

Universidad Nacional Autónoma de México

Facultad de Ingeniería



Sistema Generador de Prototipos bajo Tecnología CASE
GPCASE

Tesis Profesional
Que para obtener el título de
Ingeniero en Computación

PRESENTAN:

Tomás
Martín Torices Jiménez
Santiago Rubio Solís
Victor Manuel Ortiz Soriano

FALLA DE ORIGEN

Director de Tesis : Ing. Sergio Noble Camargo

Ciudad Universitaria, México D.F. Julio 1994



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

RECONOCIMIENTOS

Agradecemos al Ing. Sergio Noble sus consejos, enseñanzas, el tiempo que dedicó para asesorarnos, además de su profesionalismo y amistad en la obtención de este trabajo.

Al Dr Segio Beltrán, por las oportunidades ofrecidas , el apoyo para la realización de esta tesfs y por su valiosa amistad

Al Ing.sergio Ambriz por su apoyo y amistad.

Agradecemos a la Facultad de Ingeniería por permitirnos obtener una carrera.

Agradecemos en general a todos aquéllos profesores que con sus enseñanzas, paciencia y consejos lograron motivarnos para lograr el objetivo que nos trazamos.

Dedicatorias

La realización de este trabajo fue hecho con el mejor de nuestros esfuerzos, para que de esta forma sirviera como una muestra de nuestro agradecimiento a nuestros seres queridos:

A mis padres:

Felipe y Teresa, por que los quiero mucho.

A mis hermanos:

Yolanda, Santos †, Elvira, Ana, Juan, Orlando; Amigos de toda la vida.

A mis casi hermanos; mis amigos; mis cuñados:

Juan Carlos y Vicente.

A mis inquietos sobrinos:

David, Israel y Beto.

A mis compañeros de Tesis y mejores amigos:

Víctor y Santiago.

A mi director de Tesis:

Sergio Noble Camargo.

A Todos mis amigos de los cuales omito sus nombres para evitar que se me olvide alguno, ellos ya saben quienes son y se que me comprenderan.

A mi novia Magoy.

A la facultad de Ingeniería.

Sinceramente

Martín Torices Jiménez.

DEDICATORIAS

A mis papás:

De todo corazón les dedico este trabajo. Este es el resultado de un esfuerzo conjunto que hemos realizado juntos año tras año y el cual ahora ha tomado forma gracias a sus oportunos consejos, su decisivo apoyo y sobre todo su amor y cariño.

Gracias por formarme el hábito de la disciplina, alimentar mi fuerza de voluntad, a ser siempre justo, y motivarme en nuevas empresas, en fin tantas cosas que me han permitido llegar a esto y más.

A Mamá:

Por toda esa calma que mantiene aún en momentos de apremio, por esa energía y fuerza de voluntad que posee y que siempre me irradia para seguir adelante en todos mis objetivos.

A Papá:

Por su integridad, su excelente sentido del orden y por su incansable búsqueda de nuestro bienestar que me han permitido desenvolverme en todos los terrenos.

Los quiere mucho:

Víctor Manuel

DEDICATORIAS

A mis hermanos:

Por tantas cosas que hemos pasado juntos y que no importa que situación prive, siempre hemos estado unidos para resolver las contingencias.

A ambos les agradezco las desveladas que a veces pasamos juntos para ayudarme en algunos proyectos. Gracias.

A José Manuel:

Por sus impredecibles y agradables ocurrencias, su carácter para las actividades y discusiones, pero sobre todo por brindarme su apoyo en los momentos más pesados de la carrera.

A Juan Manuel:

Por su sentido práctico, su plática amena, su buen humor y por estar siempre dispuesto a poner en práctica nuevas actividades juntos.

Gracias por soportar mi carácter:

Víctor Manuel

DEDICATORIAS

A Lucy:

Con mucho cariño para la chica más agradable e importante de mi vida y que es la motivación de mi ser.

Por comprender mi forma de pensar, por su paciencia para escucharme y darme ideas, por brindarme tantos momentos agradables, por estar conmigo cuando lo necesito, por impulsarme siempre a superarme y muy en especial por darme ánimos constantes durante la realización de este trabajo.

Gracias por ser como eres:

Víctor Manuel

DEDICATORIAS

A mis Amigos: Por apoyarme en las buenas y en las malas, que respetan mi forma de pensar y que siempre me han demostrado su amistad.

A Nacho: A mi gran amigo, por el gusto de haber recibido juntos satisfacciones personales, escolares y laborales.

A Lupita: Por su espontaneidad, por transmitirme su alegría natural, por entender mi forma de pensar y por apoyar a los seres importantes para mí.

A Rodolfo y Chente: Con afecto a mis grandes cuates, por tantos momentos agradables que convivimos.

A Laura: Por su sencillez, por su franqueza para hacerme ver mis errores, por su gran compañerismo y por su apoyo.

A Luz María: Por su objetividad para analizar las cosas, por hacerme reflexionar otras ideas y por animarme a seguir adelante.

A Juan Bosco y Claudia: Por acordarse de mí en los momentos importantes y en especial a **Juan** por ser una de las personas que más contribuyó a mi superación como profesionista.

A Mario Chacón: Por incrementar mi espíritu de lucha, por ayudarme y darme consejos de gran utilidad en el aspecto personal y profesional.

A Margarita Balcázar, Juan Navarro, Roel, Yazmín y Héctor por su valiosa amistad y su apoyo.

A Olga, Elizabeth, Carlos, Miguel Angel, Martha, Vicente, Armando, Roberto y toda la flota, por ser un grupo de amigos que cualquiera envidiaría tener, gracias por siempre.

Gracias por su sinceridad y verdadero apoyo:

Víctor Manuel

AGRADECIMIENTOS

A mi madre, la que me dio la vida , quien me enseñó las primeras lecciones de la vida y de quien he tomado los valores que me hacen diferente de cada persona .

A mi padre que es un ejemplo para mí, y mi mejor amigo, al cual admiro por ser como es y del cual he aprendido muchas de las cosas que me caracterizan como individuo.

A ambos por darme todo, por ayudarme a nacer por segunda vez, por lograr juntos la realización de esta meta, por su fuerza para poder continuar adelante en los momentos más difíciles sin tratar de demostrar sus preocupaciones, por su espíritu incansable, por su amor y cariño, por todo eso y mucho más, gracias

A mi hermana Ana por el constante apoyo y cariño que ha tenido no solo conmigo, sino con toda la Familia. Gracias a ti puedo lograr esto y mucho más.

A Elvira y Juan Carlos por su ayuda y consejos

A Yolanda y Vicente.

A Santos † por los recuerdos de la infancia.

A Juan con el cual quiero compartir un logro más, ya que gracias a él puedo hacerlo, y al cual quiero ayudar a encontrar su camino .

A Orlando con el que quiero compartir un anhelo que influya en su vida personal .

A mis Sobrinos :

A David :

Por ser un niño muy inquieto, el que busca el por qué de las cosas, el que se interesa por lo que uno hace, el que busca su camino viendo la vida de sus mayores.

A Israel :

Que es un niño que tiene muchos valores, muchos de los cuales iguales a los de mi madre, el que es como el hijo de mis padres y hermano pequeño de nosotros.

A Beto :

El niño más pequeño de la familia y el más inquieto, a el por regalarme siempre una sonrisa.

A mis Amigos :

A Chuchín por tenderme siempre una mano , por ayudarme y aconsejarme en cualquier momento.

A Victor y Santiago por aportar una parte importante de sus vidas a la mía, por apoyarme en las metas que nos hemos fijado, por entenderme como soy, por compartir los buenos momentos, por ser solo eso, unos grandes amigos .

Quiero agradecer también a todos mis amigos que con su afecto, amistad y cariño lograran que los momentos cotidianos de la vida se convirtieran en los mejores de una etapa de la misma. Quisiera mencionar sus nombres, pero como tú² sabes , no quisiera olvidar ninguno, pero yo confío que con tu ayuda sepas comprenderme , ya que tú lo sabes más que yo, regularmente no acostumbro a jugar con palabras, pero yo se que mis amigos, las comprenderán.

Sinceramente

Martín Torices Jiménez .

ebee/tamm.

² vale por dos .

Agradecimientos

A mis padres por sus consejos, por su tiempo que me han dedicado, por sus regalos, porque me han ubicado en el mejor camino, por dirigir mis primeros pasos, por educarme, por su amistad, por su confianza en que esto era posible, y por otras tantas cosas, por la vida y porque los quiero mucho. Muchas Gracias.

A Juan por el ánimo y espíritu para lograr siempre lo que se propone, porque con ese espíritu levanta el ánimo de cualquiera y por que se que esperaba que yo lograré esta meta Gracias carnal.

A todos mis hermanos por todas las sonrisas que me han brindado, por sus enojos, ya que gracias a ellos he comprendido tantas cosas que con palabras no se pueden explicar; por saber ser los mejores amigos y por comprenderme Gracias.

A Martín por su amistad por su forma de ser por que he aprendido otras formas de pensar, he aprendido otra forma de ver la vida y he comprendido que no a todos nos va mal ni tampoco bien, por escucharme cuando necesite hablar, por comprenderme y ayudarme siempre y mucho más Gracias.

A Víctor por su comprensión, por brindarme su amistad, por haberme apoyado cuando fue necesario por esos días de escuela (esas desveladas), y los días de juego también porque la risa se contagia y es parte de la conversación Gracias.

A mis primas por el afecto y la confianza y Muy en especial a Elena por tomarme siempre en cuenta, por su amistad y por saber que cuento con ella Gracias.

A Alicia porque siempre tiene una sonrisa para mi, por el esfuerzo al ayudarme, por su comprensión, por su confianza, por los momentos felices que me ha brindado y por muchas otras cosas más Gracias.

A mi tío "Chindo" por estar siempre con nosotros porque siempre nos ha brindado su comprensión y apoyo Gracias.

A "Purungo" que con su amistad que me brindo durante todo este tiempo que pasamos juntos en la universidad he aprendido que aún existen amigos que dan todo por la amistad, porque siempre me apoyo cuando a mi me faltó tiempo y aprovecho para agradecer también la amistad de su hermano Hugo y de toda su familia que aunque lejos se que me consideran un amigo Gracias.

Al Lic. Carlos González por el tiempo que me permitió dedicarle a la tesis y por el gran apoyo y flexibilidad, por su amistad y por comprenderme Gracias.

A Todos los compañeros de la Oficina por su amistad y apoyo, por esos momentos tan agradables que me hacen pasar y por los desagradables también porque a final de cuentas los considero una familia y en una familia hay de todo Gracias.

Sinceramente Santiago Rubio Solís

AGRADECIMIENTOS

A Santiago:

Por manifestarme siempre su amistad, su buen humor en todos nuestros proyectos, convivencias y diversiones, por su profesionalismo, pero sobre todo por entender mi forma de pensar. Gracias.

A Martín:

Por su profesionalismo, por su gran sentido del trabajo en equipo y por hacerme reflexionar sobre diversos puntos de vista, pero sobre todo por haber contribuido a mi superación como profesionista, Gracias.

Al Ing. Sergio Noble:

Por haber encauzado esta idea, asesorarnos adecuadamente en este proyecto, por brindarnos parte de su tiempo libre para ayudarnos a salir adelante, Gracias.

Al Ing. Alejandro Jiménez Hernández

Por haberme enseñado y orientado en un nueva forma de desarrollar proyectos y acelerar y enriquecer mis conocimientos sobre mi carrera, Gracias.

A Elsa, Marysol, Eduardo y Efraín:

Por hacerme llevadero mi trabajo en Motorola, por tenerme paciencia durante todo este año de tanta presión y tensiones personales. Muy en especial a **Eduardo** y **Efraín** por contribuir a mi crecimiento como profesionista, por apoyarme cuando lo necesité para terminar este trabajo y en fin gracias a todos por su profesionalismo y comprensión.

Y a **todas aquellas personas** que contribuyeron a mi superación y facilitaron mi desenvolvimiento como profesionista, Gracias:

Víctor Manuel

Prologo

Introducción

I.- Sistemas de Información y Nuevas Metodologías	CAP. I	
I.1 Sistemas de Información		1
I.2 CASE		5
I.2.1 CASE Ideal		5
I.2.2 Metodología CASE		6
Requerimientos del Análisis		7
Diseño Externo		7
Diseño Interno		7
Codificación		7
Verificación		7
Implementación		8
Mantenimiento		8
Uppercase		9
Lowercase		10
I.2.3 Ventajas y Desventajas de las Herramientas CASE		11
II.- Prototipos	CAP.II	1
II.1 Clasificación de los Prototipos más Comunes		4
Prototipo Throwaway		4
Prototipo Quick & Dhirty		5
Prototipo Detallado Design Driven		5
Prototipo Not Functionating Mock-Ups Modelo Réplica		6
III.- Análisis	CAP.III	
III.1 Estudio Preliminar		1
III.2 Encuesta		2
Conclusiones de la Encuesta		13
Conclusiones Generales de la Población		13
Conclusiones sobre CASE		13
III.3 Propuesta		14
III.3.1 Sistema GPCASE		14
III.3.2 Requerimientos		14

III.3.2.1	Requerimientos Para Operación	14
III.3.2.2	Requerimientos Para el Desarrollo de GPCASE	15
III.3.2.3	Perfil del Usuario de GPCASE	17
Nivel Profesional		17
Nivel de Experiencia		17
Habilidad		17
Visión del Negocio		18
III.4	Elementos Gráficos Para Análisis y Diseño	19
III.5	Diagramas de Flujo de Datos (DFD's)	21
Diagrama General		22
Descripción de los diagramas		23
Diagramas de Procesos		29
Descripción de los Procesos		34
Descripción de Entidades Externas		39

IV.- Planeación

CAP.IV

V.- Diseño

CAP.V

V.1	Notación Húngara	1
V.2	Organización Empleada Para el Desarrollo	4
V.2.1	Interacción Modular de GPCASE (gráfica)	6
V.2.2	Diagrama Estructurado (DE)	7
Diagrama General de Módulos		8
Diagramas de Módulos		9
Descripción de los Módulos		14
V.3	Diagrama Entidad Relación	16
Diagrama		17
Atributos de Entidades de Datos		18
Modelos de Datos		22
Relaciones		23
Contenido de Entidades de Datos		24
Tablas de Bases de Datos del Sistema GPCASE		26

VI.- Manual Del Usuario

CAP.VI 1

VI.1	Menú Principal	3
VI.1.1	Proyecto	5
VI.1.2	Menús	7
VI.1.3	Diseñador	12
VI.1.4	Variables	17
VI.1.5	Proceso	20
VI.1.6	Ayuda	25

VI.2 Diseñador de Menú	27
VI.3 Área de Diseño de Módulos Pantalla	27
VI.4 Barra de herramientas	28
VI.5 ToolBox	33
VI.6 Línea de Mensaje de Ayuda	37
VI.7 Línea de Mensajes de proceso	37

VII.- Mapa de Menús

CAP.VII

VIII.- Manual de Referencia Técnica

CAP.VIII

VIII.1 Referencia Técnica de GPCASE	1
VIII.2 Descripción del Archivo GPDEF.H	5
VIII.3 Descripción del Archivo GPCASE.H	17

IX.- Ejemplo

CAP.IX

IX. Ejemplo Realizado en GPCASE	1
Creación de un Nuevo Proyecto	2
Creación del Menú	3
Definición de Conexiones	4
Creación de Variables	5
Diseño de Pantallas y Reportes	7
Al terminar el Prototipo	13

Conclusiones

Glosario

Bibliografía

PROLOGO

La idea de desarrollar GPCASE surge como respuesta a una inquietud profesional. En el área de la ingeniería de software, al desarrollar sistemas para diversas plataformas es necesario que el usuario y el desarrollador estén de acuerdo en el objetivo final del sistema, para ello el diseñador se dispondrá a generar una aplicación prototipo la cual someterá a evaluación del usuario.

En la forma tradicional, en la primera entrevista se encuentra el diseñador con que el usuario en ocasiones no se imagina como es un sistema a la medida de sus necesidades y esto forza al diseñador a tener que escribir código al terminar la entrevista, con el fin de presentar al usuario una primera visión de lo que podrá realizar el sistema y como lo hará.

Esto es muy problemático porque si al usuario no le agrada mucho la primera impresión, se requerirá volver a escribir o modificar código para hacer las correcciones de presentación solicitadas por él, antes de que se comprometa a cerrar un contrato o acuerdo con el diseñador para llevar a cabo un proyecto, en el peor de los casos el usuario no valora el esfuerzo realizado y sencillamente se retira, con lo que el diseñador ya generó código, gastó recursos, esfuerzo y tiempo.

Por consiguiente un sistema que facilite este proceso de retroalimentación es sumamente útil, ya que se puede presentar al usuario con el mínimo de esfuerzo una visión global de lo que sería el sistema que satisfaga sus necesidades.

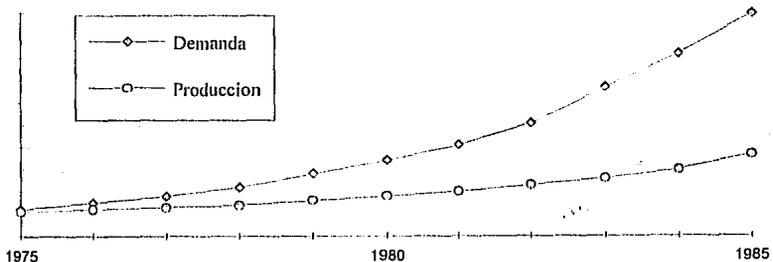
De esta forma se consideraron las principales características que dieron origen a GPCASE.

INTRODUCCION

Uno de los principales problemas que ha presentado la computación en todo el mundo es la crisis del software debido a un importante avance en planeación por medio de proyectos de software a gran escala. Desde los inicios de 1950, las espectaculares mejoras en la tecnología de la producción de hardware han conducido a la fabricación de máquinas cada vez más poderosas, disponibles a precios cada vez más bajos. En cambio la demanda social de software cada día más grande y complejo ha sufrido una atención deficitaria y los costos para obtener este software han crecido aún más rápidamente. Mientras apenas se han logrado algunas mejoras modestas en la calidad y productividad del software, el patrón general de actuación industrial en esta materia sigue siendo el de los calendarios incumplidos, los sobregiros presupuestales y los problemas de calidad en la aplicación de los principios tradicionales de la ingeniería de software, por ejemplo, descuido en el detalle de la administración del proyecto, uso inadecuado de software, etc. El uso de prácticas impropias de programación han contribuido enormemente al desarrollo de dicha crisis. Los viejos estilos de técnicas de programación:

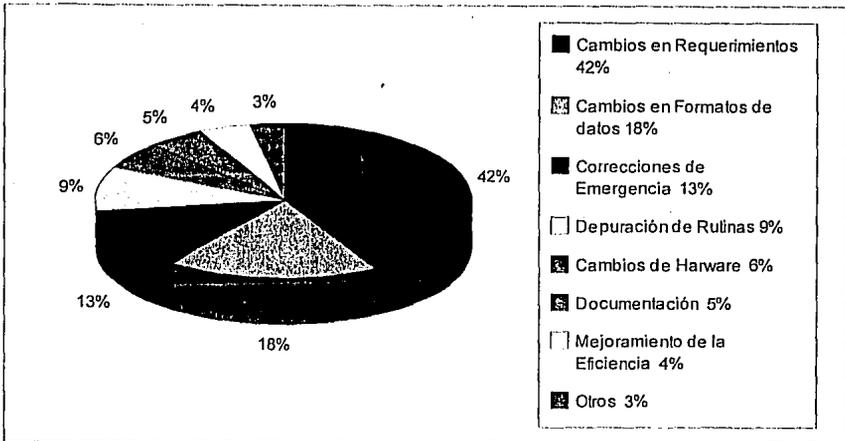
- Uso extendido de variables globales.
- Procedimientos con demasiadas variables locales.
- Descomposición impropia de módulos.

Con relación a los *problemas de productividad*, hacia 1970 se estimaba que la demanda de software, que de 1975 a 1985 crecería entre el 21 y 23% anual, no podría ser atendida por la capacidad de producción, que apenas aumentaría entre el 11.5 y 17% anual



Diversos autores han coincidido en señalar que, en promedio, las instalaciones gastan el 70% de su presupuesto para desarrollo en el mantenimiento de software existente, el cual, para continuar en operación, requiere que el tiempo medio para efectuar una modificación o corregir una falla sea menor que el tiempo medio entre dos solicitudes de cambio o entre dos fallas. Y es, que la probabilidad de que un programador logre corregir una falla al primer intento no es muy alta: si modifica solamente de cinco a diez instrucciones, la probabilidad máxima es de 50%; si modifica entre 40 y 50 instrucciones, la probabilidad de

acertar baja a 20% ¹. Lientz y Swanson ² han establecido la forma como se distribuye el esfuerzo de mantenimiento, tal como se muestra en la siguiente gráfica :



En lo que respecta a los *problemas de costo*, de entrada hay que señalar que en 1985 los costos del software a nivel mundial ya excedían los 140 mil millones de dólares. A partir de entonces han crecido a una tasa del 12% anual, de modo que para 1995 se habrán alcanzado los 435 mil millones de dólares ³. Particularmente, en lo que concierne al costo del software respecto del hardware, es preciso tener presente que en 1955 el costo del primero era menor al 10% del costo del segundo; que 20 años después, el costo del software era ya tres veces mayor que el del hardware y que para 1985, la diferencia se incrementó nueve veces.

El objetivo primordial de la ingeniería de software, fundada en el seno de una conferencia de la OTAN celebrada en Roma en 1968, es responder a los desafíos de productividad y calidad aplicando disciplinas de ingeniería al desarrollo de software. Hasta antes de su creación, la producción de software era practicada como una actividad artesanal. La entonces naciente *Programación Estructurada*, obra del profesor Dijkstra, se constituía como condición de posibilidad de la nueva disciplina de ingeniería .

¹ Boehm, B.W: "Software and its Impact: A quantitative Study", Datamation, mayo 1993.

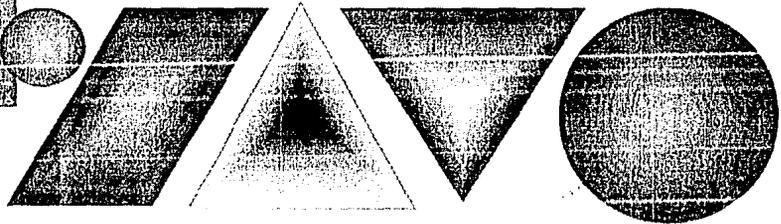
² Lientz, B.P. y E.B. Swanson : "Software Maintenance: A user/Management Tug of War", Data Management, abril de 1979

³Sommerville, I.: Software Engineering, Addison-Wesley, 4a. Edición, 1992

Durante la década de 1970 se realizaron importantes avances, tanto en la industria como en el mundo académico, para caracterizar mejor a los productos de software y establecer una base empresarial para su desarrollo a gran escala. Se refinaron los conceptos de la programación estructurada, se crearon diversas metodologías estructuradas efectivas para el análisis y diseño, entre las que alcanzaron gran popularidad SA/SD (Structured analysis and structured design) de Edward Yourdon y JSD/JSP (Jackson system development / Jackson structured programming) de Michael Jackson, con lo cual se avanzó notablemente en materia de métodos formales.

Durante la década de 1980, continuó el avance en métodos formales, se desarrollaron las métricas de software, las herramientas CASE y el empleo de prototipos y se introdujeron la reingeniería y un nuevo paradigma en la práctica del desarrollo de software: el paradigma de los objetos.

Sistemas de Información y Nuevas Metodologías



1.1 SISTEMAS DE INFORMACIÓN

El creciente avance de la tecnología ha permitido el aprovechamiento de las computadoras para la solución de problemas con las actividades humanas mediante la implementación de sistemas de información, los cuales tienen el propósito de automatizarlas y generar resultados para la toma de decisiones. Sin embargo, debido a que el ser humano demanda mejoras en cada avance que logra, los sistemas de información se vuelven proporcionalmente más complejos, llegando incluso a perderse el control de los mismos.

Para resolver este problema se desarrollaron diferentes metodologías, las cuales tienen como propósito el establecer reglas y normas para el desarrollo de sistemas de información de forma eficiente. Pero por desgracia, no siempre se cumplen en el orden establecido.

Una metodología básica y tradicionalmente empleada es la del seguimiento del ciclo de vida del sistema de información. El cual establece los siguientes puntos:

- ① Identificación del problema .
- ② Determinación de los requerimientos de información.
- ③ Análisis de las necesidades del sistema.
- ④ Diseño del sistema recomendado.
- ⑤ Desarrollo y documentación del Software.
- ⑥ Pruebas .
- ⑦ Instalación y evaluación.
- ⑧ Mantenimiento.

El seguimiento ordenado y conciso de cada una de estas etapas permite generar sistemas de información eficientes y versátiles. Sin embargo, debido a que estas etapas se presentan interrelacionadas y de manera simultánea, es difícil conocer los límites y alcances de cada etapa, dificultando asimismo darle el peso correspondiente a cada una para su correcta ejecución. Por consecuencia gran parte de los responsables de su aplicación, no le dan la importancia suficiente a éstas, ocasionando omisiones de unas y repeticiones de otras y bajo este tenor el sistema desarrollado puede potencialmente presentar defectos que requerirán desde una simple enmienda o la total reconstrucción del sistema, incluso llegando a desechar éste.

Por otro lado, debemos ahora analizar los elementos restantes, es decir, la evolución de las herramientas de Software, el Hardware y los usuarios, para darnos una idea global de los puntos a tratar. Hoy día, la información se ha convertido en todos los ámbitos de la actividad humana en un recurso de alto valor, por lo que existen dos cosas importantes que la caracterizan :

- Su procesamiento y almacenamiento
- Su distribución, entre los miembros de una empresa, comunidad o país.

En el año de 1964 con la introducción de la 360 de IBM, se hizo común el ofrecimiento de computadoras con procesadores más rápidos ya que las necesidades de los usuarios aumentaban cada vez más, provocando el advenimiento de nuevas familias de computadoras, se han ido generando microprocesadores más rápidos con sistemas de almacenamiento de mayor capacidad y de más fácil manejo. Con el incremento de velocidad se ha hecho común categorizar a una tecnología o equipo de computo en particular por el número de instrucciones por segundo (IPS, MIPS, BIPS, MFLOAT, etc.).

Ha sido tanto el desarrollo en las dos últimas décadas, que hemos podido observar la evolución de las grandes computadoras (MainFrames) a las computadoras portátiles, las cuales muchas veces superan las características de las primeras, dentro de la línea de las computadoras de escritorio y portátiles podemos mencionar a la ya muy conocida familia XXX86, Pentium, PowerPc, Machintos y estaciones de trabajo.

La principal idea de crear a una computadora de escritorio (idea propuesta por IBM) era la de independizar con sus propios recursos a una computadora de un Mainframe, la idea fue excelente, solo que IBM cometió un gran error, ver a la PC como un elemento aislado y hacer todos sus sistemas solamente dependientes de un hardware propio del mismo fabricante "sistemas cerrados", lo anterior trajo como consecuencia que muchos fabricantes tuvieran la inquietud de poder comunicar varias máquinas, lo que dio como consecuencia el nacimiento de las redes de datos y la arquitectura cliente servidor para estaciones de trabajo.

A continuación se enuncian algunas de los principales aspectos de los servidores :

- Dentro de los procesadores con tecnología RISC existe en la actualidad una variedad de éstos, dentro de los cuales destacan el SPARC, fabricado por SUN Microsystems y el 88000 fabricado por Motorola .
- Los equipos pueden tener uno o más procesadores trabajando en paralelo .

En términos generales los equipos que actualmente corren con el sistema operativo Unix, tienen procesadores de 32 bits, generalmente basados en tecnología RISC (Reduced Instruction Set of Commands) .

En el campo del Software éste ha sufrido una gran transformación, en sus inicios estaba limitado al lenguaje ensamblador de la máquina a utilizar, que es sumamente tedioso y complicado. Esto forzó al desarrollo de nuevas alternativas de Software para mejorar la interfase usuario-máquina, generándose así primeramente los intérpretes que traducen instrucciones entendibles para el programador a código máquina en tiempo de ejecución; pero era indispensable la presencia del intérprete. Posteriormente a éstos nos encontramos a los compiladores que tienen la particularidad de generar código ejecutable independiente de la presencia del compilador que lo generó. Tanto los intérpretes como los compiladores se desarrollaron para lenguajes de alto nivel a partir de los 60's, pero esto no es suficiente para cubrir las nuevas necesidades que se siguen presentando y finalmente surgen así los

lenguajes de cuarta generación a finales de los 70's y que actualmente tienen gran aceptación.

