5
1ej

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

MANEJADOR Y DISEÑADOR DE PANTALLAS EN AMBIENTE UNIX

TESIS PROFESIONAL

Que para obtener el Título de

INGENIERO EN COMPUTACION

p r e s e n t a

VICTOR HUGO AMAYA GALVAN

Dir. de Tesis:

Ing Alejandro Ramírez Lozada

México,D.F. 1993

UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

MANEJADOR Y DISEÑADOR DE PANTALLAS EN AMBIENTE UNIX

Objetivos:

Crear un manejador y un diseñador de pantallas que sustituya al EASEMAP que se encuentra en los equipos Cromemco, los cuales deberán ser capaces de utilizar los formatos de pantalla generados por éste. Dichos programas serán utilizados por el compilador RMCobol 85 en ambiente UNIX y podrán ejecutarse en diferentes tipos de terminal.

CONTENIDO

I.-INTRODUCCION

ANTECEDENTES DEL EASEMAP.

EASEMAP UNIX es una herramienta para la administración, diseño, generación y manejo de pantallas en programas escritos en lenguaje COBOL bajo ambiente UNIX.

Las ventajas que ofrece este paquete, permiten al programador de aplicaciones, una reducción sustancial en el tiempo de diseño y elaboración de código para pantallas; y al usuario final, una gama de funciones para que la captura y validación de datos resulte una tarea más sencilla.

EASEMAP UNIX cuenta con un mecanismo para el diseño (dibujo) de cualquier pantalla directamente en la terminal del usuario.

El diseñador puede moverse libremente por toda la pantalla, teclear las constantes que desea contenga el diseño, declarar el atributo y edición para estas variables, ajustar el diseño, etc., estas facilidades permiten que el diseño sea una tarea muy sencilla y muy rápida, de hecho, lo que el diseñador dibuja en la pantalla, es lo que verá el usuario final, con contenidos reales en las variables.

El código COBOL consiste en un mapa de control y un conjunto de variables que hacen posible que la laboriosa tarea de programar el manejo de la pantalla se traduzca en unas cuantas instrucciones MOVE. El mapa de control es accesado solamente por el manejador de pantallas y contiene información de control inherente al diseño de pantalla en cuestión.

El conjunto de variables corresponde a todos los datos de entrada y/o salida, exceptuando títulos o constantes que se especificaron en el diseño de la pantalla. En este conjunto de variables, el programador encontrará en esas variables los valores que fueron tecleados por el usuario final en la terminal. El resto, manejo de terminal, despliegue, aceptación, edición, validación y ajuste de datos, es hecho en forma automática por el manejador de pantallas del paquete.

Asociada a cada variable de entrada y/o salida, existe otra variable que corresponde al

atributo del dato. El paquete cuenta con 18 atributos diferentes y sólo basta mover la clave correspondiente para que ese dato sea protegido o modificado, que parpadee o permanezca fijo, etc. Esta facilidad permite que la presentación de pantallas sea cambiada dinámicamente por el programa en el tiempo de ejecución. Además, por ejemplo, una misma pantalla puede ser utilizada para un programa de altas y para otro de consultas.

MANEJO DE PANTALLAS.

EASEMAP cuenta con un manejador de pantallas que lleva a cabo todas las operaciones de entrada y salida, edición de datos numéricos y fecha, validación y ajuste de los datos introducidos por el usuario final y navegación por la pantalla.

El manejador de pantallas opera un modo de página completa logrando con ello un óptimo tiempo de respuesta y una variedad de facilidades para que la captura de datos resulte más sencilla. El programa de aplicación invoca al manejador de pantallas mediante la instrucción CALL utilizando como argumentos el mapa de control y el conjunto de variables así como un código de función. Los valores que contiene dicho conjunto de variables serán tomados por el manejador de pantallas para llevar a cabo la edición correspondiente y la presentación deseada en la pantalla. Es entonces cuando el usuario final empieza a interactuar con la pantalla, contando con facilidades para moverse por distintos sectores de la misma, capturar datos con edición inmediata, regresar a modificar un dato ya capturado e ir a un campo específico.

Una vez que se ha terminado esta interacción, el control es regresado al programa de aplicación y los datos capturados, validados y ajustados estarán a disposición en el conjunto de variables ya mencionado.

El manejador de pantallas está diseñado para trabajar en distintos tipos de terminales. Este obtendrá las características de la terminal con la cual trabajará, mismas que deberán estar

especificadas en el ambiente UNIX. Específicamente, las secuencias de control para borrar la pantalla, mover el cursor, activar y desactivar atributos e información relacionada con las funciones de tab,backtab y home, serán tomadas de la variable TERMCAP y el archivo /etc/termcap de UNIX, por lo tanto, si ya se cuenta con una definición de esa terminal, no se tendrá que elaborar otra para este paquete. La variable TERMCAP deberá estar presente en el medio ambiente asociado al login con el cual el usuario final entró al sistema.

SITUACION DE SISTEMAS.

De acuerdo al Plan Estratégico, se decidió la adquisición de equipo de cómputo para la automatización de los procesos operativos en sucursales. Se hizo un estudio acerca de las opciones existentes en el mercado y se llegó a la conclusión de que lo más recomendable eran los equipos multiusuarios con Sistema Operativo UNIX los cuales ofrecían las siguientes ventajas:

1) Estos equipos pertenecían a una corriente tecnológica que garantizaba la disponibilidad de equipos cada vez más poderosos.

2) Existía una gran variedad de equipos funcionando bajo el Sistema Operativo UNIX, lo que brindaba un alto nivel de compatibilidad y transportabilidad de software.

3) El mercado de Software para estos equipos era abundante, lo que permitía seleccionar la estrategia más adecuada para el desarrollo de nuestro sistema.

En este concurso se destacó el equipo Crommco el cual fue seleccionado por sus características y por su rendimiento en las pruebas realizadas a todos los equipos.

PROBLEMATICA.

En la operación de los sistemas, se presentaron gran cantidad de problemas en los equipos Crommco. Estos ocasionaron una gran inestabilidad en la operación y la frecuencia de fallas de los

equipos originó que se llevaran a cabo todo tipo de medidas tendientes a sostener la producción. En la operación diaria de los equipos Cromemco se presentaron ininterrumpidamente una serie de problemas ocasionados por fallas de diversa índole. Las fallas pueden agruparse en tres grupos de acuerdo al análisis de los reportes de fallas.

1) Proveedor.

2) Software.

3) Equipo.

PROVEEDOR.

Debido al deficiente Soporte Técnico se ocasionó una gran cantidad de diagnósticos equivocados provocando que el tiempo de respuesta de algunas fallas se alargara considerablemente. Además se presentaron fallas que nunca se pudieron diagnosticar. Otro problema fue el servicio a sucursales. El proveedor no tenía las piezas necesarias para reposición y era necesario quitar piezas de algún equipo para reemplazarlo temporalmente en otro y así mantener ininterrumpida la producción. Finalmente, el problema más serio se presentó cuando el proovedor cerró sus operaciones en México, por lo que era de esperarse un mayor deterioro en el servicio y soporte a estos equipos.

SOFTWARE.

Estos fueron ocasionados porque la implementación de UNIX estaba sobre otro sistema operativo denominado Cromix, lo cual volvía complicados muchos procesos de administración. Por otro lado, cada equipo que llegaba tenía un Sistema Operativo diferente a los anteriores. Esto provocó problemas de compatibilidad en respaldos, archivos, etc. Además se detectaron problemas

con la implementación del compilador RM-Cobol especialmente en el manejo de archivos indexados. Se encontró que para archivos con más de 5000 registros los tiempos de acceso aumentaban desmesuradamente y que los archivos de llaves se corrompían frecuentemente, bloqueando el acceso a los datos.

EQUIPO.

La tarjeta controladora de la unidad de cartucho consumía todos los recursos del equipo cuando se encontraba funcionando impidiendo realizar otras labores, además, presentaba muchos problemas para la recuperación de información.

En la unidad de disco, se deben alternar pistas dañadas debido algunas veces a la caída del equipo y otras sin razón aparente.

Por todo lo anterior se convocó a un nuevo concurso en el cual se eligió al equipo HP Vectra y surgió la necesidad de realizar un diseñador de pantallas que sustituyera al EASEMAP y fuera compatible con diferentes tipos de terminales. Esta necesidad se debió a que el diseñador EASEMAP no podía ser portado debido a que no se contaba con los programas objetos del mismo y por lo tanto no era posible utilizarlo en cualquier otro equipo.

EQUIPO HP.

El equipo con el que se cuenta actualmente en las sucursales es un HP modelo RS/25C con un microprocesador Intel de 25 MHz, arquitectura de memoria caché y unidades de disco de alta velocidad.

Características:

-Microprocesador Intel 80386 de 32 bits. Velocidad de reloj de 25 MHz a

-Ranuras de expansión ISA. 2 ranuras de 8 bits y 6 ranuras de 16 bits.

-Seis espacios para unidades de almacenamiento de datos.

-Tarjeta controladora de 4 funciones.

-Conectores separados para el coprocesador 80387 y el acelerador de punto flotante Weitek WTL-3167.

-Arquitectura de memoria caché de 32 Kb basada en el controlador de memoria caché 82385 de Intel.

- Memoria principal de 32 bits expandible a 2, 4, 8,10 o 16 MB en la tarjeta procesadora.

- 64 KB de ROM para BIOS con 2 bases para adicionar 64 KB de ROM.

- Unidad de disco flexible de 5.25",1.2 MB.

- Unidad de cinta interna de respaldo de 40 MB.

II.- ANALISIS DEL SISTEMA ANTERIOR

INTERPRETACION DEL FORMATO DE PANTALLA

Como se ha mencionado en el capitulo anterior, el diseñador de pantallas genera un archivo que es un "MAPA de CONTROL" , el cual contiene las constantes y variables de la pantalla, así como los apuntadores de la localización en los arreglos donde se encuentran sus valores. Estos arreglos son los que pasa como parámetros el programa cobol al manejador de pantallas cuando ejecuta la instrucción :

CALL "easemap" <screen> <variables>"

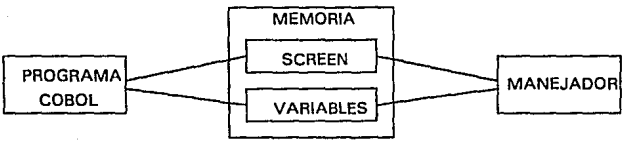
Donde:

easemap es el nombre del programa objeto que "maneja" la pantalla.

screen es la primera parte del formato de pantalla, que se explica en el siguiente párrafo, la cual contiene las descripciones de las variables y constantes.

variables son los valores de las variables dentro del programa COBOL.

Estos parámetros se "ven" en memoria como si fueran dos arreglos y los apuntadores mencionados traen los datos para localizar los valores de cada variable en el arreglo del parámetro de variables. Para ilustrar lo anterior se muestra la siguiente figura:



El formato generado tiene la estructura de la descripción de variables para un programa escrito en Cobol y contiene las siguientes partes :

1.-Un comentario entre asteriscos con una constante SCREEN seguido del nombre del formato de la pantalla .

2.-Definida a un nivel 01 la variable **SCREEN-NOMBRE**, donde NOMBRE es el nombre del archivo del formato de pantalla.

3.-Definidos a un nivel 02 los siguientes serán FILLERS de 118 posiciones tipo texto, que contienen :

a) El primer FILLER empezará con el nombre que se le haya dado al formato de pantalla hasta encontrar la cadena **N$;$s**. El FILLER mencionado continuará con una tilde (~) lo cual indica que el código ASCII de los siguientes dos caracteres menos 35 son las coordenadas renglón-columna del texto que se encuentre a partir de éstas. Se considerará que todos los siguientes caracteres son constantes de la pantalla hasta que se encuentre otra tilde, para repetir la operación anterior, si los textos más su localización se llegan a pasar de las 118 posiciones, se continuará con otro FILLER de igual o menor dimensión según sea el caso.

b) Se seguirá realizando el "pintado" de las constantes en la terminal con el método mencionado en el inciso **a)** , hasta que se encuentre la cadena **~##$+**, que indicará que los siguientes caracteres son las definiciones de las variables del formato de pantalla, las cuales vedrán en un conjunto de 11 caracteres por variable, ordenado de la siguiente forma :

El contenido de los elementos A y B indican con la fórmula que se muestra abajo, la posición

en el arreglo del parámetro de las variables, para localizar su valor.

{ [Código ASCII (A) - 35] * 91 } + [Código ASCII (B) - 35]

Los elementos **C** y **D** no se utilizan.

El contenido del elemento **E** y del **F** son la posición renglón columna de la variable, para esto se les resta 35 a su código ASCII.

El contenido del elemento **G** representa la longitud larga de la variable, esto es, el tamaño de la variable con la máscara de edición que tenga definida. Para esto se le resta 35 a su código ASCII.

El contenido del elemento **H** representa la longitud corta de la variable, es decir, sin la máscara de edición. Para esto se le resta 35 a su código ASCII.

El contenido del elemento **I** es el número de decimales, que en caso de ser real, tenga definidos. Para esto se le resta 35 a su código ASCII.

El contenido del elemento **J** es el valor del tipo de máscara de edición. Los tipos de máscara de edición se dividen en dos grupos: máscaras de edición para fechas y para cantidades numéricas. Posteriormente se muestran algunos ejemplos.

El contenido del elemento **K** es el tipo de signado que puede contener una variable numérica.

La diferencia entre longitud larga y longitud corta radica en la máscara de edición, mientras en la longitud corta sólo se presentan los números, en la longitud larga (dependiendo de la máscara de edición) se presentan además de los números otros caracteres tales como comas, signo de pesos, etc.

Cuando se ha dado la descripción de todas las variables, puede darse el caso que el último FILLER no sea de 118 posiciones.

4.-Se continuará con la descripción de las variables para el uso interno del programa COBOL, es decir son las variables a las cuales les mueven y reciben los valores del usuario. Esto empieza con la definición a un nivel **01 VARIABLES-NOMBRE,** donde NOMBRE es el nombre del formato de pantalla.

5.-A un nivel 03 se definen las variables **VAR-ALPHA-NOMBRE**, después a un nivel 05 se encuentren las descripciones de todas las variables tipo alfanuméricas que se encuentren en la pantalla.

