



11  
29

**UNIVERSIDAD NACIONAL  
AUTONOMA DE MEXICO**

---

---

**FACULTAD DE CONTADURIA Y ADMINISTRACION**

**SISTEMA DE RECIBOS DE PAGO**

**Seminario de Investigación Informática**

Que en opción al grado de :

**LICENCIADO EN INFORMATICA**

**P r e s e n t a :**

**JAVIER IÑIGO FUENTES**

Asesor del Seminario:

**L.A.E. MARIO NOVOA GAMAS**

**TESIS CON 1993  
FALLA DE ORIGEN**



Universidad Nacional  
Autónoma de México



**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 esta 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.

# **CONTENIDO**

## **PROLOGO**

## **INTRODUCCION**

### **I. SITUACION ACTUAL**

### **II. ANALISIS ESTRUCTURADO**

- II.1 Descripción
- II.2 Objetivo del Sistema
- II.3 Lista de Eventos
- II.4 Diagrama de Contexto
- II.5 Diagramas de Flujo de Datos
- II.6 Especificación de Procesos
- II.7 Diccionario de Datos

### **III. DISEÑO ESTRUCTURADO**

- III.1 Descripción
- III.2 Diagramas de Estructura

### **IV. IMPLANTACION**

- IV.1 Programación Orientada a Objetos.
- IV.2 Definición de Objetos.
- IV.3 Estructura de los Archivos de la Base de Datos.

## **CONCLUSIONES**

## **BIBLIOGRAFIA**

## **ANEXOS**

- A. Requerimientos e Instalación del Sistema.
- B. Reportes del Sistema.
- C. Procedimientos Manuales.
- D. Listado del Programa.

## INDICE

PROLOGO	
INTRODUCCION	
I. SITUACION ACTUAL .....	1
II. ANALISIS ESTRUCTURADO .....	3
II.1 Descripción .....	3
II.2 Objetivo del Sistema .....	5
II.3 Lista de Eventos .....	5
II.4 Diagrama de Contexto .....	7
II.5 Diagramas de Flujo de Datos .....	8
II.6 Especificación de Procesos .....	19
II.7 Diccionario de Datos .....	26
III. DISEÑO ESTRUCTURADO .....	32
III.1 Descripción .....	32
III.2 Diagramas de Estructura .....	34
IV. IMPLANTACION .....	39
IV.1 Programación Orientada a Objetos .....	38
IV.2 Definición de Objetos .....	41
IV.3 Estructura de los Archivos de la Base de Datos .....	45
CONCLUSIONES .....	47
BIBLIOGRAFIA .....	49
ANEXOS	
A. Requerimientos e Instalación del Sistema .....	51
B. Reportes del Sistema .....	58
C. Procedimientos Manuales .....	61
D. Listado del Programa .....	64

## PROLOGO

En el desarrollo de este trabajo observé avances significativos solamente cuando marqué horarios para trabajar en mi tesis y respetándolos, cuando no estaba seguro de algo investigaba o preguntaba a quien me pudiera ayudar, por que cuando seguía trabajando con dudas o sabía que algo no andaba bien, el trabajo se obstaculizaba mucho y siempre tenía que regresarme al punto el cual tenía dudas desde el principio, pero cuando uno trabaja seguro de lo que está haciendo es muy fluido el avance.

Considero importante incluir la situación anímica por la cual pasé, no es un trabajo fácil, y lo sabemos todos los que nos encontramos realizando nuestra tesis, sin embargo creo que nosotros mismos ponemos barreras para realizarlo, desde decidirnos a empezar, elegir tema, dedicarle tiempo, por que muchos estamos trabajando y el trabajo ¿nos absorbe totalmente?, en fin, son tantos motivos los que tenemos para no dedicarle tiempo al último requisito que nos piden para poder presentar el examen profesional, obtener un título, la cédula profesional y anteponer realmente en nuestras tarjetas de presentación L.I. (Licenciado en Informática), por que aunque nos estemos ganando un lugar importante en el mercado de trabajo, y tengamos muchas satisfacciones personales, esta satisfacción nunca va a ser completa, siempre habrá un sentimiento de que algo nos falta.

## INTRODUCCION

La descripción de la situación actual del registro de ingresos a la F.C.A. se describe en el primer capítulo, donde además se plantea de una forma general el ambiente con el cual interactúa el sistema, la fase de análisis se muestra en el capítulo dos que comprende el objetivo del sistema donde se marca el alcance del mismo, la lista de eventos que estimulan al sistema y producen que reaccione, el diagrama de contexto y de flujo de datos que son la representación del ambiente y comportamiento respectivamente, por medio de modelos que se utilizan para comunicarse con el usuario y tener una vista general o particular del sistema, la especificación de procesos que describe paso a paso los procesos utilizados y el diccionario de datos que se utiliza para describir todos los flujos de datos, flujos compuestos y almacenamientos.

La representación estructural del sistema, su jerarquía y las relaciones que existen entre los módulos que lo componen son descritos en el capítulo tres, utilizando diagramas de estructura que son una herramienta gráfica de la metodología.

La implantación del sistema se realiza bajo el concepto de Programación Orientada a Objetos utilizando el Lenguaje C++ versión 3.1 para Windows, de esta forma en el capítulo cuatro se hace una breve descripción de las características de este enfoque, así como la definición de los objetos y clases del sistema, el diseño de los reportes generados y la estructura de los archivos.

**Para finalizar se incluyeron los anexos referentes a los requerimientos del sistema y el procedimiento de su instalación, los reportes que genera, los procedimientos manuales que fueron incluidos, así como su cursograma y el listado del sistema.**

## I. SITUACION ACTUAL

El ingreso de dinero a la Facultad de Contaduría y Administración es controlado por medio de recibos de pago con un número de folio oficial asignado por Rectoría, que en respuesta a la solicitud de una dependencia entrega paquetes de recibos de pago y fichas de depósito registrando los números de folio que ampara cada paquete, las dependencias deben revisar que no falte ningún recibo del paquete te entregado, en caso que falte algún recibo o ficha de depósito se indica a rectoría para que el recibo faltante no sea considerado en el informe que solicita rectoría del uso de los recibos y fichas de pago entregados..

Los conceptos por los cuales se cobra en la Caja son Cursos Extracurriculares e Intersemestrales, Venta de Libros y Apuntes, Claves para el uso de equipo de cómputo, Venta de Recuerdos de la F.C.A., Reposición de claves de computadora, de credenciales y multa por retraso en entrega de libros a la biblioteca, Seminarios de áreas básicas, Donativos, Revista: consultorio fiscal, emprendedores y contaduría y administración, y otros

Actualmente se lleva un registro de los recibos emitidos por la Facultad describiendo el concepto, importe, fecha, no. de recibo, forma de pago.

Al cubrir una cantidad determinada de dinero se realiza un depósito al banco el cual le regresa su comprobante de depósito con su número correspondiente. Se debe registrar la cantidad, los



folios de los recibos que amparan dicho depósito, número de cuenta y fecha del depósito.

Las fichas de depósito también se solicitan a rectoría, quien las entrega con un número de folio oficial.

Se entrega al departamento de contabilidad y a la secretaría administrativa un reporte de ingresos, en el cual se describen los datos de los recibos pagados así como los depósitos realizados al banco, a la Dirección se entrega un resumen de ingresos en un periodo determinado.

Finalmente a Rectoría se le entrega un informe de los recibos utilizados y de los depósitos realizados.

Los reportes que actualmente genera el sistema son en forma manual, así que para generar un reporte de diferente mes se tiene que revisar la carpeta donde está almacenada la información y crearlo nuevamente.

## II. ANALISIS ESTRUCTURADO

### II.1 DESCRIPCION

En el desarrollo de sistemas, el análisis se refiere al estudio de alguna área organizacional o alguna aplicación determinada, que generalmente encabeza la especificación de un nuevo sistema.

"El resultado más importante del análisis de sistemas es la documentación especificada"<sup>1</sup>

De los aspectos principales del Análisis de Sistemas se encuentra el de utilizar modelos para crear una representación abstracta de algo real, las razones por las cuales se modela es porque facilita la comunicación entre el analista y los usuarios.

Para crear modelos utilizamos herramientas de las siguientes características:

Herramientas Gráficas con ayuda de textos, porque pueden ser entendidas con pocas indicaciones, y son concisas.

---

<sup>1</sup> DE MARCO, Tom. "STRUCTURED ANALYSIS AND SYSTEM SPECIFICATION". E.E.U.U.: Prentice-Hall Software Series, 1979. p.4

Flexibles, para poder revisar el sistema con una vista general o desde un punto muy detallado del mismo.

Capaces de Controlar la redundancia, es decir facilitar al lector a juntar las piezas para comprender mejor el sistema.

Con Capacidad de predecir el Comportamiento del Sistema, verificar que el comportamiento es aceptable y si no cambiar el modelo.

" Utilizamos gráficos para mostrar:

Las Transformaciones,

Las entradas que son transformadas, y de donde se originaron,

Las salidas y hacia donde se dirigen y,

Los datos que son retenidos entre las transformaciones.

Utilizamos texto para definir:

Las reglas que gobiernan las transformaciones y,

El contenido y significado de las entradas y las salidas..." 2

## MODELO ESENCIAL

El modelo esencial ilustra lo que un sistema debe hacer en condiciones normales, excluyendo los detalles de como debe ser

---

<sup>2</sup> YOURDON, Edward. " STRUCTURED ANALYSIS WORKSHOP ", E.E.U.U.: Lecture Notes. 1990, Yourdon Inc. Cap. 2, p. 5

implantado y está formado por el modelo ambiental, el modelo de comportamiento y el diccionario de datos.

## **MODELO AMBIENTAL**

El modelo ambiental nos define el alcance del sistema y consiste de:

### **II.2 OBJETIVO DEL SISTEMA,**

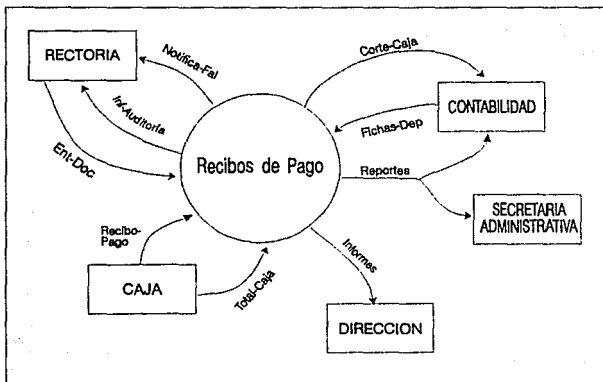
**REGISTRAR LOS INGRESOS GENERADOS POR LA  
FACULTAD DE CONTADURIA Y ADMINISTRACION  
A TRAVES DE RECIBOS DE PAGO.**

### **II.3 LISTA DE EVENTOS.**

- La caja de la Facultad de Contaduría y Administración entrega al Sistema los comprobantes de los Recibos de Pago.

- El departamento de Contabilidad entrega al Sistema las fichas de los Depósitos realizados.
- El sistema envía a la Dirección, a Contabilidad y a la Secretaría Administrativa un reporte de ingresos.
- El Sistema entrega a Rectoría un informe para la auditoría solicitada.
- El sistema entrega a la Dirección un informe de ingresos.
- La Caja de la F.C.A. envía al sistema el total de efectivo.
- El sistema envía a Contabilidad el resultado del corte de Caja.
- Rectoría entrega al sistema el total de recibos de pago y de fichas de depósito.
- El Sistema notifica a Rectoría el faltante de recibos de pago y de fichas de depósito.

## II.4 DIAGRAMA DE CONTEXTO.



## **MODELO DE COMPORTAMIENTO**

El modelo de comportamiento consiste en la descripción de procesos, que es la respuesta del sistema a los eventos en su ambiente así como su interrelación y el almacenamiento de datos, que muestra la información que debe ser almacenada para que opere el sistema y las asociaciones entre estas piezas de información.

Básicamente el modelo de comportamiento consiste de:

### **II.5 DIAGRAMAS DE FLUJO DE DATOS.**

Los diagramas de flujo nos ofrecen una vista general de los principales componentes funcionales de un sistema, pero no especifica los detalles de estos componentes. Para mostrar como se realiza la transformación de información en algún proceso, se utiliza la especificación de procesos.

## Diagrama 0

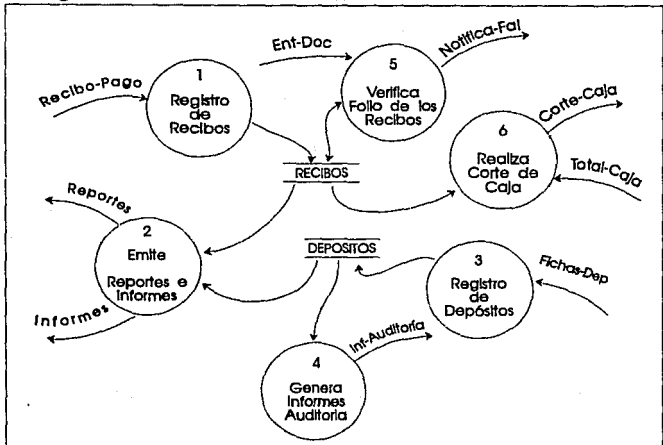
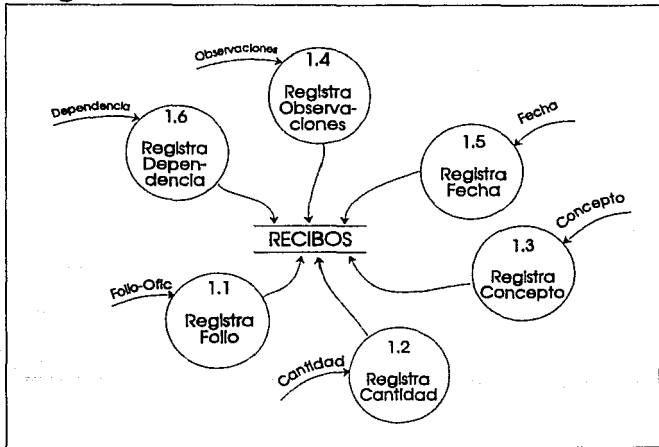
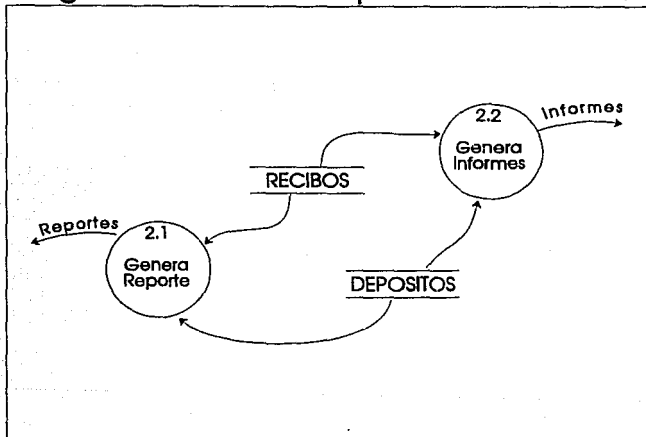




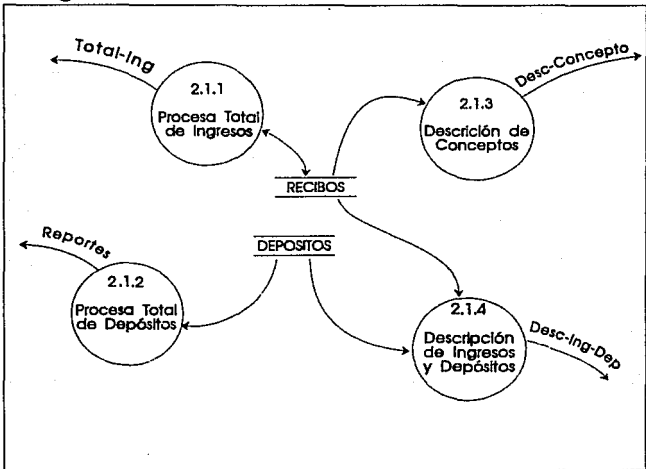
Diagrama 1. REGISTRO DE RECIBOS



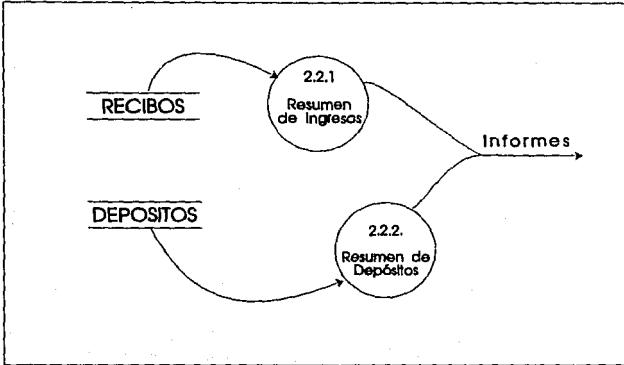
## Diagrama 2. Emite Reportes e Informes



## Diagrama 2.1 GENERA REPORTE



## Diagrama 2.2 GENERA INFORMES



## Diagrama 2.1.2 PROCESA TOTAL DEPOSITADO

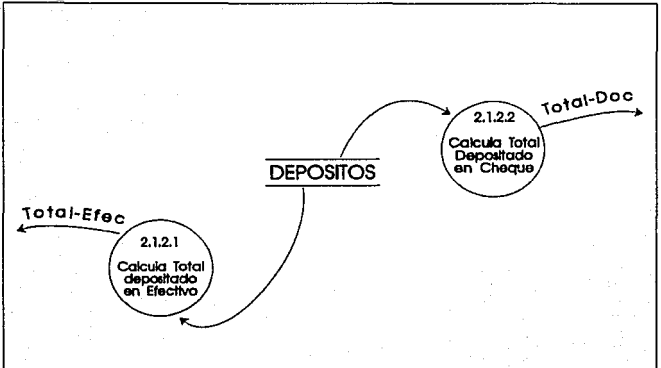
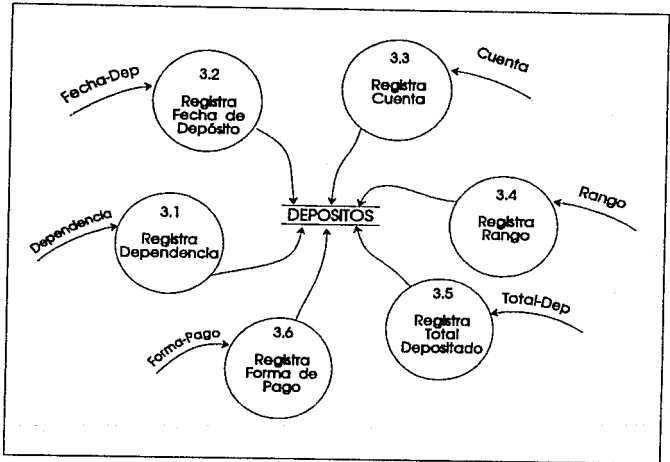


Diagrama 3. REGISTRO DE DEPOSITOS



## Diagrama 4 GENERA INFORMES AUDITORIA

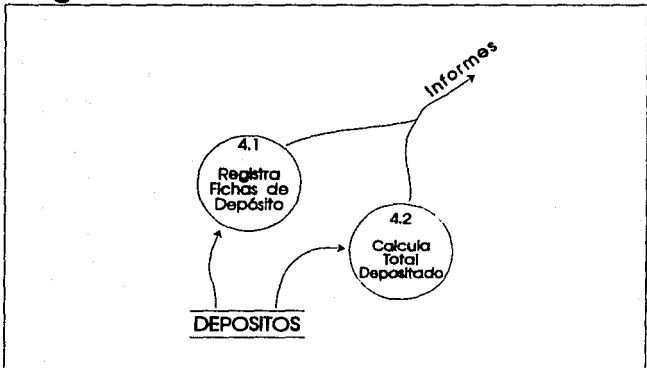


Diagrama 5. Verifica Folio de los Recibos

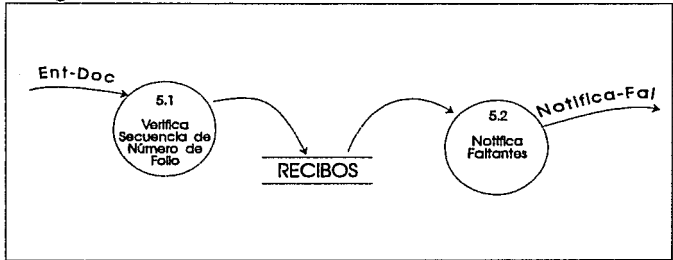
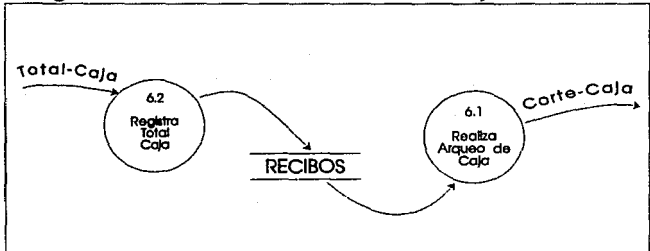




Diagrama 6. Realiza Corte de Caja



## II.6 ESPECIFICACION DE PROCESOS

Cuando se ha particionado el sistema y expresado en diagramas de flujo de datos con elementos definidos en el Diccionario de Datos nos encontramos con procesos que ya no es conveniente seguir particionando, debido a que representan actividades concretas e independientes, a éstos procesos se les denomina Procesos Primitivos y necesitamos describirlos en forma detallada por medio de la Especificación de Procesos que consiste en redactarlos, en forma algorítmica, por medio de lenguaje natural.

Por cada proceso primitivo del conjunto de diagramas de flujo de datos, debemos tener una especificación de procesos anotando el número del proceso y describiendo qué es lo que se realiza en ese proceso, así como las relaciones que existen entre los flujos de datos.

PROCESO No.	1.1
NOMBRE DEL PROCESO:	REGISTRA FOLIO
Para cada RECIBO-PAGO: Obtener FOLIO-OFIC y Escribirla en el archivo RECIBOS	

<b>PROCESO No.</b>	<b>1.2</b>
<b>NOMBRE DEL PROCESO:</b>	<b>REGISTRA CANTIDAD</b>
<b>Para cada RECIBO-PAGO:          Obtener CANTIDAD y          Escribirla en el archivo RECIBOS</b>	

<b>PROCESO</b>	<b>PROCESO No.</b>	<b>1.3</b>
<b>NOMBRE DEL PROCESO:</b>	<b>REGISTRA CONCEPTO</b>	
<b>Para cada RECIBO-PAGO:          Obtener CONCEPTO y          Escribirlo en el archivo RECIBOS</b>		

<b>PROCESO No.</b>	<b>1.4</b>
<b>NOMBRE DEL PROCESO:</b>	<b>REGISTRA OBSERVACIONES</b>
<b>Para cada RECIBO-PAGO:          Obtener OBSERVACIONES          Si el pago fue en cheque,          Obtener No-Cheque y Banco          Escribirlos en el archivo RECIBOS          Si el pago fue en efectivo dejar en blanco.</b>	

<b>PROCESO No.</b>	<b>1.5</b>
<b>NOMBRE DEL PROCESO:</b>	<b>REGISTRA FECHA</b>
Para cada <b>RECIBO-PAGO</b> : Obtener <b>FECHA</b> y Escribirla en el archivo <b>RECIBOS</b>	

<b>PROCESO No.</b>	<b>1.6</b>
<b>NOMBRE DEL PROCESO:</b>	<b>REGISTRA DEPENDENCIA</b>
Para cada <b>RECIBO-PAGO</b> : Obtener <b>DEPENDENCIA</b> y Escribirla en el archivo <b>RECIBOS</b>	

<b>PROCESO No.</b>	<b>2.1.1</b>
<b>NOMBRE DEL PROCESO:</b>	<b>PROCESA TOTAL DE INGRESOS</b>
Para Cada <b>RECIBO-PAGO</b> : Obtener <b>CANTIDAD</b> y Sumar los ingresos para obtener <b>TOTAL-ING</b>	

<b>PROCESO No.</b>	<b>2.1.2.1</b>
<b>NOMBRE DEL PROCESO:</b>	<b>CALCULA TOTAL DEPOSITADO EN EFECTIVO</b>
Para Cada <b>FICHA-DEP</b> : Obtener <b>TOTAL-EFEC</b> y Sumar los totales	

<b>PROCESO No.</b>	<b>2.1.2.2</b>
<b>NOMBRE DEL PROCESO:</b>	<b>CALCULA TOTAL DEPOSITADO EN CHEQUE</b>
Para Cada FICCHS-DEP: Obtener TOTAL-DOC y Sumar los totales	

<b>PROCESO No.</b>	<b>2.1.3</b>
<b>NOMBRE DEL PROCESO:</b>	<b>DESCRIPCION DE CONCEPTOS</b>
Para Cada RECIBO-PAGO: Obtener CONCEPTO e Identificarlos para obtener DESC-CONCEPTO	

<b>PROCESO No.</b>	<b>2.1.4</b>
<b>NOMBRE DEL PROCESO:</b>	<b>DESCRIPCION DE INGRESOS Y DEPOSITOS</b>
Obtener TOTAL-ING Del Archivo de RECIBOS Obtener TOTAL-DEP Del Archivo DEPOSITOS Genera DESC-ING-DEP	

<b>PROCESO No.</b>	<b>2.2.1</b>
<b>NOMBRE DEL PROCESO:</b>	<b>RESUMEN DE INGRESOS</b>
Para Cada RECIBO-PAGO: Obtener CONCEPTO, Obtener TOT-ING, Realizar resumen de ingresos por concepto.	

PROCESO No.	2.2.2
NOMBRE DEL PROCESO:	RESUMEN DE INGRESOS
Para Cada FICHA-DEP: Obtener TOTAL-DEP Realizar resumen de depósitos actualizado.	

PROCESO No.	3.1
NOMBRE DEL PROCESO:	REGISTRA DEPENDENCIA
Para cada FICHA-DEP: Obtener DEPENDENCIA y Escribirla en el Archivo DEPOSITOS	

PROCESO No.	3.2
NOMBRE DEL PROCESO:	REGISTRA FECHA DE DEPOSITO
Para cada FICHA-DEP: Obtener FECHA-DEP y Escribirla en el Archivo DEPOSITOS	

PROCESO No.	3.3
NOMBRE DEL PROCESO:	REGISTRA CUENTA
Para cada FICHA-DEP: Obtener CUENTA y Escribirla en el Archivo DEPOSITOS	

PROCESO No.	3.4
NOMBRE DEL PROCESO:	REGISTRA RANGO
Para cada FICHA-DEP: Obtener RANGO y Escribirla en el Archivo DEPOSITOS	

PROCESO No.	3.5
NOMBRE DEL PROCESO:	REGISTRA TOTAL DEPOSITADO
Para cada FICHA-DEP: Obtener TOTAL-DEP y Escribirla en el Archivo DEPOSITOS	

PROCESO No.	3.8
NOMBRE DEL PROCESO:	REGISTRA FORMA DE PAGO
Para cada FICHA-DEP: Obtener FORMA-PAGO y Escribirla en el Archivo DEPOSITOS	

PROCESO No.	4.1
NOMBRE DEL PROCESO:	OBTIENE FICHA DE DEPOSITO
Para cada FICHA-DEP: Leer FICHA-NUM, CUENTA, DEPENDENCIA, FECHA-DEP del Archivo de DEPOSITOS. Generar Inf-Audito.	