La tecnología basada en el sistema operativo UNIX, ha venido a cubrir un espacio vacío de gran importancia en este sentido, ya que ha permitido que la tecnología involucrada con dicho sistema genere un nivel de competencia entre los fabricantes de equipos y soluciones que giran entorno suyo, reflejándose en mejores opciones para sustentar la plataforma de información de las empresas. Por esa razón, cada vez existen mayor número de organizaciones que fundamentan sus estrategias de largo plazo en esta arquitectura a la que se ha dado llamar "**Arquitectura de Sistemas Abiertos**".

Sin embargo, después de esta gran evolución del Software y el Hardware parece no tener repercusión en verdaderas mejoras. Ya que aunque el Hardware es más accesible y el Software es más fácil de usar, el desarrollo de los sistemas de información se sigue complicando.

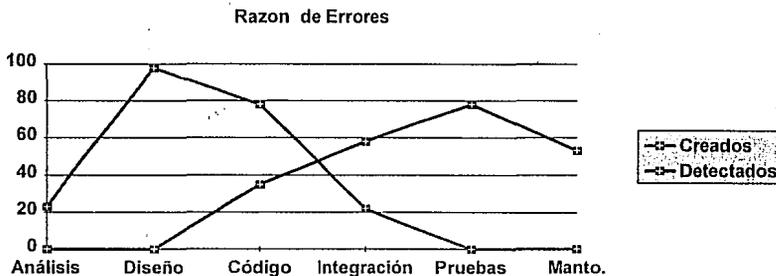
De la misma forma que han evolucionado los demás elementos, han ido surgiendo diferentes ramas de especialización en el campo del desarrollo de Software, que van desde el programador hasta el consultor. Primeramente surge el programador que se encarga de conocer un lenguaje en particular para desarrollar las aplicaciones requeridas. Después aparece el analista que tiene como objetivo analizar un problema y darle una interpretación adecuada para su posible automatización, generalmente dirigirá al programador en el desarrollo del sistema. Finalmente se va creando otro especialista, el consultor que prácticamente orientará a un usuario sobre las alternativas que están a su alcance para cubrir sus necesidades.

Esta jerarquización puede causar que un proyecto complejo se desvíe del objetivo principal, ya que generalmente se delegan responsabilidades en los diferentes niveles y es el programador, el cual aparece finalmente como el responsable de las fallas del sistema; siendo que la culpabilidad está en todas las personas que intervienen en el desarrollo del sistema, involucrando tanto a especialistas que no consideraron debidamente todas las etapas de la metodología del ciclo de vida del sistema como a los usuarios finales que no proporcionaron una idea clara de lo que solicitaron .

El precio de estos errores son : redundancia de trabajo, pérdida de tiempo, un producto final que no cumple con los requerimientos, incremento del costo del sistema en general, etc.

DESARROLLO DE SOFTWARE

ERRORES CREADOS Y DETECTADOS



En la gráfica anterior se comparan los errores que se crean y los que se detectan en las diferentes etapas. En cuanto a los *creados*, éstos se manifiestan mayor mente en el diseño, decreciendo en la codificación. Por otra parte, los errores *detectados* empiezan a manifestarse desde la codificación, creciendo casi en línea recta generando los mayores problemas en la etapa de pruebas y mantenimiento. Lo grave del asunto es que la eliminación de los errores en pruebas y mantenimiento tienen un costo de 100 a 1000 veces superior al de etapas tempranas.

Ahora bien, no implica esto que no existan sistemas de información eficientes o "buenos" actualmente, ya que de hecho existen. Sin embargo, se han desarrollado otros tantos que no satisfacen los requerimientos del solicitante, como por ejemplo alto costo tanto en operación y/o mantenimiento, fallas, obsolescencia, difícil operación e inflexibilidad. Esto ha hecho incrementarse la demanda de nuevas técnicas y el avance de nuevas áreas en el campo del desarrollo de Software que permitan la realización de sistemas más eficientes a un menor costo.

I.2 CASE

Para resolver la problemática planteada anteriormente surge un área denominada Ingeniería de Software la cual establece técnicas y metodologías debidamente formalizadas para la construcción y mantenimiento de Sistemas de Software de calidad, confiables, eficientes y económicamente aceptables.

La Ingeniería de Software se aplica en áreas de la computación tales como la construcción de compiladores y sistemas operativos, aplicaciones específicas de negocios, investigación científica, recreación, medicina, producción, finanzas, controladores, etc. El alcance de soluciones de Software van desde la simple captura de datos hasta la complicada toma de decisiones.

Una de las últimas aportaciones de la Ingeniería de Software es la integración de metodologías y técnicas desarrolladas por ella misma en un nuevo concepto conocido como CASE (Computer Aided Software Engineering, Ingeniería de Software Asistido por computadora).

CASE podemos entenderlo como un ambiente integrado por herramientas que permiten la adecuada aplicación de las técnicas y metodologías mencionadas, con esto el usuario aumenta su productividad y tiene una idea más clara del sistema final.

I.2.1 CASE IDEAL

Aunque la tecnología CASE ha avanzado enormemente en pocos años, aún está lejos de la idealidad. He aquí algunos puntos importantes de lo que se espera cumplan las herramientas CASE en un futuro.

- Todas las fases del proceso de desarrollo deben ser completamente asistidas por la computadora, incluyendo análisis, especificaciones, programación, validación; es decir, todo debe estar integrado en un mismo ambiente.
- El ambiente CASE debe promover y facilitar al usuario la elaboración de documentos suficientemente descriptivos y formales de cada etapa que realiza y estos deben ser sumamente flexibles para modificaciones en línea, verificación de consistencia, etc.
- Debe ser un ambiente abierto, es decir, permitir la posibilidad de incorporar nuevas herramientas que enriquezcan el ambiente mismo y no una colección fija de herramientas de trabajo.

- La modularización debe constituir el mecanismo principal en la estructuración de objetos complejos, no sólo en la fase de programación sino en todas y cada una de las partes del proceso.

- Para todas las herramientas integradas debe existir material de entrenamiento y descripción de sus características que permitan a los usuarios inexpertos explotar debidamente esa herramienta.

- El ambiente debe ser sumamente "amigable" y proveer una interacción computadora-humano en forma intuitiva, esto se logra através de una interfase gráfica con el Usuario (GUI : Graphical User Interface) que se emplea en muchas aplicaciones tales como: Windows, Motif, Mac's, NextSteep, OpenWindow, etc.

I.2.2 METODOLOGÍA CASE

La filosofía de las herramientas CASE se va integrando crecientemente al análisis de sistemas y metodología de diseño. Para cumplir con los objetivos de la metodología, CASE divide al proceso de análisis en una serie de tareas manejables y define elementos finales por cada tarea terminada y éstos a su vez constituyen la entrada a otras tareas que los identifican plenamente.

Significa esto que las herramientas CASE deben estar orientadas a tareas y acomodadas especialmente para cubrir requerimientos de análisis particulares.

La correlación de elementos documentados de las tareas de cada fase del proceso de análisis establece el fundamento para planear e implementar una metodología de análisis orientada a CASE.

Una Metodología CASE generalmente aceptada divide al proceso de desarrollo de software en las siguientes etapas:

- Requerimientos de Análisis.
- Diseño Externo.
- Diseño Interno.
- Codificación.
- Verificación.
- Implementación.
- Mantenimiento.

Requerimientos del Análisis

Esta etapa surge de la necesidad de solucionar un problema, del cual se establece un objetivo para el futuro sistema. En base a éste se realiza un estudio de factibilidad para justificación tanto económica como técnica, lo cual permitirá establecer una estrategia donde se proponen soluciones que se evaluarán para elegir la más apropiada.

Diseño Externo

En esta etapa se establecen las especificaciones del sistema en base a los requerimientos funcionales observados en la etapa de *requerimientos de análisis* por el usuario y el equipo de desarrollo, lo que permite controlar la subsecuente implementación del sistema asegurando que dichos requerimientos se cumplan.

** Se sugiere considerar los siguientes puntos:

- Revisión de documentación de Análisis
- Selección de Alternativas de Diseño.
- Traslación del sistema en subsistemas.
- Especificaciones técnicas del ambiente de desarrollo.
- Establecer las restricciones y alcances.
- Definir los requerimientos de interfase.
- Preparación de la documentación del diseño externo.

Diseño Interno

La información generada durante esta etapa detalla la forma en la que se deben implementar los módulos del sistema proporcionando las especificaciones necesarias de código para los programadores.

Codificación

En esta etapa se transforman las especificaciones escritas en código ejecutable. La documentación que resulta de la fase de programación incluye listados del código fuente, código objeto y referencias cruzadas, así como un manual técnico que permita su mantenimiento.

Verificación

El principal objetivo es demostrar que los requerimientos definidos funcionen acordes a las especificaciones contempladas en la etapa de *Diseño Interno*.

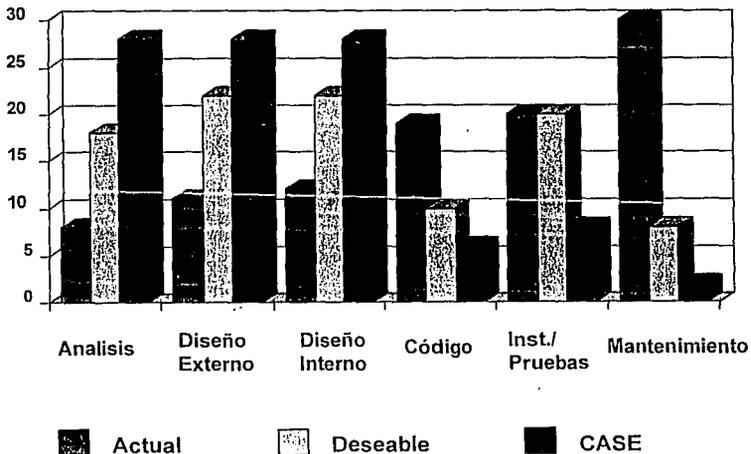
Implementación

Aquí se pone en marcha el sistema, proporcionando la correspondiente guía de uso y mantenimiento del mismo.

Mantenimiento

En esta etapa se considera el mantenimiento requerido para la operación adecuada del sistema, misma que realiza el usuario. Por otro lado se contempla el mantenimiento preventivo y correctivo que puede surgir como consecuencia de fallas u obsolescencia del mismo sistema.

DESARROLLO DE SOFTWARE Actualmente, Deseable y bajo CASE



La metodología orientada a CASE se divide en dos partes según el área de aplicación de sus herramientas: UpperCase y LowerCase.

UPPERCASE

Las herramientas UPPERCASE apoyan al usuario y al equipo de desarrollo en las etapas de requerimientos del análisis y la metodología del diseño, mediante el control formalizado y detallado de las mismas. Esto evita ambigüedades, omisiones, redundancias, excedentes, etc.; de tal manera que se garantiza que las siguientes etapas cumplan con el objetivo previamente establecido. Entre las herramientas que existen para UPPERCASE están:

- Control de Proyecto.
- Diagramas de Control/Flujo de Datos.
- Verificación de Sintaxis.
- Reposición.
- Prototipaje.
- Control de Configuración.
- Metodología del Desarrollo del Sistema.

Para el *Control del Proyecto* las herramientas CASE pueden utilizarse para verificar el progreso en el desarrollo, manejo de esquemas de control y de recursos.

Las herramientas de *Diagramación de Control/Flujo de Datos* pueden utilizarse para la identificación de relaciones organizacionales, describen el flujo de datos entre los usuarios de la organización y el grupo de desarrollo, muestran de forma gráfica la función y rendimiento de entradas, salidas, interfaces y métodos.

Los *Verificadores de Sintaxis* se utilizan para analizar estructuras que muestran la composición funcional del sistema presente, definen los requerimientos del sistema propuesto y maneja un referencia cruzada de lógica de diseño contra reglas de diseño y requerimientos funcionales.

Las herramientas de *Reposición* facilitan el almacenamiento de diagramas de estructuras; definiciones de pantallas y menús; esquemas de reportes; descripciones de registros, lógica de proceso; modelos de datos, organizacionales, de proceso; código fuente; reglas de negocios; elementos de datos y componentes del sistema de información.

Un *Prototipador* se utiliza para construir modelos que definen los requerimientos del sistema, permite simular el prototipo para enfocar requerimientos del futuro sistema y probar su comportamiento.

Los *Controladores de Configuración* llevan registro de los módulos; esto nos permite identificar todas aquellas relaciones intermodulares, dependencias, redundancias.

Las herramientas de *Metodología de Desarrollo del Sistema* son las responsables de vigilar que las etapas de desarrollo del mismo se cumplan en el orden establecido por la filosofía CASE.

LOWERCASE

Proporcionan ayuda a los programadores en la implementación y desarrollo del sistema. Entre sus herramientas nos encontramos a:

- Simulación
- Generación de Código
- Reposición
- Ingeniería Inversa

Las herramientas de *Simulación* toman como base prototipos y los utilizan para una fase de implementación y diseño.

Las herramientas de *Generación de Código* son simplemente ambientes que facilitan la generación y control de código.

Las herramientas de *Reposición* en este caso tienen gran aplicación para el código fuente y registra todos los movimientos originados por la *Ingeniería Inversa*. La Ingeniería Inversa se utiliza para actualizar el software o cambiar de plataforma, reutilizando la mayor parte del software anterior, haciendo los cambios mínimos en las partes que sean necesarias. Todo esto basado en el concepto RCR (Reusable Complement Recovery) lo que significa reutilizar en lo posible el código recuperado del sistema anterior o también para el desarrollo de nuevos sistemas.

I.2.3 VENTAJAS Y DESVENTAJAS DE LAS HERRAMIENTAS CASE

Entre las ventajas podemos mencionar:

- Promueve el cumplimiento de la metodología de desarrollo de sistemas.
- Facilita la comunicación entre los usuarios que intervienen en el proceso, lo que incrementa la creatividad de los mismos enriqueciendo el producto final.
- Provee una mejor visión del sistema.
- Facilita la elaboración de documentación estándar.
- Promueve una mayor competitividad entre los usuarios de la herramienta en comparación con los que utilizan el sistema tradicional.
- Reduce costos.
- Da un margen de tiempo más amplio para revisiones.
- Permite liberar un sistema confiable, eficiente y versátil.

Entre las desventajas encontramos:

- Actualmente las herramientas CASE tienen un costo muy elevado.
- La mayoría son dependientes de determinadas plataformas de Hardware. Algunas de las cuales exigen gran cantidad de recursos para el perfecto desenvolvimiento de la herramienta CASE.
- Adopción de una nueva metodología de trabajo que contrasta con la metodología particular empleada por los desarrolladores de sistemas, ya que consideran que algunas etapas son obvias e innecesarias.
- Ya que las herramientas CASE actualmente no cumplen con el concepto de "CASE Ideal", algunas etapas deben implementarse manualmente.

II PROTOTIPOS

El prototipaje es básicamente una técnica de análisis. La mayoría de la literatura sobre la materia resalta el uso del prototipaje para descubrir una serie de requerimientos funcionales del sistema de software propuesto.

El prototipaje fue formalmente introducido en los inicios de esta década (de los 90's) como una alternativa para resolver la problemática de la entrega forzada de sistemas de software funcionalmente incorrectos. Por lo cual surgieron nuevas herramientas de software rápidamente.

Estas herramientas hacen económicamente factible la construcción de dos sistemas -*un prototipo y el producto final*- para liberar correctamente un sistema. Las nuevas herramientas fueron suficientemente poderosas para permitir desarrollo de prototipos rápidos y de bajo costo que contienen la esencia del sistema propuesto. Existe suficiente flexibilidad de tal forma que las modificaciones puedan ser fácilmente realizadas sin que los desarrolladores deban recurrir a otras alternativas fuera de CASE.

Es decir, el prototipaje presenta la oportunidad de dar mantenimiento al software desde etapas tempranas del ciclo de vida, haciéndolo fácil y barato. Con la herramientas correctas y el enfoque adecuado las técnicas de prototipaje pueden aplicarse para modificar el sistema de una forma sumamente económica.

El análisis estructurado a madurado más que cualquier otra técnica estructurada, tal vez se debe a tempranas experiencias frustrantes al intentar completar y detallar exactamente los requerimientos funcionales con modelos estáticos en papel. Los enfoques de análisis estructurado han sufrido frecuentes mejoras. Una mejora reciente ha sido la introducción de un enfoque tridimensional. Este enfoque fue descrito por vez primera por Tom DeMarco en "Controlling Software Projects". DeMarco usa la analogía de un arquitecto especificando una construcción en tres dimensiones:

- Elevación lateral.
- Elevación frontal.
- Planta.

El subraya que los constructores de software utilizan el mismo enfoque para preparar un modelo de interface de funciones (flujo de datos), un modelo de almacenamiento de información (entidad-relación) y un modelo de ruta crítica (estados de transición).

Bien ahora hablemos de la implementación de prototipos. Por ejemplo, la implementación "top-down" involucra el desarrollo de los primeros niveles de los diagramas de flujo de datos que después se utilizan como base para los primeros niveles de diseño que son a su vez la base del código inicial. Este código inicial puede incluir código

real tal como FORTRAN, C, LISP u otros dependiendo el propósito del sistema pero sólo para alto nivel o los módulos de control. Los detalles del sistema pueden proponerse hasta que llegue el tiempo en deba implementarse el nivel correspondiente. De esta forma, el esqueleto del sistema puede ser demostrado más rápido que con el enfoque de desarrollo de sistemas tradicional.

Bien, el prototipaje es una implementación "top-down" y esto no es una contradicción, ya que bajo la metodología CASE es posible desarrollar el prototipo inicial del sistema sin necesidad de escribir y usar código real, dada la ventaja de las herramientas CASE en este rubro. Los lenguajes de cuarta generación, la programación visual, los generadores de funciones, los generadores de pantallas, las interfaces de procedimientos y los reportes hacen posible presentar el alto nivel del sistema a los usuarios finales mucho más rápido que escribiendo código. Los módulos creados con tales herramientas podrán modificarse fácil y convenientemente durante la fase de iteración del prototipo. Todas las interfaces entre sí y entre los módulos pueden ser probadas tempranamente en el ciclo de vida, sin la producción de código el cuál se generará en el nivel adecuado. No es necesario por tanto escribir código de prueba para experimentar con detalle los módulos durante la fase de prototipaje.

Definitivamente la primera demostración de un prototipo rápido será intencionalmente imperfecta, pero a su vez será muy económica su generación y muy fácil su modificación. El prototipaje no se debe considerar como un enfoque veloz y mal hecho; más bien debe considerarse como un *complemento* de las técnicas de desarrollo estructuradas existentes más que una *alternativa* a esas técnicas.

Dentro de la comunidad de profesionales de la computación, es un hecho bien conocido que el costo de corregir un error en un sistema de cómputo incrementa drásticamente conforme el ciclo de vida del sistema progresa. Muchos autores convergen en la idea de que el costo de corregir errores se va incrementando fuertemente conforme el sistema pasa por las etapas de análisis, diseño y desarrollo, para ser muy costosa la corrección durante las fases de postimplementación y operación.

El prototipaje sirve para eliminar muchos de esos errores en las etapas tempranas del proyecto, antes de que se escriba código real alguno. El modificar un prototipo es fácil, barato y rápido. Esto obviamente no sólo representa ahorros muy significativos de dinero en el costo total del software sino una gran cantidad de beneficios asociados intangibles.

En sí, el prototipaje es un enfoque para evitar riesgos, esto es, el prototipaje no introduce una nueva política o riesgos económicos al proceso de desarrollo de software, pero sí reduce considerablemente los factores de riesgo severo asociados con el desarrollo del software.

Este nuevo y poderoso enfoque para el desarrollo de software está apenas empezando a utilizarse en algunas organizaciones para el desarrollo de nuevas, grandes y complicadas aplicaciones. Es importante establecer que en la actualidad la mayoría de los

desarrollos de software aún utilizan el ciclo de vida tradicional involucrando la utilización de productos basados en papel para el análisis y el diseño para escribir programas con lenguajes de tercera generación y procedimientos de control para implementar el diseño. Sin embargo, esto está cambiando y muchas otras organizaciones han tenido experiencias exitosas al usar el prototipaje. Estos experimentos han producido sistemas de aplicación que tienen un nivel mayor que el normal en utilidades de usuario final. El software desarrollado con prototipos ha exhibido una clara tendencia a satisfacer de mejor manera los requerimientos reales del usuario y producir información y funcionalidad más completa, más exacta y más descriptiva.

Ultimamente con respecto a la estandarización de la conceptualización del prototipaje la American National Standar Institute (ANSI) podría ofrecer una definición que puede ser considerada en pocas palabras como de "mantenimiento", la National Bureau de estándares ha decidido tres terminologías en este aspecto que son: correctivo, de perfeccionamiento y adaptativo.

El software de prototipos ha sido perfilado como :

- Rápidamente funcional.
- Un modelo mínimo del sistema propuesto.
- Fácil de modificar.
- El modelo de trabajo será escrito en un lenguaje de cuarta generación.
- Siempre será reescrito en un lenguaje procedural para su implementación.
- Un modelo usado para determinar correcciones de diseño.
- Una forma rápida para aproximarse a una solución del problema.
- Siempre desechable.
- Generalmente existe una interface humano-computadora.
- Generalmente promueve la comunicación entre los desarrolladores y usuarios finales.
- No es operable para niveles de extrema complejidad de desempeño del sistema.
- No intenta ser el sistema final.
- No presenta avanzadas características del sistema.
- Presenta solamente un modelo de reportes y ventanas.
- Un estudio de viabilidad.
- Definición de la estrategia de requerimientos.

Varios escritores y expertos piensan que :

- Deben ser hechos por experimentados prototipadores.
- Reemplazará el ciclo tradicional de vida de desarrollo de un sistema.
- Añade fases al ciclo de vida tradicional del sistema.
- Involucra el uso de lenguajes de cuarta generación.
- Puede realizarse paralelamente con aplicaciones realizadas en forma tradicional.
- Requiere de un diccionario de datos activo y un directorio de sistemas.

- Requiere de mecanismos de control de liberación para su administración.
- Aplicable solamente a pequeños sistemas no muy complicados.
- No puede ser usado para el desarrollo de sistemas en tiempo real.

El valor real de los prototipos viene con presentación del mismo, ya que de esta forma se pueden realizar varias transformaciones familiares al usuario, el cual debe ser capaz de decidir el tipo de información que es adecuada, esto es, decir si es suficiente, legible y fácil de obtener. Estas determinaciones del usuario son posibles solamente cuando puede obtenerse una respuesta inmediatamente verificable por parte del prototipo a sus cuestiones tales como: ¿ Qué pasa si hago esto ?

II.1 CLASIFICACION DE PROTOTIPOS MAS COMUNES

Existen 4 filosofías establecidas acerca del prototipaje, cabe aclarar que no son las únicas, sino las mas comunes para poder clasificar a un prototipo. Dichas filosofías son las siguientes:

- THROWAWAY
- QUICK&DIRTY
- DESIGN DRIVEN
- MOCK-UP

Prototipo THROWAWAY

El prototipo throwaway describe el diseño de un producto a ser usado solamente para ayudar al usuario a identificar los requerimientos necesarios para un nuevo sistema. Por lo cual este tipo de prototipo puede ser idealizado como un sistema dinámico de especificación de requerimientos.

Un ejemplo de este tipo de prototipos son los sistemas stand-alone (máquinas dedicadas con disco y software propio) de lenguajes de cuarta generación con características de desempeño muy pobre. Otro ejemplo son los toolkit (herramientas de software) que pueden operar solamente en computadoras personales en forma stand-alone (el resultado no puede ser operado en ambientes multiusuario).

Algunas fuentes literarias indican que los prototipos throwaway tienden a incrementar cerca de un 10 por ciento el costo del presupuesto previsto, actualmente los prototipos de este tipo pueden reducir significativamente la revisión de los requerimientos y diseño de documentos y pueden a la larga evitar sorpresas durante la fase de pruebas del sistema.

Prototipo QUICK&DIRTY

Quick&dirty es un viejo término empleado para describir a un sistema que permite la realización rápida de un prototipo el cual puede ser modificado hasta obtener la aprobación del usuario. En el pasado esto significaba el tener que escribir programas en lenguajes estándares para poder obtener un prototipo que diera una idea del sistema final, todo esto sin un correspondiente análisis o diseño, ya que los programas eran modificados en base a una retroalimentación por parte del usuario el cual daba sus puntos de vista de acuerdo a los resultados generados por el incipiente sistema. Por consecuencia este tipo de desarrollo de sistemas resultaba ser demasiado caro, ya que los programas tenían que ser constantemente modificados ("parchados") antes de ser liberados y raramente acompañados de cualquier documentación.

Sin adecuadas herramientas para la realización de prototipos cualquier modelo realizado de prototipo al inicio del desarrollo del proyecto es por definición *quick&dirty*.

El desarrollo de sistemas de prototipos *quick&dirty* es particularmente capcioso; ya que el primer diseño tiene la característica de ser un trabajo fácil y con buena presentación, pero misteriosamente en cuanto comienza a crecer en cuanto a desarrollo, el esfuerzo para la elaboración de una siguiente etapa se torna más difícil que la anterior, lo que incrementa en gran medida el costo del sistema.

El desarrollo de sistemas prototipo *quick&dirty* no es nada mas que la reencarnación de los métodos de software utilizados al principio de la utilización de computadoras.

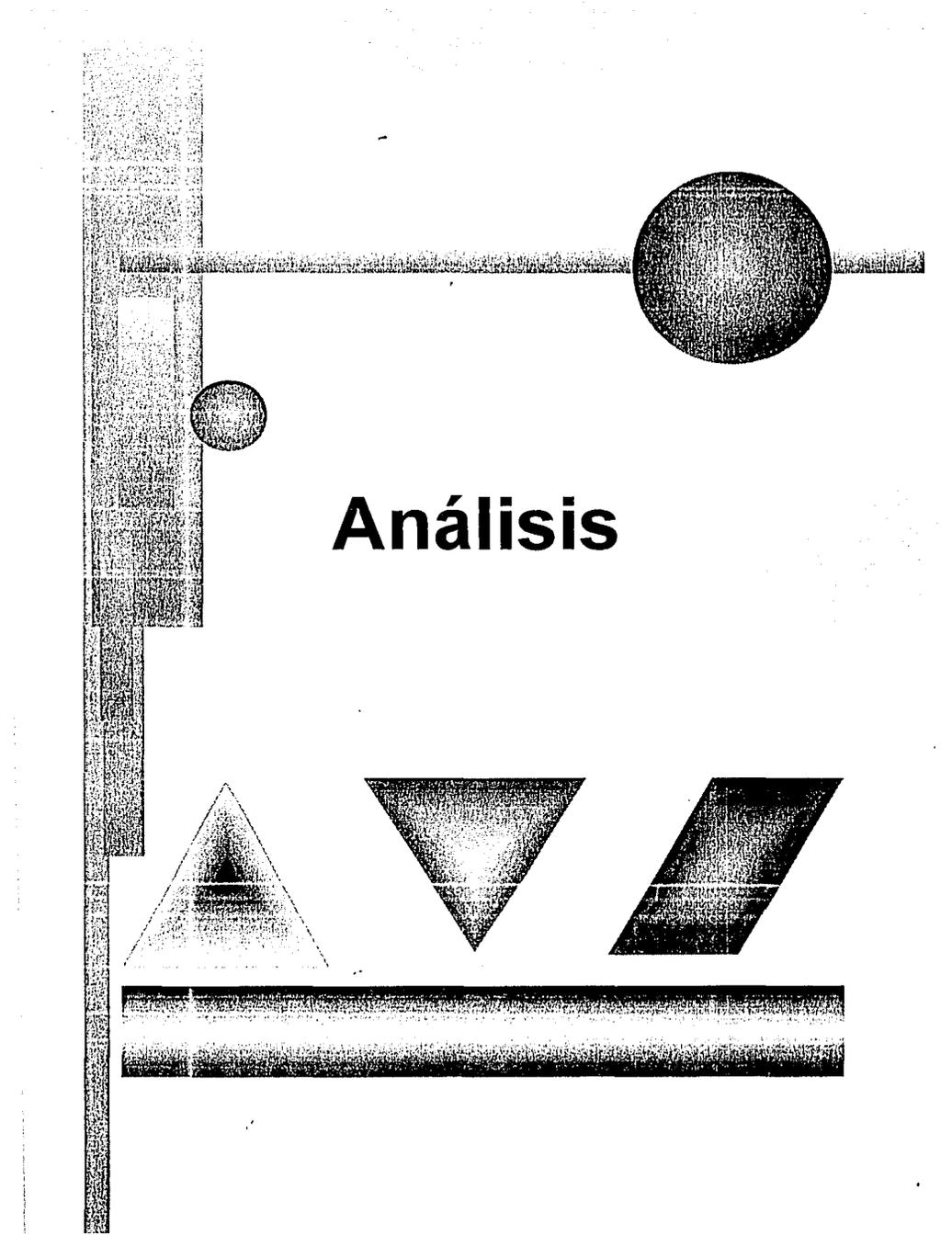
Prototipo Detallado DESIGN-DRIVEN

Este tipo de diseño se deriva de las disciplinas desarrolladas por la Ingeniería de Software donde el prototipo significa una producción previa del modelo del sistema final. El modelo es completo y lo más perfecto posible, por lo cual son utilizados métodos de prueba para evitar cualquier defecto en el sistema antes de la liberación final del producto. En términos de cómputo, un sistema es desarrollado usando técnicas tradicionales para el análisis, diseño y desarrollo. Los usuarios pueden ver los resultados de un análisis de alto nivel mediante diagramas de flujo; pero no pueden ver nada mas hasta después de que el diseño detallado haya sido completado. Este tipo de desarrollo de prototipos es problemático porque es demasiado detallado, documentos de análisis y diseño son demasiado específicos, por lo cual resulta demasiado caro el modificar alguna de estas etapas, además que la presentación del sistema puede descubrir resultados inesperados e indeseables.