6.-Con un nivel 03 se encuentra la definición de las variables **VAR-NUM-NOMBRE**, aquí se encontrarán las descripciones de las variables numéricas de la pantalla

Para los puntos 5 y 6 la estructura de las variables será la siguiente:

05 VARIABLE-FORMATO-A PIC X(01).

05 VARIABLE-FORMATO-S PIC ? (Según tipo y longitud)

En esta parte se observa como quedan los nombres de las variables para que sean manejados por el programa COBOL , el cual se compone de :

VARIABLE : Es el nombre que se le dió dentro del diseñador de pantallas.

FORMATO :Es el nombre del formato de la pantalla (que puede ser de hasta 8 caracteres).

-S es la extensión que se le pone a la variable que contendrá los datos de usuario. La descripción de ésta dependerá del tipo y longitud que se le definió en el diseño.

También se ilustra que para cada variable definida en la pantalla, automáticamente se genera otra, con el mismo nombre pero con una extensión -A, su descripción es de un caracter alfanumérico, esta variable contendrá el valor del tipo de edición que presente ese campo en la pantalla.

Se ilustra lo anterior con un ejemplo de formato de pantalla que contendrá los siguientes campos :

Variable	PICTURE	EDICION	SIGNO
NOMBRE	PIC X(30)		
VEHICULO	PIC X(20)		
PLACAS	PIC X(6)		
POLIZA	PIC 9(10)	Sin edición	Sin signo
FECHA	PIC 9(6)	DD/MM/YY	
FECHA	PIC 9(6)	DD/MM/YY	
MODELO	PIC 9(4)	Sin edición	Sin signo
SUMA	PIC 9(7)	Signo de pesos , flotante y con comas	Sin signo
PRIMA	PIC 9(7)V9(2)	Signo de pesos , flotante y con comas	Sin signo
DEDUCIBLE	PIC 9(1) OCCURS 3 TIMES	Sin edición	Sin signo

El formato que contendrá las variables mencionadas anteriormente quedará diseñado con la siguiente distribución tomando en cuenta que se ocupará en una pantalla con 80 por 24 caracteres, se indica el número de renglón y el de columna.

```
   123456789A123456789B123456789C123456789D123456789E123456789F123456789G1234567B9H
 1
 2
 3
 4                                    PANTALLA DE PRUEBA
 5
 6                                   ALTA DE POLIZA AUTOMOVILES
 7
 8
 9        NUM. POLIZA : _________     VIGENCIA DE _______ A _______
10
11       PROPIETARIO : ____________________________
12
13       VENICULO : _________________     MODELO : ____     PLACAS : _____
14
15        SUMA ASEGURADA : ___________  DEDUCIBLES
16
17                                        _ %  RESPONSABILIDAD CIVIL
18       PRIMA : ___________                    _ %  DAÑOS
19                                              _ %  ROBO TOTAL
20
21
22
23
24
```

Una vez que se ha diseñado el formato anterior, el diseñador generará el archivo de la pantalla el cual es el copy para el programa Cobol que se encarga de manejar la pantalla. El archivo de dicha pantalla se muestra a continuación.

```
*****************************************************************
*                        SCREEN PANT123                         *
*****************************************************************
    01 SCREEN-PANT123.
     02 FILLER PIC X(118) VALUE
       "PANT123 N$;$s~'@PANTALLA DE PRUEBA ~)<ALTA DE POLIZA AUTOMO
-    "VILES ~,+NUM. POLIZA : ~,HVIGENCIA DE ~,]A ~,+PROPIETARIO".
     02 FILLER PIC X(118) VALUE
       " : ~0+VEHICULO : ~0NMODELO : ~0^PLACAS : ~2+SUMA ASEGURADA :
-    " ~2MDEDUCIBLES ~4S% ~4VRESPONSABILIDAD CIVIL ~5+PRIMA :".
     02 FILLER PIC X(118) VALUE
       " ~5S% ~5VDA&OS ~6S% ~6VROBO TOTAL ~##$+#^??,9--#NN#j??,T
-    "+)#?N#r??,_+)#?N##??.9AA#AN#B??0677#AN#z??!0W"#NN#W??0g))#".
     02 FILLER PIC X(57) VALUE
       "AN$%?72</,#5N$07?4Q$$#NN$9??530,%5N$37?5Q$$#NN$67?6Q$$#NN".
     02 FILLER PIC X VALUE LOW-VALUES.
    *-----------------------------------
    01 VARIABLES-PANT123.
       03 VAR-ALPHA-PANT123.
          05 NOMBRE-PANT123-A  PIC X(01) VALUE "5".
          05 NOMBRE-PANT123-S  PIC X(30).
          05 VEHICULO-PANT123-A  PIC X(01) VALUE "5".
          05 VEHICULO-PANT123-S  PIC X(20).
          05 PLACAS-PANT123-A  PIC X(01) VALUE "5".
          05 PLACAS-PANT123-S  PIC X(6).
       03 VAR-NUM-PANT123 SIGN IS TRAILING SEPARATE.
          05 POLIZA-PANT123-A  PIC X(01) VALUE "5".
          05 POLIZA-PANT123-S  PIC S9(10).
          05 FECHA-PANT123-A  PIC X(01) VALUE "5".
          05 FECHA-PANT123-S  PIC S9(6).
          05 FECHA-PANT123-A  PIC X(01) VALUE "5".
          05 FECHA-PANT123-S  PIC S9(6).
          05 MODELO-PANT123-A  PIC X(01) VALUE "5".
          05 MODELO-PANT123-S  PIC S9(4).
          05 SUMA-PANT123-A  PIC X(01) VALUE "5".
          05 SUMA-PANT123-S  PIC S9(9).
          05 DEDUCIBLES-PANT123-INI     VALUE ALL "1".
            07 DEDUCIBLES-PANT123-VEC   OCCURS 3.
               09 DEDUCIBLES-PANT123-A  PIC X(01).
               09 DEDUCIBLES-PANT123-S  PIC S9(1).
          05 PRIMA-PANT123-A  PIC X(01) VALUE "5".
          05 PRIMA-PANT123-S  PIC S9(7)V9(2).
       03 FILLER PIC X VALUE LOW-VALUES.
    *****************************************************
```

La primera parte de este archivo es el mapa de control de la pantalla, la segunda parte esta dividida en dos tipos de definiciones, en la primera de ellas se definen las variables alfanuméricas y en la segunda se definen las variables numéricas.

Este es el archivo de COPY que genera el diseñador para que sea anexado al programa escrito en COBOL.

Analizando la primera parte de éste se observa el comentario entre asteriscos que contiene el nombre del formato de pantalla. Después en un nivel 02 se describen los FILLERS de 118 posiciones caracter, el primero comienza por el nombre de la pantalla seguido de la cadena N$;$s que indica que lo que continúa es la descripción de la pantalla, el siguiente caracter es una tilde (~) significando esto que los siguientes dos caracteres son la posición renglón-columna donde comenzará a escribir el texto, por lo tanto para el ejemplo estos dos caracteres son apóstrofe y arroba (' @) usando la fórmula de su código ASCII - 35 es :

39 -35 = 4 (renglón), 64 -35 = 29 (columna)

Por lo tanto en la posición 4,29 "pintará" los caracteres que continúen, siendo estos : **PANTALLA DE PRUEBA**, hasta que se encuentre otro caracter tilde para repetir la operación anterior, de acuerdo a esto, para el ejemplo tenemos la siguiente información :

caracteres	posición (ren, col)	texto
) <	**6,25**	**ALTA DE POLIZA AUTOMOVILES**
, +	**9,8**	**NUM. POLIZA :**
, H	**9,37**	**VIGENCIA DE**
,]	**9,58**	**A**
. +	**11,8**	**PROPIETARIO :**
0 +	**13,8**	**VEHICULO :**
0 N	**13,43**	**MODELO**
0 ^	**13,59**	**PLACAS :**
2 +	**15,8**	**SUMA ASEGURADA :**

2 M	**15,42**	**DEDUCIBLES**
4 S	**17,48**	**%**
4 V	**17,51**	**RESPONSABILIDAD CIVIL**
5 +	**18,8**	**PRIMA :**
5 S	**18,48**	**%**
5 V	**18,51**	**DA&OS**
6S	**19,48**	**%**
6V	**19,51**	**ROBO TOTAL**

A continuación se presenta como se vería la parte de screen como arreglo en memoria :

ARREGLO DEL PARAMETRO DE SCREEN

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
P	A	N	T	1	2	3		N	$	:	$	s	"	'	@	P	A	N	T	A	L	L	A		D	E		P	R	U	E	B	A

34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67
	"	)	<	A	L	T	A		D	E		P	O	L	I	Z	A		A	U	T	O	M	O	V	I	L	E	S		"	,	+

68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101
N	U	M	.		P	O	L	I	Z	A		:		"	,	H	V	I	G	E	N	C	I	A		D	E		"	,	!	A	"

102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135
.	+	P	R	O	P	I	E	T	A	R	I	O		:			"	0	+	V	E	H	I	C	U	L	O		:		"	0	N	M

136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169
O	D	E	L	O		:		"	0	"	P	L	A	C	A	S		:		"	2	+	S	U	M	A		A	S	E	G	U	R

170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203
A	D	A		:		"	2	M	D	E	D	U	C	I	B	L	E	S		:		"	4	S	%		"	4	V	R	E	S	P

2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037
O	N	S	A	B	I	L	I	D	A	D		C	I	V	I	L			-	5	+	P	R	I	M	A		:		-	5	S	%

2389	2390	2401	2412	2423	2434	2445	2456	2467	2478	2489	2490	2501	2512	2523	2534	2545	2556	2567	2578	2589	2590	2601	2612	2623	2634	2645	2656	2667	2678	2689	2690	2701	2712
-	5	V	D	A	&	O	S		-	6		D	E		-	,	)	A		-	.	+	P	R	O	P	I	E	T	A	R	I	

2723	2734	2745	2756	2767	2778	2789	2790	2801	2812	2823	2834	2845	2856	2867	2878	2889	2890	2901	2912	2923	2934	2945	2956	2967	2978	2989	2990	3001	3012	3023	3034	3045
O	:		-	O	+	V	E	H	I	C	U	L	O		:			-	O	N	M	O	D	E	L	O		:		-	O	-	P	L

3006	3007	3008	3009	3010	3011	3012	3013	3014	3015	3016	3017	3018	3019	3020	3021	3022	3023	3024	3025	3026	3027	3028	3029	3030	3031	3032	3033	3034	3035	3036	3037	3038	3039
A	C	A	S		:		-	2	+	S	U	M	A		A	S	E	G	U	R	A	D	A		:		-	2	M	D	E	D	U

3401	3412	3423	3434	3445	3456	3467	3478	3489	3490	3501	3512	3523	3534	3545	3556	3567	3578	3589	3590	3601	3612	3623	3634	3645	3656	3667	3678	3689	3690	3701	3712	3723	3734
C	I	B	L	E	S		:		-	4	S	%		-	4	V	R	E	S	P	O	N	S	A	B	I	L	I	D	A	D		C

3745	3756	3767	3778	3789	3790	3801	3812	3823	3834	3845	3856	3867	3878	3889	3890	3901	3912	3923	3934	3945	3956	3967	3978	3989	3990	4001	4012	4023	4034	4045	4056	4067
I	V	I	L		-	5	+	P	R	I	M	A		:		-	5	S	%		-	5	V	D	A	&	O	S		-	6	S	%

4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
0	0	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	4	4	4
8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
	-	V	R	O	B	O		T	O	T	A	L			-	#	#	$	+	#	-	?	?	,	9	-	-	#	N	N	#	j	?

4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5	5	5	6	6	6	6	6	6	6	6	6	6	7	7	7	7	7	7
2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5
?	.	.	T	+	)	#	?	N	#	r	?	?	.		+	)	#	?	N	#	#	?	?	.	9	A	A	#	A	N	#	B	?	?

4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5
7	7	7	7	7	7	8	8	8	8	8	8	8	8	8	8	8	9	9	9	9	9	9	9	9	9	0	0	0	0	0	0	0	0
4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7
0	6	7	7	#	A	N	#	z	?	?	0	W	"	#	N	N	N	#	W	?	?	0	g	)	)	#	A	N	$	%	?	?	2	<

5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
0	0	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	4	4	4
8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
/	.	.	#	5	N	$	0	?	?	4	Q	$	$	#	N	N	$	9	9	?	?	5	3	0	,	%	5	N	$	3	?	?	5	Q	$

5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
4	4	4	4	4	4	4	5	5	5	5	5	5	5	5
2	3	4	5	6	7	8	9	0	1	2	3	4	5	6
$	#	N	N	$	6	?	?	6	Q	$	$	#	N	N

Si se encuentra la cadena ~##S+, indicará que a continuación está la descripción de las variables contenidas en el formato, como se mencionó anteriormente, las variables están definidas con once caracteres siendo los dos primeros la posición del valor de la variable en el arreglo que se pasa del programa COBOL , sustituyendo los datos en la fórmula mencionada para estos datos se tiene :

^ (35-35) * 91 + (94 -35) = 59

Los caracteres que siguen son dos signos de interrogación que no se ocupan, los caracteres 5 y 6 son la posición de la variable en el formato de pantalla, por lo tanto :

,9 44 - 35 = 9 y 57 - 35 = 22 por lo tanto esta variable se encuentra en la posición 9, 22

Los caracteres 7 y 8 son la longitud larga y longitud corta respectivamente, entonces:

-- 45 - 35 = 10 y 45 - 35 = 10

El caracter 9 es el número de decimales, por lo tanto:

35 - 35 = 0

El caracter 10 representa el tipo de máscara de edición, para el ejemplo se tiene:

N (significa sin máscara de edición)

El caracter 11 es el tipo de signado para las variables numéricas, por lo que se tiene:

N (que es sin signo).