PROCESO No.	4.2
NOMBRE DEL PROCESO:	<b>CALCULA TOTAL DEPOSITADO</b>
Para cada INF-AUDITORIA: Sumar TOT-DEP Generar Inf-Audito.	

PROCESO No.	5.1
NOMBRE DEL PROCESO:	<b>VERIFICA FOLIO DE LOS RECIBOS</b>
Para cada ENT-DOC: Verificar Manualmente la secuencia de los números de folio en los recibos de pago Registrar resultado en archivo de RECIBOS.	

PROCESO No.	5.2
NOMBRE DEL PROCESO:	<b>NOTIFICA FALTANTES</b>
Para cada FOLIO-OFIC faltante: Generar Notifica-Fal.	

PROCESO No.	6.1
NOMBRE DEL PROCESO:	<b>REALIZA ARQUEO CAJA</b>
Para cada corte de caja: comparara el <b>TOTAL-CAJA</b> con <b>TOTAL-REGISTRADO</b> para Generar <b>CORTE-CAJA</b>	



PROCESO No.	6.2
NOMBRE DEL PROCESO:	REGISTRA TOTAL CAJA
Para cada cambio de turno: Contar el dinero que hay en caja Registrar el TOTAL-CAJA en el archivo de RECIBOS.	

## II.7 DICCIONARIO DE DATOS

" El diccionario de datos define los elementos declarados en los Diagramas de Flujo de Datos, básicamente:

- Describiendo el significado de los flujos y almacenamientos.
- Especificando la composición de almacenamientos y flujos compuestos.
- Especificando las unidades y valores de los flujos y almacenamientos elementales."<sup>3</sup>

El Diccionario de Datos es una intersección entre ambos modelos ya que empieza a elaborarse en el modelo ambiental, con el diagrama de contexto, y se concluye a lo largo del modelo de comportamiento con los diagramas de flujos de datos y la descripción de los archivos utilizados.

<sup>3</sup> YOURDON, Edward, "STRUCTURED ANALYSIS WORKSHOP", E.E.U.U.: Lecture Notes, 1990, Yourdon Inc, Cap. 7, p. 7

El diccionario de datos es la herramienta que utilizamos para definir los elementos de datos declarados en los diagramas de flujo de datos, su función principal es la de describir el significado de flujos y almacenamientos, especificar su composición y especificar sus unidades y valores.

Los símbolos utilizados para definir la composición de un flujo o almacenamiento son:

Símbolo	Significado
=	Está compuesto de
+	Y
n{..}n1	Iteraciones de los elementos, n y n1 indican el rango de las iteraciones
[   ]	Seleccionar UNA de las opciones dentro de los corchetes
(..)	Opcional puede o no estar incluido
@	Identificador de un almacenamiento, Campo LLave
*..*	Comentarios

Cantidad = \* Importe monetario del concepto a cobrar \*

Concepto = \* Motivo por el cual se paga el recibo \*

Corte-Caja = Total-Ing + Total-Caja

---

Cuenta	= * Número de Cuenta a la que se deposita en dinero * 9{Dígito}9
Dependencia	= * Número de la Facultad de Contaduría y Administración reconocido en la Universidad Nacional Autónoma de México 006 *
Depósitos	= Fichas-Dep + Total-Dep + Fecha-Dep
Desc-Concepto	= * Descripción de conceptos pagados *
Desc-Ing-Dep	= Total-Ing + Total-Dep
Ent-Doc	= Total-Recibos-Ent + Fecha-Ent + Secuencia
Fecha	= * DD/MM/AA *
Fecha-Dep	= * Fecha en que se realiza el depósito, DD/MM/AA *
Fecha-Ent	= * Fecha de entrega de Recibos por parte de Rectoría DD/MM/AA *
Fecha-Pago	= * Fecha en la que se realiza el pago del recibo DD/MM/AA *

---

<b>Ficha-Dep</b>	= Ficha-Num + Dependencia + Fecha-Dep + Cuenta + Rango + Forma-Pago + Total- Dep
<b>Ficha-Num</b>	= 7{Dígito}
<b>Folio-Ofic</b>	= * Número Foliado con el cual se identifica el Recibo de Pago *  7{Dígito}7
<b>Forma-Pago</b>	= [ Total-Efec   Total-Doc + No-Cheque   Total-Efec + Total-Doc + No-Cheque ]
<b>Inf-Audito</b>	= Fecha-Inf + Dependencia + Fichas-Dep + Total-Dep
<b>Informes</b>	= * Resumen de ingresos generados por la Facultad de Contaduría y Administración en un periodo determinado, describiendo los conceptos correspondientes. *  Periodo + Concepto + Total-Ing + (Total- Dep)
<b>Matutino</b>	= * Cobros efectuados entre las 9:00 y las 14:00 hrs. *

---

No-Cheque	= * Número de Cheque depositado * 7{Dígito}7
Notifica-Fal	= Recibo-Fal + Fecha-Ent
Observaciones	= * Si el pago se efectúa con un cheque se debe registrar el Número de cheque y el banco correspondiente. *
Periodo	= * Lapso de tiempo que cubre el Informe *
Rango	= * Total de Recibos de Pago que cubre la ficha de Depósito, DEL__AL__ *
Recibo-Fal	= * Número de folio del Recibo de pago faltante en la entrega de Rectoría *
Recibo-pago	= Folio-Ofic + Cantidad + Concepto + Dependencia + Fecha-Pago + Observaciones
Reportes	= * Relación del total de ingresos generados en una Fecha determinada. *  Total-Ing + Concepto + Turno + Fecha + (Total-Dep)

---

Secuencia	= * Números consecutivos de Folio Oficial de Recibos de Pago *
Total-Caja	= * Cuento Manual del dinero recibido en caja *
Total-Dep	= * Cantidad de Dinero depositado que cubre la Ficha de Depósito *
Total-Doc	= * Cantidad de Dinero depositado en Cheque *
Total-Efec	= * Cantidad de Dinero en efectivo depositado *
Total-Ing	= * Total de ingresos generados por la Facultad de Contaduría y Administración en una Fecha determinada. * 1{Dígito}7
Total-Ing-Dep	= Total-Dep + Total-Ing
Total-Recibo	= * Total de Recibos entregados por Rectoría *
Turno	= [ Matutino   Vespertino ]
Vespertino	= * Cobros efectuados entre las 16:00 y las 20:00 hrs. *

### III. DISEÑO ESTRUCTURADO

#### III.1 DESCRIPCION

Un diagrama de estructura es una herramienta gráfica que utilizamos para representar jerarquía, se compone principalmente de tres elementos:

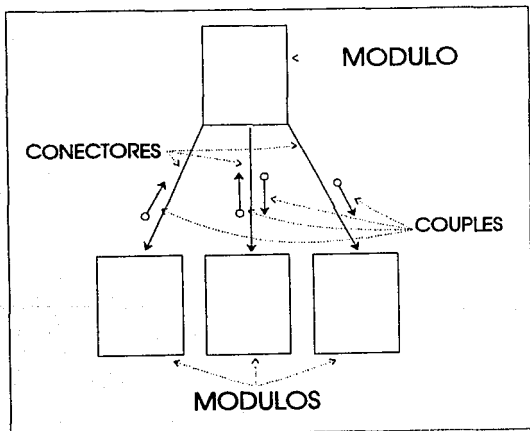
- Módulo, representado por un rectángulo con el nombre de la actividad.
- Conector, representado por un vector que une dos módulos. Un conector es una referencia de un módulo a otro, generalmente significa que un módulo hace una llamada a otro.
- Couple ó elemento dato que se representa con una flecha corta con una punta circular que va de un módulo a otro.

Un Diagrama de estructura muestra la partición del sistema en módulos, la jerarquía en la cual están ordenados y las relaciones entre ellos.

" La idea básica del Diseño Estructurado es que el Diseño debe realizarse desde la vista jerárquica de una aplicación, no desde una vista procedural. El nivel más alto muestra la principal división del trabajo, los niveles inferiores también subdividen el

trabajo hasta llegar a los niveles más bajos donde se detallan las actividades."<sup>1</sup>

El diseño estructurado es un enfoque específico para llevar a cabo el desarrollo de sistemas, determinando cuántas partes tendrá el sistema, que se debe realizar en cada una, cómo están organizadas y cuales son las interrelaciones entre ellas.

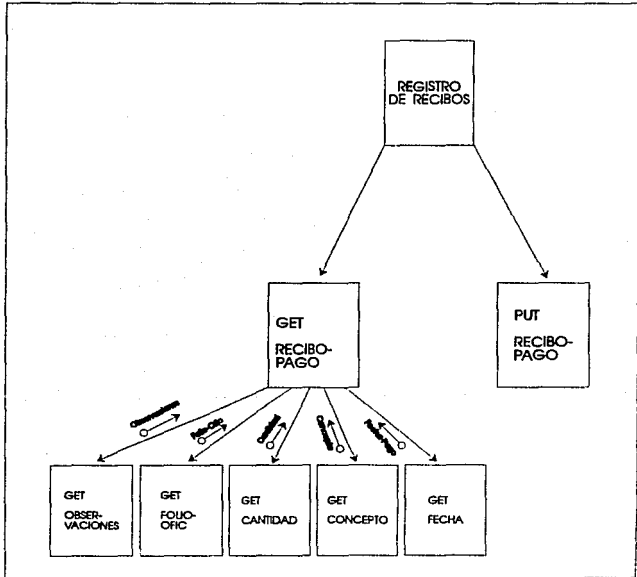


<sup>1</sup> DE MARCO, Tom. "STRUCTURED ANALYSIS AND SYSTEM SPECIFICATION". E.E.U.U.: Prentice-Hall Software Series, 1979. p. 305

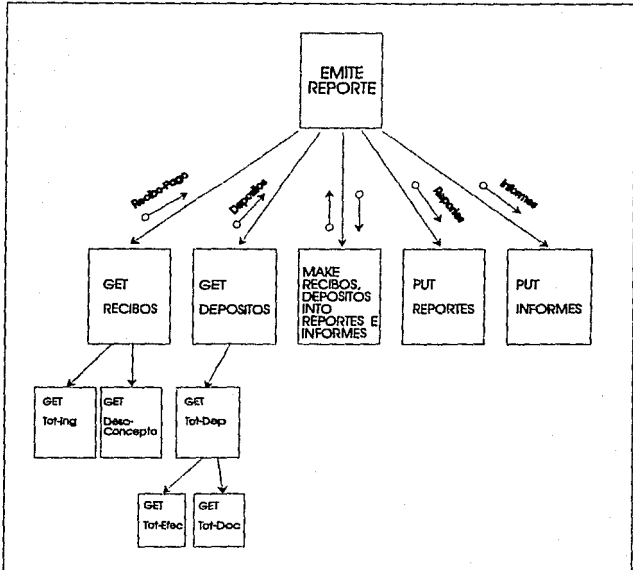


## III.2 DIAGRAMAS DE ESTRUCTURA

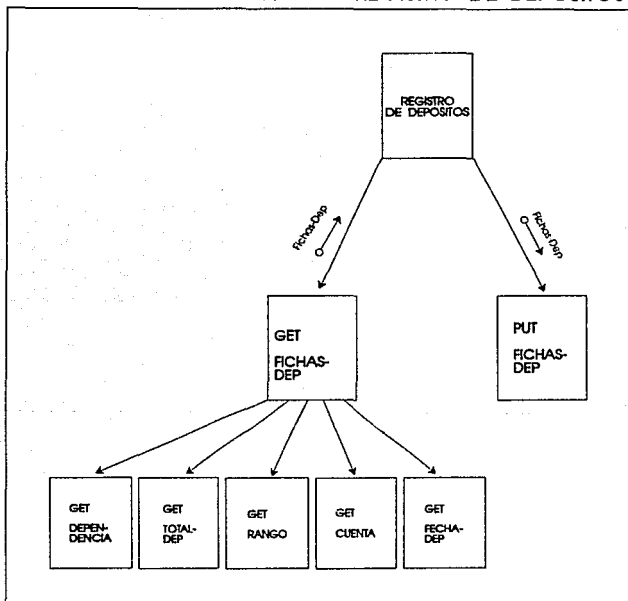
## DIAGRAMA DE ESTRUCTURA 1. REGISTRO DE RECIBOS



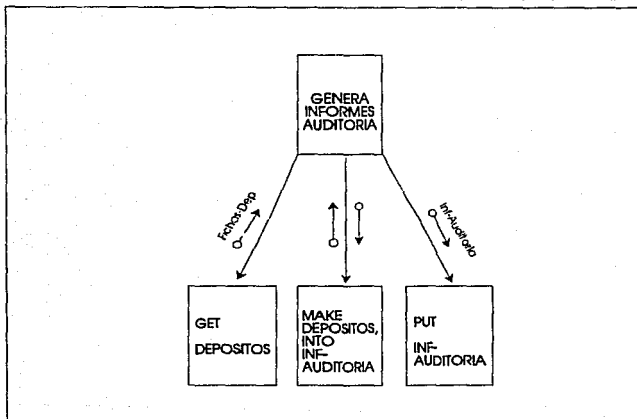
## Diagrama de Estructura 2. Emite Reportes e Informes



## DIAGRAMA DE ESTRUCTURA 3. REGISTRO DE DEPOSITOS



## DIAGRAMA DE ESTRUCTURA 4. GENERA INFORMES AUDITORIA



## IV. IMPLANTACION

### IV.1 PROGRAMACION ORIENTADA A OBJETOS

El avance tecnológico del hardware en los últimos años ha sido de gran impacto en el mundo de la informática, computadores más veloces, más baratas y compactas, pero ¿ que pasa con el software ?, como se puede observar su avance no fue en paralelo con el del hardware, cada día es más caro y el tiempo de desarrollo sigue siendo un factor determinante. Algunos autores han llamado a este fenómeno "La Crisis de Software", pero eso no es todo, tenemos que considerar el costo de mantenimiento del sistema, todo sistema necesita mantenimiento, pero ¿ que el costos de mantenimiento sea aproximadamente el 70 % del costo total de sistema ?, a este respecto Tom De Marco comenta "...No es lógico que se inviertan 2 pesos para la fase de análisis y diseño del sistema y 7 pesos para darle mantenimiento, en condiciones normales, se deberfan invertir 4 pesos para el análisis y diseño del sistema y 2 pesos para darle mantenimiento..."<sup>1</sup>, esta opinión manifiesta que el problema del costo y tiempo de mantenimiento de un sistema se reduce si se enfoca más atención a la fase de análisis y diseño del sistema, utilizando una metodología de desarrollo, que nos ofrece como otra ventaja, la documentación del sistema que ha sido otro factor por el cual se eleva el costo de mantenimiento, muchos sistemas

---

<sup>1</sup> DE MARCO, Tom, " STRUCTURED ANALYSIS AND SYSTEM SPECIFICATION ".E.E.U.U.: Prentice-Hall Software Series, 1979. p. 21

no cuentan con documentación, las personas que lo desarrollaron ya fueron cambiadas de área o trabajan en otra empresa, cuando el sistema requiere mantenimiento se tiene que investigar la estructura del programa, procesos, validaciones, etc. para determinar donde y como se debe aplicar el mantenimiento.

Otro enfoque se refiere a la poca flexibilidad del software para darle mantenimiento, es decir, para realizar cambios en requisitos del usuario, cambio en el formato de datos, cambios en hardware etc. presenta un conflicto para los encargados de mantenimiento por que si se realiza al alguna modificación a algún procedimiento del sistema, esta se refleja en el funcionamiento del mismo.

Como consecuencia de todos los factores descritos anteriormente aparece un nuevo concepto, la Programación Orientada a Objetos (OOP).

" La Programación Orientada a Objetos es un método de diseño que tiene por objetivo establecer técnicas de desarrollo de software para producir sistemas modulares "2.

El concepto básico de la OOP es la de visualizar un programa de computadora como un modelo que representa un sub conjunto del mundo real, su estructura se simplifica si cada una de sus entidades ( u objetos ) del problema que se está modelando, corresponde directamente con un objeto que se pueda manipular internamente en el programa, a este concepto se le denomina abstracción de datos y se basa principalmente en los

---

<sup>2</sup> ARRIETA MARQUEZ, Norberto. " NOTAS PARA EL CURSO PROGRAMACION ORIENTADA A OBJETOS USANDO C++ ", México, p. 9

atributos del objeto y en las operaciones que se pueden realizar con este objeto.

"La Programación Orientada a Objetos mejora la extensibilidad y el control de los programas, al proporcionar medios para la definición de una interfaz más potente."<sup>3</sup>

En lo que a lenguajes orientados a objetos se refiere, existen diversas características con las cuales deben contar, para Bertand Mayer<sup>4</sup> las principales son:

Estructura Modular Basada en Objetos.

Abstracción de Datos.

Manejo Automático de Memoria.

Clases.

Herencia.

Herencia Múltiple.

"La herencia extiende las clases existentes con la adición de nuevos métodos o modifica parte de ellos, permitiendo modificaciones y extensiones seguras al programa "<sup>5</sup>

---

<sup>3</sup> HARTMEIER, Michael. "TURBO C++", México, Computec Editores, S.A. de C.V., p.117.

<sup>4</sup> BERTAND MAYER, Carl. "OBJECT ORIENTED PROGRAM AND EIFFEL", E.E.U.U. Computer Science Textbooks, p. 123

<sup>5</sup> HARTMEIER, Michael. "TURBO C++", México, Computec Editores, S.A. de C.V., p.145.

## IV.2 DEFINICION DE OBJETOS

Es importante tener en cuenta el concepto de objeto dentro de un lenguaje de programación orientado a objetos, Schildt lo define como " una entidad lógica que contiene datos y código que manipula estos datos "<sup>6</sup>, trabajar con C++ nos brinda la oportunidad de utilizar un editor de recursos el cual sirve para crear objetos en pantalla, ventanas, menú de diálogos, botones etc. generando el código en forma automática, así se obtienen objetos y solo se necesita modificar el código para validar el tipo de eventos que va a recibir su interpretación y como va a reaccionar ante tales eventos.

Aquí debemos considerar otra de las ventajas de trabajar con objetos, la herencia, que es la capacidad de crear un objeto derivado de otro, tomando sus atributos y métodos, y la herencia múltiple donde se puede crear un objeto tomando los atributos y métodos de diferentes objetos declarados.

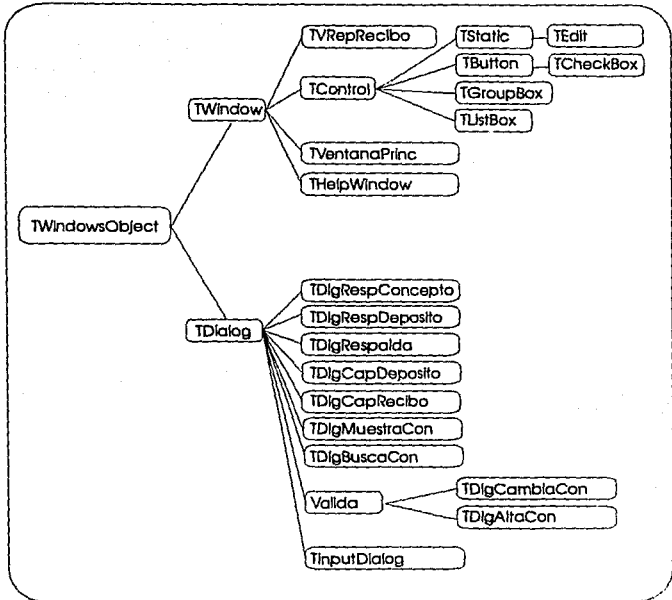
En la siguiente gráfica se muestra la estructura de los objetos que son utilizados por el sistema de recibos de pago de la F.C.A., algunos son creados en forma automática por C++, como los manejadores de ventanas y los de diálogo que toman las características del ambiente de Windows en cual estemos trabajando, a partir de estos se generan los creados para el sistema.

---

<sup>6</sup> SCHILDT, Herbert, Turbo C++ for Windows, Inside & Out, Mac Graw Hill, p. 45



## ARBOL DE OBJETOS



" **TWindowsObject** define el comportamiento fundamental de todos los objetos-ventana del programa como cajas de diálogo, ventanas y controles " <sup>7</sup>

Así **TWindowsObject** es la clase base que unifica el comportamiento de los objetos hijos, además provee funciones para manejar la creación y destrucción de objetos de ventana y procesamiento de mensajes.

**TWindow** es una clase de propósito general cuyas instancias pueden representar ventanas principales, pop-up, scroll-bar, etc. se encarga del manejo de ventanas, de ahí derivamos **TVRepRecibo**, que es la ventana de los reportes del sistema, **TVentanaPrinc** que es la ventana principal de la aplicación.

**TControl** es la clase base para todos los controles, incluyendo botones y cajas de listas, define funciones que crean controles y procesan mensajes a las clases derivadas, a partir de esta clase se deriva **TStatic** que maneja funciones para buscar, almacenar y limpiar texto de un control estático de salida, por ejemplo presentar en pantalla algún reporte, para que estos reporte pudieran ser editables utilizamos **TEdit** que contiene las capacidades de procesamiento de texto para todos los controles editables de windows.

**TButton** representa el control de todos los botones presionables en Windows, de ahí se deriva **TCheckBox** que representa cajas de verificación, por ejemplo para salir del sistema

---

<sup>7</sup> BORLAND, Object Windows for C++, Reference Guide, Borland International Inc. p. 201

se abre una ventana para verificar si realmente queremos terminar de usar la aplicación.

**TGroupBox** controla las características de los grupos de cajas de Windows, para las cajas de listas se utiliza **TListBox**, que contiene la capacidad de elegir de una caja de lista alguna opción.

**TDialog** es la clase base para todos los diálogos que se utilizan en el sistema, para el respaldo de recibos fue creada la clase **TDlgRespRec**, para el respaldo de conceptos el **TDlgRepConcepto** y para el de depósitos **TDlgRespDeposito**, para la captura de los depósitos y recibos se utiliza la clase **TDlgCapDepositos** y **TDlgCapRecibo** respectivamente.

La clases que nos sirve para mostrar los conceptos es la de **TDlgMuestraCon** y para buscar alguno es la **TDlgBuscaCon**.

La clase **Valida** contiene las funciones para validar si existe un concepto así que derivamos de esta clase **TDlgCambiaCon** para modificar conceptos y **TDlgAltaCon** para poder dar de alta otro concepto.

La última clase que se muestra es la de **TInputDialog** que es formato general para las ventanas que solicitan datos al usuario.

### IV.3 ESTRUCTURA DE LOS ARCHIVOS DE LA BASE DE DATOS

El sistema utiliza 3 archivos para el almacenamiento de información.

- a) **Conceptos**, en el cual se van a guardar todos los conceptos por los cuales se pueden realizar pagos en la caja, la estructura de este archivo es de tres campos, clave de concepto con una longitud de 2 dígitos, este campo representa el campo llave, concepto, para describir los conceptos con una longitud de 30 caracteres y el costo definido con 8 alfanuméricos, ( 2,500.<sup>00</sup> )

En el siguiente esquema se muestra la estructura de la base de datos indicando con una **D** a los dígitos, una **C** para los caracteres y una **A** para los alfanuméricos.

Clave	Concepto	Costo
2 D	30 C	8 A

- b) **Recibos**, archivo que contiene todos los datos de los recibos de pago de los cuales se realizaron cobros, con los siguientes campos, Número de Folio que es el campo llave de 7 dígitos, Clave del Concepto 2 dígitos, Costo del concepto de 8 alfanuméricos, la fecha de 10 alfanuméricos con el formato mm/dd/aaaa, cantidad total a pagar de 8 alfanuméricos, efectivo 1 caracter, turno de un caracter, número de cheque de 9 dígitos y nombre del banco de 10 caracteres.

Folio	Clave	Costo	Fecha	Total Efectivo	Turno	Cheque	Banco	
7D	2D	8A	10A	8A	1C	1C	9D	10C

c) **Depósitos**, este archivo está compuesto de los campos Ficha de Depósito campo llave de 7 dígitos, fecha de depósito con 8 caracteres, efectivo depositado 8 alfanuméricos, depósito en documentos 8 alfanuméricos, total depositado de 9 alfanuméricos, número de cheque de 9 dígitos y folio de los recibos que ampara dicho depósito.

Ficha	Fecha	Efectivo	Documentos	Total	Cheque	Folio1	Folio2
7D	2D	8A	8A	9A	9D	7D	7D

## CONCLUSIONES

Para la fase de análisis y diseño del sistema no pude contar con un CASE que me diera la facilidad de crear mis diagramas y corregirlos en forma automática así que utilicé COREL DRAW versión 2.0 para hacerlos una y otra vez hasta la versión final.

El uso de la metodología de análisis y diseño estructurado me permitió seguir una guía para el desarrollo del sistema, sin embargo consideré importante incluir procedimientos manuales ya que ésta no los contempla y así demostrar que un analista se va formando con un criterio propio, con capacidad para hacer flexible una metodología de desarrollo y utilizarla solo como una herramienta, un puente de comunicación con el usuario, que sin ser un experto, es capaz de entender el sistema a través de un modelo solo explicándole cuando se trata de un proceso, un flujo de datos de entrada o salida y los datos que se almacenan entre los procesos.

Utilicé una metodología de análisis y diseño estructurado y realicé la implantación de un sistema orientado a objetos, honestamente nunca pensé utilizar la Programación Orientada a Objetos ( OOP ), así que no utilicé una metodología orientada a objetos, sin embargo al finalizar la etapa de análisis y diseño surgió la oportunidad de entrar al mundo de los objetos y realmente fue sorprendente los resultados que se obtenían, así que decidí crear el sistema orientado a objetos utilizando el análisis y diseño que tenía listo y una vez más pude comprobar que las metodologías de desarrollo son solo una herramienta, hay

mucha gente que programa en forma estructurada o en objetos que nunca utiliza una metodología formal y sin embargo realiza sistemas, por que digo formal, por que toda la gente que desarrollamos sistemas utilizamos una metodología, en la gran mayoría de los casos, una metodología basada en la experiencia propia, no en la experiencia de un Jackson, Yourdon, Tom de Marco, etc. Así que se utilizó Borland C++ para windows que era el software que tenía en ese momento y con la asesoría de un par de expertos en la materia pude crear el sistema. Debo aclarar que no pude contar con un manejador de bases de datos para mis archivos y tuve que utilizar los archivos de C++ para mis bases de datos, no es lo más recomendable ya que no es muy sólida la base, no tiene algoritmos eficientes de búsqueda, etc. sin embargo una de las mejoras al sistema es la de exportar los archivos a un manejador de base de datos, por el momento no lo tengo.

Para la impresión de los reportes, no utilicé las características de Windows, el código necesario para que el sistema utilizara el controlador de impresoras de windows no lo comprendí totalmente, así que utilicé la impresión normal de C++ para imprimir los reportes, otra alternativa que utilicé fue la crear ventanas editables en los reportes, así se pueden copiar, cortar y pegar los reportes al portapapeles de Windows (Clipboard) y mandarlos a cualquier procesador del ambiente windows, Word for Windows, Write, Page Maker, etc. con solo marcar el reporte, copiarlo y pegarlo en cualquier aplicación de las antes descritas, incluso tener un estilo definido con el logo de la Universidad y de la Facultad, de esta forma puedo mandar a imprimir directamente los reportes o exportarlos a otra aplicación.

## BIBLIOGRAFIA

DE MARCO, Tom. " STRUCTURED ANALYSIS AND SYSTEM SPECIFICATION ".E.E.U.U.: Prentice-Hall Software Series, 1979.