Se puede observar que uno de los grandes problemas acarreados por el uso de este tipo de desarrollo de prototipos son los cambios requeridos y solicitados como resultado de la demostración del sistema, dichos cambios son típicamente mayores en número y en complejidad.

Prototipo NONFUNCTIONING MOCK-UPS (Modelo o réplica)

Un *Mock-Up* provee a un usuario con ejemplos visuales de los procesos de entrada y salida del sistema. El mock-up es muchas veces realizado en papel o puede ser implementado por pantallas en una computadora. Los ejemplos son preparados manualmente por los programadores. No existen datos reales de entrada en la computadora, así como tampoco existen datos generados para la salida, pero los programadores casi siempre incluyen datos que permiten mostrar el formato de los menus, los formatos de tipo de entrada o reportes. Aquí puede hacerse una distinción entre los prototipos hechos mediante *throwaways* y *mock-ups* basados en la presencia o ausencia de datos reales. Obviamente el *Mock-Up* no es aprovechado sino hasta después de que los requerimientos han sido definidos. El desarrollo de prototipos *Mock-ups* resultan ser no muy caros y de gran ayuda, pero no son definitivos en la identificación de los requerimientos funcionales porque no tienen la capacidad para realizar la experimentación interactiva.



Análisis

III. ANALISIS

III.1 ESTUDIO PRELIMINAR

La tecnología CASE ha tomado un fuerte impulso, cada vez aparecen en el mercado herramientas CASE que se aproximan más a la metodología ideal establecida.

Sin embargo, la gran mayoría de las herramientas actuales son muy limitadas, ya que son productos específicos para ciertas etapas de desarrollo, lo que implica que el resto deba seguirse realizando en forma tradicional o con otra herramienta. Algunas herramientas cubren la parte de análisis, otras la de diseño, algunas generan código, otras más trabajarán en forma excelente los repositorios, diccionarios de datos, documentación en general y las restantes simulación, prototipaje, ingeniería inversa, etc.

Algo que también es cierto es que en la actualidad el precio de la mayoría de estos productos es realmente alto y definitivamente esto se vuelve un obstáculo para adquirir la tecnología CASE aún cuando está comprobado que los beneficios son enormes.

Ahora bien, nos encontramos que muchas de las herramientas son dependientes de ciertas plataformas las cuales no siempre son estándares del mercado y eso definitivamente disminuye sus posibilidades de aceptación.

Ya existen por supuesto herramientas con un mercado que va creciendo y son aquéllas que se basan en plataformas estándar comercialmente hablando, el único inconveniente es que estas herramientas consumen recursos de cómputo en gran cantidad, refiriéndonos a memoria RAM, espacio en disco duro, monitores de la mejor resolución posible, el tipo de CPU, etc.

III.2 ENCUESTA

Se decidió elaborar una encuesta para conocer aspectos y opiniones de diferentes usuarios y diseñadores; sobre sus problemas en cuanto a diseño, las herramientas que utilizaban, sus metodologías para desarrollar proyectos, su conocimiento sobre tecnología CASE, etc.

Los resultados de la encuesta permitieron tener conocimiento de aspectos no visualizados por el equipo de desarrollo lo que permitió analizar la inclusión de características adicionales que podría soportar el sistema.

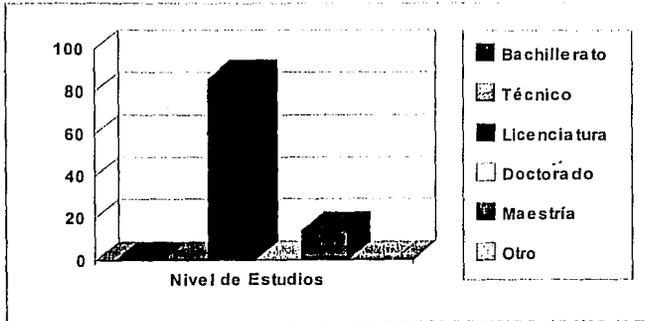
La encuesta fué aplicada en diversos sectores del área de sistemas, entre otros:

- Líderes de proyecto de ciertas compañías
- Empresas dedicadas a proporcionar servicios de software
- Población estudiantil de licenciatura ya dedicada al área de desarrollo
- Empresas de consultoría
- Centros de cómputo
- Otros

Los resultados obtenidos fueron de gran utilidad para el presente sistema, los puntos fueron ponderados y analizados, después de lo cual el equipo de trabajo los fué incluyendo de forma que satisficieran en la medida de lo posible, las características propuestas por los usuarios potenciales, para darle a este trabajo una versatilidad y mejor visión.

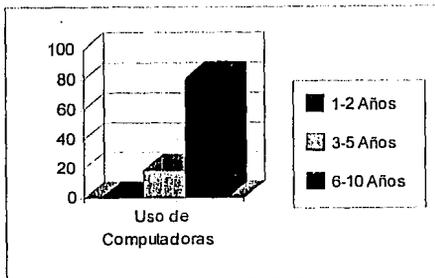
Aquí comentamos los resultados obtenidos de los encuestados:

NIVEL DE ESTUDIOS ALCANZADOS :



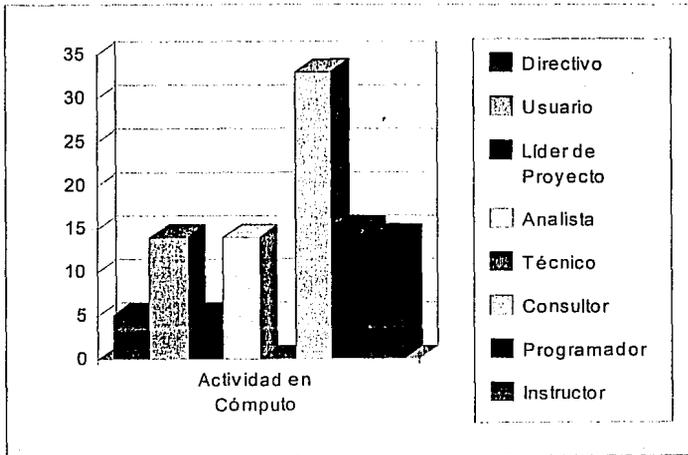
La población es mayoritaria profesionalista lo que nos permite considerar que sus respuestas son objetivas.

EXPERIENCIA EN COMPUTO :



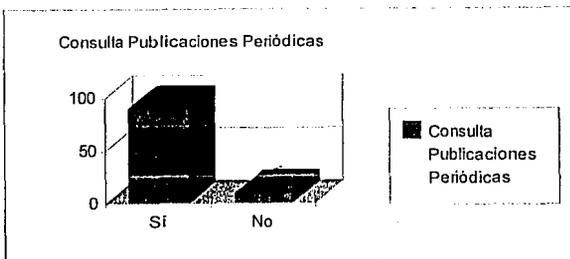
- El 80% cuenta al menos con 6 años en el área
- En estos años han ocurrido notables mejoras
- La población tiene por tanto buena experiencia en el área

ACTIVIDAD EN COMPUTO :-



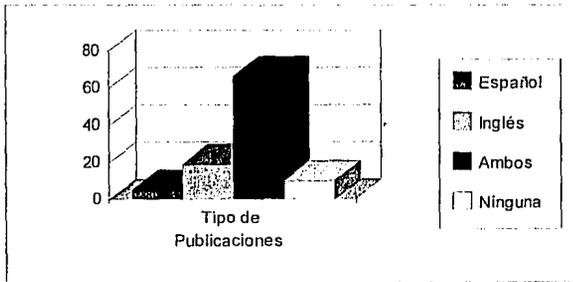
- La consultoría ha tomado gran auge
- Aproximadamente un 70% se encuentra en desarrollo de sistemas
- Fué de gran utilidad tener opiniones de diversas especialidades

¿ EXISTE ACTUALIZACION PERIODICA ?



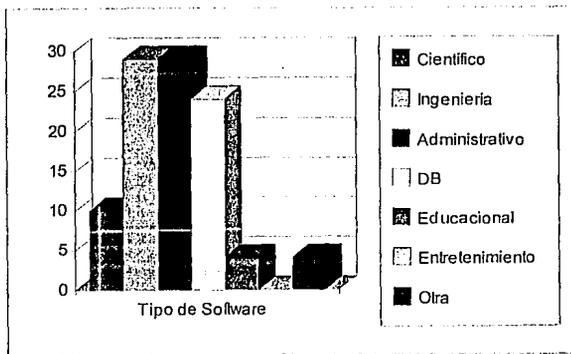
- El 90% se actualiza regularmente
- Eso permite que la curva de aprendizaje se mejore

TIPO DE PUBLICACIONES CONSULTADAS:



- Por fortuna la tendencia es consultar las publicaciones sin importar el idioma.
- Esto implica que los usuarios están interesados realmente en los artículos consultados, actualización .

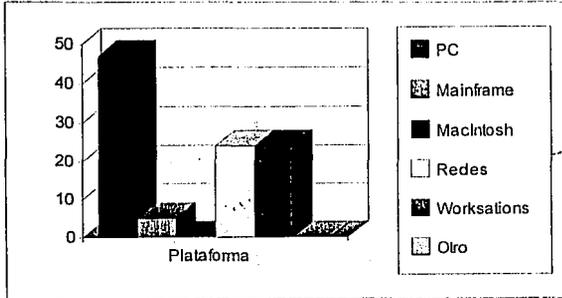
TIPO DE SOFTWARE DESARROLLADO:



Prevalece el desarrollo de software para las áreas de :

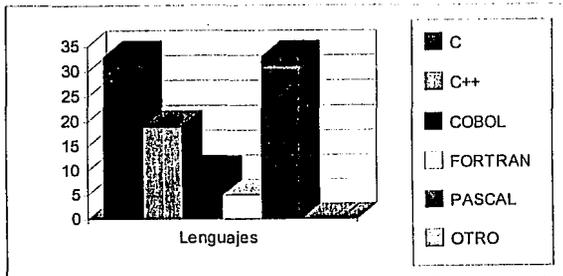
- Ingeniería
- Administración
- Bases de Datos

PLATAFORMA UTILIZADA:

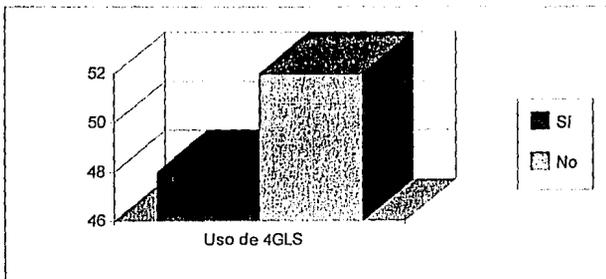


- Predomina la PC por su versatilidad y facilidad de manejo
- Aumenta la utilización de WorkStations y Redes
- En los mainframes actualmente sólo se mantienen sistemas existentes, el desarrollo es casi nulo.

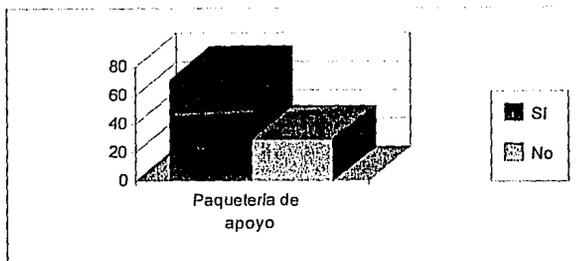
LENGUAJES DE PROGRAMACION:



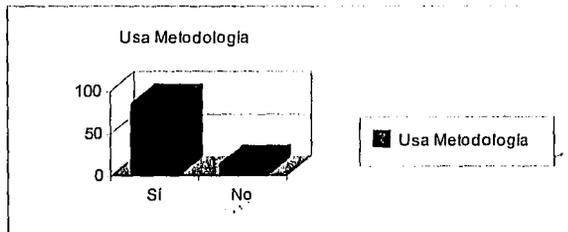
- Los mas utilizados, C y PASCAL
- C++ va en aumento
- COBOL y FORTRAN aún se utilizan, generalmente para dar mantenimiento a sistemas

¿ SE UTILIZAN 4GLS ?

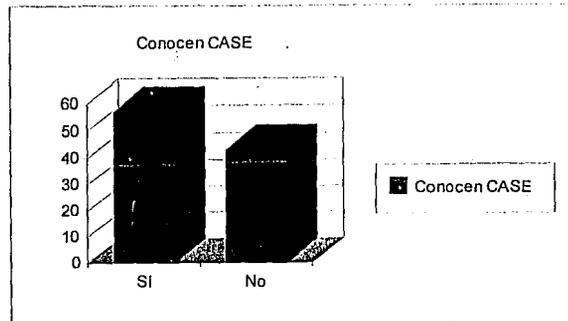
- La utilización de 4GL'S va en aumento
- Cada vez se desarrollan herramientas más amigables y específicas
- Algunas herramientas cuentan con interfaces para soportar diferentes arquitecturas

SE UTILIZA SOFTWARE DE APOYO NO ESPECIFICO PARA DESARROLLO :

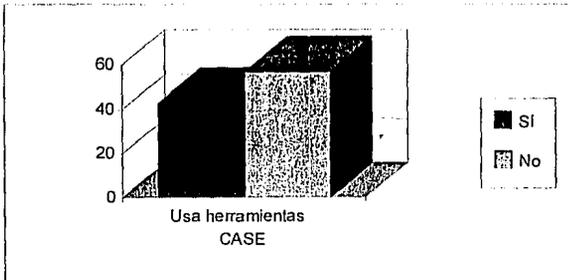
- Se utilizan los recursos de cómputo disponibles
- Gran parte de ellos no están diseñados para desarrollo
- Se observa la necesidad de contar con herramientas para desarrollo de software

¿ SE UTILIZA UNA METODOLOGIA PARA PROGRAMAR ?

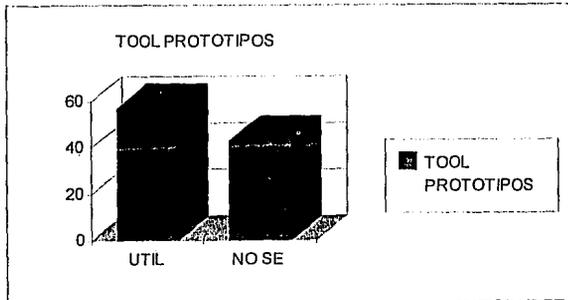
- En general si se utiliza un método
- El método no necesariamente está estandarizado
- Puede ser un método creado personalmente

¿ SE TIENE CONOCIMIENTO SOBRE CASE ?

- CASE empieza a difundirse
- Una parte de los usuarios lo conocen pero no lo han utilizado realmente
- También hay quien lo utiliza sin saberlo

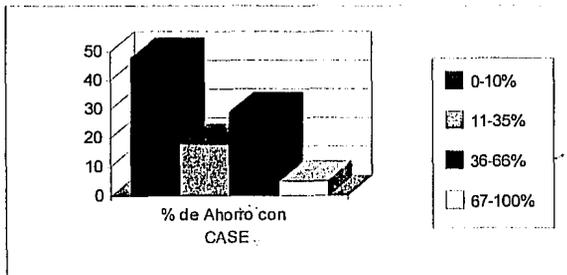
¿ SE HAN UTILIZADO HERRAMIENTAS CASE ?

- Menos de la mitad han realmente utilizado CASE

¿ SON UTILES LOS PROTOTIPOS ?

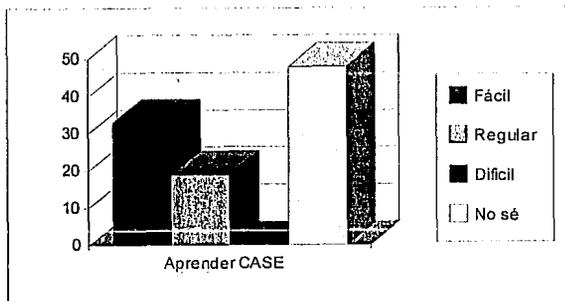
- Más de la mitad lo considera útil
- Los restantes lo desconocen o no lo identifican aún teniendo la herramienta integrada

¿ CASE AHORRA CICLO DE DESARROLLO ?



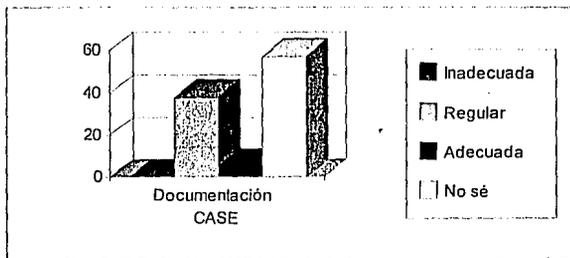
- Aproximadamente un 50% estiman muy poco ahorro
- El otro 50% estiman un ahorro considerable
- Este ahorro implica tiempo, recursos humanos, materiales y esfuerzos

CURVA DE APRENDIZAJE EN CASE :



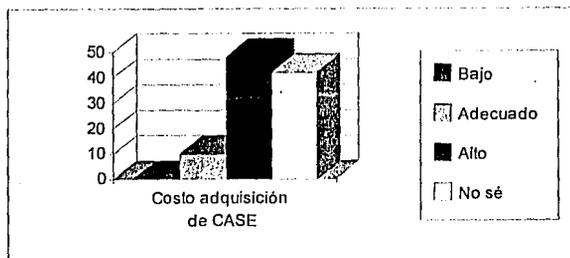
- La mayoría lo considera fácil
- En su facilidad radica parte de su potencialidad

LA DOCUMENTACION CASE ES ...



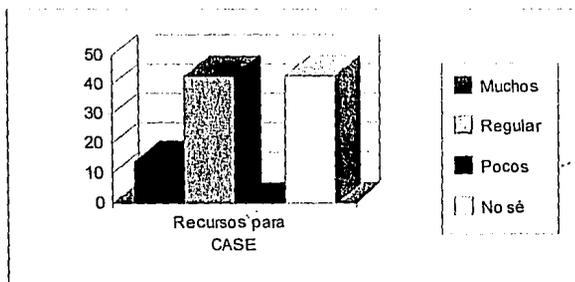
- La documentación es regular
- La mayoría no se decidieron a calificarla
- Pocos productos ofrecen documentación de fácil comprensión

ADQUIRIR CASE ES ...



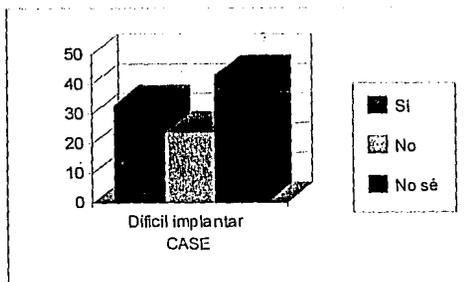
- Adquirir CASE no es muy accesible
- Una razón es que no se conoce con certeza el costo-beneficio para un producto CASE

LOS RECURSOS QUE NECESITA CASE SON ...



- La minoría los considera aceptables
- Una gran parte consideró que aún son demasiados
- Otra gran parte no se decidió a calificar los recursos que requiere

IMPLEMENTAR CASE SE CONSIDERA ...



- Aún en la actualidad se considera difícil implantar CASE
- Las razones son diversas

CONCLUSIONES DE LA ENCUESTA**CONCLUSIONES GENERALES DE LA POBLACION**

La población de sistemas ha elevado su nivel de capacitación

La población se actualiza cada vez más

El idioma inglés ya no es barrera para el progreso en el área de software

Entre los lenguajes de tercera generación se sigue utilizando C y Pascal

El FORTRAN es exclusivo para centros de investigación y desarrollos científicos

El COBOL se utiliza para mantenimiento de sistemas ya existentes, los desarrollos en este lenguaje se han sustituido por otro tipo de software

Los lenguajes de cuarta generación se han difundido enormemente

Se han utilizado todos los recursos disponibles para apoyo en el análisis y desarrollo como diagramadores, procesadores de texto, hojas de cálculo, etc, los cuales no están diseñados para este tipo de sistemas

Eso implica la necesidad de contar con herramientas específicas para análisis y desarrollo de sistemas, ésto es utilizar y difundir las herramientas CASE

CONCLUSIONES SOBRE CASE

El costo de adquisición aún es alto

La documentación de algunos productos aún deja que desear

Se requieren ciertos conocimientos para utilizar algunos productos

Trabajan bajo plataformas específicas que no son muy accesibles

Los recursos empiezan a estandarizarse

El campo de acción se incrementa

Se requiere una mayor difusión

La implantación de CASE se ha dificultado por idiosincracia, costumbre, rechazo al cambio, falta de recursos económicos, falta de infraestructura, falta de capacitación, falta de metodologías, etc.

III.3 PROPUESTA

III.3.1 Sistema GPCASE

Bien, considerando los puntos del Estudio Preliminar tenemos un panorama de las herramientas CASE en el mercado y es precisamente de esas consideraciones que nace la idea de desarrollar el GPCASE.

GPCASE surge como una herramienta CASE que cubrirá una etapa de esa metodología. Entre las características más interesantes del sistema está primeramente, que podrá operar bajo un ambiente de trabajo estándar comercialmente hablando como es Windows, esto permitirá definitivamente una gran versatilidad ya que es sumamente accesible.

Ahora bien, los requerimientos que solicita GPCASE para operar, son prácticamente los mismos que cualquier computadora debe tener para poder trabajar con Windows; es decir, si un usuario tiene los recursos suficientes para trabajar con él adecuadamente, entonces podrá utilizar GPCASE sin problema.

Por ahora, GPCASE es una herramienta que facilita el cumplimiento de una parte de la metodología.

III.3.2 REQUERIMIENTOS

III.3.2.1 REQUERIMIENTOS PARA OPERACION

El equipo recomendado para la operación óptima del sistema GPCASE es el siguiente:

Requerimientos de Hardware

- * Computadora 386 sx.
- * Memoria RAM : 2 MB.
- * Disco Duro 20 MB.
- * Monitor VGA Color.

Se recomienda la computadora 386 sx, debido a que los estándares comerciales tales como Windows 3.1, compiladores de desarrollo, etc; mismos con los que se desarrolló el sistema prototipador GPCASE, presentan su mejor rendimiento a partir de esta arquitectura. La cantidad de 2 MB de RAM es la mínima requerida por Windows para su desempeño. En

el disco duro recaerá todo el peso de la información, librerías, programas, bases de datos, etc. Se sugiere este tipo de monitor ya que el GPCASE trabaja con la misma filosofía de Windows, la cual establece una comunicación visual dinámica con el usuario.

Requerimientos de Software

- * MicroSoft Windows ver. 3.1

III.3.2.2 REQUERIMIENTOS PARA EL DESARROLLO DE GPCASE

Requerimientos de Hardware

- * Computadora 386 sx.
- * Velocidad 33 Mhz.
- * Memoria RAM : 4 MB.
- * Disco Duro 80 MB.
- * Monitor VGA Color.

Se recomienda la computadora 386 sx, debido al empleo de los programas que se enumeran en la sección de Requerimientos de Software descritos más adelante. La velocidad de 33 Mhz o mayor nos permiten como desarrolladores un buen rendimiento, debido a que el excelente tiempo de respuesta evita que nuestra atención se desvíe del objetivo propuesto en turno. Los 4 MB son necesarios debido al *Overhead* (fuerte carga de trabajo) ocasionado por el gran número de herramientas empleadas en el desarrollo del sistema, las cuales demandaban una gran cantidad de recursos en general; por experiencia propia una cantidad menor de RAM aumenta la posibilidad de saturar la computadora y disminuir su desempeño llegando incluso hasta un *Crash* (caída total del sistema). Las herramientas enumeradas necesitan aproximadamente 60 MB para su sola instalación, asimismo el Windows requiere entre 7 y 9 MB para el Swapping (manejo de memoria virtual).

Requerimientos de Software

- * MicroSoft Windows ver. 3.1 .
- * MicroSoft Programmers WorkBench C/C++ 7.0 .
- * MicroSoft Windows SDK (Software Development Kit).
- * Btrieve for Windows de Novell.
- * Windows Maker.
- * Drover's Professional ToolBox for Windows.

Windows 3.1 es el medio ambiente para la integración de las herramientas de desarrollo. Es un ambiente gráfico que se ha convertido en el estándar del mercado de las PC's compatibles y por tanto es obvia su inclusión para el desarrollo de GPCASE.

Microsoft C/C++ 7.0 incluye el Windows SDK completo, el cual es el sistema de desarrollo de software para aplicaciones Windows. El C/C++ 7.0 contiene además entre otras cosas una librería llamada "Foundation Classes" que provee herramientas orientadas a objetos para programar directamente Windows. El ambiente de desarrollo integrado del C/C++ 7.0 es el PWB (Programmer's Work Bench) que posee características de edición poderosas, permite edición múltiple de archivos, tres modos de operación para copia o borrado, una de las cuales es la edición en columna. Se manejan fácilmente las macros y el remapeo de teclado. A diferencia de otros compiladores como Borland C++, el PWB tiene un generador MAKE que usa menu tipo pop-up para seleccionar los archivos fuente de la aplicación correspondiente. Además el archivo que genera MAKE es tipo ASCII lo que permite su edición fuera del ambiente. El depurador de Microsoft ya está diseñado para trabajar con un sólo monitor lo que es una gran mejora ya que antes se necesitaban dos monitores para depurar programas Windows. El CodeView tiene además la capacidad de revisar datos dentro de los objetos C++ y activación de puntos de ruptura y rastreo dentro de las clases C++. Aunque El C/C++ 7.0 no es el más rápido en compilación en el mercado si lo es en cuanto a ejecución y lo que es más importante para desarrollos orientados a C++ y Windows el código ejecutable es compacto, óptimo y rápido.

Btrieve for Windows de Novell es un manejador sumamente flexible para desarrollar aplicaciones, los accesos a las bases de datos son veloces y sencillas, la interface se realiza a través de una sola función que hace las llamadas al manejador y la cual se codifica donde se requiere dentro del código, en este caso C/C++ y Windows SDK. El manejo de una sola función facilita enormemente el control del código y evita la generación de errores adicionales ya que además la función provee manejo de errores.

WindowsMaker ayuda al diseñador de aplicaciones generar código mucho más rápido que de la forma tradicional, ya que le permite modificar, hacer pruebas y eliminar código en forma fácil, esto no quiere decir que esta herramienta cubra todas las necesidades del desarrollador desde la etapa de diseño hasta la etapa de mantenimiento, sino que únicamente le ahorra tiempo en la definición de menus, ventanas, colores y tamaños de las ventanas, etc., ya que muchas veces estos factores son los que se consideran y en los cuales el programador emplea la mayor parte de su tiempo en atenderlos, además para el programador que se inicie en el ambiente Windows le es de gran ayuda, ya que le da una idea de como funciona Windows así como una apreciación de la utilidad de WindowsMaker. WindowsMaker elimina el trabajo tedioso en el desarrollo de aplicaciones Windows, incrementa la productividad en los usuarios expertos, así como el aumento en la curva de aprendizaje de los programadores novatos, para el desarrollador es de gran ayuda, ya que el sistema presenta una etapa de simulación lo cual evita que se tengan que hacer un gran número de compilaciones para ver como se comporta el diseño desarrollado.

