_y\_estado,reverse,upshift;

f7 = clientes.contacto,reverse,upshift;

end

instructions

delimiters " "

#En el sistema la llamamos con el nombre de Alt\_fcli#

## TESIS EL CODIGO DE BARRAS Y LA AUTOMATIZACION

#Esta forma sirve para generar la pantalla de alta de los clientes#

database TEBARRAS

screen

{

Para salir Ctrl-C

ALTAS DE CLIENTES

-----  
-----  
nombre [f000 ]

clave [f001 ]

direccion [f002 ]

telefono [f003 ]

rfc [f004 ]

codigo\_postal [f005 ]

cd\_y\_estado [f006 ]

contacto [f7]

}

end

tables

clientes

attributes

f000 = clientes.nombre,comments = " Dame el nombre ",reverse,upshift;

f001 = clientes.clave,reverse,upshift;

f002 = clientes.direccion,upshift,reverse,comments = "Dame la dirección";

f003 = clientes.telefono,upshift,reverse,picture = "###-###-##-###";

f004 = clientes.rfc,reverse,upshift;

f005 = clientes.codigo\_postal,reverse,upshift;

f006 = clientes.cd\_y\_estado,reverse,upshift;

f7 = clientes.contacto,reverse,upshift;

end

instructions

delimiters " "

#En el sistema la llamamos con el nombre de Alt\_for#





#Esta forma sirve para generar cambios a proveedores#

```

database TEBARRAS
screen
{
                                Ctrl_C para abortar

                                CAMBIOS  PROVEEDORES
-----
clave      [f000      ]
nombre     [f001      ]
direccion  [f002      ]
cd_y_estado [f003      ]
rfc        [f004      ]
contacto   [f005      ]
telefono   [f006      ]
}
end
tables
proveedores
attributes
f000 = proveedores.clave,upshift,reverse;
f001 = proveedores.nombre,upshift,reverse;
f002 = proveedores.direccion,upshift,reverse;
f003 = proveedores.cd_y_estado,upshift,reverse;
f004 = proveedores.rfc,upshift,reverse;
f005 = proveedores.contacto,upshift,reverse;
f006 = proveedores.telefono,upshift,reverse;
end
instructions
delimiters " "

```

#En el sistema la llamamos con el nombre de Alt\_ppro#

## TESIS EL CODIGO DE BARRAS Y LA AUTOMATIZACION

#Esta forma sirve para generar la pantalla de consulta de los proveedores#

```
.database TEBARRAS
screen
{
-----
CONSULTA PROVEEDORES
-----
clave      [f000      ]
nombre     [f001      ]
direccion  [f002      ]
cd_y_estado [f003      ]
rfc        [f004      ]
contacto   [f005      ]
telefono   [f006      ]
}
end
tables
proveedores
attributes
f000 = proveedores.clave,upshift,reverse;
f001 = proveedores.nombre,upshift,reverse;
f002 = proveedores.direccion,upshift,reverse;
f003 = proveedores.cd_y_estado,upshift,reverse;
f004 = proveedores.rfc,upshift,reverse;
f005 = proveedores.contacto,upshift,reverse;
f006 = proveedores.telefono,upshift,reverse;
end
instructions
delimiters " "
```

#En el sistema la llamamos con el nombre de Alt\_prep#

#Esta forma sirve para generar la pantalla de alta de los proveedores#

```

database TEBARRAS
screen
{
    Para salir Ctrl-C
    ALTA DE PROVEEDORES
    -----
    clave      [f000      ]
    nombre     [f001      ]
    direccion  [f002      ]
    cd_y_estado [f003      ]
    rfc        [f004      ]
    contacto   [f005      ]
    telefono   [f006      ]
}
end
tables
proveedores
attributes
f000 = proveedores.clave,upshift,reverse;
f001 = proveedores.nombre,upshift,reverse;
f002 = proveedores.direccion,upshift,reverse;
f003 = proveedores.cd_y_estado,upshift,reverse;
f004 = proveedores.rfc,upshift,reverse;
f005 = proveedores.contacto,upshift,reverse;
f006 = proveedores.telefono,upshift,reverse;
end
instructions
delimiters " "
    
```