YOURDON, Edward. " STRUCTURED ANALYSIS WORKSHOP ", E.E.U.U.: Lecture Notes. 1990, Yourdon Inc.

ARRIETA MARQUEZ, Norberto. " NOTAS PARA EL CURSO PROGRAMACION ORIENTADA A OBJETOS USANDO C++ ", México.

HARTMEIER, Michael. " TURBO C++ ", México, Computec Editores, S.A. de C.V.

BERTAND MAYER, Carl. " OBJECT ORIENTED PROGRAM AND EIFFEL ",E.E.U.U. Computer Science Textbooks.

SCHILD, Herbert, Turbo C++ for Windows, Inside & Out. Mac Graw Hill.

BORLAND, Object Windows for C++, Reference Guide, Borland International Inc.

MICROSOFT WINDOWS 3.1, Guía del Usuario, Microsoft Corporation, 1990-1992.

MICROSOFT Word 5.0, User Guide, Microsoft Corporation, 1988.

OFICINA INTERNACIONAL DEL TRABAJO, Introducción al Estudio del Trabajo. Tercera Edición, Ed. Limusa. México, 1987



## **ANEXOS**

## **A) REQUERIMIENTOS E INSTALACION DEL SISTEMA**

### **REQUERIMIENTOS DE HARDWARE**

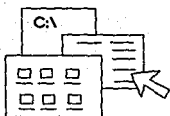
- Computadora Personal con micro procesador modelo 386 en adelante.
- 2 Mb. de Memoria RAM.
- Capacidad de almacenamiento de 3 Mb.  
*(Exclusivamente para el sistema)*
- Impresora, matriz de punto ó laser.
- Se recomienda el uso de Mouse para facilitar el uso del sistema.

### **REQUERIMIENTOS DE SOFTWARE**

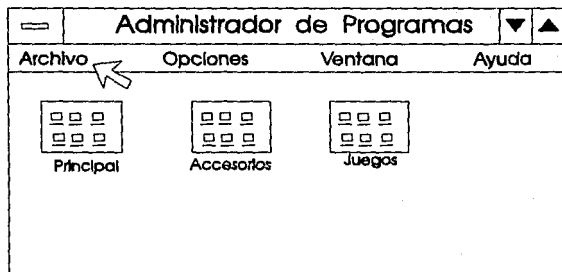
- Sistema Operativo MS-DOS, versión 4.01 en adelante.
- MicroSoft Windows versión 3.1.

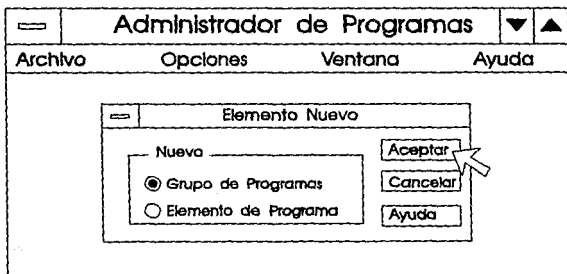
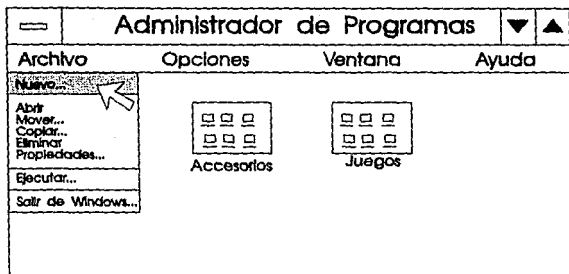
- **Sistema de Recibos de Pago.**
  
- **Se recomienda tener instalado algún procesador de texto para el ambiente windows ( Word for Windows, Page Maker, Write, etc. ), al ser editables en pantalla los reportes facilita el utilizarlos en algún documento que se requieran.**

Para instalar el sistema es necesario que tenga cargado Windows versión 3.1 previamente, inserte el disco flexible del sistema de recibos de pago y teclee RECIBOS.BAT, esto creará el directorio donde se almacenará los archivos, para crear el grupo de trabajo dentro de windows siga las siguientes gráficas.



Administrador de  
Programas



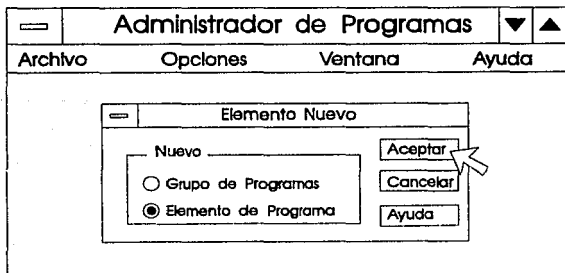
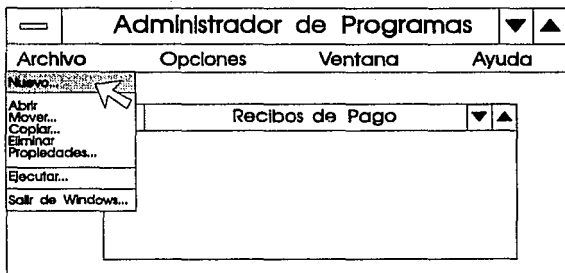


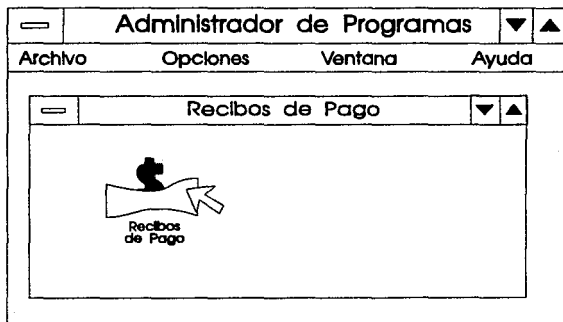
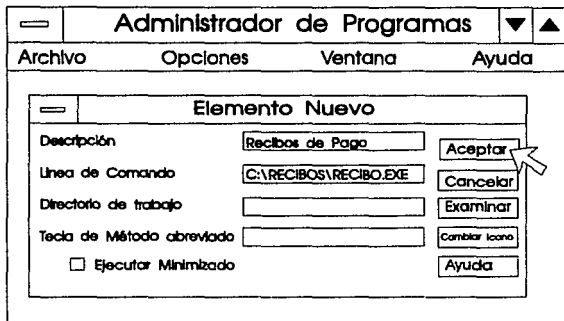
1. Teclar

Elemento Propiedades del Grupo de Programa	
Descripción:	<input type="text" value="Recibos de Pago"/>
Archivo de Grupos:	<input type="text"/>
	<input type="button" value="Aceptar"/>
	<input type="button" value="Cancelar"/>
	<input type="button" value="Ayuda"/>

2. Click

Administrador de Programas							
Archivo	Opciones	Ventana	Ayuda				
<table border="1"><thead><tr><th colspan="2">Recibos de Pago</th></tr></thead><tbody><tr><td colspan="2"> </td></tr></tbody></table>				Recibos de Pago			
Recibos de Pago							







**B) REPORTES DEL SISTEMA**

FACULTAD DE CONTADURIA Y ADMINISTRACION

## REPORTE DIARIO DE RECIBOS

FECHA: 19/03/1993

CONCEPTO	CANTIDAD	
Curso de Lotus	2,650.00	
Seminario de Titulación	3,250.00	
Recuerdos de la F.C.A	800.00	
Revista Emprendedores	630.00	
Total:	N\$ 4,680.00	Efectivo:2,680.00 Cheque: 2,000.00

## FACULTAD DE CONTADURIA Y ADMINISTRACION

## REPORTE MENSUAL DE RECIBOS

MES: ABRIL

CONCEPTO	CANTIDAD	
Curso de Lotus	12,650.00	
Seminario de Titulación	21,250.00	
Recuerdos de la F.C.A	3,800.00	
Revista Emprendedores	1,630.00	
Clave Computadoras	1,350.00	
Fondo Editorial	13,345.50	
Total:	N\$ 54,025.50	Efectivo:37,723.00 Cheque:16,302.50

## FACULTAD DE CONTADURIA Y ADMINISTRACION

## REPORTE MENSUAL DE DEPOSITOS

FECHA: 03/93

Ficha	Del No. de Folio	Al No. de Folio	Efectivo	Documentos	Total Depositado
1928374	22345673	3746532	4,575.00	18,372.00	22,947.00
2128377	38273642	4183737	5,934.50	17,838.00	23,772.50
2232423	42422342	4324244	3,233.00	19,324.00	22,557.00
Totales:			13,742.50	55,534.00	69,276.50

### **C) PROCEDIMIENTOS MANUALES**

Los procedimientos manuales que utiliza el sistema se identifican en el diagrama 5, " Verifica Folio de los Recibos " y en el diagrama 6, " Realiza Corte de Caja ".

Cuando rectoría, a solicitud del sistema envía el conjunto de recibos en blanco para que la caja de la F.C.A. pueda realizar los cobros, anexa una relación donde informa la cantidad de recibos enviados y la secuencia de los números de folio de los mismos, cuando se entregan los recibos es necesario que se verifique que no falte ningún recibo, en caso de que falte alguno se debe registrar indicando el folio del recibo que falte, una vez registrado se realiza una notificación a Rectoría para comunicarle el faltante.

En el Diagrama 6 se muestra el procedimiento manual para el corte de caja, el usuario consulta el reporte diario de recibos y debe comparar el total contra el dinero que existe físicamente en la caja al realizar el corte diario de caja, de esta forma se puede saber si falta dinero en la caja ó no se registró alguna operación, en este caso se deberá de revisar los recibos físicamente contra el reporte para encontrar la diferencia.

## Cursograma 1.

Actividad: Verifica folio de los Recibos		○ Operación ⇒ Transporte □ Inspección D Espera ▽ Almacenamiento				
Descripción	Símbolo					Observaciones
	○	⇒	□	D	▽	
El sistema solicita recibos de pago	●					
Rectoría envía recibos de pago		●				
Se reciben los comprobantes de pago	●					
Verifica la relación de recibos			●			
Registra el total de recibos	●					
Verifica los números de folio de cada			●			
Notifica a rectoría los recibos faltantes	●					

## Cursograma 2.

Actividad: Realiza corte de caja		○ Operación ⇒ Transporte □ Inspección D Espera ▽ Almacenamiento				
Descripción	Símbolo					Observaciones
	○	⇒	□	D	▽	
El usuario consulta el reporte diario de recibos	●					
El usuario cuenta el dinero que ingresa al día	●					
El usuario verifica el reporte de ingresos contra el efectivo			●			
Almacena los recibos					●	
Notifica resultado del arqueo	●					

## D) LISTADO DEL PROGRAMA

A continuación se lista el listado principal del código utilizado para el sistema de recibos de pago.

```
//Headers de windows
#include <owl.h>
#include <dialog.h>
#include <edit.h>
#include <inputdia.h>
#include <listbox.h>
#include <mdi.h>
#include <window.h>
//Headers de la aplicación
#include "recibo.h" // declaraciones de recursos
#include "reciboap.h" // declaraciones de TVentanaPrinc y TRecibo,
// son las clases principales de la aplicación
#include "caprec.h" // declaraciones de las funciones para captura de recibos
#include "capdep.h"
#include "errores.h"
#include "ayuda.h"
#include "respalda.h"
#include "reporte.h"
FVentana PRecibo=NULL;
int func_comp(const void *a, const void *b);
/*
Implementaciones de TVentanaPrinc:
*/
void TVentanaPrinc::GetWindowClass(WNDCLASS& wc){
TWindow::GetWindowClass( wc );
wc.lpszMenuName = "COMMANDS";
wc.hIcon=LoadIcon(wc.hInstance,"IC_RECIBO");
wc.style=CS_HREDRAW | CS_VREDRAW;
```

```

    hinst = wc.hInstance;
}
void TVentanaPrinc::SetupWindow()
{
    TWindow::SetupWindow();
    HMENU hMenu = GetMenu(HWindow);
    EnableMenuItem(hMenu, CM_EDITCOPY ,
MF_BYCOMMAND|MF_GRAYED);
    EnableMenuItem(hMenu, CM_EDITCUT , MF_BYCOMMAND|MF_GRAYED);
    EnableMenuItem(hMenu, CM_EDITUNDO ,
MF_BYCOMMAND|MF_GRAYED);
    EnableMenuItem(hMenu, CM_EDITPASTE ,
MF_BYCOMMAND|MF_GRAYED);
    EnableMenuItem(hMenu, CM_EDITDELETE,
MF_BYCOMMAND|MF_GRAYED);
    // TMessage x;
    // About(x);
}
BOOL TVentanaPrinc::CanClose()
{
    if(!sa){
        MessageBeep(MB_ICONQUESTION);
        return MessageBox(HWindow, "¿Realmente desea salir?",
            "Esto termina su aplicación",
            MB_YESNO | MB_ICONQUESTION) != IDNO;
    }
    return TRUE;
}
void TVentanaPrinc::Recibos(RTMessage){
    FILE *salida, *sal, *tmp;
    date fecha;
    char nom[12];
    char claves[100] | NFOLIO;
    getdate(&fecha);
    strcpy(EstrucTransf.Efectivo, "");
    strcpy(EstrucTransf.Cheque, "");
    strcpy(EstrucTransf.Banco, "");
    strcpy(EstrucTransf.Folio, "");
    strcpy(EstrucTransf.Clave1, "");
    if (GetModule()-> ExecDialog(
        new TDlgCapRecibo(this, "CAPTURARECIBO")) == IDOK ){

```



```

if((sal = fopen("temp.tmp","r")) == NULL){
    ErrorMenS(HWindow,"No se pudo abrir archivo temporal.,"ERROR de E/S");
    return;
}
fscanf(sal,"%s",&EstrucTransf.Efectivo);
fscanf(sal,"%s",&EstrucTransf.Turno);
fclose(sal);
remove("temp.tmp");
if(strcmp(EstrucTransf.Efectivo,"S") == 0){
    strcpy(EstrucTransf.Cheque,"");
    strcpy(EstrucTransf.Banco,"");
}
if((fecha.da_mon < 10) sprintf(nom,"rec0%i%i.dat",fecha.da_mon,(fecha.da_year-
1900));
else sprintf(nom,"rec%i%i.dat",fecha.da_mon,(fecha.da_year-1900));
if((salida = fopen(nom,"a+b")) != NULL){
    fwrite(&EstrucTransf,sizeof(EstrucTransf),1,salida);
    fclose(salida);
}else{
    ErrorMenS(HWindow,"No se pudo abrir el
archivo\nCONCEPTO.DAT,"ERROR de E/S");
    return;
}
void TVentanaPrinc::AltaConc(RTMessage){
    char claves[100][MAXLONCLA];
    FILE *salida,"tmp";
    if((GetModule()->ExecDialog(new
TDlgAltaCon(this,"ALTACONCEPTO")) == IDOK){
        salida = fopen("concepto.dat","a+b");
        if(salida == NULL){
            ErrorMenS(HWindow,"No se pudo abrir el
archivo\nCONCEPTO.DAT,"ERROR de E/S");
            return;
        }
        fwrite(&EstrucConcepto,sizeof(EstrucConcepto),1,salida);
        fclose(salida);
        if((salida = fopen("concepto.dat","rb")) == NULL){
            ErrorMenS(HWindow,"No se pudo abrir el
archivo\nCONCEPTO.DAT,"ERROR de E/S");

```

```

    return;
} else {
    int i=0,j;
    do {
        fread(&EstrucConcepto,sizeof(EstrucConcepto),1,salida);
        strcpy(claves[i],EstrucConcepto.Clave);
        i++;
    } while(!feof(salida));
    i--;
    qsort((void *)claves,i,sizeof(claves[0]),func_comp);
    if((tmp=fopen("temporal.dat","wb"))==NULL){
        ErrorMens(HWindow,"No se pudo abrir archivo.,"ERROR de E/S");
        return;
    } else {
        j=0;
        do {
            rewind(salida);
            do {
                fread(&EstrucConcepto,sizeof(EstrucConcepto),1,salida);
            } while( (strcmp(claves[j],EstrucConcepto.Clave)!=0);
            fwrite(&EstrucConcepto,sizeof(EstrucConcepto),1,tmp);
            j++;
        } while(j<i);
        fclose(salida);
        fclose(tmp);
        remove("concepto.dat");
        rename("temporal.dat","concepto.dat");
    }
}
}

void TVentanaPrinc::BorraConc(RTMessage){
    char ConInfo[MAXLONCLA + MAXLONCON + MAXLONCOS + 55];
    char Texto[MAXLONCLA];
    FILE *entrada;
    int enc=0;
    TEstrucConcepto EstConc;
    strcpy(Texto,"00");
    if ( GetApplication()->ExecDialog(new TInputDialog(this, "Borrar Concepto",

```

```

"Clave a borrar:",Texto,sizeof(Texto))) == IDOK ){
    if((entrada = fopen("concepto.dat","rb")) == NULL){
        ErrorMenS(HWindow,"No se pudo abrir el
archivo\nCONCEPTO.DAT","ERROR de E/S");
        return;
    }else{
        fread(&EstConc,sizeof(EstConc),1,entrada);
        while(!feof(entrada) && !enc)
            if( strcmp(EstConc.Clave,Texto) == 0){
                enc = 1;
                strcpy(ConInfo, "Clave  : ");
                strcat(ConInfo, EstConc.Clave);
                strcat(ConInfo, "\nConcepto : ");
                strcat(ConInfo, EstConc.Concepto);
                strcat(ConInfo, "\nCosto  : ");
                strcat(ConInfo, EstConc.Costo);
                strcat(ConInfo, "\n\nEstá seguro de borrar este registro?");
                fclose(entrada);
                if(MessageBox(HWindow, ConInfo,"Información almacenada",
MB_YESNO) == IDYES){
                    FILE *tmp;
                    entrada = fopen("concepto.dat","rb");
                    if( (tmp = fopen("tmp.dat","wb")) != NULL){
                        fread(&EstConc,sizeof(EstConc),1,entrada);
                        while(!feof(entrada)){
                            if(strcmp(EstConc.Clave,Texto) != 0)
                                fwrite(&EstConc,sizeof(EstConc),1,tmp);
                            fread(&EstConc,sizeof(EstConc),1,entrada);
                        }
                        fclose(entrada);
                        fclose(tmp);
                        remove("concepto.dat");
                        rename("tmp.dat","concepto.dat");
                    }else{
                        ErrorMenS(HWindow,"No se pudo abrir el archivo.","ERROR de E/S");
                        return;
                    }
                }else return;
            }else fread(&EstConc,sizeof(EstConc),1,entrada);

```

```

fclose(entrada);
if(!enc) ErrorMenS(HWindow,"No se encontr la clave","ERROR");
}
}

void TVentanaPrinc::BuscaConc(RTMessage){
char Texto[MAXLONCLA];
FILE *entrada;
int enc=0;
strcpy(Texto,"00");
if ( GetApplication()-> ExecDialog(new TInputDialog(this, "Buscar Concepto",
"Clave a buscar:",Texto,sizeof(Texto))) == IDOK){
if((entrada = fopen("concepto.dat","rb")) == NULL){
ErrorMenS(HWindow,"No se pudo abrir el
archivo\nCONCEPTO.DAT","ERROR");
return;
}
}else return;
fread(&EstrucConcepto,sizeof(EstrucConcepto),1,entrada);
while(!feof(entrada) && !enc)
if( strcmp(EstrucConcepto.Clave,Texto) == 0) // Encontr la clave
enc = 1;
else fread(&EstrucConcepto,sizeof(EstrucConcepto),1,entrada);
fclose(entrada);
if(enc) GetModule()-> ExecDialog(new
TDlgBuscaCon(this,"CONSULTA CONCEPTO");
else ErrorMenS(HWindow,"No se encontr la clave","ERROR");
}

void TVentanaPrinc::CambConc(RTMessage){
FILE *entrada, *salida;
int enc=0;
char Texto[MAXLONCLA];
TEstrucConcepto Conc;
strcpy(Texto,"00");
if ( GetApplication()-> ExecDialog(new TInputDialog(this, "Cambiar Concepto",
"Clave del concepto a cambiar:",Texto,sizeof(Texto))) == IDOK ){
if((entrada = fopen("concepto.dat","rb")) == NULL){
ErrorMenS(HWindow,"No se pudo abrir el
archivo\nCONCEPTO.DAT","ERROR");

```

```

return;
}
fread(&EstrucConcepto,sizeof(EstrucConcepto),1,entrada);
while(!feof(entrada) && !enc)
if( strcmp(EstrucConcepto.Clave,Texto) = 0) // Encontr la clave
enc = 1;
else fread(&EstrucConcepto,sizeof(EstrucConcepto),1,entrada);
}else return;
fclose(entrada);
//AQUI DEBE DE CAMBIAR EL CONCEPTO EN EL ARCHIVO
//BUSCANDO EL REGISTRO Y REESCRIBIENDO.
if(enc){
if(GetModule()-> ExecDialog(new
TDlgCambiaCon(this,"CAMBIA CONCEPTO")) = IDOK);
if((salida = fopen("tmp.dat","wb")) != NULL){
if( (entrada = fopen("concepto.dat","rb")) = NULL){
ErrorMenS(HWindow,"No se pudo abrir el archivo\nCONCEPTO.DAT","ERROR
de E/S");
return;
}
fread(&Conc,sizeof(Conc),1,entrada);
while(!feof(entrada)){
if( strcmp(Conc.Clave,EstrucConcepto.Clave) != 0)
fwrite(&Conc,sizeof(Conc),1,salida);
else
fwrite(&EstrucConcepto,sizeof(EstrucConcepto),1,salida);
fread(&Conc,sizeof(Conc),1,entrada);
}
fclose(entrada);
fclose(salida);
remove("concepto.dat");
rename("tmp.dat","concepto.dat");
}else{
ErrorMenS(HWindow,"No se pudo abrir el archivo","ERROR de E/S");
return;
}
}else ErrorMenS(HWindow,"No se encontr la clave","ERROR");
}
void TVentanaPrinc::MostrConc(RTMessage){

```

```

if(GetModule()->ExecDialog(new
TDlgMuestraCon(this,"LISTA CONCEPTO")) == IDOK);
}
void TVentanaPrinc::Deposito(RTMessage){
char fichas[100][NFICHA];
FILE *salida,*tmp;
if(GetModule()->ExecDialog(new
TDlgCapDeposito(this,"CAPTURADEPOSITO")) == IDOK){
if((salida = fopen("depo.dat","a+b")) == NULL){
ErrorMenS(HWindow,"No se pudo abrir el archivo\ndePO.DAT","ERROR de
E/S");
return;
}
else{
fwrite(&EstrucDeposito,sizeof(EstrucDeposito),1,salida);
fclose(salida);
if((salida = fopen("depo.dat","rb")) == NULL){
ErrorMenS(HWindow,"No se pudo abrir el archivo\ndePO.DAT","ERROR de
E/S");
return;
}
else{
int i=0,j;
do{
fread(&EstrucDeposito,sizeof(EstrucDeposito),1,salida);
strcpy(fichas[i],EstrucDeposito.Ficha);
i++;
}while(!feof(salida));
i--;
qsort((void *)fichas,i,sizeof(fichas[0]),func_comp);
if((tmp = fopen("temporal.dat","wb")) == NULL){
ErrorMenS(HWindow,"No se pudo abrir archivo. ","ERROR de E/S");
return;
}
else{
j=0;
do{
rewind(salida);
do{
fread(&EstrucDeposito,sizeof(EstrucDeposito),1,salida);
}while( (strcmp(fichas[j],EstrucDeposito.Ficha))!=0);
fwrite(&EstrucDeposito,sizeof(EstrucDeposito),1,tmp);

```

```

        j++;
    }while(j<i);
    fclose(salida);
    fclose(tmp);
    remove("depo.dat");
    rename("temporal.dat","depo.dat");
    }
    }
}
void TVentanaPrinc::Respalda(RTMessage){
    GetModule()->ExecDialog(new TDlgRespalda(this,"RESPALDO"));
}
void TVentanaPrinc::RespDepo(RTMessage){
    GetModule()->ExecDialog(new TDlgRespDeposito(this,"RESPDEPOSITO"));
}
void TVentanaPrinc::RespConc(RTMessage){
    GetModule()->ExecDialog(new TDlgRespConcepto(this,"RESPCONCEPTO"));
}
void TVentanaPrinc::RepRecibo(RTMessage){
    char Texto[12]="";
    if(PRecibo!=NULL){
        PRecibo->ShutDownWindow();
        PRecibo=NULL;
    }
    if ( GetApplication()->ExecDialog(new TInputDialog(this,"Reporte Diario de
Recibos",
        "Dsa, mes y A±o del reporte (dd/mm/aaaa);",Texto,sizeof(Texto))) == IDOK )
        if ( Texto[0] && (Texto[2]='/' && Texto[5]='/') && strlen(Texto)==10 ){
            MuestraVentana(this,"Reporte Diario de Recibos",1,Texto);
            HMENU hMenu = GetMenu(HWindow);
            EnableMenuItem(hMenu,CM_EDITCOPY ,
MF_BYCOMMAND|MF_ENABLED);
            EnableMenuItem(hMenu,CM_EDITCUT ,
MF_BYCOMMAND|MF_ENABLED);
            EnableMenuItem(hMenu,CM_EDITUNDO ,
MF_BYCOMMAND|MF_ENABLED);
            EnableMenuItem(hMenu,CM_EDITPASTE ,
MF_BYCOMMAND|MF_ENABLED);

```

```

        EnableMenuItem(hMenu,CM_EDITDELETE,
MF_BYCOMMAND|MF_ENABLED);
    )else ErrorMessageS(HWindow,"Mal formato de fecha, debe ser:\r\n01/01/1999.",
        "Error de Entrada");
    }
void TVentanaPrinc::RepMRecibo(RTMessage){
    char Texto[5]="";
    if(PRecibo!=NULL){
        PRecibo->CloseWindow();
        PRecibo=NULL;
    }
    if ( GetApplication()->ExecDialog(new TInputDialog(this,"Reporte Mensual de
Recibos",
        "Mes y A ± o del reporte (mmaa):",Texto,sizeof(Texto))) == IDOK )
        if(Texto[0]){
            MuestraVentana(this,"Reporte Mensual de Recibos",2,Texto);
            HMENU hMenu = GetMenu(HWindow);
            EnableMenuItem(hMenu,CM_EDITCOPY ,
MF_BYCOMMAND|MF_ENABLED);
            EnableMenuItem(hMenu,CM_EDITCUT ,
MF_BYCOMMAND|MF_ENABLED);
            EnableMenuItem(hMenu,CM_EDITUNDO ,
MF_BYCOMMAND|MF_ENABLED);
            EnableMenuItem(hMenu,CM_EDITPASTE ,
MF_BYCOMMAND|MF_ENABLED);
            EnableMenuItem(hMenu,CM_EDITDELETE,
MF_BYCOMMAND|MF_ENABLED);
        }
    }
void TVentanaPrinc::RepDepo(RTMessage){
    char Texto[5]="";
    if(PRecibo!=NULL){
        PRecibo->CloseWindow();
        PRecibo=NULL;
    }
    if ( GetApplication()->ExecDialog(new TInputDialog(this,"Reporte Mensual de
Dep sites",
        "Mes y A ± o del reporte (mmaa):",Texto,sizeof(Texto))) == IDOK )
        if(Texto[0]){