Las Drover's Professional ToolBox for Windows son rutinas de interface para programas de Microsoft C, Actor, WindowsMaker y Borland C++. Consiste esencialmente de 6 librerías dinámicas de liga (Dynamic Link Libraries, DLLs). Estas sirven de ayuda para incrementar la funcionalidad del Microsoft Windows Software Development Kit (SDK) y para simplificar la tarea del desarrollador profesional, con aplicaciones de alta calidad; por medio del uso de controles y funciones propias del ToolBox el profesional puede hacer más fácil la etapa de desarrollo. Así mismo ToolBox ha sido desarrollado lo más vinculado posible a las funciones del SDK, las convenciones de los nombres son similares, así como un gran número de funciones de bajo nivel, muchas de las cuales reemplazan a las ya existentes en C las cuales no aceptan FAR pointers (requeridas por windows). La solución que emplea ToolBox es el uso del modelo de memoria Large (fuerza automáticamente al uso de far Pointers).

III.3.2.3 PERFIL DEL USUARIO DE GPCASE

Es de suma importancia considerar ciertos aspectos sobre el usuario que explotará el producto. A continuación sugerimos analizar los siguientes puntos:

NIVEL PROFESIONAL

El GPCASE no está dirigido a un tipo de profesional o técnico en particular. Sin embargo, bien es cierto que un profesional, dedicado al área de desarrollo, obtendrá mayores beneficios de GPCASE debido a que tiene un panorama más amplio de posibles aplicaciones que un técnico.

NIVEL DE EXPERIENCIA

La experiencia es un factor determinante en este caso, ya que el análisis del problema, la elección de posibles soluciones y la toma de decisiones se lleva a cabo de una forma más rápida y eficiente.

HABILIDAD

Respecto a la habilidad nosotros consideramos que existen personas que no importando su preparación, tienen una facilidad que se va acrecentando con la práctica para desenvolverse en cualquier terreno, esto significa que un usuario hábil puede obtener buenos resultados al utilizar el GPCASE.

VISION DEL NEGOCIO

Este es un concepto que cada vez cobra mayor importancia, se refiere a la capacidad que un individuo posee de observar, analizar y obtener conclusiones del movimiento en general del ambiente en el cuál se desenvuelve. Este individuo genera nuevas ideas, cambia enfoques de trabajo, agiliza procesos, en fin puede realmente revolucionar el negocio.

En conclusión, el GPCASE no está limitado a que sólo sea explotado por cierto tipo de usuarios, mas bien, puede ser explotado por cualquier tipo de usuario que de alguna forma esté involucrado al desarrollo de software. Pero, lo que es una realidad, es que a mayor preparación, mayor experiencia, mayor habilidad y mayor visión posea el usuario, mejores serán los resultados que se obtengan con GPCASE.

III.4 Elementos Gráficos Para Análisis y Diseño

Elementos

En esta sección mostraremos el Diagrama de Flujo de Datos, el Diagrama Entidad/Relación, el Diccionario de Datos y el Diagrama Estructurado que representan el sistema GPCASE.

Diagrama de Flujo de Datos (DFD)

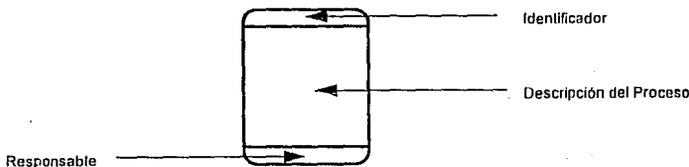
Los Diagramas de Flujo de Datos permiten obtener una visión global de los orígenes y destinos del sistema, pudiendo incluir partes manuales y automatizadas.

Elementos que componen el DFD son:

Flujo de Datos: Es una transferencia de información entre los elementos de diseño, se simboliza mediante una flecha horizontal o vertical con la punta indicando la dirección del flujo.



Proceso: Un proceso representa un cambio que sufren los datos, es necesario describir una función de cada proceso para una fácil referencia, una única identificación vinculada en lo posible, los procesos se simbolizan con un rectángulo vertical y dividido en tres áreas



Identificador : Es un número cuyo objetivo es identificar al proceso.

Descripción del proceso : Debe de hacerse de una manera sencilla, pero que indique lo que ese proceso realiza.

Responsable : Tiene el objetivo de llevar un control y una relación de quien realiza ese proceso.

Almacenamiento : Es el lugar donde se almacenan los datos durante el proceso, se simboliza de la siguiente manera .



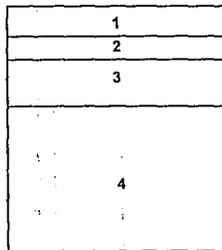
Entidad Externa : Representa una fuente o un destino de transacción, que son generalmente clases lógicas de cosa o personas, puede simbolizarse con un cuadrado con los lados superior e izquierdo de doble ancho para hacer resaltar el símbolo del resto del diagrama .



Diagrama de entidad relación : Es una representación gráfica de las entidades, sus atributos y las relaciones entre ellas, que sirven para modelar los archivos o bases de datos.

Los elementos que forman un diagrama de entidad-relación son :

Entidades : Las entidades representan cualquier objeto o evento, acerca del cual se almacena información .



1. Identificador de la entidad.

2.- Nombre de la entidad .

3.- Llaves

4.- Atributos

Relación : Son asociaciones entre entidades .

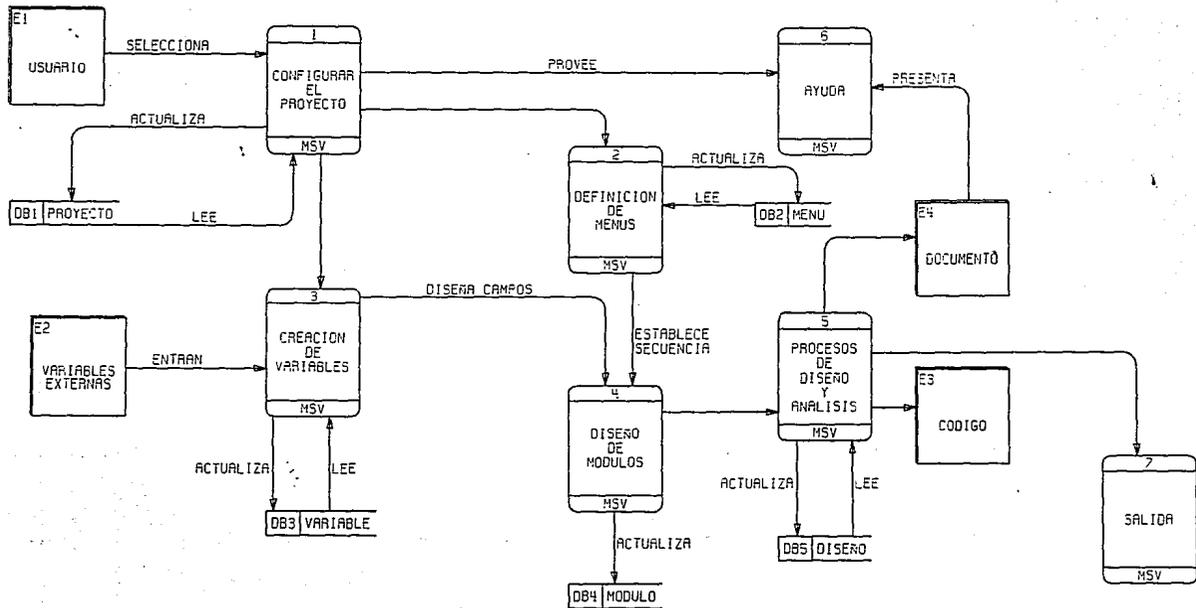


V.2.2. Diagrama de Flujo de Datos (DFD)

A continuación se presenta una serie de elementos que permitieron el análisis del sistema GPCASE, los cuales fueron obtenidos utilizando Pro Kit WorkBench.

Pro Kit generó los siguientes elementos de análisis:

- Diagrama General de Flujo de Datos
- Descripción de los diagramas
- Diagrama de cada Proceso
 - Configurar el Proyecto
 - Definición de Menús
 - Creación de Variables
 - Diseño de Módulos
 - Procesos de Diseño y Análisis
- Descripción de los Procesos
- Descripción de Entidades Externas



PROJECT :GPCASE
 VERSION :2
 WORKSTATION:MASTER
 MACRO NAME :DFD.ALL

Diagramas De GPCASE
 DATA FLOW DIAGRAM

PAGE 1
 DATE:17/06/93
 TIME:21:22:35
 USER:SYSTEM

COMPLETE LISTING OF ALL DIAGRAMS

-----DFD-----		WORKSTATION	LAST
-----ID-----	-----RELATED PROCESS NAME-----	--ID-- --OWNER- ACCESS	UPDATE
1	CONFIGURAR EL PROYECTO	MASTER SYSTEM W	17/06/1993
		ORIGIN	ORIGIN
		FROM	DATE
		DIAGRAM	17/06/1993

DESCRIPTION: *** RECORD ADDED VIA DIAGRAM EDITOR

-----CONTAINS EXTERNAL ENTITIES (EE)-----

-----ID-----	-----EXTERNAL ENTITY NAME-----
E1	USUARIO

TOTAL ITEMS : 1

-----CONTAINS PROCESSES (PR)-----

-----ID-----	-----PROCESS NAME-----	IS ALSO DFD
1	CONFIGURAR EL PROYECTO	Y
1.1	DESCRIPCION DEL PROYECTO	N
1.2	CONFIGURAR	N

TOTAL ITEMS : 3

-----CONTAINS DATA STORES (DS)-----

-----ID-----	-----DATA STORE NAME-----	-----ID-----	-----DATA STORE NAME-----
DB1	PROYECTO		

TOTAL ITEMS : 1

-----CONTAINS DATA FLOW OCCURENCES (DF)-----

-----DATA FLOW NAME-----	-----SOURCE-----	-----DESTINATION-----
PROPORCIONA	E1	1.1
CREAR PROYECTO	1.2	DB1
LEE	DB1	1.2
LEE	DB1	1.1

TOTAL ITEMS : 4

-----DFD-----		WORKSTATION	LAST
-----ID-----	-----RELATED PROCESS NAME-----	--ID-- --OWNER- ACCESS	UPDATE
2	DEFINICION DE MENUS	MASTER SYSTEM W	17/06/1993
		ORIGIN	ORIGIN
		FROM	DATE
		DIAGRAM	17/06/1993

DESCRIPTION: *** RECORD ADDED VIA DIAGRAM EDITOR

PROJECT :GPCASE
 VERSION :2
 WORKSTATION:MASTER
 MACRO NAME :DFD.ALL

Diagramas De GPCASE
 DATA FLOW DIAGRAM

PAGE 2
 DATE:17/06/93
 TIME:21:22:35
 USER:SYSTEM

COMPLETE LISTING OF ALL DIAGRAMS

-----DFD-----RELATED PROCESS NAME-----
 2 DEFINICION DE MENUS

-----CONTAINS EXTERNAL ENTITIES (EE)-----

NO EXTERNAL ENTITIES DEFINED

-----CONTAINS PROCESSES (PR)-----

ID	PROCESS NAME	IS ALSO DFD	
2	DEFINICION DE MENUS		Y
2.1	DISEÑO		N
2.2	MANTENIMIEN		N

TOTAL ITEMS : 3

-----CONTAINS DATA STORES (DS)-----

ID	DATA STORE NAME	ID	DATA STORE NAME
	DB2 MENU		

TOTAL ITEMS : 1

-----CONTAINS DATA FLOW OCCURRENCES (DF)-----

DATA FLOW NAME	SOURCE	DESTINATION
GENERA	2.1	DB2
ACTUALIZA	2.2	DB2
LEE	DB2	2.2

TOTAL ITEMS : 3

DFD	RELATED PROCESS NAME	WORKSTATION	OWNER	ACCESS	LAST UPDATE
3	CREACION DE VARIABLES	MASTER	SYSTEM	W	17/06/1993

ORIGIN FROM	ORIGIN DATE
DIAGRAM	17/06/1993

DESCRIPTION: *** RECORD ADDED VIA DIAGRAM EDITOR

-----CONTAINS EXTERNAL ENTITIES (EE)-----

ID	EXTERNAL ENTITY NAME
E2	VARIABLES EXTERNAS

TOTAL ITEMS : 1

-----CONTAINS PROCESSES (PR)-----

ID	PROCESS NAME	IS ALSO DFD	
3	CREACION DE VARIABLES		Y
3.1	EDICION		N
3.2	SELECCION DE VARIABLES		N
3.3	CARGA EXTERNA		N

PROJECT :GPCASE
 VERSION :2
 WORKSTATION:MASTER
 MACRO NAME :DFD.ALL

Diagramas De GPCASE
 DATA FLOW DIAGRAM

PAGE 3
 DATE:17/06/93
 TIME:21:22:35
 USER:SYSTEM

COMPLETE LISTING OF ALL DIAGRAMS

-----DFD-----RELATED PROCESS NAME-----
 3 CREACION DE VARIABLES

-----CONTAINS PROCESSES (PR)-----

TOTAL ITEMS : 4

-----CONTAINS DATA STORES (DS)-----

-----ID-----DATA STORE NAME----- ID-----DATA STORE NAME-----
 DB3 VARIABLE

TOTAL ITEMS : 1

-----CONTAINS DATA FLOW OCCURRENCES (DF)-----

DATA FLOW NAME	SOURCE	DESTINATION
GENERA	3.3	DB3
ACTUALIZA	3.2	DB3
ACTUALIZA	3.1	DB3
CONSULTA	DB3	3.2
CONSULTA	DB3	3.1
ENTRAN	E2	3.3

TOTAL ITEMS : 6

WORKSTATION	LAST UPDATE
MASTER SYSTEM W	17/06/1993

-----DFD-----RELATED PROCESS NAME-----
 4 DISEÑO DE MODULOS

ORIGIN FROM DIAGRAM	ORIGIN DATE
	17/06/1993

DESCRIPTION: *** RECORD ADDED VIA DIAGRAM EDITOR

-----CONTAINS EXTERNAL ENTITIES (EE)-----

NO EXTERNAL ENTITIES DEFINED

-----CONTAINS PROCESSES (PR)-----

ID	PROCESS NAME	IS ALSO DFD
4	DISEÑO DE MODULOS	Y
4.1	GENERACION DE OBJETOS	N
4.2	CONTROL DE HERRAMIENTA	N

TOTAL ITEMS : 3

PROJECT :GPCASE
 VERSION :2
 WORKSTATION:MASTER
 MACRO NAME :DFD.ALL

Diagramas De GPCASE
 DATA FLOW DIAGRAM

PAGE 4
 DATE:17/06/93
 TIME:21:22:35
 USER:SYSTEM

COMPLETE LISTING OF ALL DIAGRAMS

 DFD-----RELATED PROCESS NAME-----
 4 DISEÑO DE MODULOS

-----CONTAINS DATA STORES (DS)-----
 ID-----DATA STORE NAME-----ID-----DATA STORE NAME-----
 DB5 DISEÑO DB3 VARIABLE

TOTAL ITEMS : 2

-----CONTAINS DATA FLOW OCCURRENCES (DF)-----
 DATA FLOW NAME-----SOURCE-----DESTINATION-----
 ACTUALIZA 4.1 DB5
 PROPORCIONA DB3 4.1
 EMPLEA 4.1 4.2
 LEE DB5 4.1
 FACILITA 4.2 4.1

TOTAL ITEMS : 5

 WORKSTATION LAST
 --ID-- --OWNER- ACCESS UPDATE
 MASTER SYSTEM W 17/06/1993
 DFD-----RELATED PROCESS NAME-----
 5 PROCESOS DE DISEÑO Y ANALISIS
 ORIGIN ORIGIN
 FROM DATE
 DIAGRAM 17/06/1993

DESCRIPTION: *** RECORD ADDED VIA DIAGRAM EDITOR

-----CONTAINS EXTERNAL ENTITIES (EE)-----
 ID-----EXTERNAL ENTITY NAME-----
 E3 CODIGO

TOTAL ITEMS : 1

-----CONTAINS PROCESSES (PR)-----
 ID-----PROCESS NAME----- IS ALSO DFD
 5 PROCESOS DE DISEÑO Y ANALISIS Y
 5.1 EXPLOTACION N
 5.2 CONSULTA N
 5.3 REPRESENTAC GRAFICA N
 5.4 GENERACION REPORTES CODIGO N

TOTAL ITEMS : 5

-----CONTAINS DATA STORES (DS)-----
 ID-----DATA STORE NAME-----ID-----DATA STORE NAME-----
 DB5 DISEÑO

TOTAL ITEMS : 1

COMPLETE LISTING OF ALL DIAGRAMS

-----DFD-----RELATED PROCESS NAME-----
 5 PROCESOS DE DISEÑO Y ANALISIS

-----CONTAINS DATA FLOW OCCURRENCES (DF)-----
 -----DATA FLOW NAME-----SOURCE-----DESTINATION-----
 ACTUALIZA 5.1 DB5
 CONSULTA DB5 5.1
 PERMITE 5.1 5.2
 GENERACION DE CODIGO 5.4 E3

TOTAL ITEMS : 4

-----WORKSTATION-----LAST
 -----ID--- --OWNER- ACCESS UPDATE
 MASTER SYSTEM W 17/06/1993
 -----DFD-----DIAGRAM NAME-----
 DFD-GPCASE-2V OVERVIEW DATA FLOW DIAGRAM
 ORIGIN ORIGIN
 FROM DATE
 DICT-DIRECT 28/05/1994

DESCRIPTION:

-----CONTAINS EXTERNAL ENTITIES (EE)-----
 -----ID-----EXTERNAL ENTITY NAME-----
 E1 USUARIO
 E2 VARIABLES EXTERNAS
 E3 CODIGO
 E4 DOCUMENTO

TOTAL ITEMS : 4

-----CONTAINS PROCESSES (PR)-----
 -----ID-----PROCESS NAME----- IS ALSO DFD
 1 CONFIGURAR EL PROYECTO Y
 2 DEFINICION DE MENUS Y
 3 CREACION DE VARIABLES Y
 4 DISEÑO DE MODULOS Y
 5 PROCESOS DE DISEÑO Y ANALISIS Y
 6 AYUDA N
 7 SALIDA N

TOTAL ITEMS : 7

-----CONTAINS DATA STORES (DS)-----
 -----ID-----DATA STORE NAME-----ID-----DATA STORE NAME-----
 DB2 MENU DB1 PROYECTO
 DB4 MODULO DB3 VARIABLE
 DB5 DISEÑO

TOTAL ITEMS : 5

PROJECT :GPCASE
VERSION :2
WORKSTATION:MASTER
MACRO NAME :DFD.ALL

Diagramas De GPCASE
DATA FLOW DIAGRAM

PAGE 6
DATE:17/06/93
TIME:21:22:35
USER:SYSTEM

COMPLETE LISTING OF ALL DIAGRAMS

-----DFD-----DIAGRAM NAME-----
DFD-GPCASE-2V OVERVIEW DATA FLOW DIAGRAM

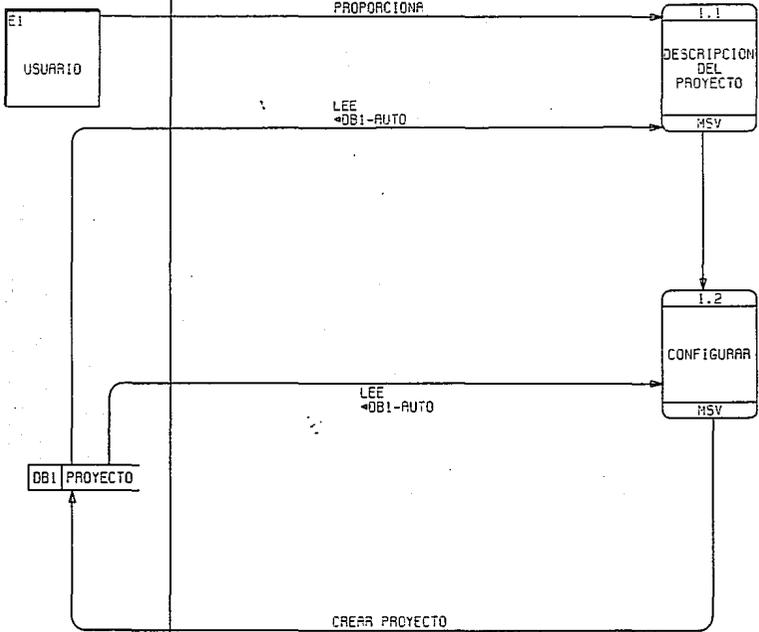
-----CONTAINS DATA FLOW OCCURRENCES (DF)-----

-----DATA FLOW NAME-----	-----SOURCE-----	-----DESTINATION-----
ACTUALIZA	1	DB1
ACTUALIZA	3	DB3
ACTUALIZA	2	DB2
ACTUALIZA	5	DB5
ACTUALIZA	4	DB4
SELECCIONA	E1	1
ENTRAN	E2	3
LEE	DB3	3
LEE	DB1	1
LEE	DB2	2
LEE	DB5	5
DISEÑA CAMPOS	3	4
PROVEE	1	6
ESTABLECE SECUENCIA	2	4
PRESENTA	E4	6

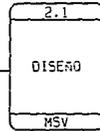
TOTAL ITEMS : 15

TOTAL NUMBER OF DIAGRAMS ON REPORT : 6

1 CONFIGURAR EL PROYECTO



2 DEFINICION DE MENUS



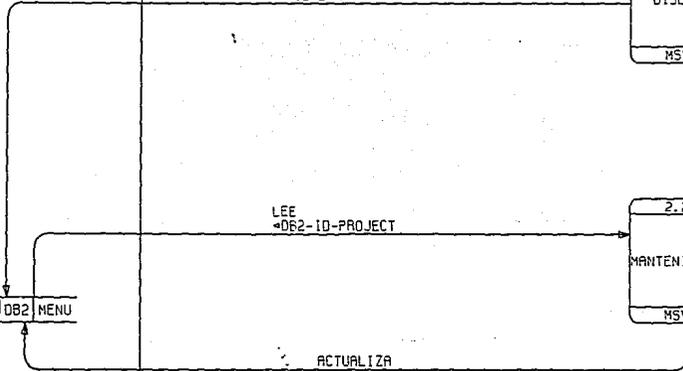
GENERA



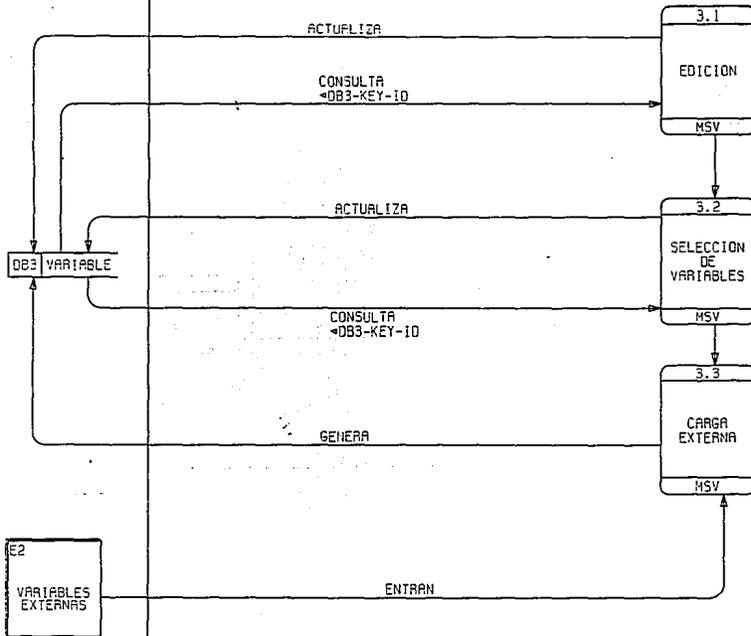
LEE
DB2-10-PROJECT

ACTUALIZA

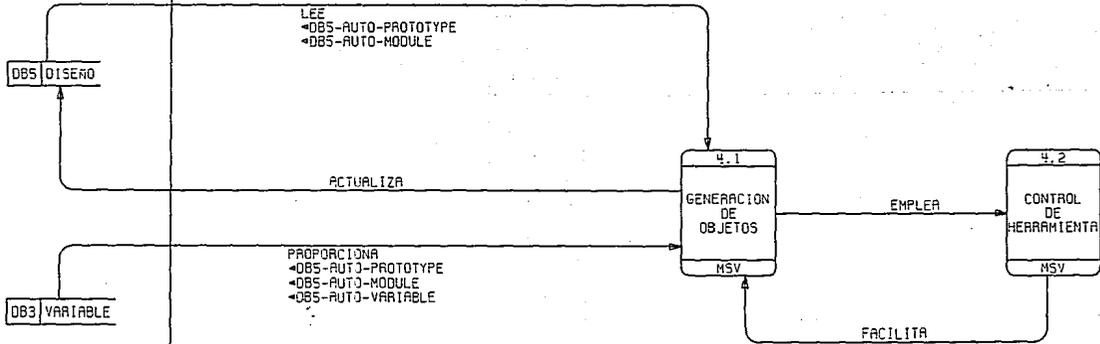
DB2 | MENU



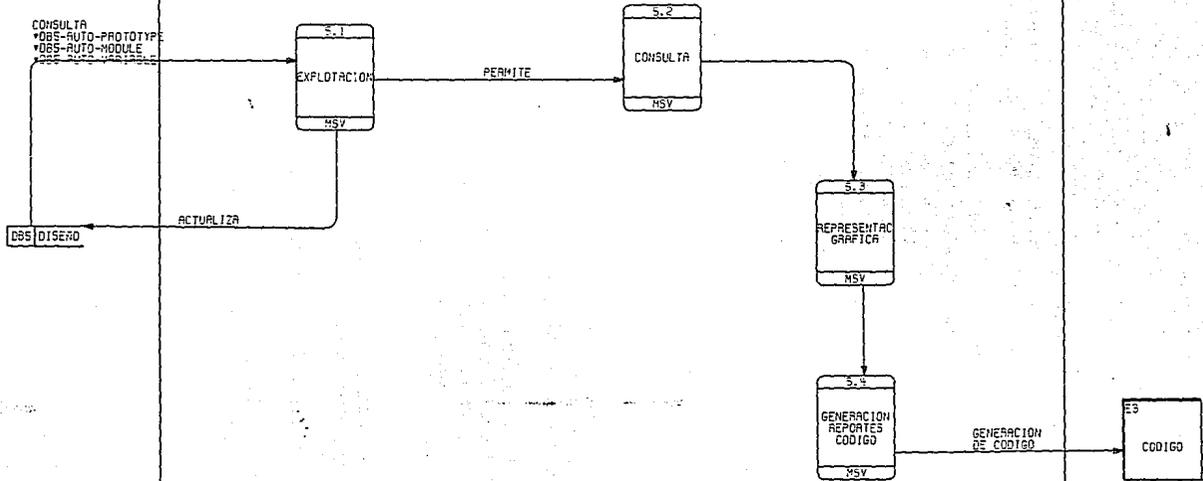
3 CREACION DE VARIABLES



4 DISEÑO DE MODULOS



5
PROCESOS DE DISEÑO Y ANALISIS



PROJECT :GPCASE
VERSION :2
WORKSTATION:MASTER
MACRO NAME :PR.ALL

Diagramas De GPCASE
PROCESS ATTRIBUTES

PAGE 1
DATE:21/06/93
TIME:21:43:14
USER:SYSTEM

COMPLETE LISTING OF ALL PROCESSES

```
=====
                                WORKSTATION
                                ---ID--- --OWNER- ACCESS
                                MASTER SYSTEM W
                                LAST
                                UPDATE
                                21/06/1993
-----ID----- PROCESS NAME-----
1             CONFIGURAR EL PROYECTO
--IMPL BY-- TYPE CLASS FORM VOLATILITY EXEC TIME   ORIGIN   ORIGIN
MSV          CLASS FORM VOLATILITY FREQ INTERVAL FROM     DATE
                                DICT-DIRECT 21/06/1993
----- PROCESS NARRATIVE-----
CONTROLA LA RELACION DE PROYECTOS ASI COMO LA CONFIGURACION Y DESCRIPCION DE ESTOS.
=====
```

```
=====
                                WORKSTATION
                                ---ID--- --OWNER- ACCESS
                                MASTER SYSTEM W
                                LAST
                                UPDATE
                                21/06/1993
-----ID----- PROCESS NAME-----
1.1          DESCRIPCION DEL PROYECTO
--IMPL BY-- TYPE CLASS FORM VOLATILITY EXEC TIME   ORIGIN   ORIGIN
MSV          CLASS FORM VOLATILITY FREQ INTERVAL FROM     DATE
                                DICT-DIRECT 21/06/1993
----- PROCESS NARRATIVE-----
EL USUARIO PROPORCIONA LA DOCUMENTACION NECESARIA PARA EXPRESAR CLARAMENTE
EL OBJETIVO DEL PROYECTO, MEDIANTE UN SISTEMA DE ARCHIVOS (HASTA 1 KB POR ARCHIVO)
=====
```

```
=====
                                WORKSTATION
                                ---ID--- --OWNER- ACCESS
                                MASTER SYSTEM W
                                LAST
                                UPDATE
                                21/06/1993
-----ID----- PROCESS NAME-----
1.2          CONFIGURAR
--IMPL BY-- TYPE CLASS FORM VOLATILITY EXEC TIME   ORIGIN   ORIGIN
MSV          CLASS FORM VOLATILITY FREQ INTERVAL FROM     DATE
                                DICT-DIRECT 21/06/1993
----- PROCESS NARRATIVE-----
ESTABLECIMIENTO DE LOS PARAMETROS DE CADA UNO DE LOS ARCHIVOS: RESPONSABLES DEL
SISTEMA, TIEMPOS DE DISEÑO Y CLAVES DE ACCESO.
=====
```

```
=====
                                WORKSTATION
                                ---ID--- --OWNER- ACCESS
                                MASTER SYSTEM W
                                LAST
                                UPDATE
                                21/06/1993
-----ID----- PROCESS NAME-----
2           DEFINICION DE MENUS
--IMPL BY-- TYPE CLASS FORM VOLATILITY EXEC TIME   ORIGIN   ORIGIN
MSV          CLASS FORM VOLATILITY FREQ INTERVAL FROM     DATE
                                DICT-DIRECT 21/06/1993
----- PROCESS NARRATIVE-----
PERMITE EL DISEÑO Y MANTENIMIENTO DE LOS MENUS DEL PROTOTIPO.
=====
```