# En el sistema la llamamos con el nombre de Alt\_prov#

#Esta forma sirve para generar la pantalla de consultas de productos#

```

database TEBARRAS
screen
{
-----
CONSULTAS PRODUCTOS
-----
clave      {f000      }
codigo_de_barras {f001      }
nombre     {f002      }
descripcion {f003      }

maximo     {f004      } minimo     {f005      }

precio1    {f006      }
precio2    {f007      }
precio3    {f008      }

descuento_maximo {f009      }

proveedor1 {f010      }proveedor2 {f011      }

stock      {f012      }
}
end
tables
prodold
attributes
f000 = prodold.clave,upshift,reverse;
f001 = prodold.codigo_de_barras,upshift,reverse;
f002 = prodold.nombre,upshift,reverse;
f003 = prodold.descripcion,upshift,reverse;
f004 = prodold.maximo,upshift,reverse;
f005 = prodold.minimo,upshift,reverse;
f006 = prodold.precio1,upshift,reverse;
f007 = prodold.precio2,upshift,reverse;
f008 = prodold.precio3,upshift,reverse;
f009 = prodold.descuento_maximo,upshift,reverse;
f010 = prodold.proveedor1,upshift,reverse;
f011 = prodold.proveedor2,upshift,reverse;
f012 = prodold.stock,upshift,reverse;
end
instructions
delimiters" "

```

#En el sistema la llamamos con el nombre de Auxicp#

## TESIS EL CODIGO DE BARRAS Y LA AUTOMATIZACION

#Esta forma sirve para generar cambios a los productos#

```
database TEBARRAS
screen
(                               Ctrl_C para abortar

-----
-----
clave          {f000      }
codigo_de_barras {f001    } ]
nombre         {f002      } ]
descripcion    {f003      } ]

maximo         {f004      }  minimo         {f005      }

precio1        {f006      }
precio2        {f007      }
precio3        {f008      }

descuento_maximo {f009    }

proveedor1     {f010      }]proveedor2 {f011      }

stock          {f012      }

)
end
tables
prodold
attributes
f000 = prodold.clave,upshift,reverse;
f001 = prodold.codigo_de_barras,upshift,reverse;
f002 = prodold.nombre,upshift,reverse;
f003 = prodold.descripcion,upshift,reverse;
f004 = prodold.maximo,upshift,reverse;
f005 = prodold.minimo,upshift,reverse;
f006 = prodold.precio1,upshift,reverse;
f007 = prodold.precio2,upshift,reverse;
f008 = prodold.precio3,upshift,reverse;
f009 = prodold.descuento_maximo,upshift,reverse;
f010 = prodold.proveedor1,upshift,reverse;
f011 = prodold.proveedor2,upshift,reverse;
f012 = prodold.stock,upshift,reverse;
end
instructions
delimiters" "
```

#En el sistema la llamamos con el nombre de Auxiepl#

#Esta forma sirve para generar la pantalla de alta de los productos#

```

database TEBARRAS
screen
(
                                Para salir Ctrl-C
                                ALTA DE PRODUCTOS
-----
clave      [f000      ]
codigo_de_barras [f001      ]
nombre     [f002      ]
descripcion [f003      ]

maximo     [f004      ]   minimo     [f005      ]

precio1    [f006      ]
precio2    [f007      ]
precio3    [f008      ]

descuento_maximo [f009      ]

proveedor1 [f010      ]proveedor2 [f011      ]

stock      [f012      ]
)
end
tables
prodold
attributes
f000 = prodold.clave,upshift,reverse;
f001 = prodold.codigo_de_barras,upshift,reverse;
f002 = prodold.nombre,upshift,reverse;
f003 = prodold.descripcion,upshift,reverse;
f004 = prodold.maximo,upshift,reverse;
f005 = prodold.minimo,upshift,reverse;
f006 = prodold.precio1,upshift,reverse;
f007 = prodold.precio2,upshift,reverse;
f008 = prodold.precio3,upshift,reverse;
f009 = prodold.descuento_maximo,upshift,reverse;
f010 = prodold.proveedor1,upshift,reverse;
f011 = prodold.proveedor2,upshift,reverse;
f012 = prodold.stock,upshift,reverse;
end
instrucciones
delimiters" "