```



```

MuestraVentana(this,"Reporte Mensual de Dep sitios",3,Texto);
HMENU hMenu = GetMenu(HWindow);
EnableMenuItem(hMenu,CM_EDITCOPY ,
MF_BYCOMMAND|MF_ENABLED);
EnableMenuItem(hMenu,CM_EDITCUT ,
MF_BYCOMMAND|MF_ENABLED);
EnableMenuItem(hMenu,CM_EDITUNDO ,
MF_BYCOMMAND|MF_ENABLED);
EnableMenuItem(hMenu,CM_EDITPASTE ,
MF_BYCOMMAND|MF_ENABLED);
EnableMenuItem(hMenu,CM_EDITDELETE,
MF_BYCOMMAND|MF_ENABLED);
}
}
void TVentanaPrinc::MuestraVentana(PWindowsObject AParent,LPSTR Titulo,int
V,char *T){
if(V == 1) PRecibo = PVentana(GetApplication()->MakeWindow(
new TReporte(AParent,Titulo,1,T));
if(V == 2) PRecibo = PVentana(GetApplication()->MakeWindow(
new TReporte(AParent,Titulo,2,T));
if(V == 3) PRecibo = PVentana(GetApplication()->MakeWindow(
new TReporte(AParent,Titulo,3,T));
}
void TVentanaPrinc::CMeditUndo(RTMessage){
HMENU hMenu;
hMenu = GetMenu(HWindow);
if(PRecibo->Edit->CanUndo() == FALSE){
EnableMenuItem(hMenu,CM_EDITUNDO,MF_BYCOMMAND|MF_GRAYED);
MessageBeep(1);
}else{
EnableMenuItem(hMenu,CM_EDITUNDO,MF_BYCOMMAND|MF_ENABLED);
Precibo->Edit->Undo();
}
}
void TVentanaPrinc::CMeditCopy(RTMessage){
Precibo->Edit->Copy();
}
}

```

```

void TVentanaPrinc::CMEditCut(RTMessage){
    PRcibo->Edit->Cut();
}
void TVentanaPrinc::CMEditDelete(RTMessage){
    PRcibo->Edit->DeleteSelection();
}
void TVentanaPrinc::CMEditPaste(RTMessage){
    PRcibo->Edit->Paste();
}
void TVentanaPrinc::Ayuda(RTMessage){
}
void TVentanaPrinc::About(RTMessage){
    GetApplication()->ExecDialog(new TDialog(this, "ABOUT"));
}
void TVentanaPrinc::Salir(RTMessage){
    sal=1;
    MessageBeep(MB_ICONQUESTION);
    if( (MessageBox(HWindow, "¿Realmente desea salir?",
        "Esto termina su aplicaci n",
        MB_YESNO | MB_ICONQUESTION)) == IDYES ){
        if(PRcibo!=NULL)
            PRcibo->CloseWindow();
        CloseWindow();
    }else sal=0;
}
/*
Implementaciones de TRecibo:
*/
void TRecibo::InitInstance(){
    TApplication::InitInstance();
    HACcTable = LoadAccelerators(hInstance, "TECLAS");
};
void TRecibo::InitMainWindow( void ){
    MainWindow = new TVentanaPrinc(NULL, "Control de Recibos");
}
// Programa principal
int PASCAL WinMain(HANDLE hInstance, HANDLE hPrevInstance,
    LPSTR lpCmd, int nCmdShow){
    TRecibo Recibo("Recibo", hInstance, hPrevInstance, lpCmd, nCmdShow);
}

```

```
Recibo.Run();  
return(Recibo.Status);  
}  
int func_comp(const void *a, const void *b){  
return(strcmp((char*)a,(char *)b));  
}
```

12  
2º

**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO  
FACULTAD DE CONTADURIA Y ADMINISTRACION**

**ANALISIS DE LAS HAND-HELDS COMPUTERS**

**SEMINARIO DE INVESTIGACION INFORMATICA  
QUE EN OPCION AL GRADO DE**

**LICENCIADO EN INFORMATICA**

**PRESENTA**

**JOSE CARLOS MORALES ALVAREZ**

**DIRECTOR DEL SEMINARIO**

**MBA. JOSE ANTONIO ECHENIQUE GARCIA**

**México, D.F.**

**1993**

**TESIS CON  
FALLA DE ORIGEN**

## **INDICE**

---

<b>INTRODUCCION.....</b>	<b>3</b>
<b>I.- LA NUEVA TECNOLOGIA Y SU DESARROLLO.</b>	
I.I.....Antecedentes.....	5
I.II.....Generalidades.....	7
I.III.....Factores de Estudio.....	8
I.IV.....Productividad y Desarrollo.....	12
<b>II.- ESTRUCTURA DE LAS HAND-HELDS COMPUTERS.</b>	
II.I.....Conceptos Generales.....	16
II.II.....Especificaciones Técnicas.....	17
II.III.....Sistema Operativo (Estructura y Comandos).....	29
<b>III.- LENGUAJE DE PROGRAMACION. OPL/g.</b>	
III.I.....Conceptos Generales.....	40
III.II.....Experiencia en OPL/g.....	41
III.III.....Programación en OPL/g.....	43
III.IV.....Tipo de Variables.....	47
III.V.....Operadores Lógicos.....	51
III.VI.....Manejo de Archivos ODB.....	53
<b>IV.- SISTEMA STORE-CHECK (HHC).</b>	
IV.I.....Objetivo del Servicio.....	59
IV.II.....Beneficios.....	59
IV.III.....Metodología.....	60
IV.IV.....Flujo de Actividades.....	67
IV.V.....Flujo de Transferencia.....	69
IV.VI.....Flujo del Programa en OPL/g.....	73
IV.VII.....Operación del sistema Store-Check para HC-110.....	79
<b>CONCLUSIONES.....</b>	<b>105</b>
<b>BIBLIOGRAFIA.....</b>	<b>108</b>

## **INTRODUCCION**

Este estudio específico sobre el equipo denominado hand-held computers (Computadora portátil para mano. HHC). contiene un panorama detallado para la utilización del mismo; su descripción y componentes, así como también, de los lenguajes utilizados para la programación y de los sistemas operativos empleados para su operación, todo esto tomando en cuenta un análisis de características y especificaciones. Con esta base se derivan conclusiones sobre el potencial del uso de las hand-helds computers en el desarrollo del país y en el manejo de la información hoy en día.

Como parámetro importante para este estudio solo se tomarán en cuenta las siguientes hand-helds computers, la PSION HC-110 y la TELXON PTC-860TC, que por sus características son máquinas muy comparables en funcionamiento y operabilidad, además de estudiar otra tecnología como la de HEWLETT PACKARD 95LX denominada Palmtop (Computadora portátil para arriba de la palma de la mano), sin embargo, nos centraremos principalmente en el detalle de la primera hand-held computer ya que por su diseño características y estructura resulta ser una computadora muy adaptable a las necesidades de cualquier usuario, sin olvidar que casi todas las hand-helds computers están fabricadas bajo una arquitectura propietaria y con una estructura especial que facilita su uso dentro de la industria de la computación.

El trabajo se inicia con un capítulo que muestra el entorno de la nueva tecnología en un ambiente de servicios de información y a través del cual se observa el desarrollo de las hand-helds computers a nivel mundial, haciendo referencia al aspecto productivo y económico en que están inmersas, sin olvidar la importancia que toman estos equipos para nuestro país.

En el segundo capítulo se destacará la estructura física de las hand-helds computers, sus componentes principales y periféricos más usados, así como los sistemas operativos que utilizan, tanto el de la PSION (EPOC-OS), como el de la TELXON (RAMSaver), su funcionamiento, características y sintaxis de los comandos principales.

En este capítulo se harán importantes comparaciones con el sistema operativo de Microsoft (MS-DOS) ya que los sistemas operativos que estudiaremos están basados en ese ambiente.

En el tercer capítulo se presenta un lenguaje de programación utilizado para el desarrollo de aplicaciones en hand-helds computers desarrollado por PSION (OPL/g), sus principales procedimientos, librerías, comandos y funciones así como la sintaxis de programación, por otra parte, se analiza la creación de archivos y la estructura fundamental para el ahorro de espacio en la máquina de este mismo fabricante denominada archivos ODB.

Finalmente se analizará, en el capítulo cuarto, un sistema desarrollado en OPL/g para la recolección de datos del servicio muestreo de tiendas llamado store-check; propiedad de A.C. Nielsen Co.

En este capítulo, se comenzará con la explicación del servicio y la utilidad práctica en la industria, observando el flujo de información que se lleva a cabo desde la captura de información via hand-held computer, hasta la descarga en una computadora personal y finalizando con la explicación del programa desarrollado para esta función.

## I. LA NUEVA TECNOLOGIA Y SU DESARROLLO.

### I.1 Antecedentes.

Los avances de la ciencia y la tecnología se han dado, en función del tiempo, como una curva exponencial, de ahí la sensación de brincar desde la prehistoria, cuando los mayas utilizaron el concepto del cero, hasta nuestros días, con las computadoras, satélites y tecnología más avanzada, de una manera dramática.

El desarrollo de la informática ha tenido una evolución tremenda. En sus comienzos, muy lenta, con el nacimiento de la máquina llamada la pascalina (BLAS PASCAL en 1642), para realizar el cálculo de manera mecánica, hasta el inicio del procesamiento de datos (HERMAN HOLLERITH en 1890), con la tabulación del censo en los Estados Unidos de América. A partir de entonces, nos enfrentamos día con día con nuevas técnicas del procesamiento de información.

El éxito alcanzado por la computadora parece ser el resultado de la labor de muchas personas que han tratado de resolver múltiples problemas en diversos campos. Es la consecuencia natural del desempeño de nuestras actividades diarias, este éxito se basa en la capacidad de almacenar y procesar actividades muy grandes de información, además de poder agregar, actualizar, recuperar y transmitir información con los elementos existentes en una computadora.

Hacia 1951, la computadora ya se había introducido al mercado en forma comercial, pero ni siquiera aquellos que mantenían un estrecho contacto con ella, estaban preparados para prever lo que vendría después. Al inicio de la década de los setentas se construye el primer microprocesador a partir de circuitos integrados (Chips). Si la computadora es ocasionalmente denominada " EL MOTOR DE LA PRODUCTIVIDAD " en la sociedad moderna, ninguna parte representa mejor esa descripción que la unidad central de almacenamiento (CPU), que para 1970 está integrado de chips de silicio abriendo la puerta a la miniaturización a gran escala.



Para agosto de 1981 la IBM lanza al mercado la primera computadora personal, ganando rápidamente el favor de la industria. Esta máquina con su procesador Intel 8088 de 16 bits inicia la carrera de sofisticación del hardware al dar mayor velocidad en el procesamiento de datos y convirtiéndose en el mejor instrumento de almacenamiento de información que sus antecesores de 8 bits.

Cada ola sucesiva en la evolución de las computadoras ha sido marcada por la construcción de circuitos electrónicos más eficientes, más confiables y más pequeños. En el mundo de las computadoras, "Lo pequeño es bello.", porque generalmente significa velocidad y economía. Los componentes más pequeños pueden estar más juntos el uno del otro, acortando la distancia que las señales eléctricas deben viajar entre ellos.

El eslabón entre una lógica más sofisticada, los componentes de memoria y las mejoras del funcionamiento comenzó con la era de la computación y la sofisticación de la tecnología. Como parte de este crecimiento tecnológico tiene lugar un proceso de convergencia entre la electrónica y el mundo de la información bajo una tecnología común; este elemento clave de tal magnitud lo constituye el microprocesador. De esta manera, los nuevos desarrollos de la electrónica dan paso a la configuración de un conjunto de nuevas actividades productivas que se asocian a la informática de hoy en día.

La difusión del microprocesador en el ambiente de los servicios de información es estimulada por la necesidad de modernizar, optimizar y simplificar el manejo de crecientes volúmenes de información de diversa índole.

Estos instrumentos están proporcionando un sólido soporte a la interconectabilidad de equipos de cómputo en redes informáticas, facilitando la comunicación entre puntos muy distantes de datos y los centros de procesamiento de dicha información.

Es de ahí donde se desprende la necesidad de miniaturizar el equipo de cómputo y aumentar las facilidades que el mismo equipo ofrece para la obtención de información ya que con esto se asegura el obtener datos de una mejor calidad en un menor tiempo a un menor costo y con la mejor oportunidad.

## **I.II Generalidades.**

El enfoque generalizado de este cambio tecnológico está dando lugar a una transformación global de la estructura productiva dentro de la industria en general. ¿Qué nos reserva el futuro?... Un número creciente de usuarios de productos de computación a través de terminales unidas a computadores centrales y mayores aplicaciones que facilitarán a la gente el desempeño de su trabajo en un mundo de información cada vez más amplio y complejo.

A su vez este cambio se expresa, por un lado, en el crecimiento del sector servicios y su mayor participación en la economía y, por otro, en su diversificación al generarse una gama amplia de nuevos servicios que estimulan nuevas necesidades en el mercado.

La producción de servicios de información registra cada vez mayor diversidad y complejidad; así, de las tradicionales empresas que originalmente proporcionaron servicios de procesamiento de datos, destacan ahora las que ponen énfasis en el desarrollo de software específico utilizando hardware de igual manera específico. Tal es el caso de las Hand-Helds Computers.

Las hand-helds computers, ( del inglés que significa "Computadora que se sostiene en la mano."), son un equipo de cómputo de propósito general para el desarrollo y uso de software específico en la solución a problemas de distancias, volúmenes de información y validación de datos muy grandes en lugares donde es difícil la utilización de un equipo de cómputo que por sus características necesite de ciertas instalaciones propias para su funcionamiento.

Estos equipos, por su diseño y características dan como resultado el abaratamiento relativo del mercado de computadoras, que a su vez se han destacado por servir de enlace de una manera rápida y segura de las fuentes de información al lugar del procesamiento de los datos; aquí han adquirido un alto valor agregado, esta situación de hecho, ha dado origen a un novedoso fenómeno denominado: **REVOLUCION DE LA TECNOLOGIA DE LA INFORMACION.**

A nivel de la economía internacional, la difusión de esta nueva tecnología está determinando un cambio fundamental en la división del espacio económico. La industria electrónica moderna, en buena medida representada por la que produce equipos de cómputo y comunicaciones, ha logrado una mayor preponderancia; se encuentra hoy situada por encima de otras industrias que tradicionalmente fueron de punta, como son la metalmecánica, la de autopartes o de bienes de consumo duradero.

Un factor fundamental en el impulso y dinámica de la industria electrónica ha sido la relevancia que en ella han tenido las actividades de investigación y desarrollo que registran, en los principales países desarrollados, mayores coeficientes de gasto en relación a otras industrias en este rubro.

### **I.III Factores de Estudio.**

El surgimiento y expansión de las hand-helds computers está determinando múltiples transformaciones que es importante destacar como antecedente para este estudio:

1.- El desenvolvimiento de esta tecnología ha implicado una modificación fundamental en la estructura productiva; ha creado condiciones para la reconversión industrial con base en la tecnología de la información, en tal reconversión las hand-helds computers son necesariamente específicas ya que la innovación es el resultado de la informatización de los procesos. Asimismo, ha ampliado y diversificado los servicios de información y en particular, ha incrementado su papel estratégico. Se eleva así la productividad y oportunidad en muy diversos sectores, en especial en el sector servicios.

2.- El avance de esta tecnología está impulsando una amplia difusión de los servicios tradicionales e incorpora nuevos; con ello, los servicios adquieren mayor comercialidad y amplían su mercado.

3.- Las hand-helds computers se convierten, cada vez más, en la infraestructura de apoyo de muy diversas actividades económicas. Lo anterior es particularmente cierto en los servicios de mercadotecnia, auditoría, inventarios, pedidos, ventas, etc.

4.- Como consecuencia de lo ya señalado, las actividades productivas con las hand-helds computers se expanden rápidamente a nivel mundial y en cada país en particular, tal es el caso de México que para 1985 cuenta con 480 y para 1992 el volúmen ha crecido a cerca de 8,000 hand-helds computers, principalmente utilizadas en las industrias de servicios. Esto permite llevar a cabo la actividad productiva de las empresas de una manera más eficiente y automatizada (1).

5.- Para cada empresa, las hand-helds computers permiten, por un lado un mayor crecimiento de los mercados a trabajar y por otro, la centralización de decisiones. Esto a su vez incrementa la descentralización de la operación, pero mantiene el control de cada proceso.

El papel de esta tecnología en México ha estado fuertemente influida por las grandes empresas transnacionales, cuya estrategia de mercado ha sido implantar un desarrollo similar para sus filiales, en México, dando apoyo y asesoría técnica para el proceso de cambio. Estas se han convertido en los agentes predominantes del desarrollo de esta tecnología. Los factores fundamentales que han determinado esta situación son los siguientes:

- Mantenimiento del control de los procesos en los servicios de información.
- Disposición de una nueva y sólida estrategia de mercado.
- Mezcla adecuada entre la calidad y la eficiencia de los servicios.

Cabe destacar que además cuentan con la fortaleza de las empresas de bienes informáticos, las cuales les ofrecen la gama de servicios necesarios, tanto de hardware como de software y de asistencia técnica con el apoyo al desarrollo de programas o sistemas específicos para lograr un cambio substancial en la forma de obtener datos vía hand-held computers.

(1) Este dato se sustenta de la empresa UNIMICRO distribidora de Hand-Helds Computers TELXON (Junio 1993).

Las empresas nacionales han tenido que adquirir su tecnología de las grandes empresas transnacionales. Esto ha llevado a sustituir lo que hubiera sido una opción de desarrollo tecnológico autónomo por otra de mutuo entendimiento comercial con quienes controlan la tecnología; adicionalmente, las empresas nacionales no cuentan con una estrategia sólida de mercadotecnia y, más importante aún, no han dispuesto de los recursos técnicos y financieros para integrar un paquete completo que les permita ofrecer el apoyo a sistemas integrados de cómputo y transferencia de información. Pero a su vez la situación de la economía mexicana durante los últimos años y las modificaciones en la política económica general, han determinado importantes avances en el uso y desarrollo de las hand-helds computers hoy en día.

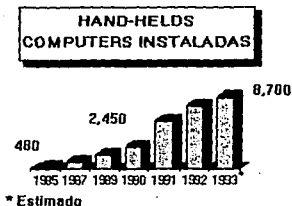
Rapidez del cambio tecnológico y convergencia de los microprocesadores con la información son los motivos por los que el análisis de las hand-helds computers se vuelve un punto importante de estudio, ya que la fusión de la tecnología de computación con las comunicaciones hacia sistemas integrados de transmisión de datos son sueños hechos realidad.

Estos equipos constituyen sin duda la base de la informática del futuro, y han puesto fin al reinado de los main-frames y las estructuras informáticas basadas en la operación de grandes centros de cómputo. Por cerca de 5,000 dólares las microcomputadoras ofrecen hoy la capacidad computacional que algunos años atrás requería equipos de gran porte con un costo superior de 900,000 dólares.<sup>(2)</sup>

Al revisar el mercado de las hand-helds computers en México, es importante visualizar el gran crecimiento que las empresas han venido observando en la adquisición, uso y distribución de estos equipos, desde la entrada de 480 hand-helds computers de fabricación sueca de la marca MICRON en 1985, pasando por el gran desarrollo impuesto por la empresa UNIMICRO en la distribución de equipos TELXON alcanzando rápidamente el liderazgo, hasta nuestros días, con la gran diversificación de marcas, estilos y modelos, pronosticando existencias instaladas superiores a las 8,700 hand-helds computers para 1993. (Figura 1).

(2) "Zoom! Here Come the New Micros" Business Week  
Diciembre 1988.

Figura 1.-



Las marcas más importantes en el mercado mexicano de hand-helds computers, clasificadas tanto por el servicio que ofrecen como por la cantidad de equipos vendidos e instalados en nuestro país son:

TELXON

PSION

SYMBOL

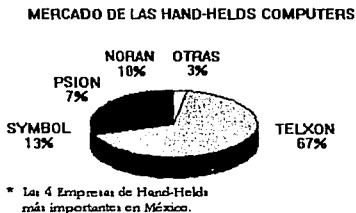
MITSUBISHI

NORAN

LXE

Estas empresas han logrado destacar en la distribución de sus equipos para diferentes usos, de igual manera cada empresa ofrece diferentes servicios adicionales, tales como la programación de aplicaciones específicas y/o la capacitación en el uso, además del soporte técnico que incluye a la venta de los equipos. (Figura 2).

Figura 2.-



#### I.IV Productividad y Desarrollo.

A todo esto, ¿ Dónde y Cómo se utilizan las Hand-Helds Computers ?...

En México:

Una de las primeras empresas en adquirir este tipo de tecnología es Nacional de Drogas S.A. (NADROSA), la TELXON PTC-700, en el inicio como auxiliar en el conteo de inventario de medicinas, y posteriormente se incluye como una herramienta de ventas ya que con este tipo de máquina se puede levantar el pedido, comunicar la información al centro de cómputo para su validación y esperar la confirmación de la venta agilizando los procesos que estén involucrados.

Dentro de las empresas paraestatales, un importante desarrollo es el que se llevó a cabo en La Compañía de Luz y Fuerza del Centro, en este desarrollo, se utilizan las hand-helds computers para tomar automáticamente la lectura del medidor de corriente, disminuyendo con esto los errores por digitación o lectura incorrecta ya que también se proporcionaron medidores con las características necesarias para descargar la información automáticamente sin necesidad de que un ser humano intervenga en la manipulación de la información; es importante que en el menor tiempo posible se logre la automatización total de este servicio ya que en este momento solo se lleva en tan solo una muestra de la población.

Otro ejemplo es el de A.C. Nielsen Company, que en 1991 introduce el uso de las hand-helds computers para el levantamiento de información para estudios mercadológicos, uno de estos servicios el store-check, lo analizaremos a detalle en el cuarto capítulo.

En Canadá:

En todas las oficinas del Forestry Department of Canada (Departamento de Silvicultura del Canadá), se utilizan las hand-helds computers para revisar y estudiar cerca de 2,400 árboles; diariamente se levanta la información y se transmite a las oficinas centrales para analizar el deterioro que sufre la naturaleza por los cambios climáticos que ocurren en el país, publicando dicho estudio más frecuentemente que antes de contar con este equipo.

En España:

En la exposición mundial celebrada en Sevilla (EXPO'92), se utilizaron este tipo de equipos, con la finalidad de mantener los stocks de productos para venta al público en niveles suficientes para aprovechar la demanda, tal es el caso de la compañía de refrescos Coca-Cola, que para reemplazar los faltantes en cada una de las máquinas despachadoras de refrescos, se equipó a los repartidores con hand-helds computers que informaran del tipo y cantidad de producto que se necesitaba, así como su localización dentro de la feria, esto permitió incrementar sustancialmente las ventas de la empresa en dicho evento.



En Holanda:

Una firma holandesa de cosméticos "SCHWARZKOPF", reconocida a nivel mundial por sus productos de cuidado para el cabello, compró, para asegurarse de que sus consumidores están provistos de todos sus productos, el desarrollo en hand-helds computers, de un sistema de muestreo y venta de sus productos, este sistema está armado en un portafolio para vendedores de la empresa; SCHWARZKOPF, no solo puede vender y distribuir mejor sus productos sino que puede conocer las preferencias del mercado, y en cuanto tiempo se desplaza su producto, con esto mejorar la calidad y entrega de los mismos.

En Estados Unidos de América:

Un importante uso de las hand-helds computers es el destinado por New York City's Transit Police Department (Departamento de Policía de Tránsito de la Ciudad de Nueva York), ya que utiliza un moderno sistema que le permite, al oficial de policía, tomar decisiones sobre arrestos, al solicitar en su hand-held computer los datos del registro criminal de un detenido, tecleando su apellido, nombre o número de licencia, este puede saber si se trata de una persona que solo tiene faltas menores, o dependiendo de su historial, ser un criminal peligroso, el sistema cuenta con una base de datos cargada que se actualiza vía telefónica o directamente en el departamento de policía, en el momento que el oficial lo solicite. Esto a disminuido la cantidad de falsos arrestos por delitos menores en esta ciudad, así como también, el desconocimiento de delinquentes peligrosos. A este sistema se le ha denominado "Angel Guardián", ya que de igual manera proporciona mayor seguridad a los oficiales en el cumplimiento de su deber.

Estos ejemplos son simplemente una muestra de la utilidad de las hand-helds computers en diferentes países, con circunstancias y proyectos muy diversos, pero de igual manera existen muchos otros países que utilizan esta tecnología en su beneficio.

De igual manera es importante reconocer los triunfos que estos equipos han recibido, tales como "Best International Product" otorgado a PSION a la mejor hand-held computer de 1992 y el "BYTE Award of Distinction" en 1991.

Cabe mencionar que, ante las oportunidades que ofrece esta nueva tecnología y conociendo los retos que arrastra, es necesario que los empresarios se convenzan de utilizar y aprovechar las estrategias de mercado que nacen de las hand-helds computers.

Al encontrarnos a una distancia muy corta de alcanzar el tratado comercial más grande de la historia, hay que pensar que lo más importante es ofrecer grandes posibilidades de desarrollo para nuestro país, ya que dentro de unos meses no será factible encontrar las condiciones de trabajo como las que tenemos hoy, en este sentido la tecnología avanza y nos envuelve de una manera casi imperceptible, por ello es de vital importancia conocer y dominar los avances tecnológicos en el menor tiempo posible.

## II. ESTRUCTURA DE LAS HAND-HELDS COMPUTERS.

### II.1 Conceptos Generales.

Una hand-held computer en su traducción literal, es una computadora que se usa con una mano, en un término más amplio, son computadoras portátiles que principalmente operan con baterías y generalmente son utilizadas para procesos y desarrollos específicos, sin embargo, cabe mencionar la existencia de otro concepto tecnológico similar al de las hand-helds computers pero este, es desarrollado por la empresa **HEWLETT PACKARD**, denominado **PALMTOP PC**, esta tecnología, aunque sigue siendo una computadora portátil que cabe en la palma de la mano, ya no es de uso específico, por que principalmente se utiliza como una extensión de la computadora personal.

La función principal de las hand-helds computers es la recolección de información, dependiendo del sistema desarrollado, para ser almacenada, procesada y validada otorgando una respuesta rápida y eficiente mientras la información es descargada, transmitida y/o actualizada a un equipo de cómputo más grande.

Este concepto se diseñó para resolver cualquier requerimiento de aplicación e integrarse dentro de un sistema de información como herramienta de trabajo. Las hand-helds computers combinan el diseño de un hardware modular, que representa un nuevo recurso para su utilización, con las facilidades de desarrollo con un software estructurado. Cada elemento del hardware es altamente configurable, desde el espacio en disco y/o memoria hasta los periféricos tales como, modems, scanners, lectores de códigos de barra e impresoras.

Para conocer más de cerca el concepto de **HAND-HELDS COMPUTERS** se analizarán dos equipos con una estructura y manejo similar, además de hacer mención a un tercer equipo con otras características y especificaciones para su uso.

- \* **PSION** HC - 110 ( Hand-Held Computer )
- \* **TELXON** PTC-860TC ( Hand-Held Computer )
- \* **HP** HP 95LX ( Palmtop PC )

## **II.II Especificaciones Técnicas.**

El análisis y determinación de los requerimientos deberá reflejar en términos de cantidad y calidad las características indispensables y deseables del equipo físico (hardware), en este caso el de las hand-helds computers así como, las del equipo adicional o complementario utilizado junto con este equipo.