COMPLETE LISTING OF ALL PROCESSES

-----ID-----		-----PROCESS NAME-----			WORKSTATION			LAST
					--ID--	--OWNER-	ACCESS	UPDATE
					MASTER	SYSTEM	W	21/06/1993
2.1		DISEÑO						
--IMPL BY--	TYPE	CLASS	FORM	VOLATILITY	EXEC FREQ	TIME INTERVAL	ORIGIN FROM	ORIGIN DATE
MSV							DICT-DIRECT	21/06/1993
-----PROCESS NARRATIVE-----								
CREACION DEL DISEÑO DEL MENU DEL PROTOTIPO.								
-----ID-----		-----PROCESS NAME-----			WORKSTATION			LAST
					--ID--	--OWNER-	ACCESS	UPDATE
					MASTER	SYSTEM	W	21/06/1993
2.2		MANTENIMIEN						
--IMPL BY--	TYPE	CLASS	FORM	VOLATILITY	EXEC FREQ	TIME INTERVAL	ORIGIN FROM	ORIGIN DATE
MSV							DICT-DIRECT	21/06/1993
-----PROCESS NARRATIVE-----								
REALIZA BASICAMENTE LAS FUNCIONES DE ACTUALIZACION, INSERCIÓN Y ELIMINACION DE MENUS.								
-----ID-----		-----PROCESS NAME-----			WORKSTATION			LAST
					--ID--	--OWNER-	ACCESS	UPDATE
					MASTER	SYSTEM	W	21/06/1993
3		CREACION DE VARIABLES						
--IMPL BY--	TYPE	CLASS	FORM	VOLATILITY	EXEC FREQ	TIME INTERVAL	ORIGIN FROM	ORIGIN DATE
MSV							DICT-DIRECT	21/06/1993
-----PROCESS NARRATIVE-----								
CONTROL DE LA CARGA EXTERNA, EDICION Y DEFINICION DEL BANCO DE VARIABLES DEL PROTOTIPO								
-----ID-----		-----PROCESS NAME-----			WORKSTATION			LAST
					--ID--	--OWNER-	ACCESS	UPDATE
					MASTER	SYSTEM	W	21/06/1993
3.1		EDICION						
--IMPL BY--	TYPE	CLASS	FORM	VOLATILITY	EXEC FREQ	TIME INTERVAL	ORIGIN FROM	ORIGIN DATE
MSV							DICT-DIRECT	21/06/1993
-----PROCESS NARRATIVE-----								
SE REALIZA LA PARAMETRIZACION Y SE PROPORCIONAN LAS CARACTERISTICAS DE CADA VARIABLE UTILIZADA.								

COMPLETE LISTING OF ALL PROCESSES

-----ID-----		-----PROCESS NAME-----			WORKSTATION			LAST	
					---ID---	--OWNER-	ACCESS	UPDATE	
					MASTER	SYSTEM	W	21/06/1993	
3.2		SELECCION DE VARIABLES							
--IMPL BY--	TYPE	CLASS	FORM	VOLATILITY	EXEC	TIME	ORIGIN	ORIGIN	
MSV					FREQ	INTERVAL	FROM	DATE	
							DICT-DIRECT	21/06/1993	
-----PROCESS NARRATIVE-----									
PARA DECIDIR QUE VARIABLES SE UTILIZARAN EN EL PROYECTO DEL BANCO DEL SISTEMA ACTUAL.									
-----ID-----		-----PROCESS NAME-----			WORKSTATION			LAST	
					---ID---	--OWNER-	ACCESS	UPDATE	
					MASTER	SYSTEM	W	21/06/1993	
3.3		CARGA EXTERNA							
--IMPL BY--	TYPE	CLASS	FORM	VOLATILITY	EXEC	TIME	ORIGIN	ORIGIN	
MSV					FREQ	INTERVAL	FROM	DATE	
							DICT-DIRECT	21/06/1993	
-----PROCESS NARRATIVE-----									
PERMITE EL INGRESO DE VARIABLES EXTERNAS AL SISTEMA.									
-----ID-----		-----PROCESS NAME-----			WORKSTATION			LAST	
					---ID---	--OWNER-	ACCESS	UPDATE	
					MASTER	SYSTEM	W	21/06/1993	
4		DISEÑO DE MODULOS							
--IMPL BY--	TYPE	CLASS	FORM	VOLATILITY	EXEC	TIME	ORIGIN	ORIGIN	
MSV					FREQ	INTERVAL	FROM	DATE	
							DICT-DIRECT	21/06/1993	
-----PROCESS NARRATIVE-----									
DISEÑO Y ACTUALIZACION DE LOS OBJETOS CAMPO, MARCO Y ETIQUETA.									
-----ID-----		-----PROCESS NAME-----			WORKSTATION			LAST	
					---ID---	--OWNER-	ACCESS	UPDATE	
					MASTER	SYSTEM	W	21/06/1993	
4.1		GENERACION DE OBJETOS							
--IMPL BY--	TYPE	CLASS	FORM	VOLATILITY	EXEC	TIME	ORIGIN	ORIGIN	
MSV					FREQ	INTERVAL	FROM	DATE	
							DICT-DIRECT	21/06/1993	
-----PROCESS NARRATIVE-----									
PERMITE AL USUARIO EL MANEJO DE LAS ETIQUETAS CAMPOS, MARCOS, ETC PARA LA GENERACION DE UN REPORTE O UNA PANTALLA DE CAPTURA.									

PROJECT :GPCASE
 VERSION :2
 WORKSTATION:MASTER
 MACRO NAME :PR.ALL

Diagramas De GPCASE
 PROCESS ATTRIBUTES

PAGE 4
 DATE:21/06/93
 TIME:21:43:14
 USER:SYSTEM

COMPLETE LISTING OF ALL PROCESSES

```
=====
                                WORKSTATION
                                --ID-- --OWNER- ACCESS
                                MASTER  SYSTEM  W
-----ID----- PROCESS NAME-----
4.2      CONTROL DE HERRAMIENTA
--IMPL BY-- TYPE   CLASS FORM VOLATILITY EXEC TIME   ORIGIN   ORIGIN
MSV      --          --          --          FREQ  INTERVAL FROM     DATE
--          --          --          --          --          --          --
--          --          --          --          --          --          --
                                DICT-DIRECT 21/06/1993
-----
                                PROCESS NARRATIVE-----
AQUI SE ENCUENTRAN UN CONJUNTO DE HERRAMIENTAS QUE FACILITAN EL DISEÑO DE LOS
OBJETOS DEL PROTOTIPO.
=====
```

```
=====
                                WORKSTATION
                                --ID-- --OWNER- ACCESS
                                MASTER  SYSTEM  W
-----ID----- PROCESS NAME-----
5        PROCESOS DE DISEÑO Y ANALISIS
--IMPL BY-- TYPE   CLASS FORM VOLATILITY EXEC TIME   ORIGIN   ORIGIN
MSV      --          --          --          FREQ  INTERVAL FROM     DATE
--          --          --          --          --          --          --
--          --          --          --          --          --          --
                                DICT-DIRECT 21/06/1993
-----
                                PROCESS NARRATIVE-----
CONFIGURACION DE LOS MODULOS PANTALLA Y REPORTE.
=====
```

```
=====
                                WORKSTATION
                                --ID-- --OWNER- ACCESS
                                MASTER  SYSTEM  W
-----ID----- PROCESS NAME-----
5.1      EXPLOTACION
--IMPL BY-- TYPE   CLASS FORM VOLATILITY EXEC TIME   ORIGIN   ORIGIN
MSV      --          --          --          FREQ  INTERVAL FROM     DATE
--          --          --          --          --          --          --
--          --          --          --          --          --          --
                                DICT-DIRECT 21/06/1993
-----
                                PROCESS NARRATIVE-----
SE REALIZAN LAS OPERACIONES DE CONTROL DE MODULOS, ENLACES CON MENUS U OTROS
MODULOS, SIMULACION Y SEGURIDAD DEL PROTOTIPO.
=====
```

```
=====
                                WORKSTATION
                                --ID-- --OWNER- ACCESS
                                MASTER  SYSTEM  W
-----ID----- PROCESS NAME-----
5.2      CONSULTA
--IMPL BY-- TYPE   CLASS FORM VOLATILITY EXEC TIME   ORIGIN   ORIGIN
MSV      --          --          --          FREQ  INTERVAL FROM     DATE
--          --          --          --          --          --          --
--          --          --          --          --          --          --
                                DICT-DIRECT 21/06/1993
-----
                                PROCESS NARRATIVE-----
PROPORCIONA AL USUARIO UNA VISION GENERAL O TAN DETALLADA COMO EL LO DESEE, YA
QUE PUEDE VER LA ORGANIZACION DE LOS MENUS DE LOS MODULOS Y DE LAS VARIABLES, ETC.
=====
```

COMPLETE LISTING OF ALL PROCESSES

ID		PROCESS NAME			WORKSTATION			LAST
--ID--		-----			--ID--	--OWNER-	ACCESS	UPDATE
					MASTER	SYSTEM	W	21/06/1993
5.3		REPRESENTAC GRAFICA						
--IMPL BY--	TYPE	CLASS	FORM	VOLATILITY	EXEC FREQ	TIME INTERVAL	ORIGIN FROM	ORIGIN DATE
MSV							DICT-DIRECT	21/06/1993
-----PROCESS NARRATIVE-----								
PERMITE UNA VISUALIZACION GRAFICA DEL PROTOTIPO, ES DECIR EL FLUJO DE LOS MODULOS O PROCESOS.								

ID		PROCESS NAME			WORKSTATION			LAST
--ID--		-----			--ID--	--OWNER-	ACCESS	UPDATE
					MASTER	SYSTEM	W	21/06/1993
5.4		GENERACION REPORTES CODIGO						
--IMPL BY--	TYPE	CLASS	FORM	VOLATILITY	EXEC FREQ	TIME INTERVAL	ORIGIN FROM	ORIGIN DATE
MSV							DICT-DIRECT	21/06/1993
-----PROCESS NARRATIVE-----								
PROPORCIONA AL USUARIO UNA DOCUMENTACION COMPLETA DEL SISTEMA, ASI COMO TAMBIEN EL CODIGO EN LENGUAJE C DEL PROTOTIPO QUE DISEÑO.								

ID		PROCESS NAME			WORKSTATION			LAST
--ID--		-----			--ID--	--OWNER-	ACCESS	UPDATE
					MASTER	SYSTEM	W	21/06/1993
6		AYUDA						
--IMPL BY--	TYPE	CLASS	FORM	VOLATILITY	EXEC FREQ	TIME INTERVAL	ORIGIN FROM	ORIGIN DATE
MSV							DICT-DIRECT	21/06/1993
-----PROCESS NARRATIVE-----								
PERMITE CONSULTAR LOS PROCEDIMIENTOS Y FUNCIONAMIENTO DEL SISTEMA ASI COMO LOS DOCUMENTOS CREADOS POR EL USUARIO.								

ID		PROCESS NAME			WORKSTATION			LAST
--ID--		-----			--ID--	--OWNER-	ACCESS	UPDATE
					MASTER	SYSTEM	W	21/06/1993
7		SALIDA						
--IMPL BY--	TYPE	CLASS	FORM	VOLATILITY	EXEC FREQ	TIME INTERVAL	ORIGIN FROM	ORIGIN DATE
MSV							DICT-DIRECT	21/06/1993
-----PROCESS NARRATIVE-----								
SALIDA DEL SISTEMA								

TOTAL NUMBER OF PROCESSES ON REPORT : 21
 TOTAL NUMBER OF PROCESSES ON PROJECT REPOSITORY VERSION 2 : 21

PROJECT :GPCASE
VERSION :2
WORKSTATION:MASTER
MACRO NAME :EE.ALL

Diagramas De GPCASE
EXTERNAL ENTITY

PAGE 1
DATE:19/06/93
TIME:20:54:02
USER:SYSTEM

COMPLETE LISTING OF ALL EXTERNAL ENTITIES

```
=====
----ID----  -----SIMPLE EXTERNAL ENTITY NAME-----  WORKSTATION  LAST
E1          USUARIO                               ID OWNER- ACCESS  UPDATE
          ---TYPE---  ---CLASS---  ORIGIN          ORIGIN
          SIMPLE     ***N/A***   FROM            DATE
          SE ENCARGA DEL CONTROL Y MANTENIMIENTO DE LOS PROYECTOS.  19/06/1993
DESCRIPTION:
PRIMARY REPRESENTATIVE :          ALTERNATE :
```

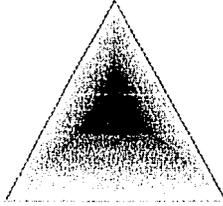
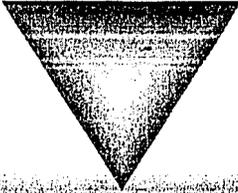
```
=====
----ID----  -----SIMPLE EXTERNAL ENTITY NAME-----  WORKSTATION  LAST
E2          VARIABLES EXTERNAS                     ID OWNER- ACCESS  UPDATE
          ---TYPE---  ---CLASS---  ORIGIN          ORIGIN
          SIMPLE     ***N/A***   FROM            DATE
          SON AQUELLAS GENERADAS POR OTROS SISTEMAS Y QUE SON ÚTILES  19/06/1993
          PARA EL NUEVO PROYECTO.
DESCRIPTION:
PRIMARY REPRESENTATIVE :          ALTERNATE :
```

```
=====
----ID----  -----SIMPLE EXTERNAL ENTITY NAME-----  WORKSTATION  LAST
E3          CODIGO                               ID OWNER- ACCESS  UPDATE
          ---TYPE---  ---CLASS---  ORIGIN          ORIGIN
          SIMPLE     ***N/A***   FROM            DATE
          ES AQUEL QUE GENERARÁ EL PRODUCTO Y QUE CONTIENE LA ESTRUCTURA Y  19/06/1993
          FUNCIONALIDAD GENERAL DEL FUTURO PROYECTO QUE COMPLETARÁ EL USUARIO.
DESCRIPTION:
PRIMARY REPRESENTATIVE :          ALTERNATE :
```

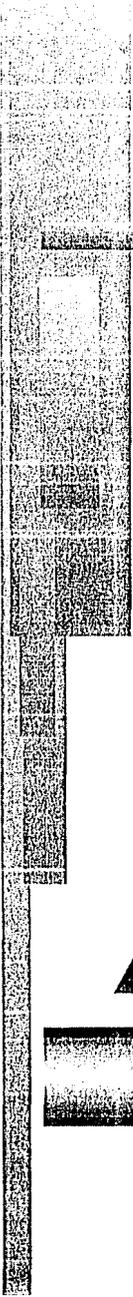
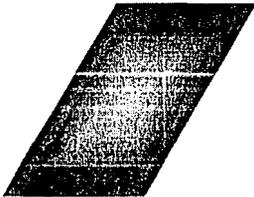
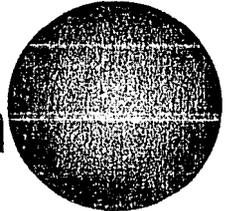
```
=====
----ID----  -----SIMPLE EXTERNAL ENTITY NAME-----  WORKSTATION  LAST
E4          DOCUMENTO                             ID OWNER- ACCESS  UPDATE
          ---TYPE---  ---CLASS---  ORIGIN          ORIGIN
          SIMPLE     ***N/A***   FROM            DATE
          ES UN CONJUNTO DE ELEMENTOS QUE PROPORCIONAN INFORMACIÓN  19/06/1993
          REFERENTE AL PROYECTO EN DISEÑO.
DESCRIPTION:
PRIMARY REPRESENTATIVE :          ALTERNATE :
```

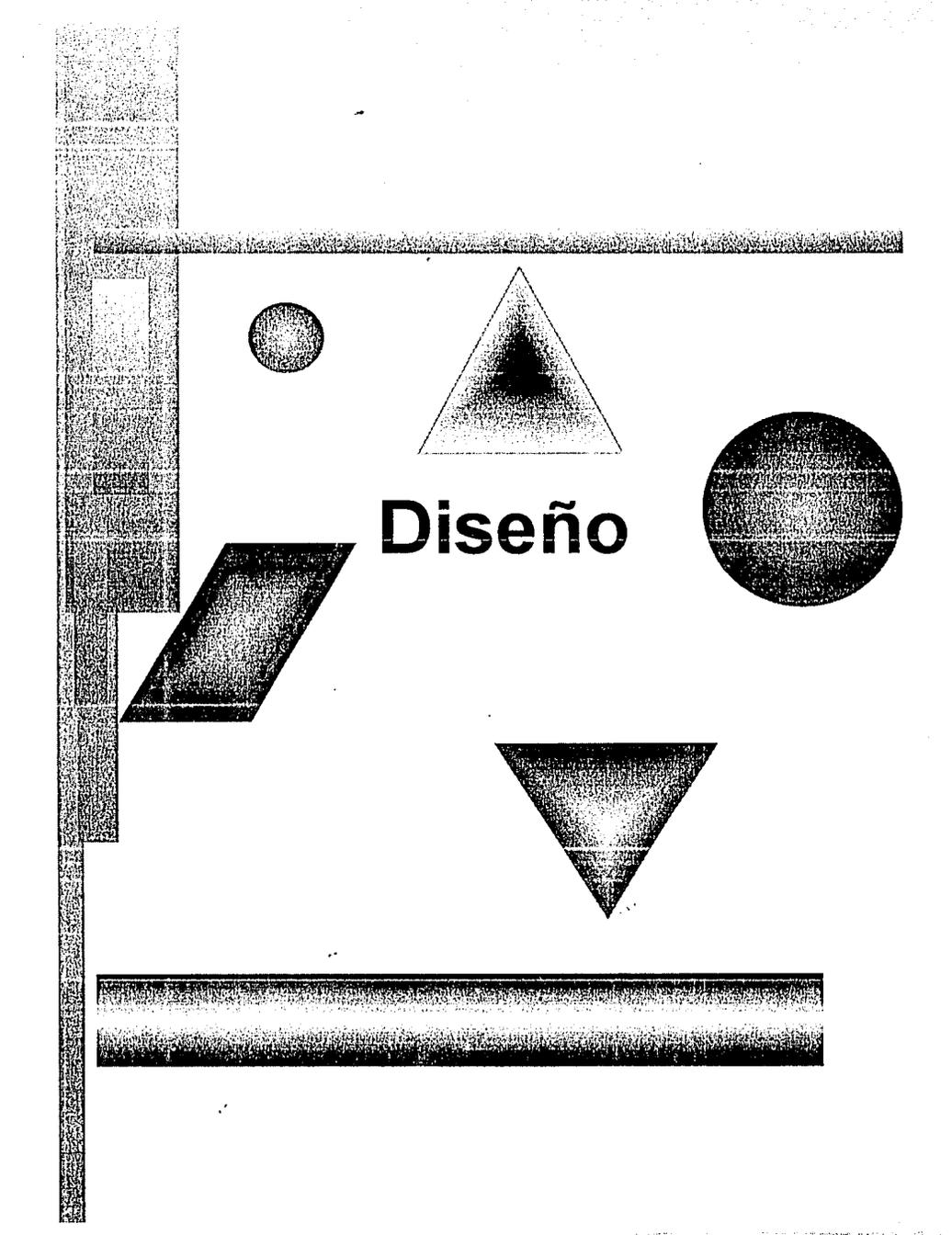
```
=====
----ID----  -----SIMPLE EXTERNAL ENTITY NAME-----
PRIMARY REPRESENTATIVE :          ALTERNATE :
```

```
=====
TOTAL NUMBER OF EXTERNAL ENTITIES ON REPORT          : 4
TOTAL NUMBER OF EXTERNAL ENTITIES ON PROJECT REPOSITORY VERSION 2 : 4
=====
```



Planeación



An abstract geometric composition featuring a vertical bar on the left, a horizontal bar at the top, and another horizontal bar at the bottom. The central area contains a circle, an upward-pointing triangle, a parallelogram, and a downward-pointing triangle. A large circle is positioned on the right side. The word "Diseño" is centered in the composition.

Diseño

V.1 NOTACION HUNGARA

Uno de los factores más importantes en el momento de empezar a desarrollar una aplicación de software es establecer patrones y convenciones de desarrollo.

Esto implica primordialmente establecer en forma general los módulos que soportarán ciertas tareas específicas y que cada integrante se apegue lo más posible a los patrones de desarrollo establecidos. Para este desarrollo uno de los estándares de trabajo fundamentales fué la adopción de la Notación Húngara para la generación de código.

La Notación Húngara es una convención sobre la forma en que se debe escribir código, de tal forma que cada nombre de variable, constante, función, etc, brinde por sí mismo con su sola lectura un concepto lo más cercano y claro posible de lo que encierra ese código, evitando al máximo que cuando se hacen revisiones globales, otro programador necesite ver a detalle cada función cada vez que existe algún problema o cada vez que se quiere readecuar un código.

¿ Por qué se adoptó la Notación Húngara y no otra convención ?. Bien, las razones son las siguientes:

- La Notación Húngara es una convención en la cual se basa actualmente el código de librerías de funciones que pertenecen a Microsoft Windows Software Development Kit.
- La Notación Húngara es de fácil asimilación al empezar a utilizarla.
- El grupo de desarrollo tiene ya experiencia con el manejo de la Notación Húngara.

Es importante comentar que para este desarrollo se adicionaron convenciones que permitiesen manejar ciertos requerimientos del diseño. Un ejemplo es que en la Notación Húngara no se utilizan subguiones a forma de espacios cuando un nombre descriptivo requiere dos palabras para autodescribirse. Sin embargo, en el caso de variables globales se utilizaron exclusivamente letras mayúsculas y el subguión debe utilizarse como separador a modo de espacio para facilitar la lectura.

A continuación se describe la Notación de código utilizada:

Variables Locales:

Estas deben iniciar con una o dos letras dependiendo el tipo de variable que sean, la letra o letras deben estar en minúsculas, en seguida vendrá el nombre de la variable empezando con letra mayúscula y el resto en minúsculas, si la variable requiere necesariamente más de una palabra para autodescribirse entonces se utilizará una letra mayúscula al iniciar cada palabra, las restantes estarán en minúsculas. Por ejemplo:

La variable es Numero Flotante l y es de tipo float, la declaración será:

```
float fNumeroFlotante l ; // f de float
```

La variable es Valor Pequeño Azul y es de tipo short , la declaración será:
 short shValorPequenoAzul ; // sh de short

La variable es un caracter LetraBeta, la declaración será:

```
char cLetraBeta ; // c de char
```

La variable es un arreglo de cadenas de caracteres Indice de Color, la declaración será:

```
char sIndiceColor[20][30] ; // s de string
ó
char szIndiceColor[20][30] ; // sz de string null ( tipo Windows)
```

La variable es una estructura de un float y dos enteros, se llama Matriz Real , la declaración será:

```
struct stMatrizReal {
    float fNumeroFlotante ;
    int iEnteroNumero1 ;
    int iEnteroNumero2 ;
} stMatrizReal ; // st de struct
```

Variables Globales:

Estas deben iniciar con una o dos letras minúsculas dependiendo el tipo de variable que sean, seguidamente vendrá el nombre de la variable utilizando sólo letras mayúsculas y a modo de espacios intercalar subguiones como se indica a continuación. Por ejemplo:

La variable es global Flotante Global 1 y es de tipo float, la declaración será:

```
float fFLOTANTE_GLOBAL1 ; // f de float
```

La variable es global de tipo caracter Bandera de Status, la declaración será:

```
char cBANDERA_STATUS ; // c de char
```

La variable es un arreglo global de caracteres Nombre de Proyectos, la declaración será:

```
char szNOMBRE_PROYECTO[10][40] ; // sz de string null (cadena terminada en nulo)
```

La siguiente es una estructura global llamada Mayor que dará lugar a dos variables globales llamadas Estructura Global Superior y Estructura Global Inferior. Ambas variables contienen un arreglo de caracteres llamado Caracteres Válidos; un arreglo de enteros largos

llamado Números Mayores, una variable de doble precisión llamada Valor Exacto, una variable de tipo *Double Word* utilizada en Windows llamada Índice de color y una variable tipo *Long Pointer String* de Windows llamada Cadena Apuntador; la declaración será:

```
struct stESTRUCTURA_MAYOR {
    char        szCaracteresValidos[25] ;
    long int    liNumerosMayores ;
    double      dValorExacto ;
    DWORD      dwIndiceColor ;
    LPSTR       lpCadenaApuntador[10] ;
} stESTRUCTURA_GLOBAL_SUPERIOR ,
  stESTRUCTURA_GLOBAL_INFERIOR ;      // st de struct
```

Funciones:

Las funciones tienen una estructura muy sencilla, se antepone al nombre de la función las letras "fn" de función, le sigue el nombre de la función con una letra mayúscula y las letras restantes con minúsculas, utilizando mayúsculas por cada inicio de palabra en el nombre de la función, en este caso no se deben usar subguiones para semejar espacios, por ejemplo:

Una función que Convierte Caracteres, la declaración es:

```
int fnConvierteCaracteres ( void ) {
    // código
    // código
};
```

Una función que crea el Menú del Prototipo para Simulación, la declaración es:

```
BOOL fnCreaMenuPrototipoSimulacion ( void ) {
    // código
    // código
};
```

Constantes:

En este desarrollo las constantes son valores que serán utilizados en todos los módulos y mantendrán valores globales para registrar cualquier evento en cualquier proceso, sencillamente se escribe el nombre de la constante en letras mayúsculas, utilizando subguiones a modo de espacios, sin antecederlas de alguna letra, por ejemplo:

Las constantes máximo de columnas, mínimo de menús y simulación de prototipo se verían así:

```
#define    MAXIMO_COLUMNAS    80
#define    MINIMO_MENU      1
#define    SIMULACION_PROTOTIPO    11
```

V.2 ORGANIZACION EMPLEADA PARA EL DESARROLLO

Una vez definido el objetivo y sentido del desarrollo de GPCASE se estableció el plan de trabajo para cada integrante, esto en función de las características del sistema, dividiendo entonces el sistema en módulos para describir los aspectos generales de la aplicación.

Aquí es donde radica la fortaleza de la modularización en cuanto a desarrollo integral y la adopción de una misma metodología para trabajar bajo el mismo patrón al desarrollar la aplicación. Conforme se avanza en la consecución de aspectos generales, se van definiendo los convenios de conexión para el futuro enlace de módulos, en su momento se definen las interfaces y conexiones correspondientes y se sigue trabajando por separado.