Siguiendo el ejemplo se tienen definidas las variables en el orden que aparecen en el COPY :

Caracteres	Posición en el arreglo	Posición		Longitud		Núm.	Máscara	Signo
		Ren.	Col	Corta	Larga	Dec.		
#^??.9- -#NN	59	9	22	10	10	0	N	N
#]??.T+)#7N	71	9	49	8	6	0	7	N
#r??,_+)#7N	79	9	60	8	6	0	7	N
##?.9AA#AN	0	11	22	30	30	0	A	N
#B?0677#AN	31	13	19	20	20	0	A	N
#z??0W' '#NN	87	13	52	9	9	0	N	N
#W??0g))#AN	52	13	68	6	6	0	A	N
$%??2</,#5N	93	15	25	12	9	0	5	N
$0??4Q$$#NN	104	17	46	1	1	0	N	N
$9??530,%5N	113	18	16	13	9	2	5	N
$3??5Q$$#NN	107	18	46	1	1	0	N	N
$6??6Q$$#NN	110	19	46	1	1	0	N	N

Suponiendo que las variables del programa Cobol que ocupa tal pantalla tuviera los siguientes datos:

VARIABLES	VALORES
NOMBRE-PANT123-S	VICTOR HUGO AMAYA GALVAN
VEHICULO-PANT123-S	CHEVROLET CAVALIER
PLACAS-PANT123-S	632ESY
POLIZA-PANT123-S	20047461
FECHINI-PANT123-S	050892
FECHATER-PANT123-S	050893
MODELO-PANT123-S	1991
SUMA-PANT123-S	37425000
DEDUCIBLES-PANT123-S (1)	5
DEDUCIBLES-PANT123-S (2)	8
DEDUCIBLES-PANT123-S (2)	9
PRIMA-PANT123-S	2500000

Entonces, el arreglo de los datos de las variables que pasa el programa Cobol al manejador se veria de la siguiente forma:

ARREGLO DE DATOS QUE PASA EL PROGRAMA COBOL AL MANEJADOR

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
X	V	I	C	T	O	R		H	U	G	O		A	M	A	Y	A		G	A	L	V	A	N						

31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58
X	C	H	E	V	R	O	L	E	T		C	A	V	A	L	I	E	R			X	6	3	2	E	S	Y

59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92
X	0	0	2	0	0	4	7	4	6	1	S	X	0	5	0	8	9	2	S	X	0	4	0	8	9	3	S	X	1	9	9	1	S

92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125
S	X	0	3	7	4	2	5	0	0	0	S	X	5	S	X	8	S	X	9	S	X	0	0	2	5	0	0	0	0	0			

Por lo tanto, si se toman los valores del arreglo la pantalla se presentará de esta manera:

```
123456789A123456789B123456789C123456789D123456789E123456789F123456789G123456789H
1
2
3
4                                  PANTALLA DE PRUEBA
5
6                                 ALTA DE POLIZA AUTOMOVILES
7
8
9       NUM. POLIZA : 0020047461              VIGENCIA DE 05/08/92 A 04/08/93
10
11      PROPIETARIO : VICTOR HUGO AMAYA GALVAN
12
13        VEHICULO : CHEVROLET CAVALIER      MODELO : 1991    PLACAS : 632ESY
14
15        SUMA ASEGURADA : $37,425,000      DEDUCIBLES
16
17                                                    5 %  RESPONSABILIDAD CIVIL
18      PRIMA : $2,500,000                            8 %  DAÑOS
19                                                    9 %  ROBO TOTAL
20
21
22
23
24
```

III.-ANALISIS DEL SISTEMA NUEVO.

NECESIDADES DEL SISTEMA

- El objetivo del proyecto del nuevo sistema que se asignó al área de soporte técnico de ASEGURADORA MEXICANA, S.A., consiste en analizar el funcionamiento del paquete EASEMAP y desarrollar uno que lo sustituya cumpliendo los siguientes requerimientos :

- El sistema nuevo debe interpretar los archivos de formato de pantalla generados por el EASEMAP de CROMEMCO y por lo tanto el diseñador de pantallas debe poder modificar cualquier pantalla que haya sido producida por éste. También el manejador de pantallas debe presentar las mismas que se realizaron en el EASEMAP.

- El diseñador de pantallas debe ser de fácil uso comprendiendo las siguientes características : Tanto el diseñador como el manejador de pantallas, deberán tener la capacidad de poder ser ejecutadas y conservar sus características desde cualquier tipo de terminal que sea de 24 renglones por 80 columnas, que sean independientes del Hardware del Procesador ,que puedan funcionar en cualquier equipo que tenga Sistema Operativo UNIX y que el código generado pueda ser ligado a un programa en lenguaje RMCobol 85.

Características del Diseñador.

Después de sostener entrevistas con los usuarios y de analizar los diseñadores de pantalla disponibles se estableció que el Diseñador debería presentar las siguientes opciones:

Para el menú número 1 referente a las opciones de archivo.

CARGAR. Cuando se ejecute esta opción, se deberá preguntar cual es el archivo de formato de pantalla para después cargar la pantalla en el programa y presentarla en la terminal del diseñador.

SALVAR. Con esta opción, se guardará el formato de pantalla tal como se encuentre al

momento de ejecutarla y quedará grabada en el archivo del cual se cargó la pantalla, si es un diseño nuevo, deberá preguntarse el nombre que se le dará al archivo de formato de pantalla.

SALVAR COMO. Cuando se haga referencia a esta opción, se preguntará el nombre del archivo con el cual se quiere grabar el diseño tal como se encuentre al llamar esta opción.

DOCUMENTAR. Al ejecutar esta opción, se generarán dos archivos, uno que contendrá la "imagen" de la pantalla y otro que presentará la pantalla con la descripción y ubicación de todas las variables que tenga definidas.

Para las opciones de **EDICION** del menú número dos :

DESCRIPCION. Esta opción presentará en pantalla la descripción de la variable en la cual se encuentre posicionado el cursor al momento de ejecutarla.

DEFINIR. En esta opción, se definirán las variables requeridas para la pantalla, esto presentará primero una ventana para seleccionar que tipo de variable se desea, pudiendo ser:

Alfanumérica

Entera

Real

Fecha

Para las variables tipo alfanuméricas deberá pedir :

Nombre

Longitud

Ocurrencias.

Para las variables del tipo enteras se deberá pedir:

Nombre

Longitud

Ocurrencias

Máscara

Las variables de tipo real pedirán los mismos datos que las variables enteras y además :

Número de decimales.

Para las variables tipo fecha se deberá pedir los siguientes datos :

Nombre

Ocurrencias

Máscara

Pudiendo ser :

YY/MM/DD

DD/MM/YY

MM/DD/YY

Donde YY es el año

MM es el mes

DD es el día

En la opción de definir se deberán hacer las siguientes validaciones:

-El nombre de cualquier variable debe empezar con una letra.

-Después de la primera letra del nombre de una variable, sólo serán válidas letras, números y guiones.

-No deben traslaparse las variables.

-No deben repetirse los nombres de las variables para una pantalla.

-Al definir una variable no se deberá rebasar en su longitud la columna 80.

-Al definir las ocurrencias de una variable no deberá rebasar el renglón 24.

-Cuando se defina el tipo de máscara de una variable deberá ser uno existente.

-La longitud máxima de una variable tipo entera deberá ser de 18 caracteres.

-La longitud de una variable tipo real será de 18 incluyendo el número de decimales.

La opción de **BORRAR.** Deberá borrar la variable en la cual se encuentre el cursor posicionado.

Cuando se ejecute la opción de **POSICION,** deberá presentar en la pantalla las coordenadas renglón columna del cursor al momento de llamarla.

El último menú contendrá las opciones de **SALIR** y serán :

SALIR. Al ejecutar esta opción se deberá salvar el formato de pantalla tal como se encuentre al ser seleccionada, quedando con el nombre de archivo con el cual se llamó y si es un diseño nuevo, deberá pedir el nombre con el que se desea que se guarde el formato de pantalla, después de esto se terminará de ejecutar el programa para regresar el shell del sistema operativo.

ABANDONAR. Con esta opción, se abandonará el diseñador sin grabar los cambios que se hayan realizado en éste, antes de salir deberá preguntarse la confirmación de la operación.

Para que el diseñador posicione el cursor sobre cualquier variable, deberá hacerlo entrando por arriba o por abajo, pero no por las orillas de la variable, ya que si se intenta de esta forma, se hará sonar la campana de la terminal y se moverá el cursor al final de ésta.

Cuando se encuentre ejecutando el diseño y se intente poner cualquier constante (texto) sobre una variable previamente definida, deberá hacerse sonar la campana de la terminal y se moverá el cursor al final de ésta.

Características del Manejador.

El manejador de pantallas debe utilizar los atributos de acuerdo a la siguiente tabla :

Clave del atributo	Descripción
0	Oculto, no despliega el dato aceptado, modificable
1	Alta intensidad,modificable
2	Alta intensidad parpadeante, modificable
3	Alta intensidad con video inverso, modificable
4	Alta intensidad con video inverso y parpadeante, modificable
5	Baja intensidad, modificable
6	Baja intensidad parpadeante, modificable
7	Baja intensidad con video inverso, modificable
8	Baja intensidad con video inverso y parpadeante, modificable
A	Oculto, no despliega el dato aceptado, protegido
B	Alta intensidad, protegido
C	Alta intensidad parpadeante, protegido
D	Alta intensidad con video inverso, protegido
E	Alta intensidad con video inverso y parpadeante, protegido
F	Baja intensidad,protegido
G	Baja intensidad parpadeante, protegido
H	Baja intensidad con video inverso, protegido
I	Baja intensidad con video inverso y parpadeante, protegido

Se deben formatear los datos con los tipos de máscara válidos que se muestran en la siguiente tabla, se tomará como ejemplo un campo de 11 posiciones numérico y cómo se presentaría el número 40527389 y un campo tipo fecha con el dato de 17 de Diciembre de 1992.

	Descripción	Ejemplo
0	Sin edición	00040527389
1	Supresión de ceros	40527389
2	Signo de pesos flotante	$40527389
3	Protección con asteriscos	***40527389
4	Supresión de ceros con comas	40.527,389
5	signo de pesos flotante y con comas	$40,527,389
6	Protección de asteriscos con comas	***40,527,389
7	DD/MM/YY	17/12/92
8	MM/DD/YY	12/17/92
9	YY/MM/DD	92/12/27

Debe impedir la captura de datos para los campos con atributo de protegido.

Debe ser capaz de aceptar sólo números en los campos de tipo entero, real y de tipo fecha.

Debe validar que se acepten fechas reales en los campos de tipo fecha.

Los campos numéricos se justificarán a la izquierda y los alfanuméricos a la derecha.

Al teclearse un **ENTER** se aceptarán los caracteres tecleados hasta ese momento y se debe formatear con el tipo de máscara que se haya definido para éste y avanzar al siguiente campo.

Si se llena un campo de caracteres a su longitud corta, se deberá formatear según su máscara de edición y avanzar al siguiente campo.

Cuando se teclee un **BACKSPACE** se deberá borrar el caracter anterior al cursor y si es el caso de estar en la primera posición de una variable, se pasará el cursor a la variable anterior, se dejará el valor original y se pasará el cursor a la variable anterior.

En los campos de tipo numérico se aceptará el signo negativo (-) sólo en la primera posición

del campo, si es uno de tipo real únicamente se deberá aceptar un punto decimal y cuando se esté capturando se deberá insertar automáticamente cuando se hayan tecleado los números de la parte entera, también si se ha tecleado un punto decimal se dará por capturado el campo cuando se hayan tecleado los números de la parte decimal.

Al capturarse un campo de tipo fecha deberá evitarse que se tecleen datos sobre las diagonales de separación.

Cuando se haya tecleado el último campo de la pantalla se terminará la ejecución del programa manejador y se regresará el control al programa COBOL.

Debe manejar un parámetro más, éste debe comportarse como si se presionars una tecla de función, es decir, al teclear un escape y dos números deberá terminarse la ejecución del programa manejador y pasar al programa Cobol el valor de las variables de la función.

METODOLOGIA

La Metodología de Análisis Estructurado indica que el producto del Análisis es la Especificación estructurada. Como lo indica Tom de Marco: *El Análisis Estructurado es el uso de herramientas como diagramas de flujo de datos, diccionario de datos, inglés Estructurado, tablas y árboles de decisión con la finalidad de construir la Especificación Estructurada del sistema.* Esta Especificación es la principal interfase entre el usuario final y el área de desarrollo. A través de esta el usuario expresa sus requerimientos a los desarrolladores de Software.

Para llevar a cabo esta Especificación, la metodología se apoya en tres herramientas que son: **Los Diagramas de Flujo de Datos** (DFD'S), que nos permiten particionar los requerimientos en base a Flujos de Datos y Procesos. Las herramientas propuestas para definir estos Flujos y Procesos son **El Diccionario de Datos** y la **Descripción de Miniespecificaciones** respectivamente.

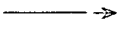
El uso de estas herramientas para desarrollar el Análisis, nos permitirá obtener una Documentación adecuada del sistema. Además nos dará la interfase adecuada para empezar con el diseño.

DIAGRAMAS DE FLUJO DE DATOS

El primer paso del Análisis consistirá en un estudio de la situación actual de la empresa y del área involucrada. Como resultado de este estudio, se generó el diagrama de contexto. Este diagrama representa la Coordinación una vez desarrollado el Sistema.

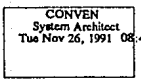
Posteriormente, se presentan cuatro diagramas de flujo de datos que son el resultado del Análisis del Sistema. El diagrama 0, representa el sistema en forma global, donde se muestran los principales flujos de datos y procesos. A partir de este diagrama se presentan dos DFD'S que son : Diseñador y Manejador de pantallas.