Por esta razón se tomaron en cuenta las siguientes puntos para el desarrollo de las especificaciones de los equipos:

- a) Capacidad de memoria.
- b) Número y tipo de unidades de entrada y salida.
- c) Cantidad de unidades de disco y características.
- d) Tipo de sistema operativo y lenguajes de programación.
- e) Requerimientos de paquetes especiales.
- f) Requerimientos necesarios para las comunicaciones.
- g) Almacenamiento y vida útil del equipo.

Los resultados permiten analizar objetivamente, dentro de las posibles alternativas de solución, los equipos más convenientes.

Un punto de igual importancia es el uso que tendrán las hand-helds computers, ya que algunos modelos son de menor resistencia que otros; su ergonometría está diseñada para uso ejecutivo o para trabajo rudo dependiendo del modelo y con esto también su desempeño, aunque el rendimiento principal esté en el desarrollo de los programas que se necesiten.

A continuación se listan las características de cada uno de los equipos:

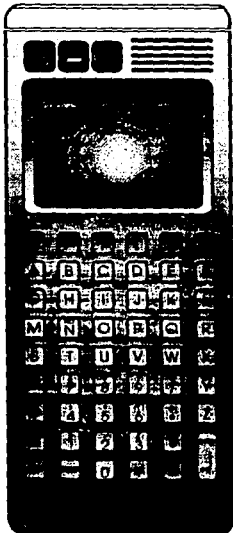
**PSION HC-110A.**

<b>Procesador:</b>	80C86 (16 bits).
<b>Velocidad:</b>	3.84 MHz.
<b>Memoria Interna :</b>	ROM 256 Kb. RAM 256 Kb CMOS.
<b>Memoria Externa:</b>	2 Unidades de Disco de Estado Sólido SSD. 128 Kb - 2 Mb (RAM o Flash)
<b>Pantalla:</b>	9 renglones x 26 columnas.
<b>Teclado:</b>	53 Teclas US. (Sin Teclas de Función).
<b>Fuente de Poder:</b>	Pila recargable de 7 Volts. (PSION) Eliminador de baterías.
<b>Periféricos:</b>	Puerto Serial (1200 - 19200 bps) (9 pins) Puerto Paralelo (25 pins) Modem (300 - 19200 bps) Lector de Códigos de Barra
<b>Sistema Operativo:</b>	Epoc/OS (Multitarea) Interprete de MS-DOS
<b>Lenguajes de Programación:</b>	OPL/g, C
<b>Software de Comunicaciones:</b>	Link - MCLink
<b>Archivos:</b>	ODB - ASCII - ISAM Library
<b>Borrado Físico de Archivos:</b>	Solo en RAM
<b>Tamaño:</b>	20 cm (largo) x 8 cm (ancho) x 3.5 cm (profundidad)
<b>Peso:</b>	640 gr.

Cada elemento del hardware de la HC-110 de PSION es altamente configurable, desde las unidades de disco de estado sólido hasta los dispositivos de expansión o periféricos, de manera similar es importante el sistema operativo multitarea con capacidad para desplegar gráficos, y permitir el monitoreo de procesos concurrentes al mismo tiempo.

La Hand-held computer de PSION es un modelo diseñado con el teclado de forma secuencial y no **qwerty** o standard, con suficiente espacio en la pantalla para el despliegue de información, por otro lado, permite el acceso a la parte interna de la misma, con el objeto de utilizar los 2 drives y la recarga o intercambio de batería de una manera fácil y rápida. ( Figura 3 ).

Figura 3.-



### Los Discos de Estado Sólido.-

La hand-held computer HC-110 tiene 2 drives equivalentes a los de una microcomputadora, estos drives de estado sólido permiten almacenar información, dependiendo de la capacidad del disco, en una tecnología que integra chips de memoria en forma de discos removibles y fácilmente almacenables. Esta memoria puede ser de dos tipos, una memoria volátil RAM que necesita de una pila de 3 voltios adicional para su uso y que permite el borrado físico de la información y/o la memoria permanente o Flash, en este tipo de memoria no se logra un borrado físico de la información a menos que se inicialice el disco. ( Figura 4 ).

Estas unidades de almacenamiento son totalmente independientes de la memoria interna de la máquina, ya que además contiene una memoria ROM de 256 Kb. donde está cargado el sistema operativo y las principales funciones de la hand-held computer, y una memoria RAM de 256 Kb. para poder ser utilizada en procesos, programas o almacenamiento antes de realizar la descarga al disco de estado sólido.

### Pila Recargable y de Respaldo.-

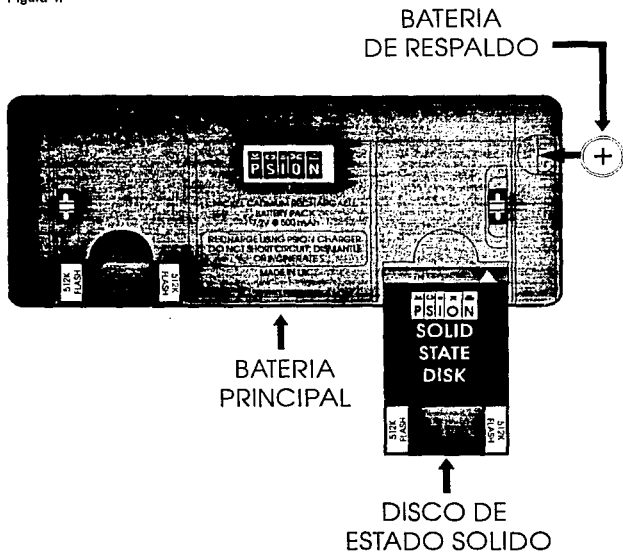
La fuente de poder principal de la HC-110 es una pila recargable de nickel-cadmio de 7 voltios, diseño especial para PSION. La manera de recargar de energía la pila puede ser conectando la hand-held computer a la corriente, y el indicador se encenderá para indicar la carga de batería, o retirar la pila de su lugar para conectarla con el eliminador directamente a la corriente.

La duración de la batería en promedio es de 8 a 10 horas de uso continuo, a diferencia de la pila de respaldo, esta es una pila de litio de 3 voltios no recargable que se utiliza principalmente para mantener la información de la memoria RAM mientras no se utiliza la pila principal. ( Figura 4 ).

### Puertos de Comunicación.-

La HC-110 para la comunicación con una microcomputadora utiliza el puerto de alta velocidad para integrarse a otro aparato llamado **CRADLE** (Cuna), para transmitir información de hasta 9600 bps. ( Figura 5 ).

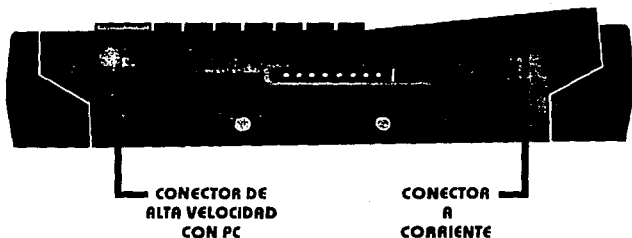
Figura 4.-



Adicionalmente a la configuración original se permite agregar un puerto serial a la hand-held computer HC-110 para la comunicación vía puerto serial de 25 pins, y sin la necesidad de adquirir el **CRADLE** (Cuna) que además contiene un puerto paralelo para conectar directamente una impresora a este puerto.



Figura 5.-



Una de las hand-helds computers con más integración y flexibilidad en el hardware es la HC-110 de PSION, por que integra todo lo necesario para su operación básica, además de poder agregar periféricos o discos de memoria de una manera dinámica sin tener que reconfigurar nada.

Por otro lado el tipo de hand-held computer práctica y de uso rudo en el trabajo, son conceptos que se encuentran entrelazados con esta máquina.

Para este equipo cabe mencionar lo siguiente: La vida útil pronosticada sobrepasa los 3 años de trabajo constante solo requiriendo el mantenimiento preventivo que se otorga de fábrica, pero es necesario adquirir cada año y medio una pila nueva para el buen funcionamiento de la hand-held computer, además, si se usan discos de memoria RAM se necesita de igual forma cambiar su pila cada año, por otro lado el teclado es uno de los componentes que se desgasta considerablemente por tratarse de un equipo que su uso principal es la captura de información, pero también, es fácil su intercambio al término de un año.

**TELXON PTC-860TC.**

<b>Procesador:</b>	80C88 (16 bits).
<b>Velocidad:</b>	4.74 MHz.
<b>Memoria Interna :</b>	ROM 128 Kb. RAM 2Mb-4Mb CMOS. EPROM 128Kb.
<b>Pantalla:</b>	16 renglones x 21 columnas.
<b>Teclado:</b>	55 Teclas US. (5 Teclas de Función).
<b>Fuente de Poder:</b>	Pila recargable de 6 Volts. Pilas Alcalinas (Comerciales). Eliminador de baterías.
<b>Periféricos:</b>	Puerto Serial (1200 - 2400 bps) (25 pins) Modem (300 - 2400 bps) Lector de Códigos de Barra
<b>Sistema Operativo:</b>	RAMSaver Interprete de MS-DOS 3.20
<b>Lenguajes de Programación:</b>	Basic, C, Pascal, Cobol
<b>Software de Comunicaciones:</b>	TCAL
<b>Archivos:</b>	ASCII - ISAM Library
<b>Borrado Físico de Archivos:</b>	Solo en RAM
<b>Tamaño:</b>	22 cm (largo) x 8.4 cm (ancho) x 4.3 cm (profundidad)
<b>Peso:</b>	680 gr.

Todos los componentes del hardware de la PTC-860TC de TELXON no son configurables fácilmente, ya que la estructura de la hand-held computer se inicia con los dispositivos de expansión o periféricos que se adquiera, de manera similar es importante señalar que todo el almacenamiento de información es interno por lo que no tiene posibilidad de ser incrementada esa capacidad de una forma manual sin requerir del fabricante. El sistema operativo es un desarrollo especial del MS-DOS 3.20 de Microsoft, pero con características propias para el desempeño de la hand-held computer.

El teclado de la hand-held computer de TELXON está diseñado de manera standard o **qwerty**, con esto facilita el acomodo para el uso de las teclas. En la pantalla despliega información, de 21 columnas por 16 renglones pero sin la capacidad de utilizar tipografía, por otro lado, no permite el acceso a la parte interna de la misma, solo el intercambio de batería de una manera fácil y rápida ya que utiliza pilas comerciales. ( Figura 6 ).

#### Unidades de Almacenamiento.-

La hand-held computer PTC-860TC solo puede utilizar la memoria RAM con que se configuró para almacenamiento pero además cuenta con una memoria EPROM para la carga de programas ya que en esta tecnología la memoria EPROM solo puede ser borrada mediante la exposición a luz ultravioleta.

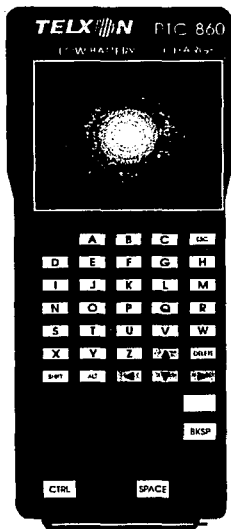
Estas unidades de almacenamiento son totalmente independientes de la memoria interna de la máquina, ya que además contiene una memoria ROM de 128 Kb. donde está cargado el sistema operativo.

#### Pila Recargable .-

La fuente de poder principal de esta hand-held computer son 4 pilas recargables de nickel-cadmio o no recargables de 6 voltios. La manera de recargar las baterías va a depender de las pilas que se compren ya que es totalmente intercambiable por cualquier pila comercial, y el momento para realizar este mecanismo lo indica encendiendo un foco con el mensaje de low battery (Batería baja) .

La duración de la batería en promedio es de 6 a 8 horas de uso continuo.

Figura 6.-



Esta hand-held computer en su configuración original no necesita de ningún tipo de periférico adicional para su funcionamiento en la recolección de información, pero cabe mencionar que al igual que otras hand-held computers, otros fabricantes han desarrollado hardware para poder ser utilizados con este equipo, tal es el caso de impresoras, modems o lectores de códigos de barras.

A diferencia de la hand-held computer de PSION, la TELXON tiene una vida útil que sobrepasa los 5 años considerando solo el intercambio de pila, ya que su estructura es de mayor durabilidad pero su ergonomía dificulta su manejo.

**HP 95-LX.**

<b>Procesador:</b>	Nec V20H (16 bits).
<b>Velocidad:</b>	5.37 MHz.
<b>Memoria Interna :</b>	ROM 1Mb. RAM 512Kb-1Mb.
<b>Memoria Externa:</b>	Unidad de Disco Card. RAM 128Kb-1Mb.
<b>Pantalla:</b>	16 renglones x 40 columnas.
<b>Teclado:</b>	78 Teclas US. (10 Teclas de Función). 7 para acceso a aplicaciones.
<b>Fuente de Poder:</b>	Pila recargable de 3 Volts. Pilas Alcalinas (Comerciales). Eliminador de baterías.
<b>Periféricos:</b>	Puerto Serial (1200 - 9600 bps) (9 pins) Modem (300 - 19200 bps) Lector de Códigos de Barra
<b>Sistema Operativo:</b>	MS-DOS 3.22
<b>Lenguajes de Programación:</b>	Basic, C, Pascal, Cobol
<b>Archivos:</b>	ASCII
<b>Borrado Físico de Archivos:</b>	Solo en RAM
<b>Tamaño:</b>	16 cm (largo) x 8.64 cm (ancho) x 2.5 cm (profundidad)
<b>Peso:</b>	312 gr.

Para realizar una comparación equivalente entre hand-helds computer es importante señalar las características y usos de otro tipo de equipos como es el caso de las **PALMTOPS**, de Hewlett-Packard.

Mientras las hand-helds computers están diseñadas para un trabajo rudo con aplicaciones únicas de servicio, a personas que solo se dedican a la recolección de información para un proceso posterior en otro tipo de computadoras ya sea de mayor capacidad o proceso, surgen otro tipo de computadoras con características similares pero orientadas a altos ejecutivos que no pueden cargar con todo su equipo de cómputo, pero si necesitan de las herramientas necesarias para trabajar con información administrativa, en pocas palabras estas máquinas se convierten en una extensión de las microcomputadoras para ejecutivos.

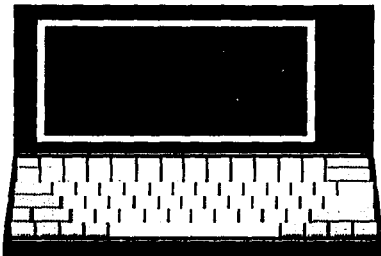
Por esta razón los componentes del hardware de la HP-95 LX de HEWLETT PACKARD están integrados de una manera única y no son configurables fácilmente por el usuario, ya que la estructura de la máquina se inicializa de acuerdo a los dispositivos de expansión o periféricos que se adquieran en la compra, el almacenamiento de información tanto puede ser interno como se pueden utilizar RAM Cards.

El sistema operativo es un desarrollo especial del MS-DOS 3.22 de Microsoft, pero con características propias para su desempeño.

El teclado está diseñado de manera standard o **qwerty**, incluyendo teclas de inicio rápido de aplicaciones tales como Lotus-123, con esto facilita el acomodo para el uso de las teclas.

En la pantalla despliega información de 40 columnas por 16 renglones con la capacidad de utilizar tipografía, por otro lado, no permite el acceso a la parte interna de la máquina, ya que solo es necesario el intercambio de batería y el drive de RAM Card es externo. ( Figura 7 ).

Figura 7.-



#### RAM Cards.-

La HP-95 LX solo puede utilizar memoria RAM ya sea con la que se configuró y es interna de hasta 1 MB o discos de hasta 1 Mb de memoria RAM llamados RAM Cards, que utilizan pila de 2.5 voltios para su uso.

Estas unidades de almacenamiento son totalmente independientes de la memoria interna de la máquina, ya que además contiene una memoria ROM de 1Mb. donde está cargado el sistema operativo y las aplicaciones de uso ejecutivo.

#### Pila Recargable.-

La fuente de poder principal de esta máquina es una pila recargable de nickel-cadmio de 3 voltios. La manera de recargar la batería va a depender de la pila que se compren ya que es totalmente intercambiable por cualquier pila comercial.

La duración de la batería en promedio es de 10 a 20 días de uso continuo.

#### Puertos de Comunicación.-

La configuración original contiene un puerto serial no standard de 9 pins.

## II.III Sistema Operativo.

¿ Qué es un Sistema Operativo ? ...Una de las definiciones más sencillas, pero que representan con mayor detenimiento el funcionamiento de los sistemas operativos esta descrita por H. M. Deitel en el libro Operating Systems... " *Es el software que controla el hardware* " (3). De una forma más detallada el sistema operativo de una computadora es el recurso principal que administra el hardware de una computadora en forma de procesos, almacenamiento, dispositivos de entrada/salida (I/O), dispositivos de comunicación y datos, ejecuta muchas funciones, tantas como interfases tenga el usuario en la interacción con el hardware, ordenando, contando y facilitando los recursos.

MS-DOS (Microsoft-Disk Operating System), aparece en 1981 siendo el primer sistema operativo para microcomputadoras con sistema de 16 bits, integrando conceptos como la administración de archivos basada en el uso de memoria residente denominado tabla de localización de archivos (FAT). ...Esto le permite convertirse en la pieza más popular de software en el mundo, ejecutandose en más de 10 millones de microcomputadoras en el mundo entero y obteniendo más de 20,000 aplicaciones para este sistema, este es el conjunto de aplicaciones más grande en el ambiente de la computación... Bill Gates (3).

Los sistemas operativos empleados en las hand-helds computers son principalmente, sistemas que en su estructura principal toman como ejemplo el MS-DOS, pero con características propias como en el caso de EPOC/OS de PSION, que su característica principal está la de ser un sistema operativo multitarea. En general los principales fabricantes de hand-helds computers proveen de sistemas operativos con una utilidad de comandos y funciones parecida a la de MS-DOS.

### EPOC / OS.

EPOC-OS muestra características importantes para ser un sistema operativo de hand-helds computers, las opciones de sus comandos dan información extra sobre la operación del sistema operativo. En el caso de un comando, existen valores que son asumidos si algunas opciones no son suplidas por otras llamando a estos valores: valores por default.

(3) "19.- Caso de Estudio: MS-DOS" Operating Systems  
H.M. Deitel Febrero 1990.



## Nombres y rutas de Archivos.

Los nombres de los archivos en las hand-helds computers no pueden ser mayores a 8 caracteres, por otro lado, puede utilizarse una extensión después de un punto de separación, la extensión no podrá ser mayor a tres caracteres.

Cualquier caracter puede ser utilizado, a menos que sea un espacio en blanco o un caracter acentuado o cualquiera de estos siete caracteres reservados para el funcionamiento del sistema operativo: \* ? . : \ ! /

Para las extensiones no puede usarse ninguna de las siguientes ya reservadas por su uso en las hand-helds computers .APP .LDD .FON .PIC .PNT .GMF .RST .DYL .IMG .BTF .OPO .ODB

Las unidades de disco en la hand-held computer HHC-110 se identifican de la siguiente manera:

LOC::M: (Para la unidad de memoria RAM)

LOC::A: y LOC::B: (Para unidades de disco de estado sólido)

LOC:: o REM:: significan abreviaturas de la localización de esas unidades, local o remota.

Los comandos principales y sus valores de entrada de comandos particulares de este sistema operativo se dan en descripciones individuales como sigue:

Sintaxis de la lista de comandos:

**COMANDO** parámetro [ *parámetro opcional* ]

parámetro:

**drive:** Nombre de la unidad de disco de estado sólido. Sólo especifica el nombre de una unidad de disco si se usa un archivo o comando que no está en la unidad de inicio.

**filename:** Nombre de un archivo que incluye la extensión esto si existe.

**path:** El nombre de la ruta de un archivo.

[directorio][\directorio...]nombre

#### Convenciones usadas en la lista de comandos:

- Donde sean aplicables, la forma corta de un comando es dada como una abreviatura.
- El texto para cualquiera archivo variable será mostrado en tipo de letra itálica.
- La información mostrada en brackets ( [ ] ) es opcional.
- Los puntos suspensivos (...) significan que un comando puede repetirse tantas veces sea necesario.

### **ATTRIBUTE**

Abreviatura: **ATT**

Objetivo: Activar o desactivar los atributos de cada archivo.,

Sintaxis: ATT [unidad:\ruta\]archivo [(+/-)<h>] [(+/-)<s>]  
[(+/-)<m>] [(+/-)<r>] <Entrada>

- +h habilita atributo para esconder archivo
- h inhabilita el atributo para esconder
- +s habilita el atributo del sistema
- s inhabilita el atributo del sistema
- +m habilita la modificación de atributos
- m inhabilita la modificación de los atributos
- +r habilita el atributo de solo lectura
- r inhabilita el atributo de solo lectura

**&**

**Objetivo:** Ejecuta o corre un programa del lenguaje OPL/g o C.

**Sintáxis:** &archivo <Entrada>

**Comentarios:** Ejecuta el programa (archivo) y espera para que este sea terminado o completado en su ejecución.

**AUTO**

**Objetivo:** Regular el tiempo para apagarse automáticamente.

**Sintáxis:** AUTO (segundos) <Entrada>

**Comentarios:** El tiempo del auto-apagado se digita en segundos. Si los segundos han sido puestos en 0, el auto-apagado se inhabilita. El máximo valor en segundos que puede aceptar es 32767 seg.

**@**

**Objetivo:** Ejecutar archivos denominados Batch-Files (BTF).

**Sintáxis:** @archivo <Entrada>

**Comentarios:** El nombre del archivo contiene un conjunto de archivos o comandos, este es un archivo de texto ASCII. Esto provee de un método de secuencia de diseño de comandos sin tener que accesarlos al prompt del sistema operativo.

Por ejemplo, para comenzar una aplicación automáticamente después del encendido total (boot) de la hand-held computer se debe de crear un archivo llamado AUTOEXEC.BTF, que contenga el o los nombre de los programas que desen ejecutarse al inicio de la hand-held computer.

**CD**

**Objetivo:** Cambiar de un directorio a otro diferente.

**Sintáxis:** CD [unidad:\]ruta <Entrada>

Para desplegar el directorio corriente,

CD <Entrada>

**COPY**

**Abreviatura:** COP

**Objetivo:** Copia uno o más archivos

**Sintáxis:** COPY [unidad:\ruta]archivo [unidad:\ruta]archivo <Entrada>

**Comentarios:** Si la segunda ruta o unidad no se especifican, la copia del archivo es creada en el directorio donde se encuentre ubicado el cursor, y tiene el mismo nombre que en el archivo original.

Un archivo no puede ser copiado en si mismo. Así que si el original está en el directorio corriente y en la misma unidad, despliega el siguiente mensaje de error:

**Create file failed - Resource locked**

Si el archivo original no existe, el comando COPY cesa y aparece el siguiente mensaje de error:

**No files copied**

El uso de caracteres especiales, (Wildcards) \* y ? puede utilizarse en la opción de nombres de archivo o rutas para generalizar.

**D**

Objetivo: Listar los nombres de archivo(s) en un directorio.

Sintáxis: D [/p] [unidad:\ruta\] <Entrada>

Comentarios: Tecleando D los nombres de los archivos por si mismos, en el directorio de ruta serán listados,

El uso de caracteres especiales, (Wildcards) \* y ? puede utilizarse en la opción de nombres de archivos o rutas para generalizar.

El parámetro opcional /p provoca que el desplegado haga una pausa al final de cada pantalla.

**DATE**

Objetivo: Desplegar la fecha y la hora en la hand-held computer.

Sintáxis: DATE <Entrada>

Comentarios: La fecha y hora se despliegan en el siguiente formato:

" Mon 16 Oct 1990 16:25:30 "

**DELETE**

Abreviatura: DEL

Objetivo: Borrar archivos

Sintáxis: DEL [unidad:\ruta\]archivo(s) <Entrada>

Comentarios: Para borrar mas de un archivo al mismo tiempo, el caracter \* puede ser usado.

Nota: Los archivos borrados continuan ocupando espacio en el Flash disk ya que no se borran de una manera física sino que solo quedan señalados, esto se tiene que tomar en cuenta para la medición del espacio a utilizar.

**DEVICE**

Abreviatura: **DEV**

Objetivo: Listar todos los dispositivos disponibles o instalados.

Sintaxis: **DEVICE** <Entrada>

**DIR**

Objetivo: Listar todos los archivos y sus atributos en un directorio.

Sintaxis: **DIR** [/p] [unidad:\ruta][archivo(s)] <Entrada>

Comentarios: El nombre completo de un nombre de ruta, de inicio es \*.\* del directorio corriente, y son desplegados todos los archivos, tamaños de los archivos y las fechas de creación así como sus atributos.

El parámetro /p provoca que el despliegue tenga una pausa en el final de cada pantalla.

El uso de caracteres especiales, (Wildcards) \* y ? puede utilizarse en la opción de nombres de archivos o rutas para generalizar.

**ENVIRONMENT**

Abreviatura: **ENV**

Objetivo: Desplegar el valor de las variables momentaneas o del ambiente de programación.

Sintaxis: **ENV** <Entrada> o **ENV** <variable>=<valor> <Entrada>

Comentarios: Ver tipos de variables (Capítulo 3).

**EXIT**

**Objetivo:** Salir del COMMAND-SHELL (Sistema de Comandos).

**Sintaxis:** EXIT <Entrada>

**Comentarios:** Salir del sistema de comandos y regresar a la siguiente aplicación en la lista de tareas. Si no existe otro sistema de comandos habilitado el comando no tendrá éxito.

**FORMAT**

**Abreviatura:** FOR

**Objetivo:** Formatear Discos de Estado Sólido RAM o Flash (SSD).

**Sintaxis:** FOR unidad:[etiqueta] <Entrada>

**Comentarios:** El comando FORMAT detecta el tipo de disco y posiciona la información apropiada en el mismo. Esta información difiere de Flash a Ram ya que sus características de la lista de acceso a archivos y el uso de memoria se localiza de manera diferente.

El nombre del volumen o etiqueta es opcional.

**FREE**

**Objetivo:** Desplegar la cantidad de bytes libres en la memoria RAM.

**Sintaxis:** FREE <Entrada>

**KILL**

**Abreviatura:** KIL

**Objetivo:** Terminar con el primer proceso llamado disponible.

**Sintaxis:** KIL archivo <Entrada>

**Comentarios:** Termina el primer proceso disponible con el nombre de archivo dado. Este sólo puede utilizarse como último recurso, ya que de la manera en que KILL actúa no permite al proceso terminar correctamente antes de salir.

El uso de caracteres especiales, (Wildcards) \* y ? puede utilizarse.

## **LPROC**

**Objetivo:** Listar todos los segmentos de memoria usados por un programa.

**Sintaxis:** LPROC [archivo] <Entrada>

**Comentarios** Los segmentos de memoria usados por procedimientos o programas, identificados como un archivo, son enlistados por nombre, número de proceso ('.\$xx'), tamaño del segmento de información usada en bytes, y el estado (listo, semáforo, corriente o suspendido).

El uso de caracteres especiales, (Wildcards) \* y ? puede utilizarse.

## **MD**

**Objetivo:** Crear un directorio

**Sintaxis:** MD [unidad:\ruta\]directorio <Entrada>

**Comentarios:** Con éste comando se puede crear una estructura de directorios multi-nivel.

## **RD**

**Objetivo:** Borrar un directorio

**Sintaxis:** RD [unidad:\ruta\]directorio <Entrada>

**Comentarios:** Antes de borrar un directorio todos sus archivos y subdirectorios deben ser borrados.



**RENAME**

Abreviatura: **REN**

Objetivo: Cambiar el nombre de un archivo.