Terminado un módulo y previstas las interfaces y conexiones necesarias, se procede a enlazar uno a uno cada módulo en una aplicación prototipo, cada módulo que se fué agregando a esta aplicación fué verificado a detalle, compilado y ligado, finalmente al estar de acuerdo en la consistencia de la aplicación fué liberado y esto fué repetido con cada uno de los módulos restantes.

MODULOS

En un proyecto de desarrollo de aplicaciones Windows, definitivamente se debe modularizar el código, para ello se establecieron los siguientes módulos:

GLOBALES.H **GLOB_EXT.H**

Es el control de todas las variables globales, en este caso se debe tener un archivo que contenga la declaración única de una variable global (GLOBALES.H) y otro que debe incluirse en todos los demás archivos de código para que se haga referencia a la misma variable brindándole consistencia al código, en este caso es el archivo GLOB_EXT.H.

GP_BTRV.C

Hay un módulo separado para el manejo de bases de datos, como es el caso de Btrieve.

GP_CODE.C

Un módulo exclusivo para la parte de generación de código, esta es una tarea delicada porque se estructura código fuente a partir del diseño del usuario y esto causa la generación de un archivo.

GP_DEF.H

Aquí se mantienen definiciones de constantes y prototipos de las funciones que se construyeron para la aplicación, se encuentra una explicación detallada en la Referencia Técnica.

GP_DL.G.C**GP_DLGM.C**

Estos dos módulos controlan todos los procesos, mensajes y operaciones que se hagan sobre los *dialogs* de la aplicación, se tienen dos porque al ir generando más código en un solo archivo, éste empezó a crear sobrecarga y problemas de compilación, por ello se decidió manejar por separado los *dialogs* de control de menús en GP_DLGM.C y los restantes en GP_DL.G.C.

GP_MAIN.C Crea la definición y características de cada una de las ventanas de GPCASE.

GP_MSG.C

En Windows todo se controla vía mensajes y aquí se procesan los de la ventana principal.

GP_SSTBF.C Funciones que utilizan librerías gráficas de SpreadSheet.

GP_UDCD.C

GPCASE se apoya sobremanera en la interfaz gráfica de Windows y uno de los elementos más interesantes del producto es la posibilidad de visualizar un diagrama esquemático del flujo del diseño del prototipo, por ello se decidió tener este módulo separado, porque es netamente gráfico.

GP_UDF.C**GP_UDFM.C**

La construcción de funciones forma gran parte del desarrollo de GPCASE y asimismo se tuvo que separar lo que fueron funciones para manejo de menús (GP_UDFM.C) y otro donde se controlarían todas las funciones restantes (GP_UDF.C).

GP_WPROC.C

Definición de las funciones de otras ventanas que integran el sistema GPCASE.

GPCASE.C**GPCASE.DEF****GPCASE.H****GPCASE.RC**

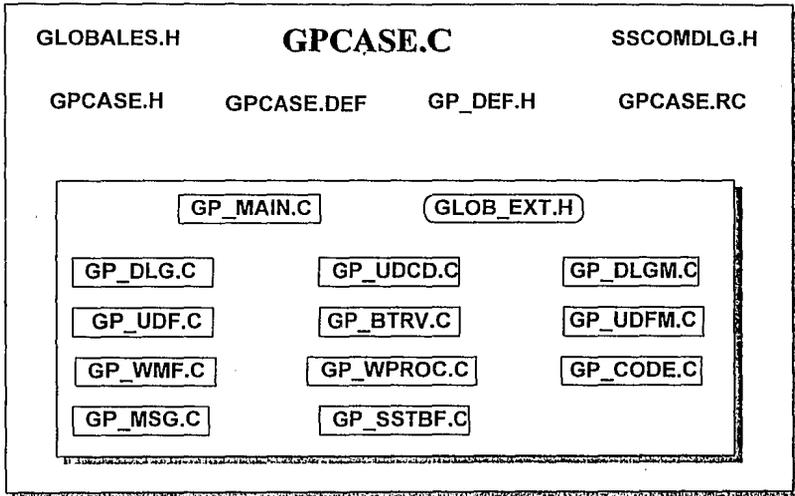
Estos cuatro archivos son indispensables para la aplicación, en ellos se controla la ventana principal y sus características, se tienen las definiciones de recursos, tipo de modularización, etc.

SSCOMDLG.H

Es en donde residen los *CommonDialogs* que son aquellos tipos de *dialogs* de uso común para las aplicaciones Windows.

Todos los módulos anteriores mantienen una interacción que se ejemplifica en el siguiente esquema:

V.2.1 INTERACCION MODULAR DE GPCASE



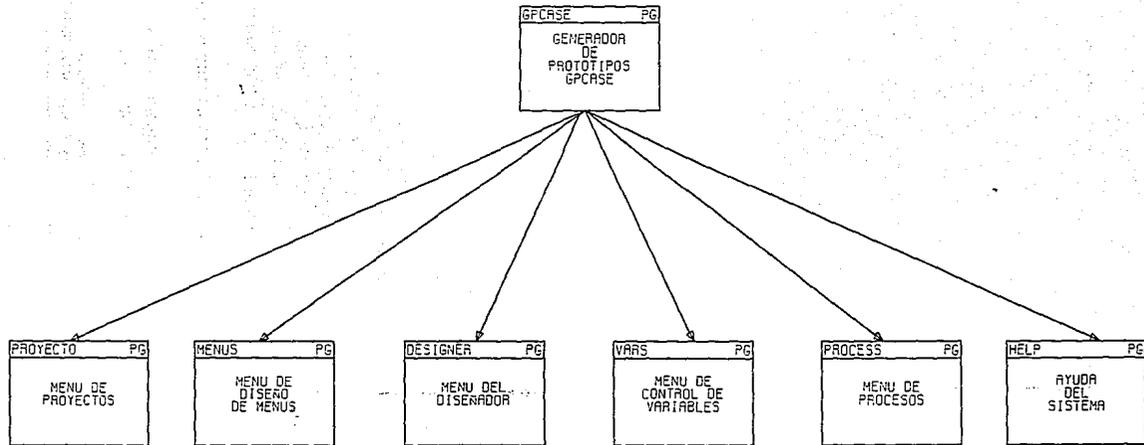
V.2.2. Diagrama Estructurado (DE)

A continuación se presenta una serie de elementos que permitieron el diseño del sistema GPCASE, los cuales fueron obtenidos utilizando Pro Kit WorkBench.

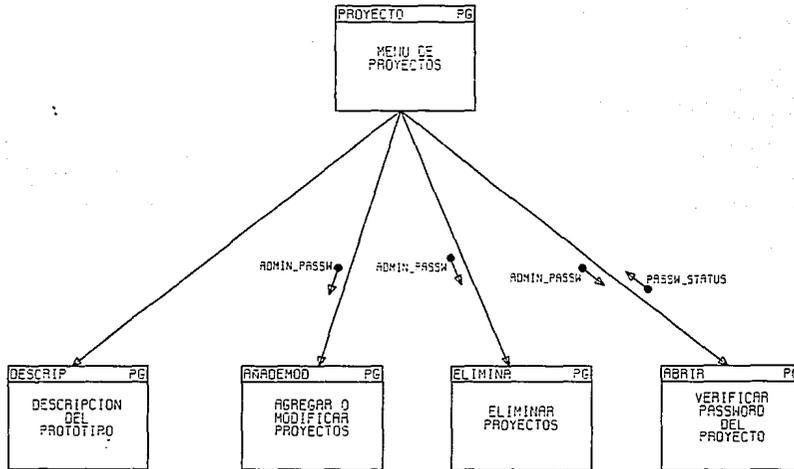
Pro Kit generó los siguientes elementos de diseño:

- Diagrama General de Módulos
- Diagramas de Módulos:
 - Menú de Proyectos
 - Menú de Diseño de Menús
 - Menú del Diseñador
 - Menú de Control de Variables
 - Menú de Procesos
- Descripción de los Módulos

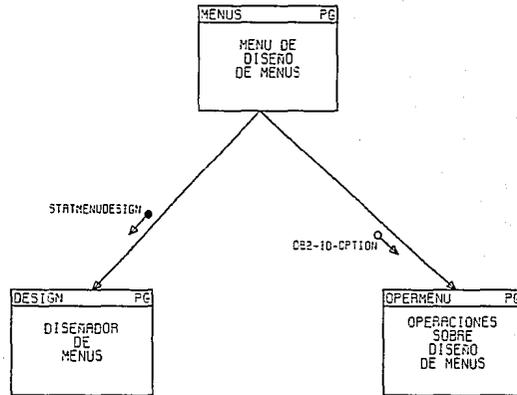
DIAGRAMA ESTRUCTURADO DE GPCASE



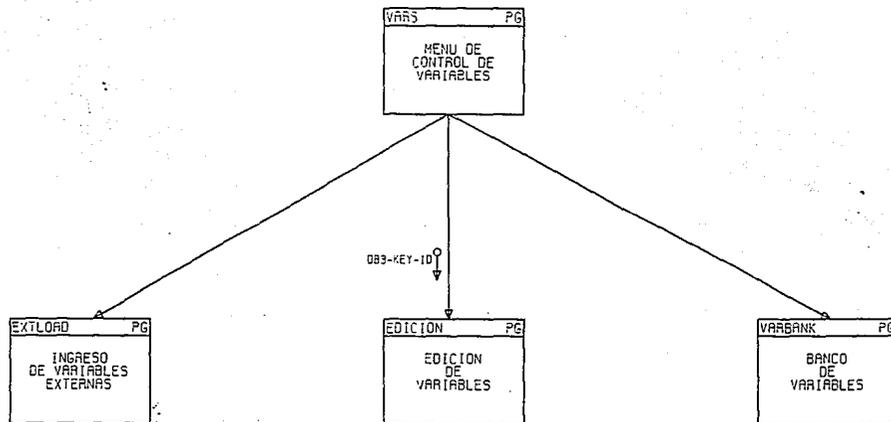
MENU DE PROYECTOS



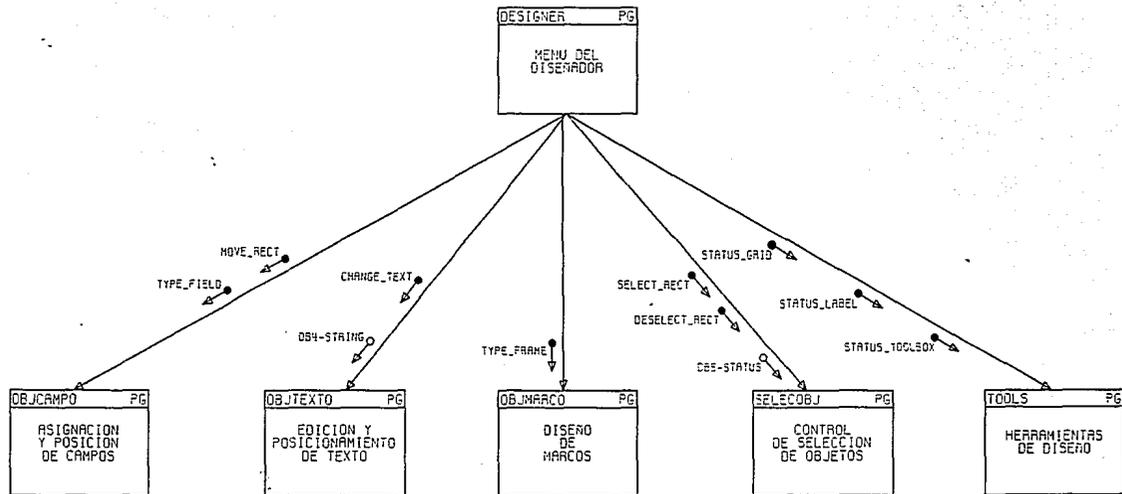
MENU DE DISEÑO DE MENUS



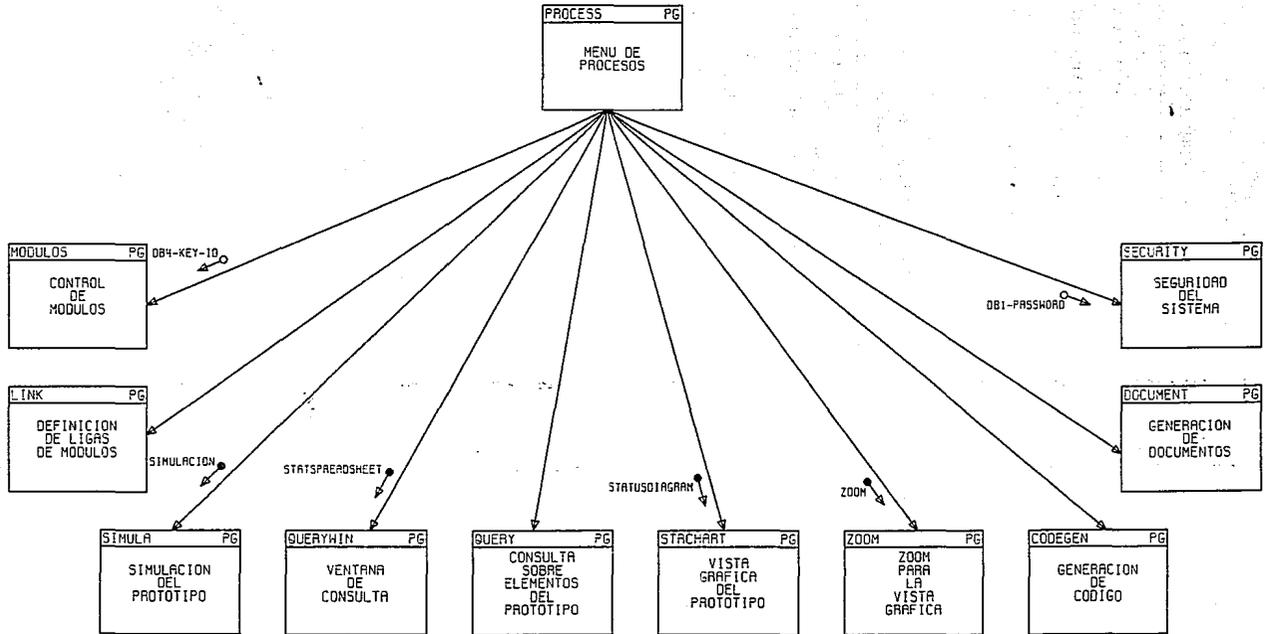
MENU DE VARIABLES



MENU DEL DISEÑADOR



MENU DE PROCESOS



PROJECT :GPCASE
VERSION :2
WORKSTATION:MASTER
MACRO NAME :SC.ALL

Diagramas De GPCASE
STRUCTURE CHART

PAGE 1
DATE:12/07/93
TIME:23:01:40
USER:SYSTEM

COMPLETE REPORT OF ALL STRUCTURE CHARTS

```
=====
--NAME--      -----RELATED MODULE LABEL-----      ---FROM-----
DESIGNER      MENU DEL DISEÑADOR      DIAGRAM
-----CONTAINS MODULES-----
--NAME--      -----MODULE LABEL-----      ALSO IS STRUCTURE CHART
OBJCAMPO      ASIGNACION Y POSICION DE CAMPOS      N
OBJMARCO      DISEÑO DE MARCOS      N
OBJTEXTO      EDICION Y POSICIONAMIENTO DE TEXTO      N
SELECOBJ      CONTROL DE SELECCION DE OBJETOS      N
TOOLS         HERRAMIENTAS DE DISEÑO      N
=====
```

TOTAL MODULES IN STRUCTURE CHART : 5
TOTAL RELATED STRUCTURE CHARTS FOR ABOVE MODULES: 0

```
=====
--NAME--      -----RELATED MODULE LABEL-----      ---FROM-----
MENUS         MENU DE DISEÑO DE MENUS      DIAGRAM
-----CONTAINS MODULES-----
--NAME--      -----MODULE LABEL-----      ALSO IS STRUCTURE CHART
DESIGN        DISEÑADOR DE MENUS      N
MENUS         MENU DE DISEÑO DE MENUS      Y
OPERMENU      OPERACIONES SOBRE DISEÑO DE MENUS      N
=====
```

TOTAL MODULES IN STRUCTURE CHART : 3
TOTAL RELATED STRUCTURE CHARTS FOR ABOVE MODULES: 1

```
=====
--NAME--      -----RELATED MODULE LABEL-----      ---FROM-----
PROCESS       MENU DE PROCESOS      DIAGRAM
-----CONTAINS MODULES-----
--NAME--      -----MODULE LABEL-----      ALSO IS STRUCTURE CHART
CODEGEN       GENERACION DE CODIGO      N
DOCUMENT       GENERACION DE DOCUMENTOS      N
LINK          DEFINICION DE LIGAS DE MODULOS      N
MODULOS       CONTROL DE MODULOS      N
QUERY         CONSULTA SOBRE ELEMENTOS DEL PROTOTIPO      N
QUBRYWIN      VENTANA DE CONSULTA      N
SECURITY      SEGURIDAD DEL SISTEMA      N
SIMULA        SIMULACION DEL PROTOTIPO      N
STRCHART      VISTA GRAFICA DEL PROTOTIPO      N
ZOOM          ZOOM PARA LA VISTA GRAFICA      N
=====
```

TOTAL MODULES IN STRUCTURE CHART: 10
TOTAL RELATED STRUCTURE CHARTS FOR ABOVE MODULES: 0

PROJECT :GPCASE
VERSION :2
WORKSTATION:MASTER
MACRO NAME :SC.ALL

Diagramas De GPCASE
STRUCTURE CHART

PAGE 2
DATE:12/07/93
TIME:23:01:40
USER:SYSTEM

COMPLETE REPORT OF ALL STRUCTURE CHARTS

```
=====
--NAME--      RELATED MODULE LABEL-----  --FROM-----
PROYECTO      MENU DE PROYECTOS                    DIAGRAM

-----CONTAINS MODULES-----
--NAME--      MODULE LABEL-----  ALSO IS STRUCTURE CHART
ABRIR          VERIFICAR PASSWORD DEL PROYECTO      N
AÑADEMOD       AGREGAR O MODIFICAR PROYECTOS        N
DESCRIP        DESCRIPCION DEL PROTOTIPO             N
ELIMINA        ELIMINAR PROYECTOS                   N
PROYECTO      MENU DE PROYECTOS                     Y
=====
```

TOTAL MODULES IN STRUCTURE CHART : 5
TOTAL RELATED STRUCTURE CHARTS FOR ABOVE MODULES: 1

```
=====
--NAME--      RELATED MODULE LABEL-----  --FROM-----
GPCASE        GENERADOR DE PROTOTIPOS GPCASE        DICT-DIRECT

-----CONTAINS MODULES-----
--NAME--      MODULE LABEL-----  ALSO IS STRUCTURE CHART
DESIGNER      MENU DEL DISEÑADOR                    Y
GPCASE        GENERADOR DE PROTOTIPOS GPCASE        Y
HELP          AYUDA DEL SISTEMA                     Y
MENUS         MENU DE DISEÑO DE MENUS               Y
PROCESS       MENU DE PROCESOS                      Y
PROYECTO      MENU DE PROYECTOS                     Y
VARS          MENU DE CONTROL DE VARIABLES          Y
=====
```

TOTAL MODULES IN STRUCTURE CHART : 7
TOTAL RELATED STRUCTURE CHARTS FOR ABOVE MODULES: 7

```
=====
--NAME--      RELATED MODULE LABEL-----  --FROM-----
VARS          MENU DE CONTROL DE VARIABLES          DIAGRAM

-----CONTAINS MODULES-----
--NAME--      MODULE LABEL-----  ALSO IS STRUCTURE CHART
EDICION       EDICION DE VARIABLES                  N
EXTLOAD       INGRESO DE VARIABLES EXTERNAS        N
VARBANK       BANCO DE VARIABLES                   N
VARS          MENU DE CONTROL DE VARIABLES          Y
=====
```

TOTAL MODULES IN STRUCTURE CHART : 4
TOTAL RELATED STRUCTURE CHARTS FOR ABOVE MODULES: 1

=====

TOTAL NUMBER OF STRUCTURE CHARTS ON REPORT : 7

V.2.2. Diagrama Estructurado (DE)

A continuación se presenta una serie de elementos que permitieron el diseño del sistema GPCASE, los cuales fueron obtenidos utilizando Pro Kit WorkBench.

Pro Kit generó los siguientes elementos de diseño:

- Diagrama General de Módulos
- Diagramas de Módulos:
 - Menú de Proyectos
 - Menú de Diseño de Menús
 - Menú del Diseñador
 - Menú de Control de Variables
 - Menú de Procesos
- Descripción de los Módulos

DIAGRAMA DE ENTIDAD-RELACION

GPCASE

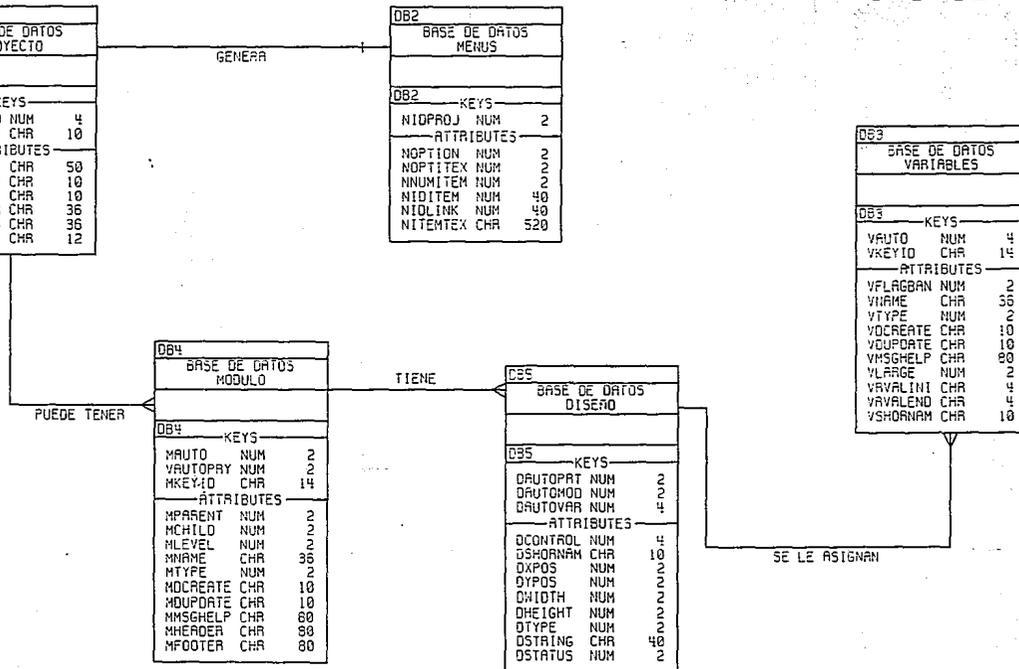
DB1	
BASE DE DATOS PROYECTO	
DB1 KEYS	
DB1-AUTO NUM	4
PKEYID CHR	10
ATTRIBUTES	
PNAME CHR	50
PORTEINI CHR	10
PORTEND CHR	10
PNAMEIDIA CHR	36
PNUMERES CHR	36
PPWD CHR	12

DB2	
BASE DE DATOS MENUS	
DB2 KEYS	
NIDPROJ NUM	2
ATTRIBUTES	
NOPTION NUM	2
NOPTITEM NUM	2
NUMITEM NUM	2
NIDITEM NUM	40
NIDLINK NUM	40
NITEMTEX CHR	520

DB3	
BASE DE DATOS VARIABLES	
DB3 KEYS	
VAUTO NUM	4
VKEYID CHR	14
ATTRIBUTES	
VFLAGBAN NUM	2
VNAME CHR	55
VTYPE NUM	2
VDCREATE CHR	10
VUPDATE CHR	10
VMSGHELP CHR	80
VLRAGE NUM	2
VVALINI CHR	4
VVALEND CHR	4
VSHORNAM CHR	10

DB4	
BASE DE DATOS MODULO	
DB4 KEYS	
MAUTO NUM	2
VAUTOPRY NUM	2
MKEY-ID CHR	14
ATTRIBUTES	
MPARENT NUM	2
MCHILD NUM	2
MLEVEL NUM	2
MNAME CHR	36
MTYPE NUM	2
MDCREATE CHR	10
MUPDATE CHR	10
MMSGHELP CHR	80
MHEADER CHR	80
MFOOTER CHR	80

DB5	
BASE DE DATOS DISEÑO	
DB5 KEYS	
DAUTOPRT NUM	2
DAUTOMOD NUM	2
DAUTOVAR NUM	4
ATTRIBUTES	
DCONTROL NUM	10
DSHORNAM CHR	10
DXPDS NUM	2
DYPOS NUM	2
DWIDTH NUM	2
DHEIGHT NUM	2
DTYPE NUM	2
DSTRING CHR	40
DSTATUS NUM	2



PUEDA TENER

TIENE

SE LE ASIGNAN

PROJECT :GPCASE
 VERSION :2
 WORKSTATION:MASTER
 MACRO NAME :ATTRIBUTES.DN.ALL

Diagramas De GPCASE
 DATA ENTITY ATTRIBUTES

PAGE 1
 DATE:30/06/93
 TIME:20:38:21
 USER:SYSTEM

COMPLETE ATTRIBUTE LISTING OF ALL DATA ENTITIES

```
=====
----ID----  ---SIMPLE DATA ENTITY NAME---          WORKSTATION          LAST
DB1         BASE DE DATOS PROYECTO              ORIGIN              UPDATE
- - - - -   -----DATA STORE NAME-----          FROM              ORIGIN
DB1         PROYECTO                            DICT-DIRECT        DATE---
                                           30/06/1993
-----DATA ENTITY USAGE-----
DEPENDENCY HIERARCHY CHARACTERISTIC ASSOCIATIVE
D
```

DESCRIPTION: ESTA BASE DE DATOS CONTENDRA LA INFORMACION REFERENTE A LOS PROYECTOS QUE GPCASE MANEJA.

```
-----DATA ENTITY CHARACTERISTICS-----
***NO STORAGE ATTRIBUTES DEFINED ***
-----SIMPLE DATA ENTITY CONTENTS (DX,DE)-----
MIG
KEY  ---DATA STRUCTURE, ELEMENT NAME          KEY  DATA LENGTH/  PHYSICAL  MAXIMUM
      SEQ  TYPE  PRECISION  SIZE  ITER  ITERATION
      DB1-AUTO              1  NUMERIC      4      4
      DB1-KEY-ID           2  CHARACTE    10     10
      DB1-NAME             CHARACTE    50     50
      DB1-DATE-INI        CHARACTE    10     10
      DB1-DATE-END        CHARACTE    10     10
      DB1-NAME-DIRECTOR   CHARACTE    36     36
      DB1-NAME-RESPONSIBLE CHARACTE    36     36
      DB1-PASSWORD        CHARACTE    12     12

TOTAL ITEMS IN DATA ENTITY: 8
TOTAL DATA ENTITY PHYSICAL SIZE - MIN: 168  MAX: 168
```

```
-----HAS DECLARED ACCESS PROFILES (AP)-----
----ID----  -KEY TYPE-  -ORDER--  -FREQ --PTI--  -----ACCESS BY DE,DX-  SEQ
LLAVE DB1  NATURAL    0          DB1-AUTO      DB1-KEY-ID      A
                                           A
```

-----COMMENT-----
 LLAVE PRINCIPAL DE LA BASE DE DATOS

```
=====
----ID----  ---SIMPLE DATA ENTITY NAME---          WORKSTATION          LAST
DB2         BASE DE DATOS MENUS              ORIGIN              UPDATE
- - - - -   -----DATA STORE NAME-----          FROM              ORIGIN
DB2         MENU                            DICT-DIRECT        DATE---
                                           30/06/1993
-----DATA ENTITY USAGE-----
DEPENDENCY HIERARCHY CHARACTERISTIC ASSOCIATIVE
D
```

```
-----DATA ENTITY CHARACTERISTICS-----
***NO STORAGE ATTRIBUTES DEFINED ***
DESCRIPTION: TABLA DE MANTENIMIENTO DE MENUS (CREACION , MODIFICACION, ELIMINACION, ETC) DEL PROTOTIPO.
```

PROJECT :GPCASE
 VERSION :2
 WORKSTATION:MASTER
 MACRO NAME :ATTRIBUTES.DN.ALL

Diagramas De GPCASE
 DATA ENTITY ATTRIBUTES

PAGE 2
 DATE:30/06/93
 TIME:20:38:21
 USER:SYSTEM

COMPLETE ATTRIBUTE LISTING OF ALL DATA ENTITIES

-----SIMPLE DATA ENTITY CONTENTS (DX,DE)-----

MIG KEY	---DATA STRUCTURE, ELEMENT NAME	KEY SEQ	DATA TYPE	LENGTH/ PRECISION	PHYSICAL SIZE	ITER	MAXIMUM ITERATION
	DB2-OPTION		NUMERIC	2	2		
	DB2-OPTION-TEXT		NUMERIC	2	2		
	DB2-NUM-ITEMS		NUMERIC	2	2		
	DB2-ID-ITEM		NUMERIC	40	40		
	DB2-ID-LINK		NUMERIC	40	40		
	DB2-ITEM-TEXT		CHARACTE	520	520		
	DB2-ID-PROJECT	1	NUMERIC		2		2

TOTAL ITEMS IN DATA ENTITY: 7

TOTAL DATA ENTITY PHYSICAL SIZE - MIN: 608 MAX: 608

-----HAS DECLARED ACCESS PROFILES (AP)-----

----ID----	-KEY TYPE-	--ORDER--	-FREQ--PTI--	-----ACCESS BY DE,DX-	SEQ
LLAVEDB2	NATURAL		0	DB2-ID-PROJECT	A

-----COMMENT-----

LLAVE PRIMARIA DE LA BASE DE DATOS DE MENU

----ID----	----SIMPLE DATA ENTITY NAME--	WORKSTATION	---ID---	--OWNER-	ACCESS	LAST UPDATE
DB3	BASE DE DATOS VARIABLES			ORIGIN		ORIGIN
-SYNC WITH-	-----DATA STORE NAME-----	FROM				DATE---
DB3	VARIABLE	DICT-DIRECT				30/06/1993

-----DATA ENTITY USAGE-----

DEPENDENCY HIERARCHY CHARACTERISTIC ASSOCIATIVE

D

DESCRIPTION: CONTIENE EL BANCO DE VARIABLES TANTO DEFINIDAS POR EL USUARIO, COMO LAS OBTENIDAS FUERA DEL SISTEMA, TOMANDO EN CUENTA PARA ELLO LAS CARACTERISTICAS DE ESTAS.