```

#En el sistema la llamamos con el nombre de Auxilia#



TESIS EL CODIGO DE BARRAS Y LA AUTOMATIZACION

#Esta forma nos sirve para generar la pantalla del punto de venta#

```

database tebarras
screen
{
-----
Fecha          PROGRAMA PUNTO DE VENTA          No. Factura
[f001          ] AUTOMATIZANDO LA VENTA CON CODIGO DE BARRAS [fact ]

CLAVE CLIENTE :[f000          ] [nom1          ]

Cve. Prod.   Descripcion          P.U.   Unidades   Importe

[a0          ] [desc          ] [pv   ] [unit  ] [cost  ]
[a0          ] [desc          ] [pv   ] [unit  ] [cost  ]
[a0          ] [desc          ] [pv   ] [unit  ] [cost  ]
[a0          ] [desc          ] [pv   ] [unit  ] [cost  ]
[a0          ] [desc          ] [pv   ] [unit  ] [cost  ]
[a0          ] [desc          ] [pv   ] [unit  ] [cost  ]
[a0          ] [desc          ] [pv   ] [unit  ] [cost  ]
[a0          ] [desc          ] [pv   ] [unit  ] [cost  ]

-----
Subtotal   +   iva   = total
[sub       ] + [iva  ] = N${tot   }
}
end
tables
clientes
prodold
attributes
f001 = formonly.vfe1 type char;
f000 = clientes.clave,upshift,
      comments = "Dame la clave del cliente";
a0   = prodold.clave,upshift,
      comments = "Dame la clave del producto";
nom1 = formonly.nom1 type char;
fact = formonly.fact1 type integer;
sub  = formonly.sub type float;
iva  = formonly.iva type float;
tot  = formonly.tot type float;
desc = formonly.desc1 type char;
unit = formonly.unit1 type smallint;
pv   = formonly.pv1 type integer;
cost = formonly.cost1 type float;
end
instructions
screen record s_perce[8] (prodold.clave,formonly.desc1,formonly.pv1,
                        formonly.unit1,formonly.cost1)
delimiters " "
end

```

#En el sistema lo llamamos con el nombre de Fpercep#

***CAPITULO VII***

**CONCLUSIONES**, despues de haber realizado este trabajo de investigacion solo queda el dar conclusiones, las cuales son las siguientes:

- El código de barras es una de las herramientas mas importantes para la automatización de procesos.

- El código de barras es una tecnologia facil de manejar y que esta al alcance de cualquier persona.

- El código de barras se puede implementar en cualquier proceso en el cual exista la captura de información.

- El código de barras nos reduce los errores de captura a un promedio de un dato en un millon de datos capturados.

- Con la tecnologia de las recolectoras de datos aunado al código de barras podemos utilizarlas en el control de inventarios ahorrando hasta en un 95% el tiempo de toma de inventarios.

- En México es una tecnologia nueva y se deve de dar mayor importancia a este tipo de tecnologia.

- En Mexico hay mucho trabajo en este campo y por lo tanto una gran necesidad de personal capacitado en este campo.

***BIBLIOGRAFIA***

**TESIS EL CODIGO DE BARRAS Y LA AUTOMATIZACION**

1. **THE BARCODE BOOK**  
Roger C. Palmer  
Helmners Publishing, inc.  
1991
  
2. **BARCODE STANDARS WITHIN ANSI SUPPORTED COMMITTEES**  
Gilligan Alan  
Scantech  
1987
  
3. **INFORMIX 4GL REFERENCE MANUAL** VOL.1  
Informix  
ver. 4.0
  
4. **INFORMIX 4GL REFERENCE MANUAL** VOL.2  
Informix  
ver. 4.0  
Julio 1991
  
5. **INFORMIX 4GL USER GUIDE**  
Informix  
ver. 4.0  
Julio 1991
  
6. **INFORMIX GUIDE TO SQL REFERENCE**  
Informix  
ver 4.0  
Julio 1991
  
7. **INFORMIX GUIDE TO SQL TUTORIAL**  
Informix  
ver 4.0  
julio 1991