Sintaxis: **REN** [unidad:\ruta]archivo archivo <Entrada>

Comentarios: Este comando renombra todos los archivos que coincidan con el nombre del primer archivo.

Dado que no se puede renombrar archivos através de las unidades de disco, el comando ignora cualquier nombre de unidad especificado con el segundo archivo.

El caracter especial \* puede usarse en cualquier opción del archivo. pero si se utiliza en una diferente posición en el segundo archivo, el comando no cambiará la posición del caracter correspondiente.

**SETDATE**

Abreviatura: **SETD**

Objetivo: Poner la fecha y hora

Sintaxis: **SETDATE** dd/mm/aa hh:mm:ss <Entrada>

Comentarios: Todos los parámetros del comando deben estar presentes.

**TERMINATE**

Abreviatura: **TER**

Objetivo: Terminar un proceso de una manera completa

Sintaxis: **TER** archivo <Entrada>

Comentarios: Un proceso identificado por el nombre de archivo es terminado de una manera ordenada. Este comando debería usarse preferentemente al comando **KILL**.

**TYPE**

Abreviatura: **TY**

Objetivo: Imprimir un archivo de texto en pantalla.

Sintáxis: **TY** [unidad:\ruta]archivo <Entrada>

**VERSION**

Abreviatura: **VER**

Objetivo: Desplegar los números de versión del sistema operativo y sistema de comandos.

Sintáxis: **VER** <Entrada>

### III. LENGUAJE DE PROGRAMACION OPL/g.

#### III.1 Conceptos Generales.

¿ Qué es un lenguaje de programación ? ...Un lenguaje de programación es una serie de caracteres, palabras, y reglas sintácticas que se pueden emplear para desarrollar un programa... (4). Estas características instruyen a una computadora, al ser traducidos al lenguaje de máquina, para que esta desarrolle o efectúe ciertas operaciones o procesos. Desde un lenguaje, que permite al programador diseñar y estructurar aplicaciones en términos similares al lenguaje cotidiano, hasta los lenguajes que actualmente se especializan en instrucciones visuales, rápidas y sencillas son entendidos como lenguajes de programación.

Casi todos los lenguajes de programación necesitan para poder ser entendidos por una computadora, un proceso de traducción al lenguaje de máquina y este proceso lo lleva a cabo otro programa llamado compilador o intérprete.

Existen en la actualidad un número bastante extenso de lenguajes de programación como los denominados de alto nivel, tales como: Pascal, Cobol, Basic, además de otros, que por sus características son denominados de bajo nivel tales como el Ensamblador o C.

Unos lenguajes fueron desarrollados expresamente para un tipo de desarrollo tal es el caso de COBOL, que se crea con el fin de desarrollar programas de tipo administrativo orientado a los negocios, otro ejemplo es BASIC que su desarrollo principal esta orientado a programadores con poco conocimiento en la materia y es de fácil codificación principalmente empleado para enseñar a programar.

El OPL/g está diseñado con características esenciales de BASIC, por ello se convierte en un lenguaje de programación muy sencillo de utilizar, pero a su vez integra conceptos del lenguaje de programación C, el cual introduce un aspecto de gran fuerza, sobre todo en la sintaxis del lenguaje.

(4) "1.- Visión de la programación" COBOL Estructurado  
Andreas S. Philippakis y Leonard J. Kazmier. (Abril 1988).

..." Cuanto más pronto se comience a codificar un programa, mayor será el tiempo que le llevará"... (5).

### III.II Experiencia con OPL/g.

OPL/g se diseñó originalmente para permitir a cualquier persona crear y correr programas así como también, producir aplicaciones en las hand-held computers de PSION. Ha sido rediseñado y modificado desde su primera versión para correr más eficientemente desde cualquier tipo de hand-held computer PSION.

Mientras se mantiene su simplicidad, OPL/g se ha extendido para ser una herramienta sofisticada que puede utilizarse en el desarrollo de una aplicación de ambiente para producir desde sistemas simples hasta los más complejos en una hand-held computer.

OPL/g es un lenguaje de programación basado en procedimientos, PBPL (Procedure Based Programming Language), el cual permite a programas bien diseñados desenvolverse fácilmente. El código fuente se translada a código Q antes de iniciar su ejecución en las hand-held computers. El código Q es un código intermedio, una forma comprimida de OPL/g, donde los comandos son traducidos a "Single Byte Tokens" (Instrucciones simples del lenguaje de máquina), los cuales se interpretan en la ejecución permitiendo que la ejecución del programa sea más rápida que un lenguaje interpretado directamente, mientras que retiene la estructura inicial con OPL/g.

OPL/g, comparado con otros lenguajes, puede usarse más fácilmente para producir aplicaciones rápidamente, ya que está provisto de un conjunto de funciones para el manejo de caracteres, de mecanismos en el manejo de archivos, de comandos y funciones para manipular información, de construcción de pantalla con funciones gráficas, con mecanismos para producir salidas, y con funciones matemáticas y científicas. Como lenguaje basado en procedimientos, el único límite en el número de procedimientos anidados es la memoria disponible. Cuenta con parámetros que pueden pasarse a procedimientos desde los cuales su valor siempre es regresado, permitiendo construir aplicaciones modulares.

(5) "LEY de Murphy de la Programación" Introducción a la programación de Computadoras Henry Ledgard. (1975).

## OPL/g y sus fortalezas:

\*Está diseñado para la ejecución rápida como un lenguaje general, con procedimientos funcionales en el manejo de los datos.

\*Por estar basado en procedimientos, permite el buen diseño de programas.

\*Tiene un manejo eficiente del registro en los archivos ya que incluye una base de datos con manejo de comandos para guardar, recuperar, y leer información en una forma estructurada, convirtiéndose en un acceso rápido y eficiente.

\*El código de origen al ser traducido o compilado al código Q se compacta y las expresiones matemáticas se convierten a la notación polaca, así que los operadores son ejecutados en la forma más rápida posible.

\*Las funciones matemáticas, científicas y de caracteres están provistas como parte del lenguaje.

OPL/g es un lenguaje que incluye un conjunto de funciones y comandos los cuales hacen uso del sistema operativo y la memoria de la hand-held computer para permitir la salida de variadas pantallas gráficas, cargando hasta 8 ventanas de tipo bitmap o mapeo de caracteres.

La forma de generar un programa con OPL/g para la hand-held computer de PSION, se lleva de la siguiente manera:

-Crear un archivo con el lenguaje fuente, en una microcomputadora o PC, dándole una extensión **OPL**.

-Compilar el archivo de origen al formato que el interpretador del OPL/g pueda correr, usando el comando **TRAN** que lo convierte a código **Q**.

-Cargar el módulo objeto con extensión **OPO** en la hand-held computer.

-Correr y probar el programa en la hand-held computer.

### III.III Programación con OPL/g.

\*Un procedimiento es una sola unidad de programa. Los programas pueden construirse de muchos procedimientos. Un procedimiento comienza con la palabra **PROC** y terminan con la palabra **ENDP**.

\*Un módulo es el origen de un archivo, el cual contiene uno o más procedimientos. Un programa es un conjunto de procedimientos y/o módulos que puede ser ejecutados.

#### MODULO01.OPL

**PROC UNO:**

...

...

... } INSTRUCCIONES

...

...

**ENDP**

**PROC DOS:**

...

...

... } INSTRUCCIONES

...

...

**ENDP**

Un módulo es un archivo de texto ASCII, con extensión OPL. Contiene los comandos, órdenes y funciones del lenguaje fuente, posteriormente serán compilados y convertidos al código Q, cada módulo puede estar compuesto de un número ilimitado (solo por memoria), de procedimientos. Además puede incluir definiciones o librerías propias del OPL/g.

La declaración general para un módulo es:

**/\* Comentarios \*/**

**#include <archivo>**

**#define nombre\_simbólico valor**

**PROC UNO:**

**GLOBAL** declaración de variables globales al programa.

**LOCAL** declaración de variables para este procedimiento.

**INSTRUCCIONES (Código)**

**DOS:** Llamada a otro procedimiento

**ENDP**

**REM** Comentarios

**PROC DOS:**

**#include "proc2.opb."**

...

...

...

**ENDP**

**#include <archivo>** o **#include "proc2.opb"** Causa la inclusión de todo un archivo origen específico para ser agregado hasta este punto de compilación. Por ejemplo, las definiciones estándar pueden agruparse en un archivo con extensión .OPB, o las librerías de OPL/g.

Sin embargo sería mejor usar el comando `LOADM` para cargar una librería `OPL/g` o bien compartir el código, esto es porque cualquier cambio al módulo insertado, requerirá de módulos dependientes, para ser recompilados y con el comando `LOADM`, la inclusión ocurre en el momento de correr la aplicación, así que depende de que los módulos no estén afectados, excepto los cambios funcionales.

Las " comillas " alrededor de los archivos incluye, buscan primero la ruta indicada o de origen, mientras que las < llaves > le indican al compilador que busque el lugar estándar para el archivo. El lugar estándar se posiciona por la bandera `-I` en el comando del compilador.

**#define** Permite asignar nombres simbólicos a constantes o expresiones para su uso dentro del programa. Por ejemplo:

```
#define MAXLEN 20

...
...
...
IF len(var1) < MAXLEN
    ...
...
ELSE
    ...
...
ENDIF
...
...
...
```

Además se puede definir expresiones con código `OPL/g` para incrementarse en su valor.

```
#define VARKEY WHILE key : ENDWH
```



**PROC UNO:** Existen algunas restricciones en cuanto al nombre que se le da al procedimiento:

- Puede tener 8 caracteres.
- Puede incluir números, pero debe iniciar con letra.
- Puede estar en cualquier combinación de mayor o menor caso.
- Debe finalizar con dos puntos.
- Debe estar precedido por la palabra PROC.

**REM Comentarios o /\* Comentarios \*/** Es la forma de incluir comentarios al programa y pueden estar en cualquier parte del módulo o procedimiento.

**GLOBAL** Es la forma de realizar declaración de variables para ser utilizadas en todo el programa.

**LOCAL** Es la forma de realizar declaración de variables para ser utilizadas únicamente en el procedimiento donde se encuentran declaradas.

¿Qué ocurre cuando se compila un archivo OPL?...

\* Primero: El módulo es preprocesado...

El preprocesador despoja cualquier comentario del programa, inserta archivos que haya sido incluido por la declaración `#include`, como si fueran parte del programa mismo, reemplaza nombres simbólicos de constantes con sus valores equivalentes, y provee de una revisión condicional usando la sintaxis.

\* Segundo: Los procedimientos se verifican por errores...

Si existen errores en la sintaxis de los procedimientos en el módulo, un mensaje describiendo el error es desplegado en la pantalla. Existen dos tipos de error reportados por el compilador: errores de sintaxis o errores de tecleo, y fallas en la estructura del programa, por ejemplo si el máximo nivel de anidamiento ha excedido de 8 niveles o no existe un procedimiento a utilizarse. Por otro lado verifica la aplicación entera analizando todos los módulos asociados con el programa revisando estructuras incorrectas, alcanzando errores de variables, y nombres conflictivos.

\* Tercero: Cuando no hay mas errores, se produce un archivo OPO...

El compilador produce un archivo .OPO del mismo nombre que su archivo OPL original, esto sólo si la compilación ha sido exitosa. Con este archivo se puede trabajar en las hand-helds computers.

Cuando un programa se ha compilado exitosamente y se ha cargado en la hand-held computer, se puede ejecutar, y para ello existen tres opciones en las hand-helds computers.

1.-Crear un archivo, **AUTOEXEC.BTF**, el cual contiene comandos tipo batch file y debe contener el nombre del programa extensión .OPO, para correr la aplicación desde el encendido principal o boot de la hand-held computer.

2.-Teclear el nombre del programa extensión .OPO en el prompt \$ del sistema operativo de la hand-held computer.

3.-Correr el programa desde la microcomputadora o PC usando MCLINK.

### III.IV Tipos de variables.

#### DECLARANDO VARIABLES.

**LOCAL** es el comando que se utiliza para declarar variables dentro de un procedimiento, las cuales deben ir separadas por comas. Las variables con este comando, serán reconocidas dentro del procedimiento donde se definen, (si se utilizan en otro procedimiento, no serán reconocidas.)

Con este comando se reserva suficiente memoria para archivar los valores de éstas variables, además de inicializar las variables al valor cero.

Para la declaración de variables se pueden usar nombres descriptivos, con las siguientes condiciones:

- NO pueden ser de más de 8 caracteres de longitud.
- Deben empezar con cualquier letra del abecedario, aun que después de eso se puede utilizar cualquier combinación de números y letras.
- Adicionalmente, no se debe usar ninguna función de OPL/g como nombres de comandos o nombres de librerías y procedimientos.

#### TIPOS DE VARIABLES.

Existen diferentes tipos de variables para declarar a las variables, de acuerdo a su contenido, pero se necesita agregar un símbolo al nombre de la variable para definir el tipo de esa variable:

+ Texto o Strings.- Para este tipo de variables es necesario agregar el símbolo **\$** al final del nombre de la variable; por ejemplo: **nombre\$**.

+ Números Enteros Cortos o Small Integers.- Para este tipo de variables es necesario agregar el símbolo **%** al final del nombre de la variable; por ejemplo: **numero%**.

+ Números Enteros Largos o Long Integers.- Para este tipo de variables es necesario agregar el símbolo **&** al final del nombre de la variable; por ejemplo: **numero&**.

+ Números Decimales o Floating Points.- Para este tipo de variables no es necesario agregar ningún carácter al final del nombre de la variable; por ejemplo: **precio**.

Cualquiera de los símbolos del tipo de variables (**\$**, **&** y **%**) deben estar incluidos dentro de los 8 caracteres del nombre de las variables.

Para declarar una variable de tipo texto o string, debe seguir al símbolo \$ la máxima longitud de caracteres que se quiera manejar, esta longitud se debe de representar entre paréntesis; por ejemplo: `nombre$(15)`.

La longitud de este tipo de variables no puede ser mayor de 255 caracteres, ya que este es el número máximo que se permite.

Para declarar una variable de tipo numérico se deben de tomar en cuenta las siguientes características:

Los números enteros cortos o small integers solo aceptan números de un rango de -32768 a +32767.

Los números enteros largos o long integers aceptan números del rango de 2147483648 a +2147483647.

Los números decimales o floating-point eceptan números en el rango  $-10^{\pm 307}$  a  $+10^{\pm 307}$ .

Para la definición de las variables se puede utilizar un grupo de variables para archivar listas de valores. En vez de tener que declarar por separado una variable para cada valor se puede declarar arreglos de variables.

El número entre paréntesis para cualquier tipo de variable numérica es el número de elementos en el arreglo. Así que la declaración LOCAL `a%(5)` crea cinco variables enteras: `a%(1)`, `a%(2)`, `a%(3)`, `a%(4)` y `a%(5)`.

En el caso de la declaración de variables de tipo texto, entre paréntesis, el primer número es el número de elementos en el arreglo y el segundo número especifica la máxima longitud de los strings; todos los elementos en el arreglo del texto tienen la misma longitud, por ejemplo: LOCAL `texto$(5,10)` aloja espacio en memoria para cinco variables de tipo texto, con diez caracteres de longitud.

## ASIGNANDO VALORES A LAS VARIABLES.

Se puede asignar un valor a una variable directamente, o por medio de operaciones entre variables de tipo numérico; por ejemplo:

PROC suma:

```
LOCAL x%, y%, z%
```

```
x%=569
```

```
y%=203
```

```
z%=x%+y%
```

```
PRINT z%
```

```
GET
```

ENDP

suma: Es el nombre del procedimiento

La declaración LOCAL dá x%, y% y z% el valor 0.

Se asigna el valor 569 a x% y 203 a y%, la suma de estos dos valores se asignan a z%.

El comando PRINT despliega el valor de z% en la pantalla.

Para asignar valores a una variable de tipo texto se necesita estar entre comillas el texto que se desee asignar:

```
var$="Texto"
```

## OPERADORES ARITMETICOS.

- +** SUMA
- RESTA o NEGATIVO
- /** DIVISION
- \*** MULTIPLICACION
- \*\*** ELEVAR A UNA POTENCIA
- %** PORCENTAGEAR

En el caso de operaciones con variables de tipo texto, solo se permite la suma y da en la variable de resultado el valor de la primera variable seguida del valor de la segunda; por ejemplo:

Si `a$ = "Choco"` y `b$ = "late"`, entonces la declaración `c$=a$+b$` significa que `c$` es igual a "Chocolate".

Alternativamente, se puede dar a `c$` el mismo valor con la declaración `c$="Choco"+"late"`.

Se debe tener en cuenta que cuando se adicionan variables de tipo texto, el resultado no debe ser más largo que la máxima longitud que se declaró para la variable destino.

### III.V Operadores lógicos.

> Mayor que  
 >= Mayor que y/o igual a  
 < Menor que  
 <= Menor que y/o igual a  
 = Igual a  
 <> Diferente a

**AND** Y  
**OR** O  
**NOT** NO

### Prioridad o Precedencia

Mayor : \*\*  
 - (Signo negativo) NOT  
 \* /  
 + -  
 = > < <> >= <=

Menor : **AND OR**

Así que, cuando existe igual precedencia la prioridad se evalúa de derecha a izquierda, con excepción de las potencias que serán evaluadas de izquierda a derecha, la forma de cambiar la precedencia es utilizando los paréntesis.

La precedencia de tipos de variables en operaciones con números de variables de diferentes tipos como entre decimales y enteros hay que tener cuidado de la variable a la que se asignará el resultado ya que el lenguaje convierte el valor de la variable de tipo entero a decimal puede rebasar el límite permitido para almacenar su valor.

#### Tipos de conversiones

\* Si asigna un valor entero a una variable decimal, el valor simplemente tiene, adicionado a él cero en los decimales.

\* Si asigna un valor decimal a una variable entera, el valor es convertido a un entero, siempre redondeado a cero.

#### Expresiones lógicas

La comparación de operadores y operadores lógicos están basados en la idea de que una cierta situación puede ser evaluada como verdadero o falso. Estos operadores son útiles para poner rutas alternativas en el desarrollo del programa.

Los valores posibles de estos operadores lógicos solo pueden ser verdadero, que está representado por un entero -1 y falso, que está representado por el entero 0.

Los operadores lógicos AND, OR y NOT tiene diferentes efectos dependiendo de sea de ser usados con números enteros o decimales. Cuando se usen números decimales unicamente los operadores lógicos tienen los siguientes efectos:

Ejemplo      Resultado

a AND b

Verdadero si ambos a y b son no cero.  
Falso si tanto a o b son cero.

a OR b

Verdadero si tanto a o b son no cero.  
Falso si ambos a y b son cero.

NOT a

Verdadero si a es cero.  
Falso si a es no cero.

Cuando se usen valores enteros unicamente AND, OR y NOT son operadores logicos con bits inteligentes. Ya que, los números enteros cortos están representados internamente con un código binario de 16-bits y los números enteros largos por un código binario de 32-bits. El bit inteligente significa que una operación es diseñada en bits individuales; un bit se pone si tiene valor 1 y se borra si tiene valor 0.

Ejemplo, la declaración PRINT 12 AND 10 despliega 8, la declaración PRINT 12 OR 10 despliega 14 y la declaración PRINT NOT 7 despliega -8:

Representación Binaria:

```
12 00000000000001100
10 00000000000001010
```

AND actúa en cada par de bits. Trabajando de izquierda a derecha sin considerar los primeros 12 bits:

```
1 AND 1 = 1
1 AND 0 = 0
0 AND 1 = 0
0 AND 1 = 0
```

El resultado del número binario 1000, es el decimal 8.

```
1 OR 1 = 1
1 OR 0 = 1
0 OR 1 = 1
0 OR 0 = 0
```



El resultado del número binario 1110, es el decimal 14.

Representación Binaria:

7 0000000000000111

NOT 7 = 111111111111000

El resultado del número binario 11111111111000, es el decimal -8.

Una forma rápida de calcular NOT para enteros es adicionar 1 al número original y revertirle el signo.

### III.VI Manejo de Archivos ODB.

Se pueden crear archivos desde el lenguaje OPL/g, llamados archivos ODB (OPL/g DataBase-File), que principalmente son utilizados como fuente de información a los programas desarrollados en OPL/g.

La forma de construir los archivos ODB consiste en:

- Un header (cabecera), de 21 bytes con la siguiente información:

- \* Byte 0 a 15 el texto OPLDatabaseFile.
- \* Byte 16 y 17 la representación ASCII del número de versión del software con que se creó el archivo.
- \* Byte 18 la representación ASCII de la distancia al primer registro.
- \* Byte 19 espacio en blanco.
- \* Byte 20 y 21 la representación ASCII de la versión del software mínima requerida.

#### - Información de cada registro

- \* Un Byte con la representación ASCII para determinar el número de campos en el archivo.
- \* Un espacio en blanco.
- \* Un Byte con la representación ASCII por cada campo del archivo dependiendo del tipo de campo:
  - 0 Entero corto.
  - 1 Entero largo.
  - 2 Decimal.
  - 3 Texto.
- \* Un Byte con la representación ASCII para determinar la longitud del registro, sumando el número de campos que existen en el archivo.
- \* Un Byte con la representación ASCII para determinar si un registro es principal o está en uso o es un registro borrado.
  - 0 Registro borrado.
  - 1 Registro principal o en uso.
- \* Un Byte con la representación ASCII de la longitud del campo.
- \* Dato de cada campo.

#### Nombres lógicos

En un programa o sistema, creado en OPL/g, se puede tener hasta 4 archivos ODB abiertos al mismo tiempo. Cada uno de ellos debe tener un nombre lógico: A, B, C o D. Este nombre es solo una abreviación para la identificación del archivo dentro del programa.

Un nombre lógico debe ser usado por cada archivo de información abierto y no puede repetirse ninguno; un archivo no tiene porque ser abierto con el mismo nombre lógico a la última vez que fué abierto ya que cuando un archivo está cerrado, su nombre lógico es liberado para ser usado por otro archivo.

### Fields (campos)

Los campos en un archivo no pueden exceder de 32 en cualquier registro y los nombres de estos son construidos con las mismas características a los de las variables, así que se debe usar estos caracteres % & o \$ para definir el tipo de variable apropiado de cada campos. En este caso no se puede usar arreglos como en las variables de programa.

La longitud no debe ser especificada, pero la longitud máxima de un campo de texto no puede exceder a 255 caracteres.

La máxima longitud de un registro en OPL/g es de 1022 caracteres.

Cuando se utilizan los campos dentro del programa, se necesita agregar el nombre del archivo lógico al frente del nombre del campo, para especificar cual de los archivos abiertos contiene este campo. Por ejemplo, A.nombre\$ es el campo nombre\$ del archivo lógico A.

### Comandos para el manejo de Archivos:

**CREATE** .- Este comando se utiliza para crear un nuevo archivo ODB desde un programa o procedimiento en OPL/g.

Por ejemplo:

```
CREATE "a:\clientes",B,nombre$,dir$,col$,tel$
```

- \* Se crea un archivo llamado CLIENTES.ODB en el drive A:
- \* El nombre del archivo es un texto , así que necesita ir encomillado.
- \* B es el nombre lógico de este archivo.
- \* Después están los nombres y tipos de cada campo del archivo.

**OPEN - OPENR** .- Estos comandos permiten abrir un archivo ya existente en la hand-held computer, el primero lo abre con características de lectura y escritura y el segundo solo de lectura. La sintaxis de ambos comandos es idéntica.

Por ejemplo:

```
OPEN "a:\clientes",B,nombre$;dir$,col$,tel$
```

**CLOSE .-** Cuando se crea por primera vez un archivo o se abre al terminar de usarse o al final del programa se debe siempre cerrar el archivo, aunque el lenguaje OPL/g cierra automáticamente los archivos cuando el programa termina, pero, si se tienen abiertos 4 archivos y se necesita tener acceso a uno más es necesario cerrar uno de ellos para poder acceder el otro.

Por ejemplo:

```
USE A  
CLOSE
```

Este comando remueve automáticamente los registros borrados del archivo lógico A, pero solo si se trata de discos de memoria Ram.

**APPEND .-** Este comando descarga y guarda la información de un registro en el archivo.

**COUNT .-** Esta función regresa el número de registros existentes y activos en el archivo.

**UPDATE .-** Actualiza la información de un registro previamente guardado con la ventaja que si se trabaja con memoria Ram este comando borra el registro y agrega el registro como nuevo al final del archivo, de otra suerte solo se identifica el registro borrado y se agrega otro registro al final.

**USE .-** Permite hacer mención del nombre lógico de archivo que en ese momento se va a utilizar.

**POSITION .-** Este comando posiciona la lectura o escritura de un registro como el registro accesado en ese momento.

Por ejemplo:

### **POSITION 3**

\* El registro 3 es el registro accesado en ese momento.

**FIRST** .- Posiciona el archivo en el registro número uno y lo habilita como el registro accesado.

**NEXT** .- Mueve al siguiente registro dentro de un archivo.

**LAST** .- Posiciona al archivo en el último registro.

**FIND** .- Posiciona al archivo en el registro donde se cumpla la sintaxis definida para su búsqueda.

Por ejemplo:

```
r%=FIND("CAFE")
```

\* Seleccionará el primer registro conteniendo un campo con el valor "CAFE". El número de ese registro se asigna a una variable r% pero, si el número de regreso es cero, la búsqueda no encontró un campo con el cual coincida el texto a buscar.

\* Solo se puede buscar campos de tipo texto, no campos de tipo numérico.

## **IV. SISTEMA STORE-CHECK (HHC).**

---

### **IV.I Objetivo del Servicio.**

La empresa A.C. Nielsen Co. desde hace más de 25 años se dedica a la realización de estudios de mercado, estudios que se encuentran divididos dependiendo de su naturaleza en estudios a nivel hogares, a nivel detallistas, o los de medición de ratings. Del grupo de estudios a nivel detallista, se desprende el estudio STORE-CHECK ( Muestreo de Tiendas - ISOS).

Proporcionar una herramienta ágil y flexible que permita al cliente, verificar rápidamente en los autoservicios más importantes, los esfuerzos de mercadotecnia tanto de sus marcas como de la competencia en los días de mayor afluencia de los consumidores, así como contar con información que les permita realizar una planeación estructurada para futuras estrategias de mercado, es el objetivo principal del servicio.

### **IV.II Beneficios.**

- Nos permite conocer la participación y ubicación de nuestras marcas en los anaqueles de las tiendas, con respecto a las marcas de la competencia.
- Verificar el precio regular y especial con que se venden los productos propios y de la competencia, ayuda a identificar discrepancias o variaciones en los mismos.
- Facilita a la fuerza de ventas, controlar el desplazamiento del producto, de la bodega al área de ventas, y prevenir agotamientos, sobre todo en los días de mayor venta en los autoservicios.
- Ayuda a administrar las ofertas y promociones de nuestros productos y permite conocer las que realiza la competencia.

- Identifica el apoyo publicitario que tienen las marcas propias y de la competencia en los días de mayor afluencia de público en los principales autoservicios.
- Facilita el control y planeación en el lanzamiento de nuevos productos.

#### **IV.III Metodología.**

La información es recabada mediante el sistema de recopilación directa o auditoría, vía hand-helds computers, y se realiza en 130 tiendas de autoservicio, las más importantes del Valle de México, Guadalajara y Monterrey, durante los días jueves, viernes y sábado más próximo después de la quincena; recibiendo el cliente la información el miércoles siguiente al fin de semana de auditoría.