Los diagramas fueron hechos basados en la siguiente simbología:

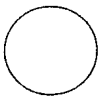


FLUJO DE DATOS

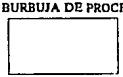
BASE DE DATOS



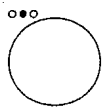
CUADRO DE DESCRIPCION



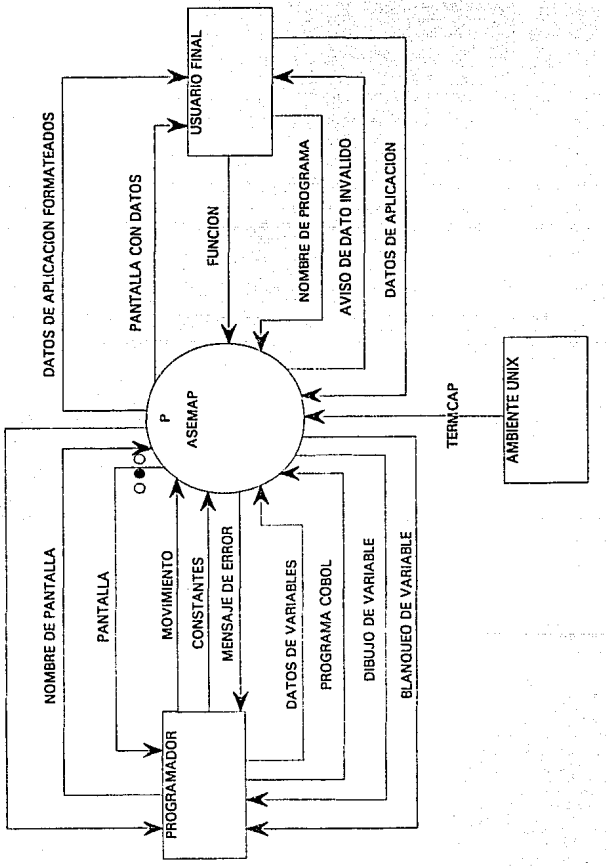
BURBUJA DE PROCESO

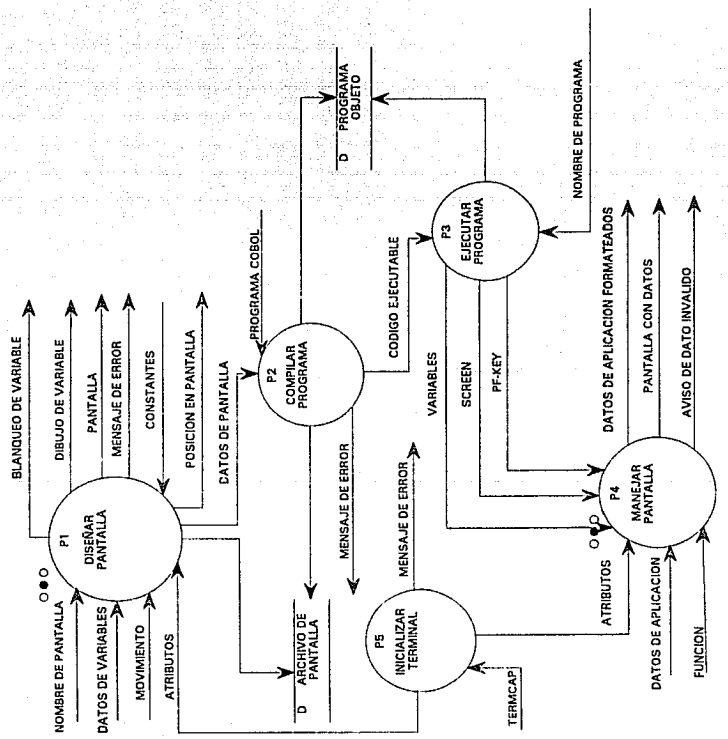


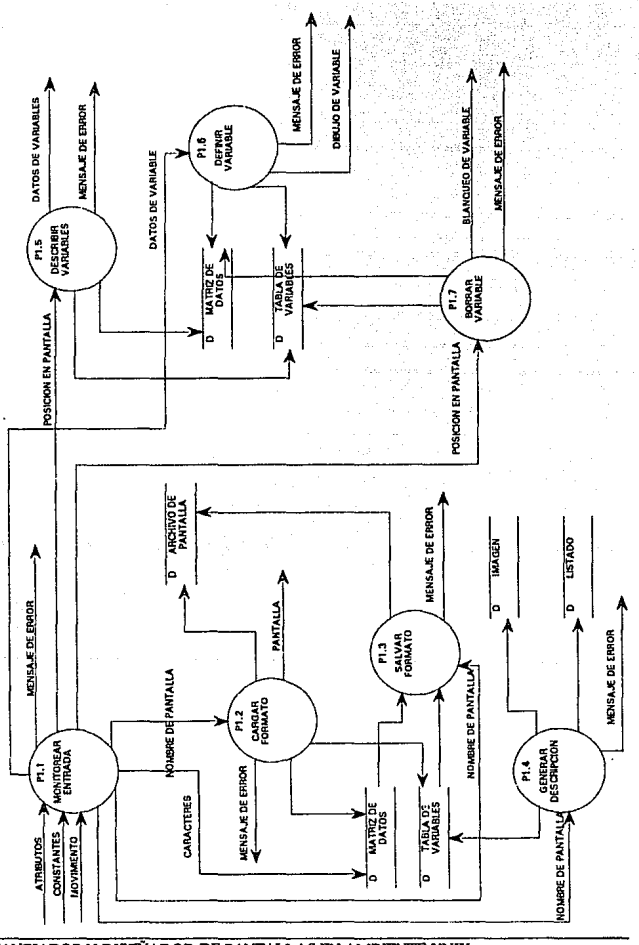
TERMINADOR/ORIGEN

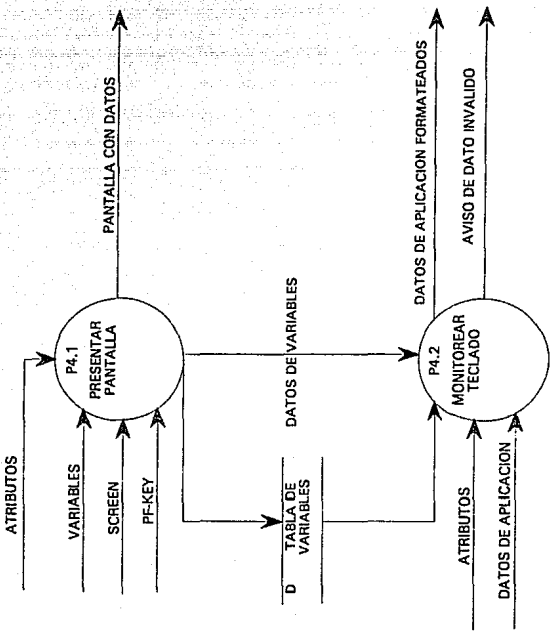


BURBUJA DE PROCESO CON PROCESOS DEPENDIENTES


USUARIO FINAL
DATOS DE APLICACION FORMATEADOS
PANTALLA CON DATOS
FUNCION
NOMBRE DE PROGRAMA
AVISO DE DATO INVALIDO
DATOS DE APLICACION
P
ASEMAP
TERMCAP
AMBIENTE UNIX
POSICION EN PANTALLA
NOMBRE DE PANTALLA
PANTALLA
MOVIMIENTO
CONSTANTES
MENSAJE DE ERROR
DATOS DE VARIABLES
PROGRAMA COBOL
DIBUJO DE VARIABLE
BLANQUEO DE VARIABLE
PROGRAMADOR


NOMBRE DE PANTALLA
DATOS DE VARIABLES
MOVIMIENTO
ATRIBUTOS
P1
DISEÑAR PANTALLA
BLANQUEO DE VARIABLE
DIBUJO DE VARIABLE
PANTALLA
MENSAJE DE ERROR
CONSTANTES
POSICION EN PANTALLA
DATOS DE PANTALLA
P2
COMPILAR PROGRAMA
PROGRAMA COBOL
D ARCHIVO DE PANTALLA
MENSAJE DE ERROR
D PROGRAMA OBJETO
CODIGO EJECUTABLE
P3
EJECUTAR PROGRAMA
NOMBRE DE PROGRAMA
VARIABLES
SCREEN
PF-KEY
DATOS DE APLICACION FORMATEADOS
PANTALLA CON DATOS
AVISO DE DATO INVALIDO
P4
MANEJAR PANTALLA
ATRIBUTOS
DATOS DE APLICACION
FUNCION
P5
INICIALIZAR TERMINAL
MENSAJE DE ERROR
TERMCAP


ATRIBUTOS
CONSTANTES
MOVIMIENTO
P1.1 MONITOREAR ENTRADA
P1.2 CARGAR FORMATO
P1.3 SALVAR FORMATO
P1.4 GENERAR DESCRIPCION
P1.5 DESCRIBIR VARIABLES
P1.6 DEFINIR VARIABLE
P1.7 BORRAR VARIABLE
CARACTERES
NOMBRE DE PANTALLA
MENSAJE DE ERROR
PANTALLA
POSICION EN PANTALLA
DATOS DE VARIABLE
DATOS DE VARIABLES
DIBUJO DE VARIABLE
BLANQUEO DE VARIABLE
D MATRIZ DE DATOS
D TABLA DE VARIABLES
D ARCHIVO DE PANTALLA
D IMAGEN
D LISTADO


ATRIBUTOS
VARIABLES
SCREEN
PF-KEY
P4.1
PRESENTAR PANTALLA
PANTALLA CON DATOS
DATOS DE VARIABLES
D
TABLA DE VARIABLES
P4.2
MONITOREAR TECLADO
DATOS DE APLICACION FORMATEADOS
AVISO DE DATO INVALIDO
ATRIBUTOS
DATOS DE APLICACION

MINIESPECIFICACIONES

Una MiniEspecificación es un *documento que satisface nuestros objetivos de Especificación de la mejor manera posible*. Es una descripción escrita de los procesos primitivos de los DFD'S; es decir, de los procesos que ya no se descompusieron en más burbujas. Por lo tanto, existirá una MiniEspecificación por cada proceso primitivo.

Las Miniespecificaciones serán hechas utilizando un Pseudo-lenguaje llamado Español Estructurado.

Las Miniespecificaciones del Sistema son las siguientes: Monitorear entrada, Cargar formato, Salvar formato, Generar descripción, Describir variable, Definir variable, Borrar variable, Presentar pantalla, Monitorear teclado, Compilar programa, Ejecutar programa e Inicializar terminal. Además de su nombre, cada MiniEspecificación tendrá un número que será el mismo utilizado en el proceso del DFD correspondiente.

Número de Proceso : 1.1
Nombre de Proceso : Monitorear Entrada.

Para cada tecla que se presione
    Validar que sea un caracter ASCII aceptado por COBOL
    Si se trata de un caracter de texto (constante)
        Si en la posición actual no esta declarada una variable
            Grabar el caracter en la Matriz de Datos.
            Escribir el caracter en la pantalla.
            Avanzar el cursor a la siguiente posición en la pantalla
    Si se trata de una tecla de movimiento de cursor horizontal.
        Si en la posición a la que se desea cambiar existe definida una variable
            Mover el cursor a la primera posición a la izquierda o derecha que no este ocupada por la variable
            Enviar un mensaje de error
        De otra manera
            Mover el cursor a la posición que corresponda a la tecla
    Si se trata de una tecla de movimiento de cursor vertical.

Continuación

Mover el cursor a la posición que corresponda a la tecla
Si es el caracter <CONTROL> A
Pintar el primer cuadro del menú de ARCHIVO
Si el siguiente caracter es una tecla de movimiento horizontal
Borrar el menú actual
Reestablecer los datos ocupados por el menú
Presentar el menú correspondiente.
Posicionar el cursor en la primera opción de tal menú
Escribir con atributo de video inverso la opción actual
Si el siguiente caracter es una tecla de movimiento vertical
Escribir con atributo de video normal la opción actual
Posicionar el cursor a la opción del menú que corresponda a la tecla
Escribir en video inverso la nueva opción
Si el siguiente caracter es un <ENTER>
Borrar el menú actual
Reestablecer los datos ocupados por el menú
Ejecutar el proceso correspondiente a la opción seleccionada
De otra manera:
Enviar un aviso de error.
De otra manera
Enviar un aviso de error

Número de Proceso : 1.2
Nombre de Proceso : Cargar Formato.

Aceptar el nombre de la pantalla.
Si existe el archivo de la pantalla
Para cada línea del archivo
Si es un FILLER con valores.
Si el primer caracter es un tilde
Si los cuatro siguientes caracteres son ##$+
Acualizar los datos en la tabla de variables tomando conjuntos de once caracteres para obtener de cada variable :
Posición en el arreglo de datos
Posición renglón columna en la pantalla,
Longitud corta y larga,
Máscara de edición

Continuación

Número de decimales
De otra manera
Tomar los siguientes caracteres constantes y actualizarlos en la matriz de pantalla
Si son las variables de la pantalla
Obtener su nombre y número de ocurrencias actualizando la tabla de variables.
Asignar un número de variable y llenar en la matriz de pantalla el espacio ocupado por ésta con dicho número.
Si es un FILLER con LOW-VALUES
Escribir la pantalla en la terminal tomando como base la matriz de pantalla.
De otra manera
Enviar un mensaje de error.

Número de Proceso : 1.3
Nombre de Proceso : Salvar Formato.

Aceptar el nombre del formato de pantalla
Escribir en el archivo de pantalla el encabezado, que consiste en el nombre de la pantalla entre asteriscos.
Para cada caracter en la matriz de pantalla
Si el caracter es uno ASCII válido en COBOL
Escribir en el archivo de formato de pantalla dentro de FILLER's de COBOL una tilde seguida de dos caracteres que correspondan a la posición renglón columna del caracter en la pantalla y todos los caracteres que continúen hasta encontrar fin de linea o un caracter no válido en COBOL.
Ordenar la tabla de variables comenzado por las alfanuméricas y éstas a su vez de acuerdo a su posición en la pantalla, esto es de izquierda a derecha y de arriba a abajo
Para cada variable en la tabla de variables
Escribir en el archivo de formato de pantalla los caracteres correspondientes a su posición, dos signos de interrogación, longitud corta y larga, máscara de edición, número de decimales y signo.
Escribir en el archivo de formato de pantalla las variables de la pantalla con el formato de COBOL y un FILLER con LOW-VALUES

Número de Proceso : 1.4
Nombre de Proceso : Generar Descripción.

Aceptar nombre de formato de pantalla
Si existe formato de pantalla
    Escribir en el archivo de imagen un marco rodeando la pantalla
        Para cada renglón en la matriz de pantalla
            Si el código ASCII del caracter es mayor a 126 escribir un subguión en el archivo de imagen
            De otra manera
            Escribir el caracter correspondiente en el archivo de imagen
        Escribir en el archivo de listado la numeración de las columnas, del uno al ochenta
    Para cada renglón en la matriz de pantalla
        Escribir el número de renglón en el archivo de listado
        Si el código ASCII del caracter es mayor a 126
            Escribir un subguión ' _ ' en el archivo de listado
        De otra manera
            Escribir el caracter correspondiente en el archivo de listado
    Para cada variable definida en la tabla de variables:
        Escribir en el archivo de listado su nombre, tipo, renglón, columna, longitud corta longitud larga, ocurrencias y máscara.
De otra manera
        Enviar un mensaje de error

Número de Proceso : 1.5
Nombre de Proceso : Describir Variable.

Aceptar la posición actual del cursor
Si en la posición en la que se encuentra el cursor existe una variable,
        Escribir un marco dando el nombre de la variable, su tipo, su longitud corta y larga, su máscara de edición, número de ocurrencias y número de decimales
        Restaurar los datos ocupados por el marco.
De otra manera
        Enviar un mensaje de error

Número de Proceso : 1.6
Nombre de Proceso : Definir Variable.