-----DATA ENTITY CHARACTERISTICS-----

----ID---- ---SIMPLE DATA ENTITY NAME---

-----SIMPLE DATA ENTITY CONTENTS (DX,DE)-----

MIG KEY	---DATA STRUCTURE, ELEMENT NAME	KEY SEQ	DATA TYPE	LENGTH/ PRECISION	PHYSICAL SIZE	ITER	MAXIMUM ITERATION
	DB3-AUTO	1	NUMERIC	4	4		
	DB3-KEY-ID	2	CHARACTE	14	14		
	DB3-FLAG-BANK		NUMERIC	2	2		
	DB3-NAME		CHARACTE	36	36		
	DB3-TYPE		NUMERIC	2	2		
	DB3-DATE-CREATE		CHARACTE	10	10		
	DB3-DATE-UPDATE		CHARACTE	10	10		
	DB3-MSG-HELP		CHARACTE	80	80		
	DB3-LARGE		NUMERIC	2	2		
	DB3-RANGE-VALUE-INI		CHARACTE	4	4		
	DB3-RANGE-VALUE-END		CHARACTE	4	4		
	DB3-SHORT-NAME		CHARACTE	10	10		

TOTAL ITEMS IN DATA ENTITY: 12

TOTAL DATA ENTITY PHYSICAL SIZE - MIN: 178 MAX: 178

-----HAS DECLARED ACCESS PROFILES (AP)-----

----ID----	-KEY TYPE-	--ORDER--	-FREQ--PTI--	-----ACCESS BY DE,DX-	SEQ
LLAVEDB3	NATURAL		0	DB3-AUTO	A
				DB3-KEY-ID	A

PROJECT :GPCASE
 VERSION :2
 WORKSTATION:MASTER
 MACRO NAME :ATTRIBUTES.DN.ALL

Diagramas De GPCASE
 DATA ENTITY ATTRIBUTES

PAGE 3
 DATE:30/06/93
 TIME:20:38:21
 USER:SYSTEM

COMPLETE ATTRIBUTE LISTING OF ALL DATA ENTITIES

```

=====
---ID---  ---SIMPLE DATA ENTITY NAME---      WORKSTATION      LAST
DB4      BASE DE DATOS MODULO              ---ID--- --OWNER- ACCESS  UPDATE
- - - - -  -----DATA STORE NAME-----      ORIGIN           ORIGIN
DB4      MODULO                             FROM             DATE---
- - - - -  -----DATA ENTITY USAGE-----      DICT-DIRECT      30/06/1993
  
```

DEPENDENCY HIERARCHY CHARACTERISTIC ASSOCIATIVE
 D

DESCRIPTION: CONTROL DE LOS OBJETOS DE DISEÑO

-----DATA ENTITY CHARACTERISTICS-----

***NO STORAGE ATTRIBUTES DEFINED ***

-----SIMPLE DATA ENTITY CONTENTS (DX,DE)-----

MIG	KEY	DATA STRUCTURE, ELEMENT NAME	SEQ	DATA TYPE	LENGTH/PRECISION	PHYSICAL SIZE	MAXIMUM ITER	ITERATION
		DB4-AUTO	1	NUMERIC	2	2		
		DB4-AUTO-PROJECT	2	NUMERIC	2	2		
		DB4-KEY-ID	3	CHARACTE	14	14		
		DB4-PARENT		NUMERIC	2	2		
		DB4-CHILD		NUMERIC	2	2		
		DB4-LEVEL		NUMERIC	2	2		

---ID--- ---SIMPLE DATA ENTITY NAME---

-----SIMPLE DATA ENTITY CONTENTS (DX,DE)-----

MIG	KEY	DATA STRUCTURE, ELEMENT NAME	SEQ	DATA TYPE	LENGTH/PRECISION	PHYSICAL SIZE	MAXIMUM ITER	ITERATION
		DB4-NAME		CHARACTE	36	36		
		DB4-TYPE		NUMERIC	2	2		
		DB4-DATE-CREATE		CHARACTE	10	10		
		DB4-DATE-UPDATE		CHARACTE	10	10		
		DB4-MSG-HELP		CHARACTE	80	80		
		DB4-HEADER		CHARACTE	80	80		
		DB4-FOOTER		CHARACTE	80	80		

TOTAL ITEMS IN DATA ENTITY: 13 TOTAL DATA ENTITY PHYSICAL SIZE - MIN:
 322 MAX: 322

-----HAS DECLARED ACCESS PROFILES (AP)-----

---ID---	-KEY TYPE-	--ORDER--	-FREQ--PTI--	-----ACCESS BY DE\DX-	SEQ
LLAVEDB4	NATURAL		0	DB4-AUTO	A
				DB4-AUTO-PROJECT	A
				DB4-KEY-ID	A

```

=====
---ID---  ---SIMPLE DATA ENTITY NAME---      WORKSTATION      LAST
DB5      BASE DE DATOS DISEÑO              ---ID--- --OWNER- ACCESS  UPDATE
- - - - -  -----DATA STORE NAME-----      ORIGIN           ORIGIN
DB5      DISEÑO                             FROM             DATE---
- - - - -  -----DATA ENTITY USAGE-----      DICT-DIRECT      30/06/1993
  
```

DEPENDENCY HIERARCHY CHARACTERISTIC ASSOCIATIVE
 D

DESCRIPTION: MANTIENE LAS CARACTERISTICAS DEL DISEÑO DEL PROTOTIPO TALES COMO:
 CAMPOS, ETIQUETAS, MARCOS, VARIABLES, ETC.

-----DATA ENTITY CHARACTERISTICS-----

PROJECT :GPCASE
 VERSION :2
 WORKSTATION:MASTER
 MACRO NAME :ATTRIBUTES.DN.ALL

Diagramas De GPCASE
 DATA ENTITY ATTRIBUTES

PAGE 4
 DATE:30/06/93
 TIME:20:38:21
 USER:SYSTEM

COMPLETE ATTRIBUTE LISTING OF ALL DATA ENTITIES

-----SIMPLE DATA ENTITY CONTENTS (DX,DE)-----

MIG	KEY	DATA STRUCTURE, ELEMENT NAME	KEY SEQ	DATA TYPE	LENGTH/ PRECISION	PHYSICAL SIZE	ITER	MAXIMUM ITERATION
		DB5-AUTO-PROTOTYPE	1	NUMERIC	2	2		
		DB5-AUTO-MODULE	2	NUMERIC	2	2		
		DB5-AUTO-VARIABLE	3	NUMERIC	4	4		
		DB5-CONTROL		NUMERIC	4	4		
		DB5-SHORT-NAME		CHARACTE	10	10		
		DB5-X-POSITION		NUMERIC	2	2		
		DB5-Y-POSITION		NUMERIC	2	2		
		DB5-WIDTH		NUMERIC	2	2		
		DB5-HEIGHT		NUMERIC	2	2		
		DB5-TYPE		NUMERIC	2	2		

----ID----- SIMPLE DATA ENTITY NAME---

-----SIMPLE DATA ENTITY CONTENTS (DX,DE)-----

MIG	KEY	DATA STRUCTURE, ELEMENT NAME	KEY SEQ	DATA TYPE	LENGTH/ PRECISION	PHYSICAL SIZE	ITER	MAXIMUM ITERATION
		DB5-STRING		CHARACTE	40	40		
		DB5-STATUS		NUMERIC	2	2		

TOTAL ITEMS IN DATA ENTITY: 12

TOTAL DATA ENTITY PHYSICAL SIZE - MIN: 74 MAX: 74

-----HAS DECLARED ACCESS PROFILES (AP)-----

----ID----	-KEY TYPE-	--ORDER--	-FREQ-	--PTI--	-----ACCESS BY DE,DX-	SEQ
LLAVEDB5	NATURAL			0	DB5-AUTO-PROTOTYPE	A
					DB5-AUTO-MODULE	A
					DB5-AUTO-VARIABLE	A

TOTAL NUMBER OF DATA ENTITIES ON REPORT : 5

TOTAL NUMBER OF DATA ENTITIES ON PROJECT REPOSITORY VERSION 2 : 5

PROJECT :GPCASE
 VERSION :2
 WORKSTATION:MASTER
 MACRO NAME :CONTENT'S.DM.ALL

Diagramas De GPCASE
 DATA MODEL CONTENTS

PAGE 1
 DATE:04/07/93
 TIME:19:55:21
 USER:SYSTEM

COMPLETE CONTENTS REPORT OF ALL DATA MODELS

=====

DATA MODEL

-----NAME----- -----TYPE-----

-----CONTAINS DATA ENTITIES (DN)-----

-----ID-----	-----DATA ENTITY NAME-----	--TYPE--
DB1	BASE DE DATOS PROYECTO	SIMPLE
DB2	BASE DE DATOS MENUS	SIMPLE
DB3	BASE DE DATOS VARIABLES	SIMPLE
DB4	BASE DE DATOS MODULO	SIMPLE
DB5	BASE DE DATOS DISEÑO	SIMPLE

TOTAL ITEMS : 5

-----CONTAINS RELATIONSHIPS (RL)-----

---RELATIONSHIP NAME---	---TYPE---	INSERTION	DEGREE	--ORDER--		
GENERA	APPLICATION	MANUAL	1:1	IMMATERIA		
-----ID-----	-----DATA ENTITY NAME (DN)-----	EXISTS	MIN	AVG	MAX	---CARDINALITY---
OWNER(S): DB1	BASE DE DATOS PROYECTO	OPTIONAL	0	0	0	
MEMBERS: DB2	BASE DE DATOS MENUS	MANDATORY	0	0	0	

---RELATIONSHIP NAME---	---TYPE---	INSERTION	DEGREE	--ORDER--		
PUEDE TENER	APPLICATION	MANUAL	1:N	IMMATERIA		
-----ID-----	-----DATA ENTITY NAME (DN)-----	EXISTS	MIN	AVG	MAX	---CARDINALITY---
OWNER(S): DB1	BASE DE DATOS PROYECTO	OPTIONAL	0	0	0	
MEMBERS: DB4	BASE DE DATOS MODULO	OPTIONAL	0	0	0	

---RELATIONSHIP NAME---	---TYPE---	INSERTION	DEGREE	--ORDER--		
TIENE	APPLICATION	MANUAL	1:N	IMMATERIA		
-----ID-----	-----DATA ENTITY NAME (DN)-----	EXISTS	MIN	AVG	MAX	---CARDINALITY---
OWNER(S): DB4	BASE DE DATOS MODULO	OPTIONAL	0	0	0	
MEMBERS: DB5	BASE DE DATOS DISEÑO	OPTIONAL	0	0	0	

---RELATIONSHIP NAME---	---TYPE---	INSERTION	DEGREE	--ORDER--		
SE LE ASIGNAN	APPLICATION	MANUAL	1:N	IMMATERIA		
-----ID-----	-----DATA ENTITY NAME (DN)-----	EXISTS	MIN	AVG	MAX	---CARDINALITY---
OWNER(S): DB5	BASE DE DATOS DISEÑO	OPTIONAL	0	0	0	
MEMBERS: DB3	BASE DE DATOS VARIABLES	OPTIONAL	0	0	0	

TOTAL ITEMS : 4

=====

TOTAL NUMBER OF DATA MODELS ON REPORT : 1

PROJECT :GPCASE
VERSION :2
WORKSTATION:MASTER
MACRO NAME :ATTRIBUTES.RL.ALL

Diagramas De GPCASE
RELATIONSHIP ATTRIBUTES

PAGE 1
DATE:01/07/93
TIME:20:08:32
USER:SYSTEM

COMPLETE ATTRIBUTE LISTING OF ALL RELATIONSHIPS

```
=====
---RELATIONSHIP NAME----- ---TYPE---
      WORKSTATION
      ---ID--- --OWNER-  ACCESS
      LAST
      UPDATE--
RETENTION  FORKED RECURSIVE DEGREE ORDER  FORM  CLASS  ORIGIN  ORIGIN
           N          N      1:1          UNDEF  UNDEF  FROM    DATE--
DESCRIPTION: *** RECORD ADDED VIA DIAGRAM EDITOR
      DICT-DIRECT 01/07/1993
-----CONTAINS OWNER/MEMBERS (DN)-----
---OWNER---
      ---MEMBER--
      ---ID--- ----DATA ENTITY NAME (DN)----
      ---ID--- ----DATA ENTITY NAME (DN)----
DB1          BASE DE DATOS PROYECTO          DB2          BASE DE DATOS MENUS
EXISTS: OPTIONAL TYPE: 0 OR MORE          EXISTS: MANDATORY TYPE: 0 OR MORE
CARDINALITY - MIN: 0 AVG: 0 MAX: 0          CARDINALITY - MIN: 1 AVG: 1 MAX: 1
=====
```

```
=====
---RELATIONSHIP NAME----- ---TYPE---
      WORKSTATION
      ---ID--- --OWNER-  ACCESS
      LAST
      UPDATE--
RETENTION  FORKED RECURSIVE DEGREE ORDER  FORM  CLASS  FROM    ORIGIN
           1:N          UNDEF  UNDEF  DICT-DIRECT 01/07/1993
-----CONTAINS OWNER/MEMBERS (DN)-----
---OWNER---
      ---MEMBER--
      ---ID--- ----DATA ENTITY NAME (DN)----
      ---ID--- ----DATA ENTITY NAME (DN)----
DB1          BASE DE DATOS PROYECTO          DB4          BASE DE DATOS MODULO
EXISTS: OPTIONAL TYPE: 0 OR MORE          EXISTS: OPTIONAL TYPE: 0 OR MORE
CARDINALITY - MIN: 0 AVG: 0 MAX: 0          CARDINALITY - MIN: 0 AVG: 0 MAX: 0
=====
```

```
=====
---RELATIONSHIP NAME----- ---TYPE---
      WORKSTATION
      ---ID--- --OWNER-  ACCESS
      LAST
      UPDATE--
RETENTION  FORKED RECURSIVE DEGREE ORDER  FORM  CLASS  FROM    ORIGIN
           1:N          UNDEF  UNDEF  DICT-DIRECT 01/07/1993
-----CONTAINS OWNER/MEMBERS (DN)-----
---OWNER---
      ---MEMBER--
      ---ID--- ----DATA ENTITY NAME (DN)----
      ---ID--- ----DATA ENTITY NAME (DN)----
DB5          BASE DE DATOS DISEÑO          DB3          BASE DE DATOS VARIABLES
EXISTS: OPTIONAL TYPE: 0 OR MORE          EXISTS: OPTIONAL TYPE: 0 OR MORE
CARDINALITY - MIN: 0 AVG: 0 MAX: 0          CARDINALITY - MIN: 0 AVG: 0 MAX: 0
=====
```

```
=====
---RELATIONSHIP NAME----- ---TYPE---
      WORKSTATION
      ---ID--- --OWNER-  ACCESS
      LAST
      UPDATE--
RETENTION  FORKED RECURSIVE DEGREE ORDER  FORM  CLASS  FROM    ORIGIN
           1:N          UNDEF  UNDEF  DICT-DIRECT 01/07/1993
-----CONTAINS OWNER/MEMBERS (DN)-----
---OWNER---
      ---MEMBER--
      ---ID--- ----DATA ENTITY NAME (DN)----
      ---ID--- ----DATA ENTITY NAME (DN)----
DB4          BASE DE DATOS MODULO          DB5          BASE DE DATOS DISEÑO
EXISTS: OPTIONAL TYPE: 0 OR MORE          EXISTS: OPTIONAL TYPE: 0 OR MORE
CARDINALITY - MIN: 0 AVG: 0 MAX: 0          CARDINALITY - MIN: 0 AVG: 0 MAX: 0
=====
```

TOTAL NUMBER OF RELATIONSHIPS ON REPORT : 4
TOTAL NUMBER OF RELATIONSHIPS ON PROJECT REPOSITORY VERSION 2 : 4

COMPLETE CONTENTS REPORT OF ALL DATA ENTITIES

```

=====
----ID-----  ---SIMPLE DATA ENTITY NAME---
DB1           BASE DE DATOS PROYECTO
-----
-----SIMPLE DATA ENTITY CONTENTS (NDX,DE)-----
MIG
KEY  DATA STRUCT., ELEMENT NAME      KEY  DATA LENGTH/  PHYSICAL  MAXIMUM
      SEQ --TYPE-- PRECISION  SIZE  ITER  ITERATION
DB1-AUTO                1  NUMERIC      4      4
DB1-KEY-ID              2  CHARACTE    10     10
DB1-NAME                 CHARACTE    50     50
DB1-DATE-INI            CHARACTE    10     10
DB1-DATE-END            CHARACTE    10     10
DB1-NAME-DIRECTOR      CHARACTE    36     36
DB1-NAME-RESPONSIBLE   CHARACTE    36     36
DB1-PASSWORD           CHARACTE    12     12
TOTAL ITEMS IN DATA ENTITY:  8
TOTAL DATA ENTITY PHYSICAL SIZE - MIN:  168  MAX:  168
=====
  
```

```

-----
----ID-----  ---SIMPLE DATA ENTITY NAME---
DB2           BASE DE DATOS MENUS
-----
-----SIMPLE DATA ENTITY CONTENTS (NDX,DE)-----
MIG
KEY  DATA STRUCT., ELEMENT NAME      KEY  DATA LENGTH/  PHYSICAL  MAXIMUM
      SEQ --TYPE-- PRECISION  SIZE  ITER  ITERATION
DB2-OPTION              NUMERIC      2      2
DB2-OPTION-TEXT        NUMERIC      2      2
DB2-NUM-ITEMS          NUMERIC      2      2
DB2-ID-ITEM            NUMERIC     40     40
DB2-ID-LINK            NUMERIC     40     40
DB2-ITEM-TEXT          CHARACTE    520    520
DB2-ID-PROJECT         1  NUMERIC      2      2
TOTAL ITEMS IN DATA ENTITY:  7
TOTAL DATA ENTITY PHYSICAL SIZE - MIN:  608  MAX:  608
=====
  
```

```

-----
----ID-----  ---SIMPLE DATA ENTITY NAME---
DB3           BASE DE DATOS VARIABLES
-----
-----SIMPLE DATA ENTITY CONTENTS (NDX,DE)-----
MIG
KEY  DATA STRUCT., ELEMENT NAME      KEY  DATA LENGTH/  PHYSICAL  MAXIMUM
      SEQ --TYPE-- PRECISION  SIZE  ITER  ITERATION
DB3-AUTO                1  NUMERIC      4      4
DB3-KEY-ID              2  CHARACTE    14     14
DB3-FLAG-BANK           NUMERIC      2      2
DB3-NAME                 CHARACTE    36     36
DB3-TYPE                NUMERIC      2      2
DB3-DATE-CREATE         CHARACTE    10     10
DB3-DATE-UPDATE         CHARACTE    10     10
DB3-MSG-HELP            CHARACTE    80     80
DB3-LARGE               NUMERIC      2      2
DB3-RANGE-VALUE-INI    CHARACTE      4      4
DB3-RANGE-VALUE-END    CHARACTE      4      4
DB3-SHORT-NAME          CHARACTE    10     10
  
```

PROJECT :GPCASE
 VERSION :2
 WORKSTATION:MASTER
 MACRO NAME :CONTENTS.DN.ALL

Diagramas De GPCASE
 DATA ENTITY CONTENTS

PAGE 2
 DATE:01/07/93
 TIME:20:15:17
 USER:SYSTEM

COMPLETE CONTENTS REPORT OF ALL DATA ENTITIES

-----SIMPLE DATA ENTITY CONTENTS (DX,DE)-----
 MIG KEY DATA LENGTH/ PHYSICAL MAXIMUM
 KEY \DATA STRUCTURE, ELEMENT NAME-- SEQ --TYPE-- PRECISION SIZE ITER ITERATION
 TOTAL ITEMS IN DATA ENTITY: 12
 TOTAL DATA ENTITY PHYSICAL SIZE - MIN: 178 MAX: 178

----ID---- ---SIMPLE DATA ENTITY NAME---
 DB4 BASE DE DATOS MODULO

-----SIMPLE DATA ENTITY CONTENTS (DX,DE)-----
 MIG KEY \DATA STRUCT, ELEMENT NAME KEY DATA LENGTH/ PHYSICAL MAXIMUM
 SEQ --TYPE-- PRECISION SIZE ITER ITERATION
 DB4-AUTO 1 NUMERIC 2 2
 DB4-AUTO-PROJECT 2 NUMERIC 2 2
 DB4-KEY-ID 3 CHARACTE 14 14
 DB4-PARENT NUMERIC 2 2
 DB4-CHILD NUMERIC 2 2
 DB4-LEVEL NUMERIC 2 2
 DB4-NAME CHARACTE 36 36
 DB4-TYPE NUMERIC 2 2
 DB4-DATE-CREATE CHARACTE 10 10
 DB4-DATE-UPDATE CHARACTE 10 10
 DB4-MSG-HELP CHARACTE 80 80
 DB4-HEADER CHARACTE 80 80
 DB4-FOOTER CHARACTE 80 80
 TOTAL ITEMS IN DATA ENTITY: 13
 TOTAL DATA ENTITY PHYSICAL SIZE - MIN: 322 MAX: 322

----ID---- ---SIMPLE DATA ENTITY NAME---
 DB5 BASE DE DATOS DISEÑO

-----SIMPLE DATA ENTITY CONTENTS (DX,DE)-----
 MIG KEY \DATA STRUCT, ELEMENT NAME KEY DATA LENGTH/ PHYSICAL MAXIMUM
 SEQ --TYPE-- PRECISION SIZE ITER ITERATION
 DB5-AUTO-PROTOTYPE 1 NUMERIC 2 2
 DB5-AUTO-MODULE 2 NUMERIC 2 2
 DB5-AUTO-VARIABLE 3 NUMERIC 4 4
 DB5-CONTROL NUMERIC 4 4
 DB5-SHORT-NAME CHARACTE 10 10
 DB5-X-POSITION NUMERIC 2 2
 DB5-Y-POSITION NUMERIC 2 2
 DB5-WIDTH NUMERIC 2 2
 DB5-HEIGHT NUMERIC 2 2
 DB5-TYPE NUMERIC 2 2
 DB5-STRING CHARACTE 40 40
 DB5-STATUS NUMERIC 2 2
 TOTAL ITEMS IN DATA ENTITY: 12
 TOTAL DATA ENTITY PHYSICAL SIZE - MIN: 74 MAX: 74

TOTAL NUMBER OF DATA ENTITIES ON REPORT : 5
 TOTAL NUMBER OF DATA ENTITIES ON PROJECT REPOSITORY VERSION 2:5

TABLAS DE BASES DE DATOS DEL SISTEMA GPCASE

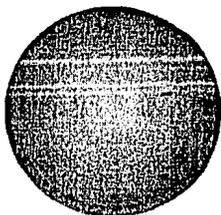
<i>Project.dat</i>					
<i>sProject</i>					
<i>Inicio</i>	<i>longitud</i>	<i>tipo</i>	<i>nombre</i>	<i>indice</i>	<i>campo</i>
1	2	int	lAuto	0	autoinc
3	10	char	sKeyID(10)	1	zstring
13	50	char	sName(50)		2
63	10	char	sDateIn(10)		5
73	10	char	sDateEnd(10)		6
83	36	char	sNameDirector(36)		3
119	36	char	sNameResponsible(36)		4
155	12	char	sPassWord		
total :	166				

<i>Module.dat</i>					
<i>sModule</i>					
<i>Inicio</i>	<i>longitud</i>	<i>tipo</i>	<i>nombre</i>	<i>indice</i>	<i>campo</i>
1	2	int	lAuto	0	autoinc
3	2	int	lAutoProject	1	integer
5	14	char	sKeyid(14)	2	zstring
19	2	int	lParent		6
21	2	int	lChild		7
23	2	int	lLevel		8
25	36	char	sName(36)		9
61	2	int	lType		10
63	10	char	sDatecreate(10)		11
73	10	char	sDateUpdate(10)		12
83	80	char	sMsgHelp(80)		13
163	80	char	sHeader(80)		14
243	80	char	sFooter(80)		15
total :	322				

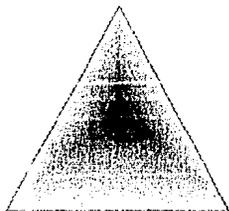
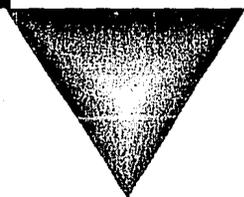
Variable.dat					
stVariable					
inicio	longitud	tipo	nombre	indice	campo
1	4	long	IAuto	0	16
5	14	char	sKeyID(14)	1	17
19	2	int	lFlagBank		18
21	36	char	sName(36)		19
57	2	int	lType		20
59	2	int	lLarge		21
61	10	char	sDateCreate(10)		22
71	10	char	sDateUpdate(10)		23
81	4	char	sRangeValueIn(4)		24
85	4	char	sRangeValueEnd(4)		25
89	80	char	sMsgHelp(80)		26
169	10	char	sShortName(10)		27
Total :	178				

Desing.dat					
stDesing					
inicio	longitud	tipo	nombre	indice	campo
1	2	int	lPrototype	0	
3	2	int	IAutoModule	0	
5	4	long	IAutoVariable	1	
9	4	long	lControl	2	
13	10	char	sShortName(10)	2	28
23	2	int	lxPosition		29
25	2	int	lyPosition		30
27	2	int	lWidIn		
29	2	int	lHeight		
31	2	int	lType		
33	40	char	sString(40)		
73	2	int	lStatus		
Total :	74				

<i>Menu.dat</i>					
<i>siMenu</i>					
<i>inicio</i>	<i>longitud</i>	<i>tipo</i>	<i>nombre</i>	<i>indice</i>	<i>campo</i>
1	2	int	idProject	0 int	
3	2	int	idOption	0 int	
5	12	char	sOptionText(12)	1 int	
9	2	int	iNumItems	2 int	
13	40	int	idItem(20)	2 int	31
53	40	int	idLink(20)		32
93	40	int	idMaxLevelModule(20)		33
133	520	char	sItemText(20)(26)		
Total :	658				



Manual del Usuario



VI.- Manual Del Usuario

Las principales características del sistema GPCASE se encuentran dentro de los siguientes tópicos :

Menús:

Los títulos para el menú principal que se pueden diseñar por proyecto están limitados a 10 como máximo y el espacio que ocupen en conjunto no deberá exceder los 80 caracteres de longitud, incluyendo los espacios de separación que deben existir entre cada título diseñado. Esto significa que GPCASE no permitirá al diseñador seguir introduciendo títulos al menú principal del proyecto al cumplirse cualquiera de las 2 condiciones mencionadas, por ejemplo:

- Si un menú de proyecto llega a 10 títulos, GPCASE no permitirá agregar más títulos aunque no se haya llegado a los 80 caracteres.
- Si un menú llega al límite de 80 caracteres de longitud incluyendo los espacios entre títulos, GPCASE no permitirá más diseños aún cuando no se haya llegado a los 10 títulos.