Actualmente los 130 autoservicios se distribuyen de la siguiente manera:

- 70 tiendas en el Valle de México
- 30 tiendas en Guadalajara
- 30 tiendas en Monterrey

Para llevar a cabo esta selección se construye un índice o calificación de la importancia de cada tienda. Este índice es un promedio adecuadamente ponderado de las siguientes variables.

- Número de carriles de salida
- Número de empleados
- Superficie del área de ventas

Y se obtiene de una técnica estadística llamada "Análisis de Componentes Principales" con la cual se describe la importancia de cada autoservicio.

Es importante mencionar que aunque se tomen variables ajenas a las ventas para construir este índice, análisis posteriores demuestran que se correlacionan directamente con la importancia en ventas.

## VARIABLES A REPORTAR.

La siguiente lista de variables son las características que se auditan en los autoservicios para posteriormente producir el reporte al cliente, cabe mencionar que también en esta lista existen variables que solo sirven para realizar el cálculo posterior de otra variable, tal es el caso de descuentos utilizada para calcular el precio de venta al público.

- 1.- Presencia en el área de ventas
- 2.- Unidades al frente
- 3.- Ubicación en el anaquel
- 4.- Precio regular
- 5.- Precio especial
- 6.- Descuentos
- 7.- Inventario en el área de ventas
- 8.- Inventario en bodega
- 9.- Ofertas del fabricante
- 10.- Número de islas y botaderos
- 11.- Número de cabeceras
- 12.- Actividad promocional
- 13.- Publicidad cooperativa
- 14.- Observaciones especiales
- 15.- Respuestas alfanuméricas

A continuación detallaremos brevemente cada una de estas variables...

### **PRESENCIA EN EL AREA DE VENTAS**

Se refiere al número neto de tiendas con falta total de producto, ya sea para cada tamaño o para el total de la marca en cualquier lugar del área de ventas, a esto también se le denomina agotamiento en el área de venta.

Al referirnos al área de ventas consideramos anaqueles, cabeceras, islas, botaderos, área de cajas o cualquier tipo de exhibidor donde se encuentre la mercancía. Para esto es obligación del auditor recorrer toda la tienda para recabar los datos de los productos o marcas solicitados, con objeto de asegurar que la información levantada no presenta omisiones o reporta agotamientos falsos.



## **NUMERO DE UNIDADES AL FRENTE**

Las unidades al frente es todo aquel producto que se encuentran al frente del anaquel o góndola localizado en la primera y/o segunda hilera del anaquel, ya sea que se encuentren dispuestas en forma vertical u horizontal.

En el caso de refrigeradores abiertos y/o anaqueles en cuya parte inferior solamente se ven las tapas, se consideran las unidades de dos primeros niveles en la parte baja del refrigerador y/o anaquel.

## **UBICACION EN EL ANAQUEL**

Corresponde al nivel de altura en que se localiza cada tamaño o el total marca de los productos a ser auditados dentro del anaquel y/o góndola, esto refiriéndose al lugar físico donde se encuentran; Ejemplo: Alto, Medio, Bajo.

## **PRECIO REGULAR**

Se considera como precio regular al precio localizado en la etiqueta del producto.

En caso de encontrar diferentes precios para un producto de la misma marca y tamaño, se deberá registrar como precio regular el que tenga mayor existencia en el anaquel. Si la diferencia en precios es muy grande, se deberá preguntar al encargado de marcaje, cual de todos es el correcto.

## **PRECIO ESPECIAL**

Se tomará como precio especial a los precios rebajados que se identifiquen con una doble etiqueta (etiqueta roja), debiéndose registrar el precio más bajo.

En caso de que el descuento al precio regular se aplique en caja o sea de otro tipo, se deberá verificar antes, si el precio de la etiqueta incluye ya el descuento anunciado por el material promocional en la tienda.

#### **FORMA DE OBTENER EL PRECIO PROMEDIO**

El procedimiento que se sigue para obtener el precio promedio está basado en el concepto estadístico de mediana. La razón principal por la que se utiliza la mediana para obtener el precio promedio, es que es insensible a valores extremos, ya sean grandes o chicos. (Los valores extremos pueden ser errores al capturar la información o bien datos reales asociados a condiciones excepcionales).

El cálculo de la mediana se define como la observación que cae en el centro cuando las observaciones se ordenan en forma creciente. Si el número de observaciones es par, se escoge como mediana al valor medio entre las dos observaciones de en medio <sup>(4)</sup>.

#### **INVENTARIO EN AREA DE VENTAS:**

Se refiere al conteo de todos los productos localizados en cualquier lugar del área de ventas.

#### **INVENTARIO EN BODEGA:**

Este se realiza de igual manera que el anterior, solo que se hará en la bodega del autoservicio.

#### **OFERTAS DEL FABRICANTE**

Ofertas ofrecidas por el fabricante o detallista, en las que el consumidor deberá obtener algo de valor en efectivo o en la mercancía en el momento de adquirir el producto.

(4) Definición "3.- Estadística Descriptiva." Estadística para administración y economía (Junio 1981).

En este caso únicamente se reportará el tipo de oferta, sin calcular el precio correspondiente, ya que con esta información el sistema realiza el cálculo de aplicación de descuentos sobre el precio correspondiente.

La siguiente es una lista del tipo de ofertas del fabricante que pueden ser auditadas con cada producto:

**1.- Oferta Múltiple:**

Venta de dos o más productos iguales o diferentes como una sola unidad de venta a un precio menor al que se pagaría si se compraran los productos por separado.

**2.- Venta de productos combinados regalando uno de ellos:**

Oferta en que se regala uno de los productos, para identificar cual es el que se está regalando utilizaremos las siguientes palabras:

"Con GRATIS" Si en la compra del producto que auditamos se proporciona uno de regalo.

"GRATIS en" Si el producto que auditamos se regala en la compra de otro producto.

**3.- Cantidad Adicional Gratis:**

Regalo del fabricante al consumidor de cantidad adicional de producto, por el mismo precio que pagaría por el tamaño normal.

**4.- Otras ofertas del fabricante:**

Es necesario describir en la forma de auditoría de que tipo de oferta se trata, ya que pueden existir diversas formas de ofertar los productos.

## NUMERO DE ISLAS Y BOTADEROS

### Islas:

Definamos así a la mercancía que se apila en forma de torre y que generalmente es colocada en lugares que llaman la atención del consumidor.

### Botaderos:

Nos referimos a las canastas que contienen producto en su interior.

## NUMERO DE CABECERAS

Se refiere a las partes laterales del anaquel y/o góndola sin tomar en cuenta los niveles en que se localiza el producto; estas partes de los anaqueles son considerados como exhibidores especiales.

## ACTIVIDAD PROMOCIONAL

Esta variable se refiere a los esfuerzos que realiza el fabricante o detallista para estimular las ventas de algún producto.

Algunas de las promociones más comunes que ofrece el fabricante o detallista son:

### 1.- Demostradoras.

Personas encargadas de distribuir muestras gratis de producto, explican como se usa o resaltan los beneficios del mismo.

### 2.- Exhibidores especiales.

Anaqueles especiales proporcionados por el fabricante, donde aparece la marca impresa en el mismo anaquel y generalmente se hace demostración de uso o prueba de producto, según sea el caso por demostradoras.

### 3.- Material punto de venta.

Se considera dentro de este material a las cenefas, posters, moños, cartulinas, boletines y todo material impreso que se localice en el área de venta.

### 4.- Cupones.

Certificados o títulos al portador que los detallistas o fabricantes dan al consumidor a través de boletines o medios impresos y los cuales tienen cierto valor en la compra de algún producto específico.

### 5.- Otras Promociones.

Cualquier promoción del fabricante o detallista no especificada anteriormente, es necesario describir claramente en la auditoría de que tipo de promoción se trata. Ejemplo: Vehículos vendiendo productos, presentaciones de artistas, etc.

## PUBLICIDAD COOPERATIVA

Nos referimos a la publicidad pagada en forma continua por fabricantes y detallistas.

En esta variable se reporta la existencia de apoyo publicitario para la marca de productos a auditarse y se identificarán los medios utilizados tales como:

**TELEVISION**  
**RADIO**  
**PRENSA**

## OBSERVACIONES ESPECIALES

Observaciones especiales es una variable que podrá responderse de manera cerrada (SI o NO) y estas son preguntas que directamente el cliente solicita y requiere saber de manera concreta.

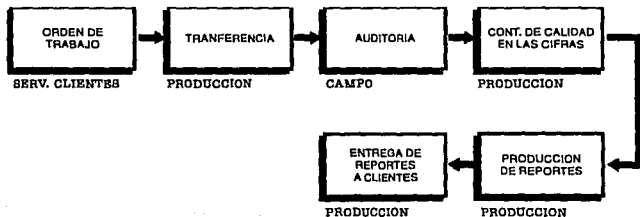
## RESPUESTAS ALFANUMERICAS

Respuestas a preguntas formuladas por el cliente, de tal manera que puedan ser contestadas en forma numérica o alfanumérica (Para la respuesta se tendrán como máximo 9 caracteres).

Aunque estas variables se utilizan normalmente para la producción de reportes del servicio de store-check, esto, no impide que se generen nuevas variables dependiendo de las necesidades de cada cliente.

### IV.IV Flujo de Actividades.

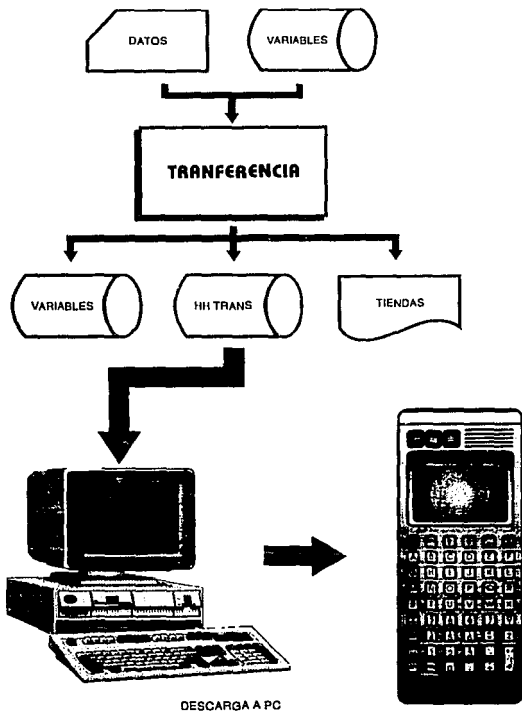
El flujo de actividades para la realización del estudio de store-check se realiza de la siguiente manera:



- 1) El ejecutivo de servicio a clientes llena una orden de trabajo con los requerimientos de cada cliente, de acuerdo a los productos que se necesite auditar.
- 2) Al recibir la orden de trabajo el coordinador de producción, carga los requerimientos en la aplicación para que se genere la transferencia de información para la auditoría.

- 3) Se realiza la carga de hand-helds computers y el área de campo audita cada tienda necesaria para el servicio.
- 4) Una vez terminada la auditoría se regresa la información al main-frame para revisión de cifras y el control de calidad.
- 5) Se realiza la producción de reportes para clientes, se revisan y se entregan a cada cliente y se prepara la aplicación para iniciar un nuevo estudio.

## IV.V Flujo de Transferencia.





El servicio del store-check, como principal fuente de información tiene la auditoría de tiendas, de esta actividad se encarga el área de campo y la realiza vía hand-held computer, para ello es indispensable cargar los archivos necesarios para la auditoría del servicio, una vez que el ejecutivo de servicio a clientes paso a producción la orden de trabajo por cada cliente se realiza un proceso en el sistema del main-frame para obtener los archivos que van en la hand-held computer, estos archivos son transferidos a una microcomputadora, en donde son preparados y convertidos al formato necesario para que al cargarse en las hand-helds computers puedan ser utilizados.

Los archivos necesarios para la auditoría son:

**Transferencia.-** Contiene las características de los productos a auditarse.

**Variables.-** Contiene las instrucciones de la auditoría.

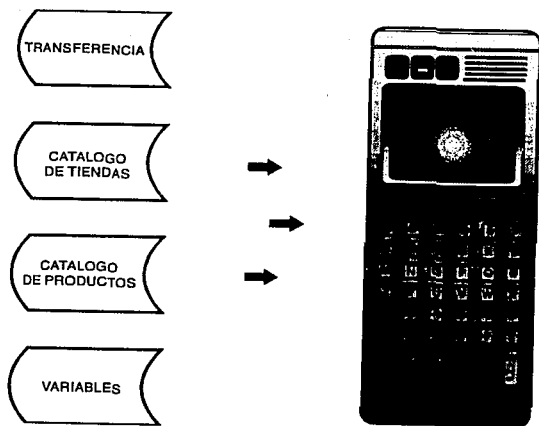
**Tiendas.-** Contiene las características de las tiendas a auditarse.

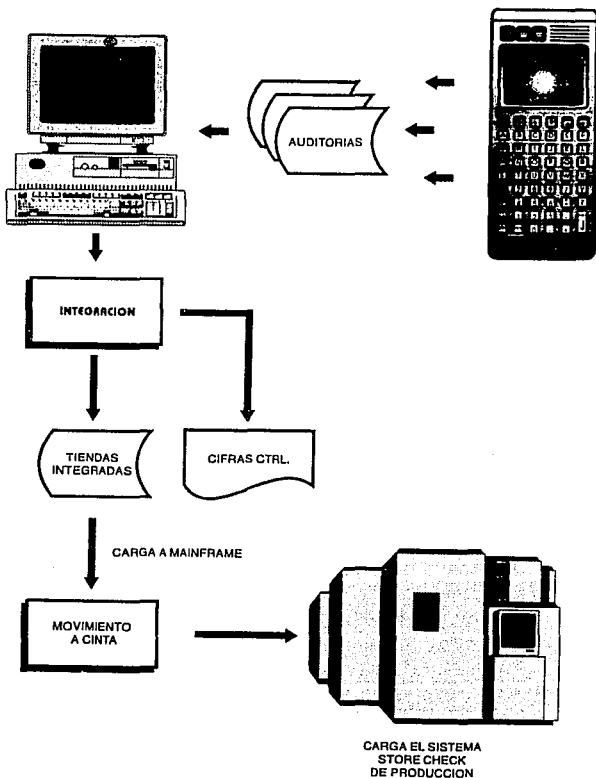
**Clases de Producto.-** Contiene las clases de producto de cada producto.

**Productos.-** Contiene las descripciones de los productos.

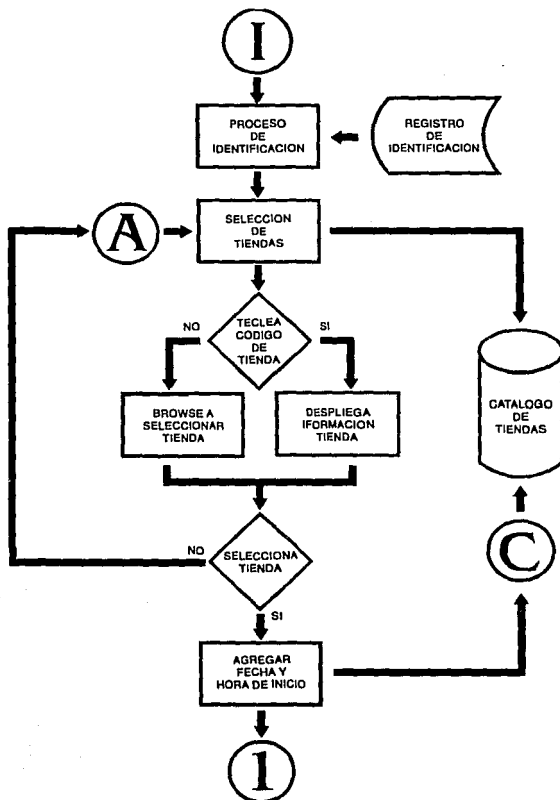
Esta información especifica todos los productos a auditarse y las características de cada preguntas que se solicita en auditoría de cada tienda existente en la muestra para este servicio; las transferencias de los archivos se realizan en un proceso de carga automatizado a cada hand-held computer para la auditoría.

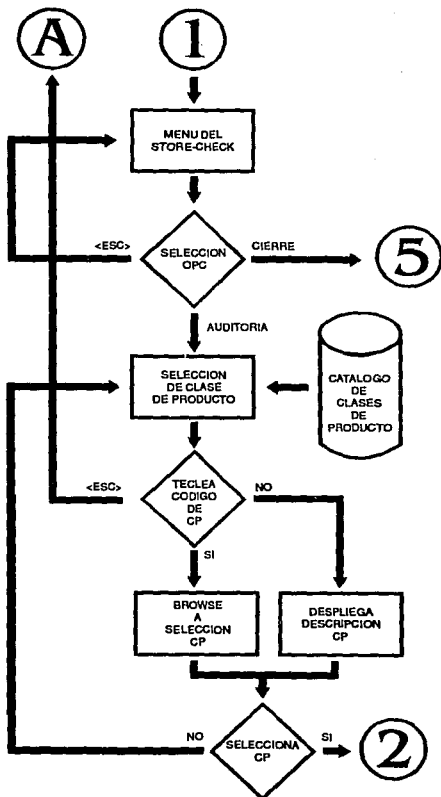
Una vez realizada la auditoría de todas las tiendas en las hand-helds computers se descargan los archivos de auditorías de cada hand-held computer en una computadora personal, este procedimiento al igual que la carga está automatizado, se integran todos los archivos y se reformatean al código ASCII estandar de las pc's para posteriormente ser cargados nuevamente al sistema de producción de reportes en el main-frame.

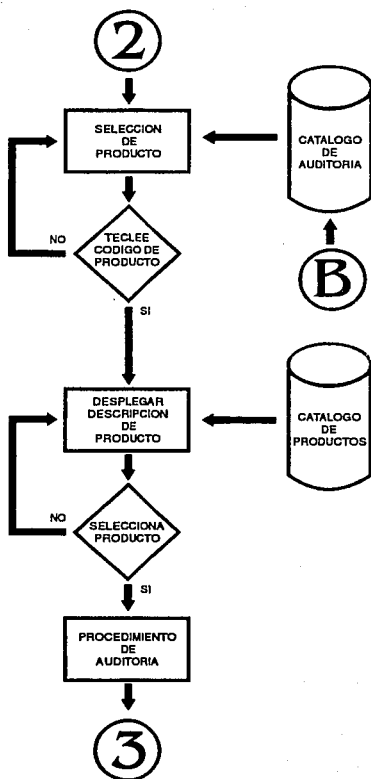


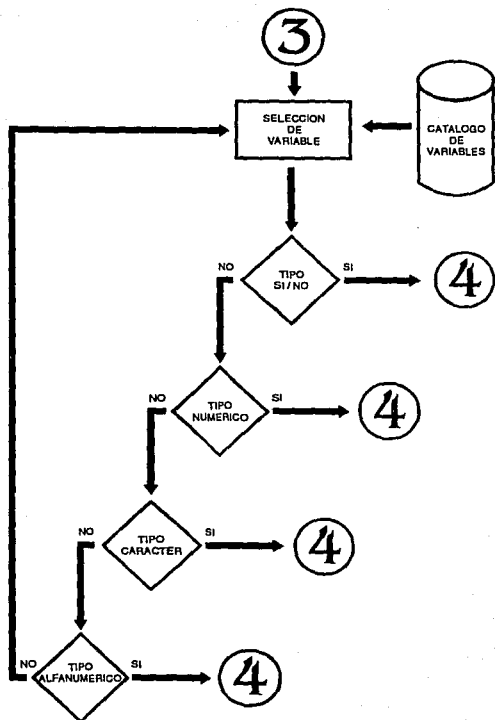


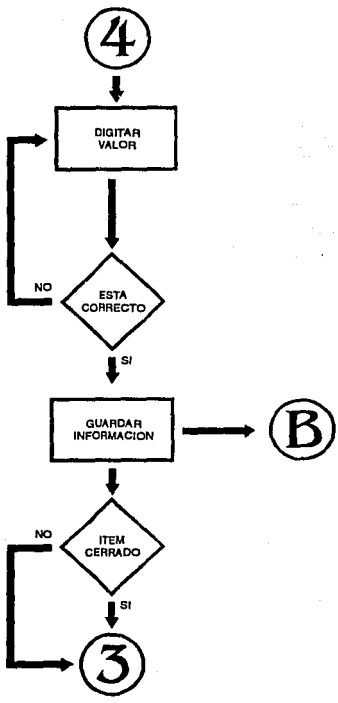
## IV.VI Flujo del programa en OPL/g.



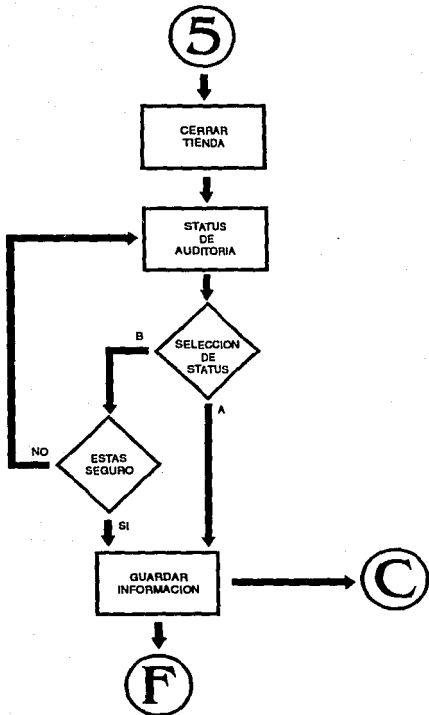












#### IV.VII Operación del sistema Store-Check para HC-110.

##### 1. Cómo acceder la aplicación.

Para poder utilizar la aplicación de la auditoría del servicio de *Store-Check*, se debe estar en la pantalla de selección de tiendas; esta opción es habilitada desde el centro regional por el operador del servicio; está como se muestra en la siguiente pantalla:

07-01-0001 GRUPO AURRERA CLUB AURRERA PERIFERICO N. 32456 SATELITE EDO. DE MEXICO 64380	<ESC>
-----	
<S> - Selecciona Tienda . . . .	

##### 2. Selección de tienda.

En la pantalla de la hand-held computer se despliega en secuencia las tiendas que están asignadas al auditor responsable de cada una de esas tiendas, pero es importante leer la descripción de cada una de ellas para hacer la selección correcta de acuerdo a la tienda que se necesite auditar en ese momento.

##### Objetivo:

Seleccionar en la hand-held computer la tienda que se va a auditar, dependiendo de su descripción: (Cadena a la que pertenece, Nombre de la tienda, dirección, etc.).

**Instrucciones:**

1. Una vez accesada la aplicación de **Store-Check**, en la hand-held computer se despliega la primera tienda disponible, como se puede observar en la siguiente pantalla:

07-01-0001	
GRUPO AURRERA	
CLUB AURRERA	
PERIFERICO N. 32456	
SATELITE	
EDO. DE MEXICO	
64380	<ESC>
-----	
<S> - Selecciona Tienda . . . .	

2. Presionar <ENTER> para que se despliegue la siguiente tienda.

**Note:** Cuando una tienda ya fue auditada en la parte superior derecha se despliega el status que se le asignó: **Auditada** (Auditoría Completa) o **NO Auditada**.

Las Tiendas con status **NO Auditada** Si pueden volver a seleccionarse para volver a trabajarla.

07-01-0001	AUDITADA
GRUPO AURRERA	
CLUB AURRERA	
PERIFERICO N. 32456	
SATELITE	
EDO. DE MEXICO	
64380	<ESC>
-----	
<S> - Selecciona Tienda . . . .	

07-01-0001	NO AUDITADA
GRUPO AURRERA	
CLUB AURRERA	
PERIFERICO N. 32456	
SATELITE	
EDO. DE MEXICO	
64380	<ESC>
-----	
<S> - Selecciona Tienda . . . .	

3. Cuando en la pantalla se despliegue la descripción de la tienda requerida, para ser auditada presionar <S> y la tienda queda seleccionada.

**Nota:** una vez seleccionada la tienda **NO** se puede cancelar la selección, ni retroceder en la aplicación, únicamente cerrando la tienda, según corresponda.

4. Se despliega el mensaje de:

***Espere un momento. . .  
Cargando Transferencia***

### 3. El menú del Store-Check.

El Menú del Store-Check, es el enlace para la ejecución de los dos procesos posibles a realizar para cada una de las tiendas seleccionadas, los cuales son:

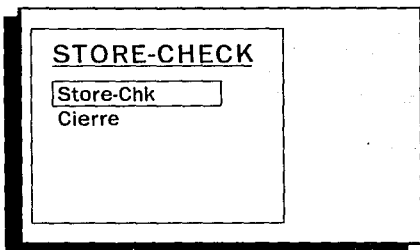
- \* **Store-Check** : Proceso para el levantamiento de información de la auditoría.
- \* **Cierre** : Conclusión de la auditoría por medio de validaciones y estatus.

#### **Objetivo:**

Permitir la selección de la actividad a realizar: **Store-Check** o **Cierre**.

**Instrucciones:**

1. Una vez seleccionada la tienda a ser auditada se despliega la siguiente pantalla:



2. Se necesita posicionar el cursor, con la flecha ascendente o descendente, en el renglón de la actividad a realizar y presionar **<ENTER>** para seleccionar.

3. Otra alternativa para la selección directa de las opciones es presionar **<S>** para seleccionar **Store-Chk**, o **<C>** para seleccionar el **Cierre**.

La selección de **Store-Chk**, es la auditoría del servicio y la selección de **Cierre** es el cierre de auditoría.

**4. Auditoría.**

En esta sección se describen los pasos necesarios para llevar a cabo la auditoría completa de un producto o ítem, por lo que la repetición de ese conjunto de pasos permite efectuar una auditoría completa, de todas las clases de producto requeridas para el estudio en particular y de todos los productos o ítems de las mismas clases de producto.

**Objetivo:**

Definir detalladamente la secuencia de pasos que se deben ejecutar para la auditoría de un producto de una clase de producto determinada.

#### 4.1. Selección de la clase de producto

##### Objetivo:

Seleccionar la clase de producto a auditar.

##### Instrucciones:

1. Una vez seleccionada la operación de **Store-Chk** del menú, se despliega la siguiente pantalla:

CLASE DE PRODUCTO
Código : [    ]
<ESC>
Código - Clase de producto por Auditor

2. Se necesita teclear el código de la clase de producto a seleccionar y presionar <ENTER>.
3. En caso de no conocer el código correspondiente, se puede dejar en blanco el campo y presionar <ENTER> para que se desplieguen en secuencia, las clases de producto disponibles para ese estudio en particular.
4. En caso de teclear un código de clase de producto no existente, la aplicación permanece en la misma pantalla, por lo que se debe reintentar la operación hasta digitar un código existente de clase de producto.

5. Una vez teclado el código de la clase de producto o presionado **<ENTER>** para el despliegado una a una de todas las clases de producto del estudio, se despliega la descripción correspondiente como se muestra en la siguiente pantalla:

<p>CLASE DE PRODUCTO</p> <p>017 SHAMPOO</p> <p style="text-align: right;"><b>&lt;ESC&gt;</b></p> <hr/> <p><b>&lt;S&gt;</b> - Selecciona Clase . . .</p>
---

6. Cuando se despliegue la clase de producto requerida, presionar **<S>** para seleccionarla.

7. Si no se requiere continuar con la selección de la clase de producto, cuando esté desplegada la descripción de la misma o estando en la pantalla donde se debe teclear el código correspondiente, presionar **<ESC>** y la aplicación regresará al **Menú del Store-Check**.