Pintar un marco pidiendo el tipo de variable
Aceptar tipo de variable
Si se eligió una variable tipo alfanumérica
    Aceptar el nombre de la variable
    Aceptar su Longitud y número de ocurrencias
    Si la longitud de la variable excede a la columna 80 o el número de ocurrencias se pasa del renglón 24
        Enviar un mensaje de error
Si se eligió una variable tipo numérica
    Aceptar el nombre de la variable
    Aceptar su Longitud, número de ocurrencias
    Si la longitud de la variable excede a la columna 80 o el número de ocurrencias se pasa del renglón 24
        Enviar un mensaje de error
    Escribir un marco en la pantalla que contenga los diferentes tipos de máscara de edición y aceptar el valor deseado
    Restaurar el marco ocupado por los tipos de máscara
    Escribir un marco en la pantalla que contenga los diferentes tipos de signado y aceptar el valor deseado.
    Restaurar el marco ocupado por los tipos de signado
    Si la variable es de tipo real
        Aceptar el número de decimales
        Si el número de decimales es incorrecto
            Enviar un mensaje de error
Si se eligió una variable tipo fecha
    Aceptar el nombre de la variable
    Aceptar el número de ocurrencias
    Si el número de ocurrencias se excede del renglón 24
        Enviar un mensaje de error
    Escribir un marco que contenga los tipos de máscara de edición válidos y aceptar el valor deseado
    Restaurar el marco ocupado por los tipos de máscara de edición

Restaurar el marco ocupado para pedir los datos de la variable

Continuación

Actualizar la tabla de variables.
Acualizar la matriz de pantalla poniendo el número de variable en el espacio que ocupa
Escribir en la pantalla subrayas en el espacio ocupado por la variable.

Número de Proceso : 1.7
Nombre de Proceso : Borrar variable.

Aceptar la posición actual del cursor
Si en la posición existe una variable
    Escribir un marco dando el nombre de la variable y un letrero de confirmación.
    Aceptar cofirmación de borrado
    Si es afirmativo el borrado
        Borrar el registro de la variable en la tabla de variables
        Blanquear el espacio ocupado por la variable en la matriz de pantalla y en la terminal
    Restaurar la parte que ocupa el marco con el nombre de la variable
De otra manera
    Enviar un mensaje de error.

Número de Proceso : 2
Nombre de Proceso : Compilar programa.

Para cada programa que se compile
    Fijar en el medio ambiente de UNIX la variable RMPATH con el valor de la trayectoria, donde se encuentran los COPYS (archivos de pantalla) y la de PATH que contenga el directorio /usr/rmcobol
    Posicionarse en el directorio en el cual se encuentra el programa fuente de COBOL
    Compilar el programa con el comando rmcobol
    Si existen errores de compilación
        Enviar los mensajes a la pantalla para que sean corregidos

Número de Proceso : 3
Nombre de Proceso : Ejecutar programa.

Para cada programa que se desee ejecutar
Fijar en el medio ambiente de UNIX la variable RUNPATH con el valor de la trayectoria donde se encuentran los programas objeto y la de PATH que contenga el directorio /usr/rmcobol
Posicionarse en el directorio en el cual se encuentran los archivos de de datos de aplicación
Fijar los parámetros de la terminal en **raw y noecho**
Ejecutar el programa con el comando **runcobol** dando como parámetro el nombre del programa
Pasar los valores de variables, screen y PF-KEY al programa manejador

Número de Proceso : 4.1
Nombre de Proceso : Presentar Pantalla.

Para cada caracter en el arreglo de SCREEN
Si el caracter es un tilde
Si los siguientes 4 caracteres son ##$+
Para cada conjunto de once caracteres que continúen
Calcular la posición en el arreglo de datos del valor de la variable con la fórmula: código ASCII del primero menos 35 por 91 más código ASCII del segundo menos 35
Obtener su posición renglón columna en la pantalla, longitud larga y corta y número de decimales siendo estos valores el codigo ASCII de los siguientes caracteres menos 35, respectivamente.
Obtener el valor del tipo de máscara y signado con los carecteres décimo y onceavo respectivamente
Actualizar la tabla de variables con los datos
Escribir en la pantalla el atributo de desplegado
Obtener el valor de la variable del arreglo de datos
Formatear el valor según la máscara de edición y escribirlo en la pantalla.
De otra manera
Posicionar el cursor en la posición renglón columna que se calcula con el código ASCII de los siguientes dos caracteres menos 35
Escribir los siguientes caracteres en la pantalla.

Número de Proceso : 4.2
Nombre de Proceso : Monitorear teclado.

Para cada tecla que se presione
Si es un <ENTER>
    Si es un campo tipo fecha
        Si la fecha es una real a partir de 1900
            Escribir el registro temporal de lo que se tecleó
        De otra manera
            Enviar un mensaje de dato inválido
    De otra manera
        Formatear el registro temporal según máscara de edición
        Escribir el atributo de despliegue
        Escribir el registro temporal en pantalla
    Si es el último campo en la pantalla
        Regresar al programa COBOL los valores de PF-KEY,
            y los de las variables, así como el control de ejecución
    De otra manera
    Avanzar al siguiente campo
De otra manera
Si es un <TABULADOR>
    Formatear el registro temporal según máscara de edición
    Escribir el atributo de despliegue
    Escribir el registro temporal en pantalla
    Regresar al campo anterior
Si es un BACKSPACE
    Si es una variable de tipo fecha y se encuentra posicionado después de una diagonal "/"
        Regresa el cursor 2 posiciones
    De otra manera
        Regresa el cursor 1 posición
Si es un guión y el campo es de tipo numérico
    Si el campo es signado y el cursor se encuentra al principio
        Escribir el guión en pantalla
        Guarda el guión en el registro temporal
        Avanza el cursor una posición
    De otra manera
    Envía un mensaje de error
Si es un punto y el campo es numérico
    Si el campo es real y previamente no se ha tecleado otro punto en este campo
        Escribir el punto en la pantalla
        Guarda el punto en el registro temporal
        Avanzar el cursor una posición

Continuación

De otra manera
Enviar un mensaje de dato inválido
Si es un caracter válido de COBOL
Si es una variable numérica o fecha y el caracter no es un número
Enviar mensaje de error
De otra manera
Si el campo es numérico real y con este caracter se llega a la longitud de la parte entera
Escribir el número y un punto en la pantalla
Guardar el número y el punto en el registro temporal
Avanzar el cursor dos posiciones
De otra manera
Escribir el caracter en la Pantalla
Guarda el caracter en el registro temporal
Si con este caracter se llega a la longitud corta del campo
Si es el último campo en la pantalla
Regresar al programa COBOL los valores de PF-KEY, y los de las variables, así como el control de ejecución
De otra manera
Si es un campo tipo fecha
Validar que la fecha sea una real a partir de 1900
Escribir el registro temporal en pantalla
De otra manera
Formatear el registro temporal según máscara de edición
Escribir el atributo de despliegue
Escribir el registro temporal en pantalla
Avanza al siguiente campo
De otra manera
Avanzar el cursor una posición
De otra manera
Enviar mensaje de error
Si es un <ESCAPE>
El programa sólo aceptará dos números
Se actualizarán estos números en el arreglo de función
Regresar al programa COBOL los valores de PF-KEY, y los de las variables, así como el control de ejecución
Si es <CONTROL-C>
Termina la ejecución de los programas manejador y del COBOL

Número de proceso : 5
Nombre de proceso : Inicializar Terminal.

Para cada ocasión que se entre al programa MANEJADOR y DISEÑADOR.
    Inicializar las variable de señales
    Si existe definición del tipo de terminal de TERMCAP
        Guardar el medio ambiente en un buffer temporal
        Fijar los valores del tiempo de reconocimiento de una tecla
        Traer el valor de la variable de ambiente "TERM"
        Asignar los valores de secuencias de escape de clear screen, cursor motion y de los atributos de video normal, normal parpadeante, normal inverso,normal subrayado, alta intensidad, alta intensidad parapadeante, alta intensidad inverso, alta intensidad subrayado.
    De otra manera
        Enviar un mensaje de error.
        Terminar el programa manejador y el programa COBOL.

DICCIONARIO DE DATOS

Un Diccionario es un conjunto ordenado de definiciones. En este caso en particular las definiciones serán de Flujos de Datos. Es decir, para cada Flujo de Dato utilizado en un DFD, existirá una Definición en el Diccionario.

La definición será hecha en función de los componentes de dicho flujo. Por ejemplo, el Flujo POSICION EN PANTALLA estará definido por los siguientes elementos: NUMERO_DE_RENGLON + NUMER_DE_COLUMNA. A su vez, dichos elementos pueden estar definidos en el Diccionario.

El formato utilizado en el diccionario será: para cada definición se utilizará un rectángulo dividido en dos; la parte superior describe el nombre del flujo y la parte inferior los elementos de dicho flujo. El signo "+" representa un "AND" lógico, el signo de "=" representa "EQUIVALENTE A", y los "[]" representan la "SELECCION" de uno de los elementos dentro de los corchetes.

Los flujos de este Sistema son los siguientes :

FLUJO DE DATOS NOMBRE DE PANTALLA =
ELEMENTOS NOMBRE + .scrn

FLUJO DE DATOS	PANTALLA =
ELEMENTOS	CONSTANTES + VARIABLES

FLUJO DE DATOS	MOVIMIENTO =
ELEMENTOS	TECLA DE CURSOR [IZQ.|DER.|ARRIBA|ABAJO]

FLUJO DE DATOS	POSICION EN PANTALLA =
ELEMENTOS	NUMERO DE RENGLON + NUMERO DE COLUMNA

FLUJO DE DATOS	CONSTANTES =
ELEMENTOS	1{CARACTERES ASCII}80

FLUJO DE DATOS
AVISO DE ERROR =
ELEMENTOS
MENSAJE [TEXTO|BEEP]

FLUJO DE DATOS
DATOS DE VARIABLE =
ELEMENTOS
TIPO [ALFANUMERICA|ENTERA|REAL|FECHA] + NOMBRE DE VARIABLE + LONGITUD + OCURRENCIAS + (SIGNADO) + (MASCARA DE EDICION) + (NUMERO DE DECIMALES)

FLUJO DE DATOS
PROGRAMA COBOL =
ELEMENTOS
INSTRUCCIONES DE RMCOBOL 85

FLUJO DE DATOS PANTALLA CON DATOS =
ELEMENTOS CONSTANTES DE LA PANTALLA + VALORES DE LAS VARIABLES

FLUJO DE DATOS AVISO DE DATO INVALIDO =
ELEMENTOS BEEP

FLUJO DE DATOS DATOS DE APLICACION =
ELEMENTOS DATOS [ALFANUMERICOS|NUMERICOS|REALES|FECHAS]

FLUJO DE DATOS TERMCAP =
ELEMENTOS SECUENCIAS DE ESCAPE

FLUJO DE DATOS BLANQUEO DE VARIABLE =
ELEMENTOS CONJUNTO DEL CARACTER BLANCO

FLUJO DE DATOS DIBUJO DE VARIABLE=
ELEMENTOS CONJUNTO DEL CARACTER SUBRAYA

FLUJO DE DATOS PANTALLA=
ELEMENTOS CONSTANTES + DIBUJO DE VARIABLES

FLUJO DE DATOS NOMBRE DE PROGRAMA =
ELEMENTOS NOMBRE + .CBL

FLUJO DE DATOS ATRIBUTOS =
ELEMENTOS NORMAL + PARPADEANTE + VIDEO INVERSO + SUBRAYADO + ALTA INTENSIDAD + ALTA INTENSIDAD PARPADEANTE + ALTA INTENSIDAD VIDEO INVERSO + ALTA INTENSIDAD SUBRAYADO

FLUJO DE DATOS CODIGO EJECUTABLE =
ELEMENTOS INSTRUCCIONES DE RUNTIME DE RMCOBOL 85

FLUJO DE DATOS DATOS DE APLICACION FORMATEADOS =
ELEMENTOS DATOS {NUMERICOS|REALES} + FORMATO [SUPRESION DE CEROS | SIGNO DE PESOS FLOTANTE | PROTECCION CON ASTERISCOS | SUPRESION DE CEROS CON COMAS | SIGNO DE PESOS FLOTANTE Y CON COMAS PROTECCION DE ASTERISCOS CON COMAS DATOS {FECHA} + FORMATO {DD/MM/YY | MM/DD/YY | YY/MM/DD}

FLUJO DE DATOS PF-KEY =
ELEMENTOS ESCAPE + NUMERO

FLUJO DE DATOS VARIABLES =
ELEMENTOS VALORES DE LAS VARIABLES

FLUJO DE DATOS SCREEN =
ELEMENTOS MAPA DE CONTROL DE LA PANTALLA

FLUJO DE DATOS DATOS DE VARIABLES =
ELEMENTOS SCREEN + VARIABLES

IV.-DISEÑO DEL SISTEMA NUEVO

METODOLOGIA

El objetivo del Diseño Estructurado es el de *construir Sistemas Modularizados que nos permitan mantenerlos y evaluarlos fácilmente.* El Diseño Estructurado está basado en dos técnicas: Análisis de Transformación y Análisis de Transacciones; además cuenta con técnicas de refinamiento que son la Cohesividad y Acoplamiento y una herramienta gráfica llamada Diagrama de Estructura.

La técnica de Transformación de Análisis será empleada para el Diseño de este Sistema, ya que nos permitirá derivar nuestros Diagramas de Estructura a partir de la Especificación hecha en el Análisis Estructurado.

Esta técnica consta de cuatro partes fundamentales:

1.- Representar el problema con un DFD,

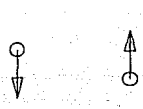
2.- Identificar los datos Aferentes (de entrada) y Eferentes (Salida),

3.- Elaborar el primer nivel de factorización,

4.- Factorizar los módulos Aferente, Eferente y de Transformación.

La primera parte; es decir el DFD se encuentra en la Especificación del Sistema. Posteriormente, los elementos Aferentes encontrados fueron el Diseñador y el Manejador de pantallas.