#### **4.2. Instrucciones de la clase de producto.**

Las instrucciones se refieren a las especificaciones que se deben seguir para la auditoría de una clase de producto determinada, los cuales varían en cada estudio, ya que dependen completamente de los requerimientos del cliente.

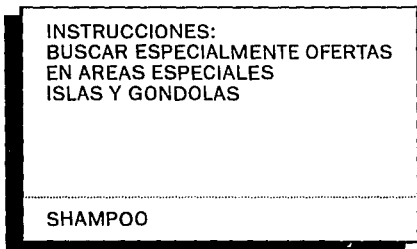
Estas instrucciones siempre van asociadas a una clase de producto en particular, es decir, pueden existir tantas instrucciones como clases de producto contenga dicho estudio, pero además también pueden existir clases de producto sin especificaciones.

**Objetivo:**

Especificar instrucciones que se deben considerar para una clase de producto.

**Instrucciones:**

1. Una vez seleccionada la clase de producto a auditar, enseguida se despliega la pantalla de instrucciones, como se muestra en la siguiente pantalla:



2. Presionar <ENTER> para continuar con la selección de productos o ítems.

**4.3. Selección de productos.**

Es importante recordar que en la hand-held computer, todos los productos o ítems están organizados en función de la clase de producto a la que pertenecen, por lo que para trabajar un producto, se debe seleccionar previamente su clase de producto; o en su defecto, ya estando trabajando con una clase de producto en específico, únicamente se podrán registrar datos para los productos de esa clase de producto en particular.

**Objetivo:**

Especificar el producto a auditar de la clase de producto seleccionada previamente.



**Instrucciones:**

1. Se debe seleccionar el producto a auditar de acuerdo a la siguiente pantalla:

ITEMS
Número : [    ]
<ESC>
Número del Item en el Reporte de Transferencia

2. Se requiere de consultar en el reporte de transferencia el **número consecutivo**, que le corresponde al producto que se requiere seleccionar.
3. Posteriormente, se debe teclear el número consecutivo correspondiente.
4. En caso de no querer continuar, presionar **<ESC>** y la aplicación regresa a la pantalla para la selección de la clase de producto.
5. En caso de teclear un número de producto no existente, la aplicación permanecerá en la misma pantalla hasta que se le teclee un número existente.
6. Cuando el número consecutivo tecleado existe, se despliega la descripción del producto, como se muestra en la siguiente pantalla:

ITEM 0002.-017-1-00210 01

SHAMPOO PERT PLUS REGULAR  
400ML

<ESC>

<S> - Selecciona Item . . .

Donde:

\* ITEM **0002.-017-1-00210 01.-** se refiere al código completo del producto seleccionado, de la siguiente manera:

<b>0002.-</b>	Número consecutivo del producto
<b>017 -</b>	Clase de producto
<b>1 -</b>	Tipo de producto
	1 - Sencillo
	2 - Marca
	3 - Observaciones especiales
	4 - Respuesta Alfanumérica
	5 - Combinado
<b>0210</b>	Código asignado al producto para la producción
<b>01</b>	Grupo de variables

\* **Shampoo Pert Plus Regular 400 ml.-** es la descripción del producto correspondiente.

7. Cuando se tecldea el número consecutivo de un **ITEM CERRADO**, se despliega la siguiente pantalla:

ITEM 0002.-017-1-00210 01

SHAMPOO PERT PLUS REGULAR  
400ML

ITEM CERRADO . . . <ESC>

<S> - Selecciona Item . . .

Ese mensaje significa que ese producto ya tiene registrado algún valor en cada una de sus variables, pero se puede volver a seleccionar para registrar o actualizar nuevos datos.

8. Cuando la descripción desplegada corresponda a la que se pretende auditar, presionar <S> para seleccionar el producto.

9. En caso contrario, presionar <ESC> y la aplicación regresa a la pantalla para la selección de productos.

#### 4.4. Las variables a auditar

Las variables a auditar son las características que se van a calificar por cada producto en el estudio vigente. Como se vió anteriormente estas variables dependen de la información que el cliente quiera obtener para cada clase de producto en particular.

Como su nombre lo indica, estas pueden variar para cada estudio y para cada clase de producto. Por ello en cada estudio se especifica un grupo de variables a utilizar.

Existen diferentes tipos de variables, los cuales se distinguen por la respuesta que le corresponde, como se explica a continuación:

a) Variable con respuesta **SI o NO**.-

Concepto que en su respuesta requiere únicamente de una **S** o **N** para contestar afirmativa o negativamente.

b) Variable con respuesta **Númerica**.-

Concepto que en su respuesta los valores posibles son números (máximo 9 dígitos).

c) Variable con respuesta **Porcentual**.-

Concepto que su respuesta requiere valores numéricos con decimales, se toman como porcentajes y contiene el signo de % posterior.

d) Variable con respuesta **Monetaria**.-

Concepto que su respuesta requiere valores numéricos con decimales, se toman como cantidades en moneda circulante y contiene el signo de **N\$** antepuesto.

e) Variable con respuesta **Alfanumérica**.-

Concepto que requiere en su respuesta la combinación de letras y/o números, máximo 9 posiciones por cada una.

f) Variable con respuesta **Caracter**.-

Concepto que en su respuesta tiene diferentes alternativas, y cada una de ellas identificada por una letra o un número a seleccionar dependiendo de su valor especificado.

**Objetivo:**

Registrar los datos de cada variable para el producto seleccionado.

**Instrucciones:**

1. Una vez seleccionado el producto, en pantalla se despliegan los siguientes datos:

ITEM 0002.-017-1-00210 01

SHAMPOO

1. PRESENCIA
2. UNID. AL FRENTE
3. UBICACION
4. PRECIO REGULAR

<ESC>

Seleccione Variable:

ITEM 0002.-017-1-00210 01

SHAMPOO

5. PRECIO ESPECIAL
6. DESCUENTOS
12. INV. A DE VTA.
13. INV. EN BODEGA

<ESC>

Seleccione Variable:

2. Cuando las variables son más de cuatro, se debe presionar **<ENTER>** para pasar a la siguiente pantalla y así hasta recorrer el total de variables a auditar.

Nótese que cuando existe otra pantalla de variables el mensaje aparece de la siguiente forma:

**+ *Seleccione Variable:***

En caso de que no existan más variables en el mensaje se omite el signo de +, quedando de la siguiente manera:

***Seleccione Variable:***

Este tipo de pantallas son cíclicas ya que al llegar al final la siguiente pantalla es la primera, todo esto es posible solo presionando <ENTER>.

3. Cuando se requiera salir de la pantalla de variables, ya sea por haber terminado la auditoría del producto seleccionado o por querer cancelar la operación, presionar <ESC> y se desplegará el siguiente mensaje:

***INFORMACION CORRECTA (S)I / (N)O...***

4. Se necesita de responder a la pregunta, si se presiona <S> es para confirmar que la información que se auditó para ese producto es correcta, con ello la aplicación regresa a la pantalla de selección de productos.

5. En caso contrario si se presiona <N> para notificar que la información no es correcta, la aplicación permanecerá en la pantalla de variables de ese producto.

***4.4.1. Variable con respuesta SI o NO:***

Para este tipo de variable se va a tomar como ejemplo la variable... ***Presencia*** .

1. Al seleccionar la variable a auditar, presionando el número que le corresponde, aparece la siguiente pantalla:

ITEM 0002.-017-1-00210 01

1. PRESENCIA

SI NO

[ ]

<ESC>

Teclee (S)i/(N)o . . .

2. Solo necesita presionar <S> o <N> para indicar que es afirmativa o negativa la respuesta para la variable de este producto.
3. La aplicación continuará con la siguiente variable secuencialmente si es que existe otra variable y que además no tenga la respuesta en su área de datos.

**4.4.2. Variable con respuesta Numérica:**

Para este tipo de variable se va a desarrollar como ejemplo la variable...  
***Inventario en el área de ventas.***

1. Al seleccionar este tipo de variable aparecerá la siguiente pantalla:

ITEM 0002.-017-1-00210 01

12. INVENTARIO A. VENTAS

[0]

0

<ESC>

Teclee la Cantidad a Agregar

2. Se debe teclear la cantidad de unidades a registrar y presionar <ENTER>.

**Nota:** Se pueden teclear cantidades negativas, las cuales serán restadas del **Total acumulado de la sesión**.

Por ejemplo:

```

ITEM 0002.-017-1-00210 01
12. INVENTARIO A. VENTAS
[000000008]
[12]           0
                0
                12
                                <ESC>

Teclee la Cantidad a Agregar
  
```

Donde los valores corresponden a lo siguiente:

**[0000008]** Total acumulado del producto

**[12]** Total acumulado en la sesión

**0** 3a. posición (cantidad parcial de la sesión)

**0** 2a. posición (cantidad parcial de la sesión)

**12** 1a. posición (cantidad parcial de la sesión)

El valor del **total acumulado en la sesión** no queda registrado como definitivo, es decir, no se adiciona al **total acumulado del producto**, sino hasta que se presiona <ESC> para salir de ésta variable.

Las cantidades parciales que se van registrando se recorren en las 3 posibles posiciones de la cantidad parcial de la sesión según corresponda, a la vez que se van adicionando al total acumulado de la sesión.



Esto es, si se tecldea otra cantidad parcial, por ejemplo de 25, el resultado sería el siguiente:

<b>[00000008]</b>	<b>0</b>
<b>[37]</b>	<b>12</b>
	<b>25</b>

Es decir, la primera cantidad tecldeada, 12, pasó a la segunda posición, y la última que se tecldeó a la primera posición, y el **total acumulado de la sesión** se incrementó.

Si se tecldea una nueva cantidad las existentes se recorrerán una posición más, se incrementa el **total acumulado de la sesión** y así sucesivamente hasta suspender la sesión.

Esta forma de presentar las variables con respuesta numérica, permite ver las cantidades parciales que se registran, lo que permite llevar un mejor control de los conteos de unidades.

3. Una vez registradas las unidades correspondientes al concepto de la variable, presionar <ESC> para dar por terminada la sesión de esa variable, y en este momento es cuando se incrementa el **total acumulado del producto**.

#### **4.4.3. Variable con respuesta Porcentual:**

Para este tipo de variable se va a desarrollar como ejemplo la variable... **Descuento**.

1. Al seleccionar la variable se despliega la siguiente pantalla:

ITEM 0002.-017-1-00210 01
5. DESCUENTOS
[00.00 %]
[00.00 %]
<ESC>
Teclee el Descuento

2. Para teclear la cantidad porcentual se debe realizar de la siguiente manera:

***Unidades (2) - punto decimal - decimales (2)***

3. En caso de requerir registrar otro porcentaje para el mismo producto, ya sea sin haber salido de esta pantalla o ya sea volviendo a seleccionar la variable, la nueva cantidad sustituye a la anterior, quedando siempre la última cantidad teclada.

**Nota:** En caso de omitir los decimales la HHC los toma como **CERO**.  
La hand-held computer **no** suma los porcentajes, únicamente los sustituye.

4. Para salir de esta variable solo se debe presionar <ESC> y la aplicación continuará.

#### ***4.4.4. Variable con respuesta Monetaria:***

Para este tipo de variable se va a desarrollar como ejemplo la variable... **Precio Regular**.

1. Al seleccionar esta variable se desplegará la siguiente pantalla:

ITEM 0002.-017-1-00210 01

4. PRECIO REGULAR

[N\$ 0.0]

N\$ 0

<ESC>

Teclée el Precio Regular.

2. Para teclear la cantidad en nuevos pesos, se debe realizar de la siguiente manera:

**Unidades (4) + punto decimal + decimales (2)**

**Nota:** En Precios no se puede omitir decimales, cuando la cantidad no tenga centavos, se deben teclear forzosamente uno de los decimales como CERO, de otra forma la hand-held computer registrará la cantidad completa en CERO.

3. En caso de requerir registrar otro precio para el mismo producto, ya sea sin haber salido de esta pantalla o ya sea volviendo a seleccionar la variable, la nueva cantidad sustituye a la anterior, quedando siempre la última tecléada.
4. Para salir de esta variable solo se debe presionar <ESC> y continuará.

#### **4.4.5. Variable con respuesta Alfanumérica:**

Para este tipo de variable se va desarrollar como ejemplo la variable...  
**Respuesta Alfanumérica .**

1. Al seleccionar la variable se desplegará la siguiente pantalla:

ITEM 0002.-017-1-00210 01

14. RESP. ALFANUM.

¿De qué color estaban  
vestidas las demostradoras?

[                    ]

[   ]

2. Solo deberá teclear la respuesta correspondiente con cualquier caracter o número.

La respuesta que usted teclee va a desplegarse dentro del corchete del último renglón.

El corchete del penúltimo renglón se refiere a la contestación definitiva de la auditoría, es decir, si se despliega algún valor en dicho corchete, significa que había una respuesta previa y se desea modificar.

Si aparece en blanco tal corchete, significa que no había ninguna respuesta previa y que la que se esté tecleando en esta ocasión, será la que se tomará como definitiva de la auditoría al salir de esta pantalla.

4. Cuando la respuesta esté completa, presionar <ENTER>.
5. En caso de requerir registrar otra respuesta para el mismo producto, ya sea sin haber salido de esta pantalla o ya sea volviendo a seleccionar la variable, la nueva respuesta sustituye a la anterior, quedando siempre la última tecleada.
6. Presionar <ESC> para salir de esta variable y continuar.

**4.4.6. Variable con respuesta Caracter:**

Para este tipo de variable se va a desarrollar como ejemplo la variable...  
**Ubicación.**

1. Al seleccionar la variable de este tipo aparecerá la siguiente pantalla:

```
ITEM 0002.-017-1-00210 01
3. UBICACION
A M B
[ ]
<ESC>
Teclee el Caracter Válido
```

Donde **A M B**, se refiere a las alternativas de respuesta para la variable, las cuales pueden ser numéricas o alfabéticas como en este caso.

**Note:** Las alternativas de respuesta son complementarias, no excluyentes, por lo que se pueden utilizar más de una de ellas para el mismo producto.

2. Se debe presionar los caracteres correspondientes a la respuesta correcta.

Para activar como respuesta un caracter, presionarlo una vez.

Para desactivar como respuesta un caracter activado, presionarlo por segunda vez.

3. Cuando no corresponda como respuesta ninguna de las alternativas de la variable, presionar **<\*>** (asterisco), para indicarle a la hand-held computer que se registró un valor inexistente.

**Note:** Una respuesta de tipo caracter *no* debe quedar sin valor, al menos debe quedar con asterisco.

4. La respuesta que usted teclee va a desplegarse dentro del corchete del último renglón.

5. Presionar <ESC> para salir de esta variable y continuar.

#### 4.5. Generalidades de la aplicación en la auditoría.

Dado que el servicio se maneja con diferentes variables, de acuerdo a los requerimientos del cliente, esto obliga a que la hand-held computer tenga un funcionamiento estandar, aunque los conceptos de las variables y la relación lógica entre ellas cambie.

##### 4.5.1. Cómo saber las respuestas de cada variable.

La Auditoría consiste en responder cada variable de cada producto; para controlar dichas respuestas la hand-held computer despliega el valor de cada una como lo muestra la siguiente pantalla:

ITEM 0002.-017-1-00210 01

SHAMPOO

1. PRESENCIA	SI
2.UNID. AL FRENTE	45
3. UBICACION	AMB
4. PRECIO REGULAR	14.05

<ESC>

Seleccione Variable:

Cada variable en su mismo renglón tiene su propia respuesta.

El siguiente ejemplo presenta las respuestas posibles que la hand-held computer asigna automáticamente cuando se determinó que no había presencia del producto en esa tienda.

```

ITEM 0002.-017-1-00210 01
SHAMPOO
1. PRESENCIA           NO
2.UNID. AL FRENTE     000000000
3. UBICACION          *
4. PRECIO REGULAR     000000000
                        <ESC>
Seleccione Variable:
  
```

```

ITEM 0002.-017-1-00210 01
SHAMPOO
5. PRECIO ESPECIAL    000000000
6. DESCUENTOS         000000000
12. INV. A DE VTA.    000000000
13. INV. EN BODEGA    000000000
                        <ESC>
Seleccione Variable:
  
```

La computadora asigna directamente el valor **NULO** correspondiente a cada variable.

**NOTA:** El valor de cada respuesta se va modificando de acuerdo a los datos registrados durante la auditoría.

## 5. Cierre de auditoría.

Para efectos de control en el funcionamiento de la hand-held computer, una tienda que ha sido seleccionada para ser auditada debe recibir un *status*. Con esto se determina el estado en el que queda la información registrada para dicha tienda, es decir, con **Auditoría Completa** o **NO Auditada**.

Este status debe ser asignado a través del módulo de **Cierre** de la aplicación.

El status de **Auditoría Completa** se debe asignar a aquellas tiendas en las que se auditaron todos las variables de todos los productos.

Cuando se asigna este status la hand-held computer verifica que todas las variables de cada producto tengan registrado algún dato o el valor nulo, si algún producto no cumple con dichas características no se le puede asignar el status de **Auditoría Completa** sino hasta que se auditen todas las variables.

El status de **NO Auditada** se debe asignar a aquellas tiendas que por alguna razón no tienen ningún dato registrado en las variables de sus productos o porque está incompleta y ya no se va a poder seguir trabajando, al menos por el momento.

La hand-held computer asigna este status en forma directa, sin verificar los datos registrados hasta el momento, únicamente verificando si el usuario está seguro de ello, ya que dicho status **BORRA** los datos que hasta el momento se habían registrado y deja a la tienda como en un inicio, sin registros en las variables de todos los productos.

Una tienda con status **NO Auditada** puede volver a seleccionarse para realizar desde el inicio la auditoría.

### Objetivo:

Asignar un status a la tienda seleccionada, de acuerdo a los datos registrados y con ello dar por terminado el trabajo de auditoría para la misma.

### Instrucciones:

1. Estando en el **Menú del Store-Check**, seleccionar la opción de **Cierre**.
2. Se despliega la pantalla para la asignación del status que corresponda a la tienda que está seleccionada, como lo muestra la siguiente pantalla:



<p>STATUS-AUDITORIA</p> <p>A. Auditoría Completa</p> <p>B. NO Auditada</p> <p style="text-align: right;">&lt;ESC&gt;</p> <hr/> <p>Teclee Status de la Tienda</p>
--

3. Para asignar a la tienda un status de **Auditoría Completa** se deberá presionar <A> para asignar a la tienda un status de **NO Auditada** se deberá presionar <B>.

En caso de asignar el status de **Auditoría Completa**:

Se despliega un instante el mensaje de:

***Espere un momento. . .  
Verificando Auditoría.***

La hand-held computer verifica que las variables de los productos tengan algún valor. Cuando encuentra alguna(s) sin valor de auditoría, despliega el producto con dicha(s) variable(s), como lo muestra la siguiente pantalla:

ITEM 0002.-017-1-00210 01

1. PRESENCIA
2. UNID. AL FRENTE
3. UBICACION
4. PRECIO REGULAR
5. PRECIO ESPECIAL
6. DESCUENTOS
9. OFERTAS FABRIC.

<ESC>

Si se requiere trabajar el producto para poder cerrar la tienda, presionar <S> para seleccionarlo.

Continuar con los pasos normales para la auditoría de un producto y automáticamente después regresará a continuar la verificación.

Una vez que se terminen de auditar los productos pendientes, y de que se verifique totalmente la tienda, se le asigna automáticamente el status de **Auditoría Completa** y se envía el siguiente mensaje:

TIENDA CON AUDITORIA

COMPLETA . . .

Cualquier tecla para Cont.

En caso de asignar el status de **NO Auditada**:

Se despliega el mensaje de:

***Está seguro de dar el  
Status de Tienda NO  
Auditada... (S) / (N)o.***

Presionar <S> para asignar el status de **NO Auditada**.

En este momento la aplicación asigna automáticamente el status de **NO Auditada** y regresa a la pantalla de **Selección de Tiendas**.

En caso de no requerir asignar el status de **NO Auditada** a la tienda seleccionada, presionar <N>.

La aplicación regresa al **Menú del Store-Check**, permitiendo asignar otro status o seguir dentro de la auditoría de la tienda.

Una vez asignado el status de auditoría la aplicación permite continuar con la auditoría de otra tienda.

## **CONCLUSIONES**

Como resultado de la difusión de las microcomputadoras, los equipo de cómputo han reducido significativamente sus costos y aumentado espectacularmente su capacidad; -- Si la industria automotriz hubiera hecho lo que la industria de las computadoras ha hecho en los últimos 30 años, un Rolls Royce costaría dos dolares y medio y rendiría un millón de kilometros por litro. --

Este fenómeno ha permitido a las organizaciones medianas y pequeñas, así como a personas físicas, adquirir computadoras con el propósito de hacer más productiva la actividad que desarrollan. En el caso de las empresas grandes, este fenómeno les a permitido ampliar la capacidad de cómputo al incorporar microcomputadoras, expandiendo y diversificando con ello sus procesos de información.

Es importante determinar, dada la situación económica actual del país, el interés de las empresas para que su inversión en tecnología de cómputo avanzada, sea más productiva; por eso, considero a las hand-helds computers como una herramienta muy importante en el crecimiento económico y productivo, no solo de las empresas sino del país. De igual manera existen otro tipo de herramientas que alcanzan el mismo objetivo, pero considero de manera especial, a las hand-helds computers como un equipo que compite tanto en precio como en eficiencia y productividad con cualquier otra herramienta en su tipo.

El conjunto de evoluciones tecnológicas en el mundo está causando un aceleramiento en la participación de la informática en todas las actividades del ser humano. Nuestro país no está ajeno a este explosivo desarrollo, se sigue fortaleciendo de toda una infraestructura nacional que se irá ampliando a medida que el país vaya consolidando sus estrategias de crecimiento, por el lado de quien utiliza estas herramientas, es necesario que siga contando con una amplia variedad de equipos y opciones, como las hand-helds computers, para satisfacer las necesidades de información de una manera más rápida y eficiente.

En lo social los impactos de la utilización de esta tecnología pueden acarrear mejores y mayores oportunidades para cualquiera de los trabajadores, ya que la profesionalización de los oficios derivará en mayor capacitación e innovación en el trabajo, todo esto gracias a la utilización de avanzados sistemas de información que permiten realizar cualquier actividad en un menor tiempo y a un menor costo.

En la década de los noventas se inicia algunos cambios en las relaciones económicas y políticas internacionales de México con sus vecinos de América del Norte, Estados Unidos y Canadá. Ya que para finales de 1990 decidieron iniciar pláticas para la formación de una área de libre comercio en América del Norte. De concretarse, el acuerdo trilateral de libre comercio será un paso significativo en el desarrollo tecnológico, económico y político de nuestro país, ya que obtendremos de una manera más rápida y barata los adelantos en esta y cualquier otra tecnología, además con las posibilidades de desarrollar nuestra propia infraestructura de desarrollo.

Este acontecimiento nos invita a mejorar nuestros sistemas de información, automatizar y optimizar nuestro procesos de producción al máximo donde podamos competir con nuestros futuros socios comerciales con las mismas herramientas, ganando tiempo y eficiencia.

Por otro lado, es necesario fortalecer proyectos que favorezcan el uso de la nueva tecnología, ya que los empresarios mexicanos deben de demostrar su madurez para afrontar los rápidos cambios que se pueden efectuar con la ayuda de las hand-helds computers, no solo para entender las ideas, que vienen del extranjero con respecto a esta tecnología, sino por el contrario, para tratar de mejorarlas y así, lograr en un futuro no muy lejano el desarrollo de tecnología mexicana que se utilice en el extranjero; demostrando un crecimiento económico nacional iniciado por el espíritu emprendedor de los mexicanos.

Dado que en México se cuenta ya con la experiencia en el diseño y operación de esta tecnología y dado el desarrollo en otras actividades de la informática se podría favorecer y hacer uso de una capacidad exportadora en los servicios de información que cada empresa ofrece.

En este contexto se presenta para México un reto y a la vez una gran oportunidad, de aprovechar la capacidad innovadora para producir desarrollos en esta tecnología, de gran calidad que satisfagan las necesidades internas del país y posteriormente puedan ganar mercados a nivel internacional.

En síntesis, las hand-helds computers tienen una sólida base y gran potencial para desarrollarse y contribuir al progreso del país y de cada empresa en lo particular. Es evidente enfatizar la relevancia que ha alcanzado esta tecnología en las actividades productivas de las empresas; asimismo, la informatización y los nuevos servicios especializados en comunicaciones son rasgos de un esquema moderno de difusión de la informática en los sectores productivos del país.

Todos estos hechos me motivaron para realizar el presente análisis, el cual pretende dar una visión actualizada y completa sobre esta nueva tecnología y sus aplicaciones; además, las hand-helds computers me hicieron ver la necesidad de la buena utilización del desarrollo de sistemas en el área de mercadotecnia donde poco se ha hecho, debemos recordar que la informática es una ciencia multidisciplinaria la cual permite el estudiante poder desarrollar un trabajo en prácticamente cualquier área.

Espero que la tesis pueda ser leída por todos aquellos estudiantes interesados en el desarrollo de la informática en cualquier área de servicio y en particular en el tema de las hand-helds computers.

Proporcionar lo que considero puede ser toda una gama de posibilidades en el desarrollo profesional relacionado con la informática es de suma importancia ya que día con día el mundo demanda más y mejor realización de todas nuestras actividades.

## **BIBLIOGRAFIA.**

---

### **Historia de la Computación.**

Ing. Rodrigo Guerra Botello.  
Lic. Héctor Meza Curiel.  
IBM de México  
1987

### **Directorio de Empresas de la Asociación Mexicana de Fabricantes de Bienes Informáticos A.C.**

AMFI  
1991

### **Business Week.**

Revista  
Zoom! Here Come the new micros.  
Diciembre 1986

### **Una Metodología para la Selección de Equipos de Cómputo.**

Eduardo Ojeda Trejo.  
Tesis de la Fac. de Ciencias  
1976

### **Ingeniería del Software.**

Richard Fairley.  
1987

### **PSION HC RANGE.**

PSION PLC  
1991

### **GUIDE TO THE PTC-860TC.**

TELXON Corporation.  
1991

### **PSION NEWS.**

PSION PLC  
1992

**Operating Systems.**

Harvey M. Deitel.  
Addison-Wesley Boston College  
1990

**Dentro y Fuera del Ordenador.**

Antonio M. Ferrer A.  
INGELEK  
1985

**La Modernización Nacional y La Inversión Extranjera.**

Lic. Mauricio Rossell.  
Porrúa  
1991

**México y el TLC.**

Dr. Eduardo Andere.  
Dra. Georgina Kessel.  
ITAM  
1992

**HP 95LX Guía de Usuario.**

Hewlett Packard  
1992

**Store-Check (ISOS).**

A.C. Nielsen Co.  
1992

**New Hand Held Computer Training.**

Nielsen University.  
1992

**COBOL Estructurado.**

Andreas S. Philippakis.  
Leonard J. Kazmier.  
Mc Graw Hill.  
1988



**Introducción a la Programación y Solución de Problemas con PASCAL.**

G. Michael Schneider.  
Steven W. Weingart.  
LIMUSA.  
1986

**Decisión BIT.**

Revista  
Problemas del desarrollo de software.  
Abril 1988

**Estadística para administración y economía.**

Mendenhall.  
Reinmuth.  
Wadsworth/Iberoamérica.  
1981