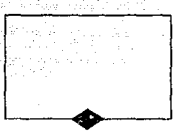
La Simbología para la elaboración de los diagramas es la siguiente:



COUPLES

LLAMADA A UN MODULO

MODULO

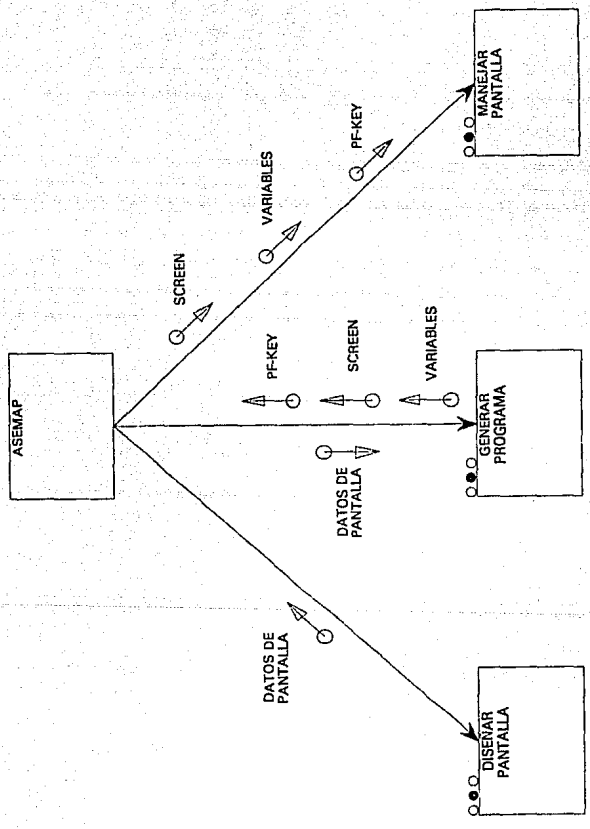


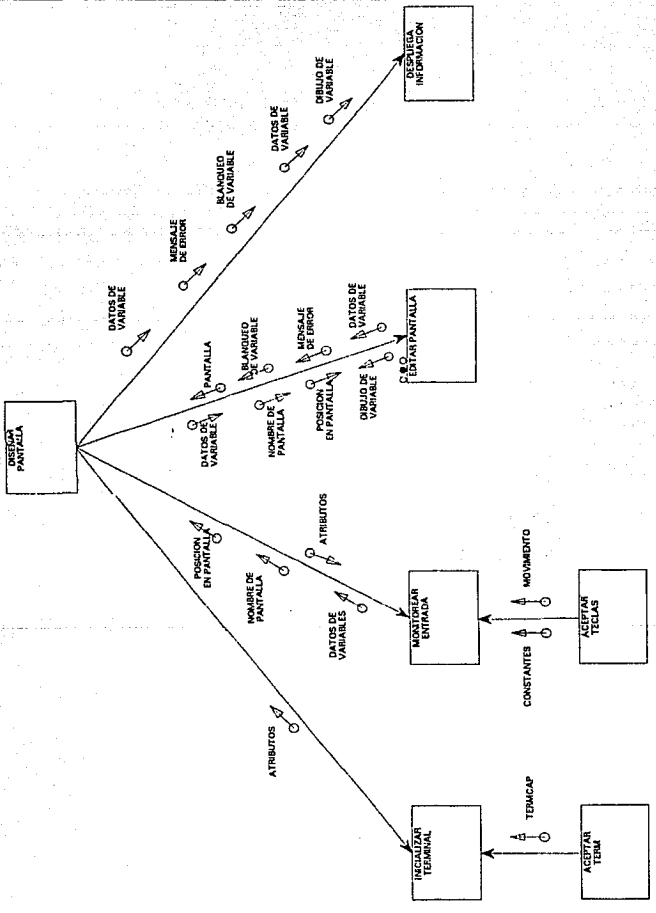
DECISION *

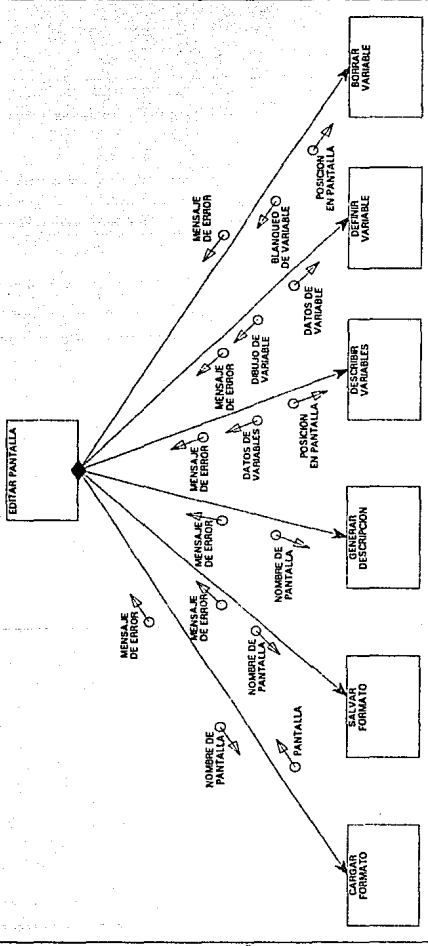
DIAGRAMAS DE ESTRUCTURA

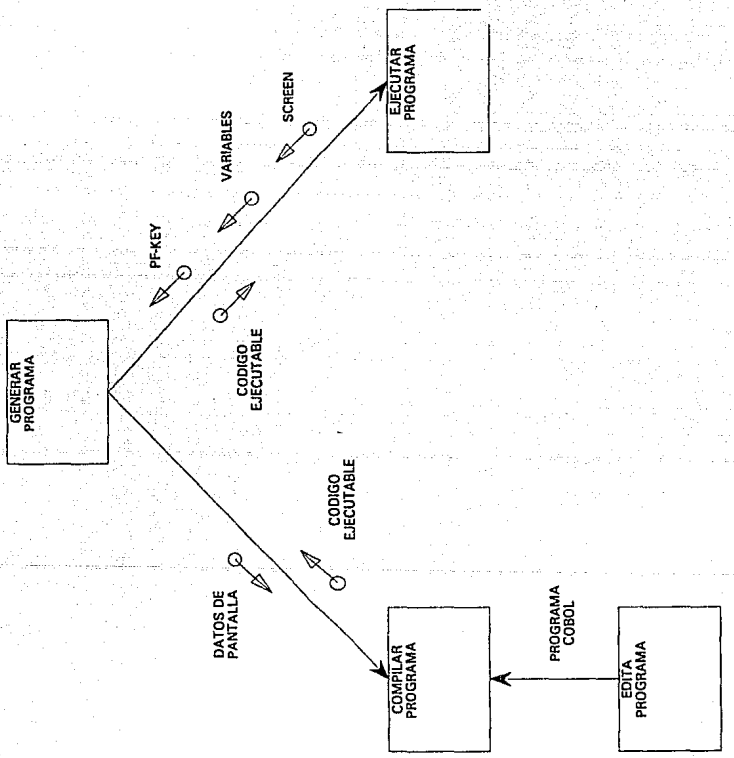
Los diagramas obtenidos al aplicar la Técnica de Transformacion de Análisis son los siguientes:

* NOTA: LA METODOLOGIA DE DISEÑO ESTRUCTURADO UTILZA UN ROMBO PARA INDICAR DECISION, SIN EMBARGO, POR NO CONTAR EL CASE CON UN ROMBO DE ESTE COLOR , SE UTILIZO UNO NEGRO PARA EL MISMO FIN


ASEMAP
SCREEN
VARIABLES
PF-KEY
MANEJAR PANTALLA
PF-KEY
SCREEN
VARIABLES
GENERAR PROGRAMA
DATOS DE PANTALLA
DATOS DE PANTALLA
DISEÑAR PANTALLA


DESPLIEGA INFORMACION
DESPLIEGA PANTALLA
DATOS DE VARIABLE
MENSAJE DE ERROR
BLANQUEO DE VARIABLE
DATOS DE VARIABLE
DIBUJO DE VARIABLE
DATOS DE VARIABLE
PANTALLA
BLANQUEO DE VARIABLE
MENSAJE DE ERROR
DATOS DE VARIABLE
DATOS DE VARIABLE
NOMBRE DE PANTALLA
POSICION EN PANTALLA
DIBUJO DE VARIABLE
EDITAR PANTALLA
POSICION EN PANTALLA
NOMBRE DE PANTALLA
ATRIBUTOS
DATOS DE VARIABLES
MONITOREAR ENTRADA
MOVIMIENTO
CONSTANTES
ACEPTAR TECLAS
ATRIBUTOS
INICIALIZAR TERMINAL
TERMCAP
ACEPTAR TERM


EDITAR PANTALLA
CARGAR FORMATO
SALVAR FORMATO
GENERAR DESCRIPCION
DESCRIBIR VARIABLES
DEFINIR VARIABLE
BORRAR VARIABLE
MENSAJE DE ERROR
PANTALLA
NOMBRE DE PANTALLA
DATOS DE VARIABLES
POSICION EN PANTALLA
DIBUJO DE VARIABLE
BLANQUEO DE VARIABLE
DATOS DE VARIABLE


EDITA PROGRAMA
PROGRAMA COBOL
COMPILAR PROGRAMA
DATOS DE PANTALLA
CODIGO EJECUTABLE
GENERAR PROGRAMA
CODIGO EJECUTABLE
PF-KEY
VARIABLES
SCREEN
EJECUTAR PROGRAMA

MANEJAR PANTALLAS
VARIABLES
PF-KEY
SCREEN
DATOS DE LAS VARIABLES
PANTALLA CON DATOS
ATRIBUTOS
DATOS DE LAS VARIABLES
AVISO DE DATO INVALIDO
DATOS DE APLICACION FORMATEADOS
ATRIBUTOS
AVISO DE DATO INVALIDO
DATOS DE APLICACION FORMATEADOS
PANTALLA CON DATOS
PRESENTAR PANTALLA
MONITOREAR TECLADO
DESPLIEGA DATOS

V.-IMPLANTACION Y PRUEBAS

IMPLANTACION

Tomando en cuenta que el equipo HP Vectra RS 25C, tiene instalado el sistema operativo SCO XENIX V3.2.3, lo siguiente es instalar el paquete SCO Development System, que es el que contiene el lenguage de programación "C". También es necesario verificar la existencia del archivo que se encuentra en el directorio /etc que se llama **termcap**.

La implantación del sistema se dividió en las siguientes partes :

1- Edición de la pantalla.

2- Generación del archivo de "COPY", del formato de pantalla.

3- Elaboración de los reportes de imagen y listado de la pantalla

4- Elaboración del manejador de pantallas.

5- Programación del sistema.

1- Edición de la pantalla.

Esta parte dará la facilidad de poder definir constantes y variables para la pantalla, así como también invocar a los menús y ejecutar la opción selecionada.

2- Generación del archivo de "COPY".

Una vez que se ha diseñado la pantalla, se procederá a crear tal archivo, para esto se toma en cuenta que existe definida una matriz de 80 por 24 que contiene los datos de la pantalla para cada punto de ésta, también, se cuenta con una tabla de variables que contiene los datos de las variables definidas.

3-Elaboración de los Reportes.

En esta parte los reportes se diseñaron básicamente tomando en cuenta que el reporte de imagen es pasar a papel la pantalla tal como se " ve " en la terminal. y el reporte de listado es la imagen de la pantalla numerando los renglones y columnas acompañado de una lista de las variables y los datos que contiene la pantalla.

4- Elaboración del manejador de pantallas.

Se inicia con la presentación de las constantes de la pantalla para después escribir los valores de las variables con sus atributos de despliegue y se continuará con la validación de los datos que vaya teeleando el usuario final.

5.-Programación del sistema

En esta etapa es cuando se realizaron los programas en lenguaje "C", que corresponden uno al diseñador y otro al manejador de pantallas, dado que los programas ocupan en total 1800 líneas de código fuente se presentará sólo la parte que obtiene la información del tipo de terminal en la cual se correrán estos programas :

Se inicia con la definicion de las funciones de :

Movimiento de cursor

```
#define x_goto(A,B) printf("%s",tgoto(cm,B-1,A-1))
```

Activar atributo de despliegue

```
#define x_att(A) printf("%s",at[A])
```

La función que inicializa el programa manejador es la que se muestra abajo y como se puede apreciar recibe tres parámetros, el primero se refiere al valor de la función con la cual el usuario

desea salir, el segundo es la cadena de screen mencionada en el capítulo 2 y por último la cadena que contiene el valor de las variables, también se muestran las definiciones de algunas variables del programa.

```
int asemap(func,screen,variable)
char *func,*screen,*variable;
{ char letr[83],t1,t2,*p,*key,pnt[9],pnta[9];
  int j,jj,no_first,si,n_dec,n_mas,var,var1,tit,tit1,l_lar,l_cor,p_mes=0;
  int  n_sig,numvar,fin,keys;
  char *at[10],atr[10][12],*cl,*terminal,*area;
  char *tgetstr(),*getenv();
  char bp1[1024],bp2[1024];
```

Con estas instrucciones se habilitan las señales de :

Interrupción

```
 signal(SIGINT,handint);
```

Quit

```
 signal(SIGQUIT,handint);
```

Segment Violation

```
 signal(SIGSEGV,handint);
```

La siguiente instrucción "abre" la terminal como si fuera un archivo,es decir, se leéra y escribirá de y hacia la terminal.

```
 if((term = open("/dev/tty",O_RDWR)) < 0)
return -3;
```

Las siguientes instrucciones son llamadas al sistema que hacen referencia a la estrucura de termio; TCGETA toma los parámetros asociados con la terminal y los almacena en itty y otty.

```
 ioctl(term,TCGETA,&itty);
 ioctl(term,TCGETA,&otty);
```

Con la siguiente instrucción se enciende la bandera que habilita el uso de señales.

```
 itty.c_lflag = ISIG;
```

Las siguientes instrucciones son valores que se ponen a caracteres de control, VMIN es el mínimo número de caracteres que se pueden recibir, y VTIME es el tiempo (en décimas de segundo) que se debe esperar cuando más para recibir un segundo caracter, es decir, cuando se recibe un caracter después de otro en un intervalo de una décima de segundo entonces se indicará que se presionó una tecla que esta compuesta por más de un caracter (por ejemplo las flechas de movimiento).

```
itty.c_cc[VMIN] = 2;
itty.c_cc[VTIME] = 1;
```

La siguiente instrucción pone en la terminal los valores que se han modificado.

```
ioctl(term,TCSETA,&itty);

 if (isatty(0))
```

Con las siguientes instrucciones se tomarán los atributos de la terminal a través de la variable TERM del ambiente UNIX, que toma las secuencias de escape del archivo /etc/termcap de acuerdo al tipo de terminal.

```
  if (terminal=getenv("TERM"))
   { ii=tgetent(bp1,terminal);
    area=bp1;
    cl=tgetstr("cl",&area);          /*  limpiado de la pantalla                    */
    cm=tgetstr("cm",&area);          /* movimiento del cursor                       */
    at[1]=tgetstr("Nm",&area)        /* atr de desp. normal                         */
    at[2]=tgetstr("Nb",&area);       /* atr de desp en parpadeante                  */
    at[3]=tgetstr("Nr",&area);       /* atr de desp en video inverso                */
    at[4]=tgetstr("Ns",&area);       /* atr de desp subrayado                       */
    at[5]=tgetstr("Al",&area);       /* atr de desp con alta intensidad             */
    at[6]=tgetstr("Ab",&area);       /* atr de desp con alta intensidad arpadeante  */
    at[7]=tgetstr("Ar",&area);       /* atr de desp con alta intensidad en video inverso */
    at[8]=tgetstr("As",&area);       /* atr de desp con alta intensidad subrayado   */
    at[9]=tgetstr("Fi",&area);
    for (ii=1;ii<=9;ii++) strcpy(atr[ii],at[ii]);
    first=1;
   }
```

La siguiente parte ilustra la forma en la cual se reciben los caracteres teclеados, como puede notarse si el primer caracter que se recibe tiene como código un 27 (que corresponde a un escape), entonces si se recibe un segundo caracter (dentro de la siguiente decima de segundo) significa que se está oprimiendo una tecla que envia más de un caracter, para lo cual se lee uno o dos caracteres más.

Esto lo checa la función rdchk, que obtiene si hay los siguientes caracteres del buffer.

```
char getch()
{
        char c;
        extern int DEBUG;
        static char buff[5];
        while(read(term,&c,1)==-1);
        if(c == 27 && rdchk(term)>0) {
                int i;
                read(term,buff,2);
                for(i = 17;i < 35; i++) {
                        if(i == 25) i = 27;
                        if(atrib[i] == NULL) continue;
                        if(!strncmp(buff,atrib[i]+1,strlen(atrib[i]+1))) return 16 - i;
                }
                bell(); return 0;
        }
        if(c == 6) {
                read(term,&c,1);
                return -(c-'0') - 10;
        }
        if(c == 1) {
                DEBUG = 1 - DEBUG;
                bell() ;
                return 0;
        }
        return c;
}
```

PRUEBAS

Para comprobar que el sistema cumple con los requisitos establecidos, se elaboró un programa en el lenguaje COBOL, que realice altas, bajas, modificaciones y consultas a un archivo de pólizas. Para esto se creó con el DISEÑADOR DE PANTALLAS una forma que permitiera manejar los campos requeridos para efectuar las transacciones al archivo mencionado anteriormente.

La forma quedó de la siguiente manera :

PANTALLA DE PRUEBA

ALTA DE POLIZA AUTOMOVILES

NUM. POLIZA : _____ VIGENCIA DE : _____ A : _____

PROPIETARIO : _____

CLAVE VEHIC.: ___ _____ MODELO : __

PLACAS : _____ DEDUCIBLES

SUMA ASEGURADA : _____ _% RESPONSABILIDAD CIVIL

_% DA&OS

PRIMA : _____ _% ROBO TOTAL

Tambien se verificó que el archivo de COPY fuera generado correctamente ,éste se presenta a continuación:

```
*****************************************************************
*                        SCREEN  PANT001                        *
*****************************************************************
 01  SCREEN-PANT001.
  02 FILLER PIC X(118) VALUE
     "PANT001 HS;$e¯!@PANTALLA DE PRUEBA  ¯)<ALTA DE POLIZA AUTOMO
-    "VILES  ¯,+NUM. POLIZA : ¯,HVIGENCIA DE : ¯.,A : ¯.+PROPIET".
  02 FILLER PIC X(118) VALUE
     "ARIO : ¯0+CLAVE VEHIC.: ¯0_MODELO : ¯2+PLACAS : ¯2MDEDUCIBLE
-    "S   ¯4+SUMA ASEGURADA : ¯4S%  ¯6VRESPONSABILIDAD CIVIL  ¯5S".
  02 FILLER PIC X(118) VALUE
     "%  ¯5VDA&COS  ¯6+PRIMA : ¯6S%  ¯6VROBO TOTAL  ¯##$+$g??,9--#1
-    "N$a??,V+)#7N$(??,c+)#7N##??.9AA#A#%(??09!'#NN#8??0>AA#A#%.".
  02 FILLER PIC X(108) VALUE
     "??0h%X#N#N#a??24))#AN%2??4<1.%6N%????4Q$$#N#N%8??5Q$$#N#NXH??630
-    ".%5N%E??6Q$$#N#N%???9+|!#AN$???;+33#AN$e??;<$$#AN".
  02 FILLER PIC X VALUE LOW-VALUES.
*-----------------------------------------
 01  VARIABLES-PANT001.
     03 VAR-ALPHA-PANT001.
        05 NOMBRE-PANT001-A  PIC X(01) VALUE "5".
        05 NOMBRE-PANT001-S  PIC X(30).
        05 DESC-PANT001-A  PIC X(01) VALUE "5".
        05 DESC-PANT001-S  PIC X(30).
        05 PLACAS-PANT001-A  PIC X(01) VALUE "5".
        05 PLACAS-PANT001-S  PIC X(6).
        05 MENSAJE-PANT001-A  PIC X(01) VALUE "5".
        05 MENSAJE-PANT001-S  PIC X(70).
        05 LET-CORRECTO-PANT001-A  PIC X(01) VALUE "5".
        05 LET-CORRECTO-PANT001-S  PIC X(16).
        05 CORRECTO-PANT001-A  PIC X(01) VALUE "5".
        05 CORRECTO-PANT001-S  PIC X(1).
     03 VAR-NUM-PANT001 SIGN IS TRAILING SEPARATE.
        05 POLIZA-PANT001-A  PIC X(01) VALUE "5".
        05 POLIZA-PANT001-S  PIC S9(10).
        05 FECHI-PANT001-A  PIC X(01) VALUE "5".
        05 FECHI-PANT001-S  PIC S9(6).
        05 FECHT-PANT001-A  PIC X(01) VALUE "5".
        05 FECHT-PANT001-S  PIC S9(6).
        05 CLAVE-PANT001-A  PIC X(01) VALUE "5".
        05 CLAVE-PANT001-S  PIC S9(4).
        05 MODELO-PANT001-A  PIC X(01) VALUE "5".
        05 MODELO-PANT001-S  PIC S9(2).
        05 SUMA-PANT001-A  PIC X(01) VALUE "5".
        05 SUMA-PANT001-S  PIC S9(9)V9(2).
        05 DEDUCIBLES-PANT001-INI      VALUE ALL "1".
           07 DEDUCIBLES-PANT001-VEC   OCCURS 3.
              09 DEDUCIBLES-PANT001-A  PIC X(01).
              09 DEDUCIBLES-PANT001-S  PIC S9(1).
        05 PRIMA-PANT001-A  PIC X(01) VALUE "5".
        05 PRIMA-PANT001-S  PIC S9(7)V9(2).
     03 FILLER PIC X VALUE LOW-VALUES.
*****************************************
```

Los reportes generados son los que se muestran a continuación :

```
123456789A123456789B123456789C123456789D123456789E123456789F123456789G123456789H
 1
 2
 3                          PANTALLA DE PRUEBA
 4
 5                       ALTA DE POLIZA AUTOMOVILES
 6
 7
 8
 9       NUM. POLIZA : _________     VIGENCIA DE : _______ A : _______
10
11       PROPIETARIO : ____________________________
12
13       CLAVE VEHIC.: ___ ____________________________  MODELO : __
14
15       PLACAS : ______                DEDUCIBLES
16
17       SUMA ASEGURADA : _____________     _ %  RESPONSABILIDAD CIVIL
18                                          _ %  DAÑOS
19       PRIMA : _____________              _ %  ROBO TOTAL
20
21
22
23
24
```

VARIABLE	TIPO	REN	COL	LON C.	LON L.	OCU.	MASCARA	SIGNO	DEC.
NOMBRE	ALFA	11	22	30	30	1			
DESC	ALFA	13	27	30	30	1			
PLACAS	ALFA	15	17	6	6	1			
MENSAJE	ALFA	22	8	70	70	1			
LET	ALFA	24	8	16	16	1			
CORRECTO	ALFA	24	25	1	1	1			
POLIZA	ENT.	9	22	10	10	1	Supresion de ceros	Sin signo	
FECHI	FECH	9	51	6	8	1	DD/MM/YY		
FECHT	FECH	9	64	6	8	1	DD/MM/YY		
CLAVE	ENT.	13	22	4	4	1	Sin edicion	Sin signo	
MODELO	ENT.	13	69	2	2	1	Sin edicion	Sin signo	
SUMA	REAL	17	25	11	14	1	Proteccion de asteriscos con comas	Sin signo	2
DEDUCIBLES	ENT.	17	46	1	1	3	Sin edicion	Sin signo	
PRIMA	REAL	19	16	9	13	1	Signo de pesos , flotante y con comas	Sin signo	2

Para manejar la forma anterior en el programa cobol se presentarán los párrafos que son requeridos tomando en cuenta la parte del programa que modifica los datos.

- Para inicializar los campos de la pantalla con los datos a modificar :

```
MOVE NOMBRE-TITULAR TO NOMBRE-PANT001-S
MOVE DESC-VEHIC     TO DESC-PANT001-S
MOVE PLACAS         TO PLACAS-PANT001-S
MOVE SPACES         TO MENSAJE-PANT001-S
MOVE NUM-POLIZA     TO POLIZA-PANT001-S
MOVE FECH-INI-VIG   TO FECHI-PANT001-S
MOVE FECH-FIN-VIG   TO FECHT-PANT001-S
MOVE CLAVE-VEHIC    TO CLAVE-PANT001-S
MOVE MODELO         TO MODELO-PANT001-S
MOVE SUMA-ASEG      TO SUMA-PANT001-S
MOVE PRIMA          TO PRIMA-PANT001-S
MOVE DEDUCIBLES (1) TO DEDUCIBLES-PANT001-S (1)
MOVE DEDUCIBLES (2) TO DEDUCIBLES-PANT001-S (2)
MOVE DEDUCIBLES (3) TO DEDUCIBLES-PANT001-S (3)
MOVE SPACES         TO LET-CORRECTO-PANT001-S
MOVE SPACES         TO CORRECTO-PANT001-S
```

- Para que los campos presenten los atributos requeridos :

```
ATR-MODIFICA.
        MOVE "3" TO NOMBRE-PANT001-A
        MOVE "3" TO DESC-PANT001-A
        MOVE "3" TO PLACAS-PANT001-A
        MOVE "3" TO MENSAJE-PANT001-A
        MOVE "D" TO POLIZA-PANT001-A
        MOVE "3" TO FECHI-PANT001-A
        MOVE "3" TO FECHT-PANT001-A
        MOVE "3" TO CLAVE-PANT001-A
        MOVE "3" TO MODELO-PANT001-A
        MOVE "3" TO SUMA-PANT001-A
        MOVE "3" TO PRIMA-PANT001-A
        MOVE "3" TO DEDUCIBLES-PANT001-A (1)
        MOVE "3" TO DEDUCIBLES-PANT001-A (2)
        MOVE "3" TO DEDUCIBLES-PANT001-A (3)
        MOVE "A" TO LET-CORRECTO-PANT001-A
        MOVE "A" TO CORRECTO-PANT001-A .
```

- Para llamar al programa manejador

```
MOVE 99 TO PF-KEY.
        PERFORM UNTIL (PF-KEY = 00 OR
                       PF-KEY = 13 OR PF-KEY = 32)
          CALL "/usr/easemap/screen" USING PF-KEY
                                           SCREEN-PANT001
                                           VARIABLES-PANT001
        END-PERFORM.
```

VI.-CONCLUSIONES

El sistema que se desarrolló durante el seminario cumplió con el 100 % de los requerimientos planteados en el capitulo 2. El sistema se desarrolló buscando la portabilidad, la cual es una de las caracteristicas de UNIX, ésta se logró exitosamente. A continuación se explican las causas por las cuales se afirma lo anterior.

a) Funcionó perfectamentes sin realizar modificaciones, en equipos con procesador INTEL 286, 386 y 486 con multiprocesador y con los sistemas operativos SCO XENIX V 2.3.2, SCO UNIX versiones 3.2.3 y 3.2.4, UNIX VR4 MP.

b) Funcionó realizando sólo algunas modificaciones en un equipo HP-9000 847 con sistema operativo HP-UX 8.02, esto se realizó haciendo uso de la librería de CURSES.

Otro ejemplo de la portabilidad logrado en base a la modularidad del sistema, es la posibilidad de utilizar distintas librerías de funciones "primitivas" para el manejo de terminales. En los sistemas operativos mencionados en el párrafo anterior inciso a), se utilizó TERMCAP. Para el inciso b), se utilizó la libreria de CURSES que ofrece ventajas sobre la de TERMCAP, ésta última no está soportada en HP-UX, por lo que fue necesario utilizar CURSES basada en la base de datos TERMINFO que hace que el desplegado en la pantalla sea más rápido ya que esta compuesta de archivos compilados.

El sistema está orientado a ser una herramienta útil para facilitar el desarrollo de sistemas en el lenguaje de programación RMCOBOL 85, aunque en la actualidad los sistemas se estan orientando al uso de bases de datos con lenguajes de cuarta generación y SQL, esto no implica que COBOL vaya a desaparecer ya que existe una gran plataforma de desarrollo y se continúan realizando nuevos proyectos con este lenguaje.

Es importante destacar que la experiencia de haber utilizado la metodología de análisis y diseño estructurado permite obtener sistemas de mejor calidad y que hacen que el mantenimiento de estos sea más sencillo. Cuando se trabaja en equipo, como el caso del desarrollo de éste sistema, se permite una mejor coordinación entre los integrantes, ya que se evitan fallas en las interfaces de los módulos del proyecto y se facilita la definición de estándares

GLOSARIO DE TERMINOS.

ACOPLAMIENTO	Medida de interdependencia entre módulos.
ASCII	American Standar Code Information Interchange.
BACKGROUND	Proceso que corre de manera independiente de la sesión que se encuentra en la terminal
CALL	Instrucción de COBOLque transfiere el control a otro programa (de COBOL o de otro tipo).
COBOL	COmmon Business Oriented Language.
COHESION	Medida de asociación de elementos de un módulo.
CONEXION	Cualquier referencia de un módulo a cualquier otra cosa definida en otro módulo. Es representado por un vector que une dos módulos.
COPY	Instrucción que incorpora texto de una libreria fuente de COBOL a un programa fuente de COBOL.
COUPLE	Dato que se mueve de un módulo a otro. Es representado por una flecha pequeña con un círculo en la parte trasera.
CROMIX	Sistema Operativo propietario de Cromemco.
CURSES	Librería de UNIX de manejo de pantallas la cual puede incluirse en cualquier programa en C.
DESCRIPCION DE PRIMITIVOS	Definición de procesos a través de comandos y acciones.
DIAGRAMAS DE ESTRUCTURAS	Representación jerárquica del sistema a través de módulos, conexiones y couples.
DIAGRAMA DE FLUJO DE DATOS	Es también llamado DFD, es una representación gráfica de los flujos datos que viajan y se transforman a través de burbujas de procesos.
DICCIONARIO DE DATOS	Conjunto de definiciones de flujo de datos, de procesos y de archivos.

/ETC/TERMCAP	Archivo del Sistema Operativo UNIX en el cual se encuentran las definiciones de varios tipos de terminales
FILLER	Nombre de dato genérico,muy utilizado para la descripción de formatos de registro de informes.
LOGIN	Indicación de aceptación de clave de usuario.
MODULO	Caja rectangular con un nombre dentro. Básicamente representa un conjunto de comandos.
RDCHK	Instrucción de UNIX que checa si hay datos que leer.
SQL	Structured Query Language.(Lenguaje Estructurado de Consulta)
TERMINFO	Base de datos en la cual se describen las características de funcionamiento de las terminales, dichas características describen como se realiza su operación
TERMCAP	Libreria para el manejo de terminales. Las definiciones de las secuencias de escape utilizadas por esta libreria se encuentran en el archivo /etc/termcap
VMIN	Número mínimo de caracteres que se pueden recibir.
VTIME	Tiempo máximo de espera para reconocer que un caracter forma parte de un conjunto de ellos.
UNIX	Sistema Operativo Multiusuario creado por Brian Ken Thompson y Dennis M. Ritchie.
VALIDAR	Revisar que en la información capturada no exista ninguna incongruencia en claves o datos.

APENDICES

APENDICE A.

MANUAL DE USUARIO

ASEMEX MAP

Para entrar al diseñador de pantallas ASEMAP deberá teclearse el siguie..te comando :

asemap

con lo cual aparecerá la siguiente pantalla :

ASEGURADORA MEXICANA, S.A.

ASEMEX MAP V1.0

SOPORTE TECNICO

Para continuar deberá oprimirse cualquier tecla , después la pantalla de la terminal se presentará limpia para que el diseñador pueda moverse libremente por toda la pantalla , teclear las constantes (titulos) que se desea contenga el diseño , definir las variables requeridas ,etc.

GENERACION DE PANTALLAS .

Después de que el diseñador ha dibujado y salvado su formato de pantalla . **ASEMEX MAP** generará el código **COBOL** el cual puede ser incluido mediante la instrucción **COPY** , en el programa de aplicación.

Opciones del menú de **ARCHIVO**

Cargar

Esta opción permite cargar un formato de pantalla previamente diseñado para cualquier modificación que se requiera. Al seleccionarla se presentará la ventana mostrada abajo ,la cual pedirá el nombre del archivo de la pantalla, después de hacer esto,se "pintará" la pantalla en la terminal para su actualización.

Salvar

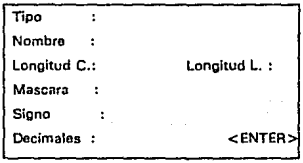
Al seleccionar esta opción , se guardará la pantalla tal como se encuentre en ese momento con el nombre del archivo que se haya asignado , en caso de ser un nuevo diseño al cual aún no se le ha asignado un nombre, se presentará la ventana mostrada en el punto anterior para guardar el diseño con el nombre deseado.

Salvar como .....

Con esta opción , se guardará el diseño de la pantalla tal como se encuentre en ese momento , con un nombre distinto de archivo de pantalla al que se tenía asignado , esto implica que en el archivo original no se actualicen los cambios generados. Para dar dicho nombre se presentará la ventana mencionada en los puntos anteriores.

Descripción

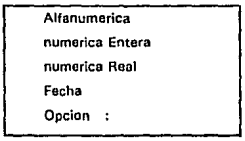
Al seleccionar esta opción se presentará una ventana con la descripción de la variable que se encuentra donde esté el cursor al momento de entrar al menú , tal ventana es como la mostrada :



Dependiendo del tipo de variable , aparecerán los datos de máscara , signo y decimales. Para continuar en el diseño se deberá presionar la tecla de **ENTER**.

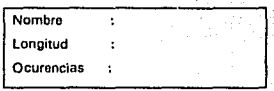
Definir

Con esta opción se definen las variables que se requieran en el diseño , al entrar se presentará la ventana mostrada abajo en la cual se elige el tipo de la variable , esto se logra presionando la tecla de la letra que aparezca en mayúscula .

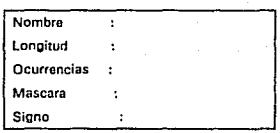


Presionando la tecla A se definirá una variable **ALFANUMERICA** y se presentará otra

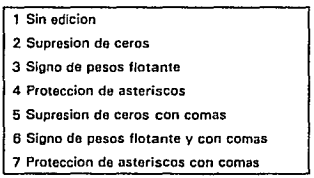
ventana que pedirá en el orden mostrado los datos de ésta.



Presionando la tecla E se definirá una variable numérica ENTERA y se presentará una ventana en la cual se deberán dar los datos de dicha variable.



Cuando se tenga que dar el tipo de máscara , se presentará la siguiente ventana para seleccionarlo.



Después de asignar el tipo de máscara se deberá dar el tipo de signo para esto se borrará la

ventana de ayuda anterior y se presentará la siguiente:

1 Sin signo
2 Signo menos a la izquierda
3 Signo menos a la derecha
4 CR a la derecha

Presionando la tecla **R** se definirá una variable numérica **REAL** y se presentará una ventana en la que se deberán dar los datos de dicha variable.

Nombre	:
Longitud	:
Ocurrencias	:
Mascara	:
Signo	:
Decimales	:

Cuando se asigne el tipo de máscara y el de signo se presentarán las ventanas respectivas de ayuda mencionadas para el caso de las variables numéricas enteras .

Presionando la tecla **F** se definirá una variable de tipo **FECHA** y se presentará otra ventana que pedirá en el orden siguiente los datos de la misma:

Nombre	:
Longitud	:
Ocurrencias	:
Mascara	:

Cuando se de el tipo de máscara , se presentará la siguiente ventana de ayuda para seleccionarlo:

```
1 DD\MM\YY
2 MM\DD\YY
3 YY\MM\DD
```

Borrar

Al seleccionar esta opción se podrá borrar alguna variable previamente definida, esto se hace posicionado el cursor sobre la variable que se quiere eliminar , después se presentará la ventana mostrada abajo para confirmar esta operación.

```
Nombre de la variable        Estas seguro?
```

Al confirmar esto , se borrará de la pantalla la variable. definida.

Posición

Al entrar a esta opción se presentará la siguiente ventana , indicando la posición del cursor al momento de entrar a ella:

```
Renglon: YY   Columna: XX
```

Para continuar deberá presionarse la tecla **ENTER**.

Opciones del menú de **SALIR**

Salir

Al escoger esta opción se salvará el diseño tal como esté en ese momento , si el diseño es nuevo , se presentará la ventana mostrada al principio pidiendo el nombre con el que se desea salvar lo hecho hasta este momento . Después se terminará la sesión de **asemex map** .

Abandonar

Con esta opción se presentará una ventana para confirmar esta operación ya que con ésta se guarda el diseño tal como estaba cuando se cargó. Si es un diseño nuevo , no se guardará lo hecho hasta este momento. Después se terminará la sesión de **asemex map**.

APENDICE B.

UNIX.

El sistema UNIX fue diseñado por un grupo de personas de AT&T influenciados por el Sistema Operativo MULTICS, que fue desarrollado en MIT a finales de los 60's. Debido a que era uno de los primeros sistemas de tiempo compartido MULTICS poseía la gran mayoría de las ideas de los sistemas multitarea actuales. Por ser tan innovador MULTICS resultó más complejo de lo que era necesario, por estas razones la AT&T abandonó la mayor parte de su participación en el proyecto MULTICS. Al dejar abandonado el proyecto, las personas que trabajaban en él se quedaron sin un sistema operativo con el cual trabajar y crearon uno nuevo.

Ken Thompson y Dennis Ritchie (los diseñadores originales) apoyados por Rudd Canaday, J.F. Ossana y R Morris consiguieron una máquina DEC PDP-7 de desecho y comenzaron a trabajar. Crearon la estructura del sistema de archivos muy similar a la estructura actual, realizaron varios procesos y completaron su sistema operativo. El nombre UNIX deriva de que su sistema fue una simplificación del sistema MULTICS.

La codificación original fue hecha en ensamblador, pero poco tiempo después se desarrolló el lenguaje de programación C. Este lenguaje fue utilizado en la continuación del desarrollo del sistema UNIX y en 1973 el Kernel (núcleo) se recodificó en C, en la actualidad sólo unas cuantas subrutinas están programadas en ensamblador. La portabilidad que se consiguió al codificar casi en su totalidad el sistema operativo en un lenguaje de alto nivel es considerada como una de las razones principales de la popularidad que tiene UNIX.

La AT&T dió copias del sistema UNIX a varias universidades en el mundo entero lo que ocasionó que una generación completa de gente relacionada con la informática aprendiera su profesión con el sistema UNIX. Por estas circunstancias apareció la implementación e innovación BSD (Berkeley Software Distribution) en la Universidad de California en Berkeley. Mientras la

AT&T optimizaba su sitema para actividades comerciales, las versiones de BSD eran dirigidas a las comunidades universitarias y técnicas.

A finales de los 70's y principios de los 80's una sola o ambas de las versiones BSD y AT&T habían sido portadas a todas las computadoras capaces de soportarlas. Para esto, las computadoras debían poseer unidades de disco de alta velocidad. En la actualidad existen versiones de UNIX tanto para las grandes computadoras como para casi todas las máquinas pequeñas a la venta.

Con el desarrollo de la velocidad y potencia de las computadoras y su disminución de precio, el UNIX ha podido llegar a ser parte de ellas. Las originales PC's con sus procesadores 8088 eran lo suficientemente potentes para soportarlo y algunas implementaciones podían ejecutarse en ellas.

El sistema operativo XENIX es una versión del UNIX menos completa para PC's que sin embargo sólo ha tomado forma en AT's 80286 y 80386.

En resumen, el sistema UNIX desarrollado originalmente por expertos para su propio uso ignoró las necesidades de los novatos en favor de la velocidad y precisión, pero sus ventajas hicieron que pasara rápidamente a los nuevos usuarios. Con las mejoras hechas a cada una de las nuevas versiones, el sistema UNIX cuenta cada vez más con mejores características como son:

Robustez: Esto significa que requiere muy poco mantenimiento software para que funcione con el máximo rendimiento. Muchas tareas que anteriormente eran realizadas por el administrador ahora se realizan automáticamente .

Administración del Sistema: Ahora la mayor parte de las implementaciones tienen herramientas sencillas que proporcionan una gran ayuda en la configuración y administración del sistema.

Documentación: El manual de usuario además de ser material de referencia conciso ahora es más explícito que en las versiones anteriores.

Consistencia: Muchas órdenes son ahora más consistentes de modo que ahora la confusión

ha disminuído.

Nuevas características: Se han introducido nuevas órdenes que incorporan ideas de BSD (Berkeley Software Development) y XENIX. También se han mejorado algunos subsistemas como el uucp (de transferencia de archivos) y nuevas características en el mecanismo de flujos (streams) para soportar la red.

Compartición del sistema operativo: Las nuevas computadoras permiten que puedan coexistir el UNIX y el MS-DOS. Esta característica significa que el sistema UNIX puede ejecutar el MS-DOS como un proceso bajo su control permitiendo que las funciones de background de UNIX se ejecuten al mismo tiempo que la máquina es utilizada por programas de MS-DOS. Esta característica está restringida en la actualidad al UNIX de las máquinas Intel 80286 y 80386.

Debido principalmente a que el sistema anterior (EASEMAP) venía funcionando bajo el Sistema Operativo XENIX y que tal Sistema Operativo (como se mencionó anteriormente) es una versión recortada del Sistema UNIX además de todas las ventajas que ofrece este Sistema Operativo, se decidió realizar el nuevo sistema (ASEMAP) en este Ambiente.

Una de las principales características con las que debía cumplir nuestro sistema era la **portabilidad**, la cual junto con la **interconectividad** son los términos asociados a los sistemas abiertos. A continuación se definen estas dos términos.

Portabilidad: Denota la capacidad para utilizar el Sistema Operativo o Software de aplicación en una amplia gama de equipos de cómputo diseñados por diferentes fabricantes. En el caso de las aplicaciones de Software éste término indica la capacidad de diseñar aplicaciones que puedan ser ejecutadas en equipos de cómputo fabricados por diferentes proveedores, basada en una **Interfase de Programación de Aplicaciones** común.

Interconectividad: También conocida como redes de computadoras indica la capacidad que poseen las computadoras de diferentes fabricantes para intercambiar información entre ellas.

En este aspecto existe una organización de estándares más directamente involucrada por la interoperabilidad o interconectividad, ella es la International Standar Organization (ISO) a través de su modelo de referencia Open System Interconnection (OSI) reference Model.

BIBLIOGRAFIA

RM/COBOL-85 LANGUAGE REFERENCE MANUAL

EU: LIANT, 1987

RM/COBOL-85 INSTALLATION & USER'S GUIDE

EU: LIANT, 1987

De Marco Tom, *Structured Analysis and System Specification.* EU: Prentice Hall, 1979.

Yourdon, Edward and L.Constantine, *Structured Design.* EU: Prentice Hall,1979.

Crane,Mark, *Word for Windows.* EU: Microsoft Press, 1990.

SCO SCO XENIX V / 386, System Administration's Guide. EU: The Santa Cruz Operation, Inc, 1988.

SCO SCO XENIX V, SYSTEM V / 386 Development System Porgrammer's Reference EU: The Santa Cruz Operation, Inc, 1988.

HP-UX Reference Volume 2, C Programming Routines HEWLETT PACKARD, 1991

HP-UX Reference Volume 3, Administration and Other Topics HEWLETT PACKARD, 1991

SYSTEM ARCHITECT, User Guide. EU: Popkin Software & Systems Incorporated, 1989.

SYSTEM ARCHITECT, Tutorial. EU: Popkin Software & Systems Incorporated, 1989.