La razón de estas limitantes es que la mayoría de las aplicaciones tanto para DOS como para Windows trabajan con menos de 10 títulos en sus menús principales. En cuanto a los 80 caracteres, es conocido que la mayoría de los monitores trabajan el estándar de 25 renglones por 80 columnas. Entonces el permitir que el diseñador genere más de 80 caracteres haría no transportable el código que genera GPCASE.

Items:

Se permite el diseño de 20 ítems como máximo por menú, esto obedece a que la mayoría de las aplicaciones actuales de software presentan menos de 20 ítems por menú. Otra de las razones es que muchas plataformas trabajan el modo texto en 80 columnas por 25 renglones, de estos 25 renglones se restan: uno por el nombre de la aplicación, uno por el renglón que ocupa el menú principal, a veces uno para tener mensajes de ayuda y otro más para mensajes de error. Esto conduce a que en el peor de los casos y hablando de sistemas típicos, quedarían solamente 21 renglones disponibles para la lista de ítems por menú. Se considera entonces que 20 renglones es muy razonable. La longitud para establecer el nombre del ítem será de 25 caracteres como máximo.

Renglones y Columnas:

Se maneja una cuadrícula de 24 renglones por 80 columnas para el diseño de pantallas de captura y reportes, considerando los puntos de los párrafos anteriores.

Mensajes, Encabezados y Pies de Página:

Estos elementos tendrán una longitud máxima de 80 caracteres que es el tamaño estándar de la aplicación GPCASE.

El sistema GPCASE se encuentra formado por las siguientes secciones :

MENÚ PRINCIPAL .

PROYECTO
 MENUS
 DISEÑADOR
 VARIABLES
 PROCESOS
 AYUDA

DISEÑADOR DE MENÚ.

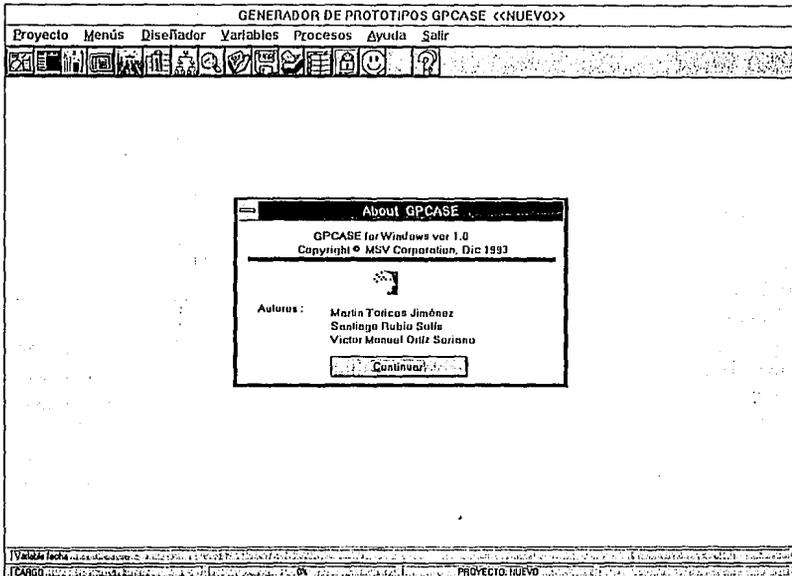
ÁREA DE DISEÑO DE MÓDULOS PANTALLA.

BARRA DE HERRAMIENTAS (Acceso rápido a funciones del menú principal).

TOOLBOX (Caja de herramientas para ejecución de funciones del menú principal).

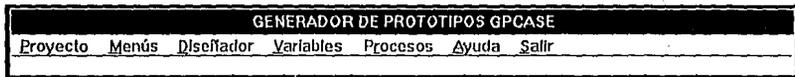
LÍNEA DE MENSAJES DE AYUDA(Mensajes de cada variable de diseño).

LÍNEA DE MENSAJES DE PROCESO (Avance de un proceso, proyecto, módulo, etc.).

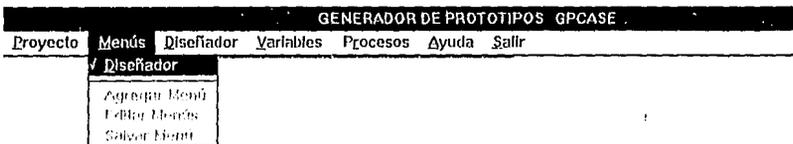


VI.1 MENÚ PRINCIPAL.

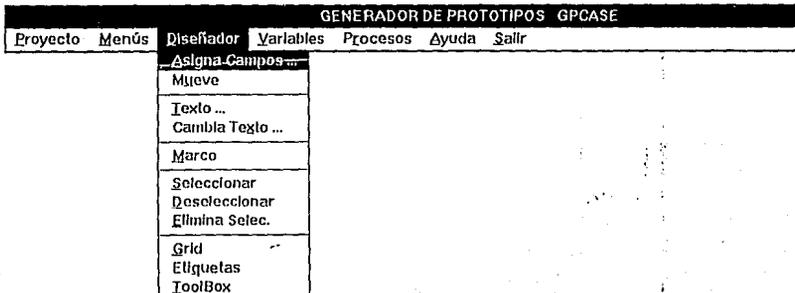
El Menú principal contiene todas las opciones para la utilización del sistema, se encuentra en la parte superior de la pantalla con el texto "GENERADOR DE PROTOTIPOS GPCASE " al cual se le adiciona el nombre del proyecto que el usuario elija.



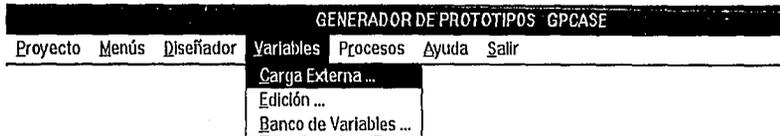
Al seleccionar la opción de **Proyecto** aparecerá el dialog para la configuración y elección de un proyecto. Si el usuario elige la opción de **Menús** entonces se desplegará una persiana con las siguientes opciones, con las cuales podrá habilitar o deshabilitar la ventana del menú, crear el diseño de su menú, editarlo (modificarlo, borrarlo, etc.), así como salvar el menú.



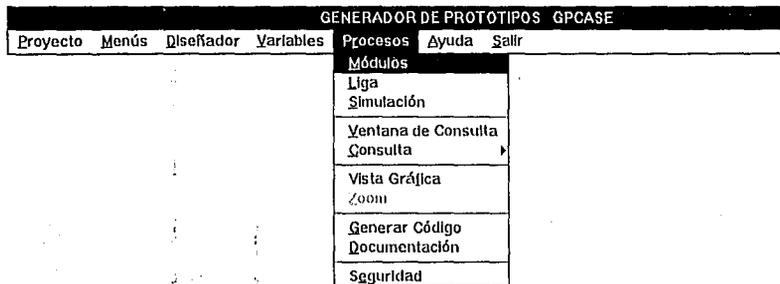
Si se elige **Diseñador** entonces aparecerá la siguiente persiana, la cual contiene los principales procesos para el diseño de módulos.



Dentro de la opción de **Variables** tenemos el manejo y configuración de variables .



Las funciones que dan una mayor versatilidad al sistema, que explotan sus recursos y dan al usuario una idea clara del mismo se encuentran dentro de la opción de **Procesos**.



La opción de **Ayuda** permite al usuario obtener una idea de la funcionalidad, convenciones empleadas, consulta de la documentación de un proyecto, glosario, tópicos, etc. En esta parte también se puede consultar el nombre, versión y autores del sistema GPCASE.



La última opción del proyecto corresponde a **Salir** que permite abandonar el sistema, cerrar bases de datos y enviar un mensaje para salvar el menú si es que éste fue modificado y no fue salvado .

VI.1.1.- PROYECTO (Dialog)

GENERADOR DE PROTOTIPOS GPCASE					
Proyecto	Menús	Diseñador	Variables	Procesos	Ayuda Salir

En esta opción el usuario podrá definir el nombre y características del proyecto, cabe hacer notar que es uno de los primeros pasos que debe realizar, ya que al definirlo, el proyecto adquiere un atributo que lo caracterizará y que permitirá que las bases de datos las cuales contienen datos y atributos del mismo se relacionen .

A continuación presentamos el *dialog* que aparecerá al elegir esta opción .

Definición de Proyectos	
ID (8 Caract.) :	NOMINA
Nombre (50 Caracteres máximo) :	
Sistema de Nomina empresarial	
Director :	Responsable Técnico :
Carlos Estrada Mina	Federico Solar Asol
Fecha	
Inicig : 05/00/93	Terminación : 07/08/94
F2 = Borrar, F3 = Actual, F4 = Calendario	
Porcentaje de Tiempo Consumido :	
0%	
Descripción	Añade/Mod
Elimina	Contingir
Cancelar	

El nombre que caracterizará al proyecto se coloca en el *edit* que tiene como etiqueta **ID** (IDentificador del proyecto), el cual tiene como máximo 8 caracteres y tienen la finalidad de funcionar como llave en el manejo de bases de datos (por lo cual se recomienda el uso de nombre *mnemónicos* que permitan identificar su funcionalidad en base al nombre), así como el de permitir el manejo de varios proyectos dentro de un mismo *dialog*, dichos proyectos se presentan en el *listbox* (lista de ítems) de la derecha, el cual le permite al usuario ver los proyectos existentes en forma alfabética, permitiéndole así mismo el cambio a otro proyecto con solo elegir a otro identificador dentro del *listbox*.

Como podemos notar muchas veces un nombre *mnemónico* no es lo suficientemente claro para describir la funcionalidad de un proyecto debido a que pueden existir ambigüedades, confusiones o simplemente no decir nada, es por ello que se añade un *edit* etiquetado con **Nombre** que permite al usuario poner una descripción de 50 caracteres con la finalidad de permitir dar una idea clara del proyecto y que evite principalmente los problemas ya mencionados, y si esto no fuese suficiente para describir la finalidad u objetivo del proyecto, el usuario tiene la facilidad de poner una descripción más amplia (1 K byte de memoria) con solo oprimir el botón de **Descripción**.

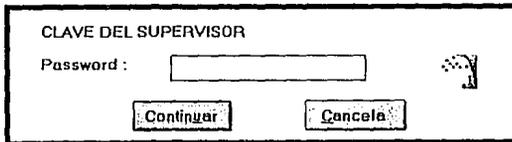
Como parte administrativa y de control aparecen los ítems de :

Director : Este ítem permite asignar el nombre del director del proyecto (35 caracteres) .

Responsable Técnico : Este ítem permite asignar el nombre del responsable técnico del proyecto (35 caracteres) .

Fecha : Aparecen dos edits que permiten asignar la fecha de inicio y terminación del proyecto, si el usuario presiona la tecla **F3** el sistema le proporcionara la fecha actual del sistema en el edit en el cual se encuentre, si presiona **F2** se borra la fecha contenida en dicho edit, si el usuario presiona **F4** aparecerá un calendario proporcionado por el sistema de la computadora el cual le permitirá elegir una fecha adecuada a sus necesidades .

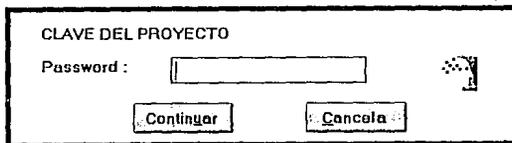
Para dar de alta los datos proporcionados por el usuario, así como para futuras correcciones basta con oprimir el botón de **Añade/Mod.** Si se desea borrar un proyecto solo deberá de presionar el botón de **Elimina**, eliminándose con ello todo lo concerniente con dicho proyecto. cabe hacer notar que estos procesos solo pueden ser hechos por el supervisor del sistema, ya que el sistema pedirá el password antes de realizar dichas acciones.



CLAVE DEL SUPERVISOR

Password :

Cuando se registre un proyecto por primera vez se pedirá también la clave para el acceso del mismo, ya que esta clave será asignada al usuario que este a cargo del mismo.



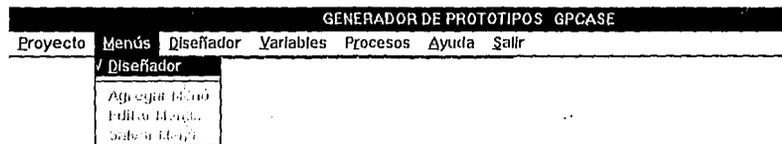
CLAVE DEL PROYECTO

Password :

Si el usuario presiona el botón de **Continuar** el sistema le pedirá la clave de acceso para poder entrar y si la clave es correcta, automáticamente todas las bases de datos harán referencia a la llave seleccionada (IDentificador del proyecto 8 caracteres) , por el contrario si el usuario elige la opción de **Cancelar** se tomara la llave que este por default.

VI.1.2.- MENÚS

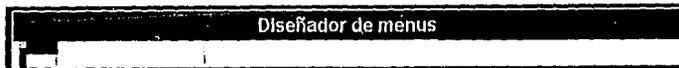
Al elegir esta opción del menú principal el sistema nos presentara un menú con las siguientes opciones:



Si no ha sido seleccionado un proyecto en el sistema, las tres últimas opciones permanecerán deshabilitadas hasta que exista una selección y la última de las tres **Salvar Menú** solo se habilitara cuando exista una modificación en el menú del proyecto. Si el usuario decide cambiar de proyecto o salir del sistema recibirá un aviso para poder salvar el diseño de dicho menú, quedando en él la responsabilidad de la modificación del menú.

DISEÑADOR (Proceso)

Permite la visualización de la ventana de diseño del menú del proyecto que se encuentre presente en ese momento. Por default esta ventana siempre aparecerá presente y desaparecerá cuando se escoga el ítem de **Diseñador** dentro del menú principal.



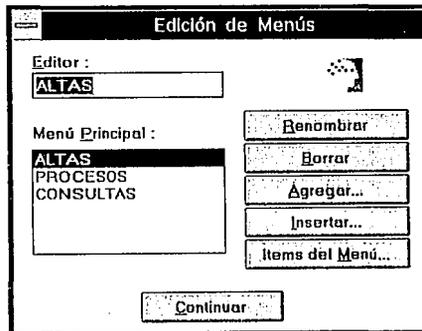
AGREGAR MENÚ (Dialog)

Este *dialog* nos da la oportunidad de agregar o insertar un nuevo menú a nuestro diseño de menús del proyecto. Sólo requiere de dos parámetros: el **Nombre** que llevará el nuevo menú que es el *edit* del mismo nombre y el **Número de ítems** que contendrá el menú.

El nombre del menú está limitado a 10 caracteres y el número de ítems a 20. Para el número de ítems existe un valor por defecto de 4. Al aceptar los datos del Dialog automáticamente se verá reflejada nuestra creación en el diseño del menú principal. El *push button Aceptar* no estará disponible si el primer carácter tecleado es diferente de alfanumérico. Los espacios no son válidos. **Cancelar** terminará el *dialog* y no hará modificaciones.

EDITAR MENÚ (Dialog)

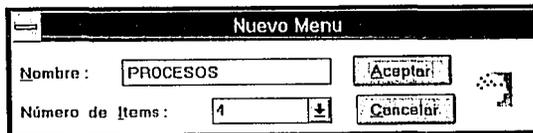
Este *Dialog* es muy completo y nos permite hacer modificaciones al diseño del menú rápidamente. Existen dos formas de acceder este *dialog*; la primera es mediante la selección de la opción *Editar Menú* del menú nombrado *Menús* de GPCASE. La segunda es cuando ya se han diseñado algunos menús para el proyecto y el usuario selecciona alguno de estos menús con lo cuál aparecerá este *dialog*. Una vez que aparece se refleja automáticamente el nombre del menú actual en el *edit Editor* y sobre el cuál se podrán hacer operaciones.



Para renombrar un menú se necesita seleccionar de *Menú Principal* el menú deseado el cuál se reflejará en *Editor*, en ese momento cambiamos el nombre y presionamos el *push button Renombrar*.

Un menú se borra sencillamente eligiéndolo de *Menú Principal*, reflejándose en *Editor* y se presiona el *push button Borrar*.

Para agregar un menú se hace presionando el *push button Agregar...* y este presentará el *Dialog NUEVO MENÚ* que fue previamente descrito, el menú no debe ser mayor a 10 caracteres de longitud, debemos aclarar que el menú proporcionado se adicionará al final de los menús actuales.



La inserción de menús se hace por medio del *push button Insertar...* que presenta el *Dialog NUEVO MENÚ*, pero a diferencia de *Agregar*, en este caso el menú se inserta en la posición en donde se encuentra actualmente la *list box* de *Menú Principal*, en consecuencia los menús se recorrerán un lugar hacia abajo en la lista y hacia la derecha en el menú principal del proyecto.

El *push button* **Ítems del Menú...** lo llevará a un *Dialog* llamado EDICIÓN DE ÍTEMS que le permitirá hacer operaciones sobre los ítems del menú.

El *push button* **Continuar** cerrará el *Dialog* y regresará al menú del proyecto en cuestión. Este *Dialog* tiene más partes que el de EDICIÓN DE MENÚS, pero en este caso nos permite hacer modificaciones a los ítems del menú que se haya seleccionado. El acceso a este menú se logra mediante la selección del *push button* **Ítems del Menú** del *dialog* EDICIÓN DE MENÚS. Una vez que aparece se refleja automáticamente el nombre del menú actual en el *edit* **Menú**. Por su parte el *edit* **Editor** siempre refleja la selección actual del cursor de la *list box* llamada **Ítems del Menú** y que contiene en orden de aparición de arriba hacia abajo los ítems que pertenecen a ese menú en particular. Existe también un *edit* llamado **Módulo Primario de Conexión** que contendrá el nombre del módulo que esté ligado al ítem enfocado en ese momento por el cursor en **Ítems del Menú**. En caso de que no haya una conexión para ese ítem aparecerá la leyenda "Sin Conexión".

Para renombrar un ítem se necesita seleccionar de **Ítems del Menú** el ítem deseado el cuál se reflejará en **Editor**, en ese momento cambiamos el nombre y presionamos el *push button* **Renombrar**.

Un ítem se borra eligiéndolo de **Ítems del Menú**, reflejándose en **Editor** y se presiona el *push button* **Borrar**.

Para agregar un ítem se presiona el *push button* **Agregar...** y este presentará el *Dialog* AGREGAR ÍTEM para realizar la operación.

La inserción de ítems lleva la misma mecánica que la inserción de menús, se hace por medio del *push button* *Inserirar...* que presenta el *Dialog* AGREGAR ITEM, a diferencia de Agregar, en este caso el ítem se inserta en la posición en donde se encuentra actualmente la selección de la *list box* de *Items del Menú..*, en consecuencia los ítems se recorrerán un lugar hacia abajo en la lista.

El *push button* *Conexión del Item...* lo llevará a un *Dialog* llamado REGISTRO DE MÓDULOS que permitirá la conexión con los módulos del proyecto.

Registro de Módulos		Descripción
ID(8 Char):	<input type="text" value="DEFAULT0"/>	<input type="button" value="Añadir/mod"/>
Tipo	<input checked="" type="radio"/> Pantalla <input type="radio"/> Reporte	<input type="button" value="Elimina"/>
Nombre:	<input type="text"/>	<input type="button" value="Continúa"/>
Fechas	Creación: <input type="text" value="05/03/94"/> Actualización: <input type="text" value="05/03/94"/>	<input type="button" value="Cancela"/>
F2 = Borrar, F3 = Actual, F4= Calendario		
Línea de ayuda:		
<input type="text"/>		

El *push button* *Continuar* cerrará el *Dialog* y regresará al menú del proyecto en cuestión. Contiene el *edit* *Nuevo Item* y el cuál acepta como máximo 25 caracteres, recordando que el ítem adicionado se anexará al final de los ítems actuales, el *push button* *Aceptar* sólo se activará si la entrada de *Nuevo Ítem* en su primer caracter es diferente de un espacio o un punto, se debe comenzar con letra o número.

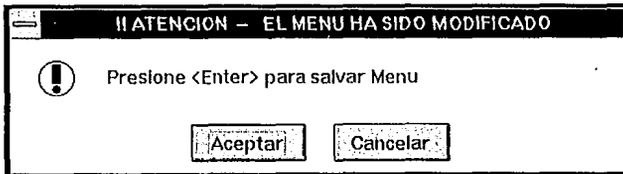
Cancelar regresa al *dialog* anterior sin hacer modificaciones.

REGISTRO DE MÓDULOS (Dialog)

Este contiene varias partes, primeramente un *edit* ID que refleja el identificador de un módulo ya existente, en su defecto se sugiere un default y el diseñador es libre de aceptarlo o modificarlo.

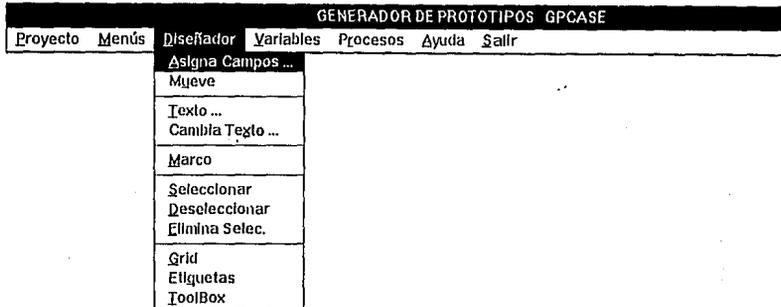
SALVAR MENÚ (Proceso)

Permite la actualización a las modificaciones hechas en el menú de diseño siempre y cuando se hallan realizado, ya que esta opción solo se habilitara cuando exista un cambio en el diseño del menú. Si el usuario decide cambiar de proyecto o salir del sistema recibirá un aviso para poder salvar el diseño de dicho menú, quedando en él la responsabilidad de la modificación de dicho menú.

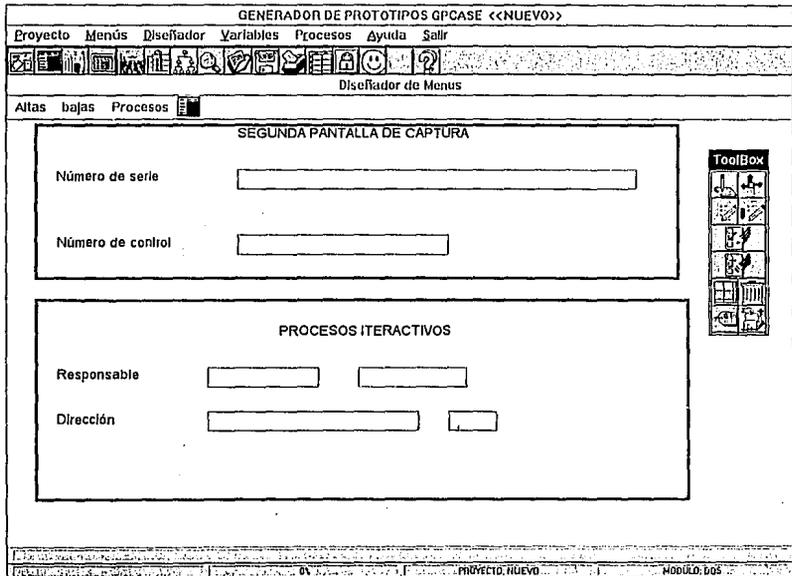


VI.1.3.- DISEÑADOR

Dentro de esta opción el usuario encontrara las principales funciones para el diseño de módulos pantalla

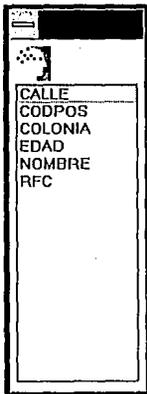


Ejemplo del diseño de un módulo pantalla :



ASIGNA CAMPOS (Dialog)

El diálogo de asignación de campos permite a el usuario el poder colocar estos dentro del diseño de un **módulo pantalla** , basta con que se elija un campo dentro de la lista para que el usuario pueda colocarlo dentro de la pantalla mediante el uso del mouse con solo presionar el botón izquierdo, cabe hacer notar que una vez hecha la selección dentro de la lista desaparece el nombre del campo dentro de ella.

**MUEVE**

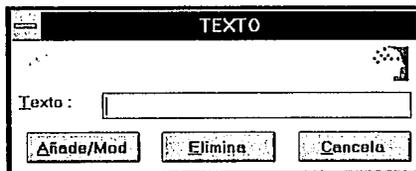
Al elegir esta opción el cursor cambiará a la forma de una mano lo que nos indicara que el sistema se encuentra listo para realizar dicha acción .



Para realizar esta acción basta con presionar el botón izquierdo del mouse dentro del objeto seleccionado, mover el mouse sin soltar el botón izquierdo hacia una posición que el usuario considere adecuada (aunque si esta no es adecuada el sistema valida que se muevan solo aquellas figuras que se encuentren dentro de los limites de diseño y no hace modificaciones de la posición original del objeto) para fijar el objeto basta con solo soltar el botón izquierdo. Cabe mencionar que el sistema trabaja con coordenadas de diseño en modo caracter (80x24) y validación del pccisionamiento de objetos.

TEXTO (Dialog Texto)

Este *dialog* permite la edición de texto para la asignación y modificación, para asignar texto basta con editar la etiqueta deseada la cual debe tener una longitud máxima de 35 caracteres, una vez editada se presiona el botón de **Añade/Mod** acto en el cual desaparecerá el *dialog* y el texto quedara grabado en la pantalla de diseño hasta que el usuario presione el botón izquierdo del mouse dentro de la posición que halla elegido.

**CAMBIA TEXTO (Dialog Texto)**

Permite al usuario el renombrar una etiqueta asignada dentro de un módulo pantalla, para hacer esto basta con seleccionar la etiqueta con el botón izquierdo del mouse, al realizar dicha acción aparecerá la ventana de dialog presentada anteriormente en **Texto**.

MARCO

Permite dibujar marcos o rectángulos dentro del área de diseño, para dibujarlos basta con elegir un punto dentro de la pantalla con solo presionar el botón izquierdo del mouse y sin soltarlo mover el mouse con lo cual se dibujara el marco hasta un punto en el que soltemos el botón del mouse. Cabe hacer notar que el marco se ajustara a las coordenadas de diseño de caracteres ASCII (80x24).

**SELECCIONAR**

Permite al usuario el seleccionar objetos para posteriormente serán borrados con la opción de **ELIMINAR SELECCIONADOS** o con la tecla **DEL**. Para seleccionar un objeto basta con marcarlos (el sistema les ponra unas pequeñas marcas en las esquinas) con el botón izquierdo del mouse.

DESELECCIONAR

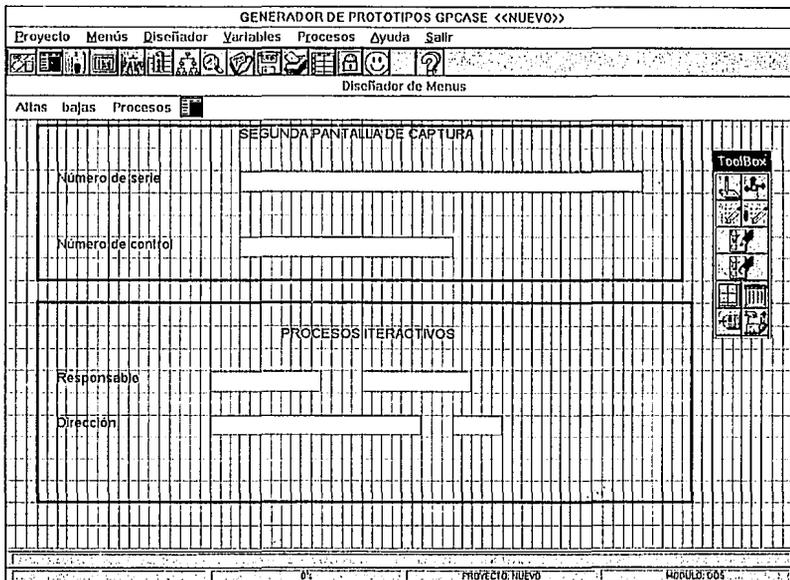
Es la acción contraria de **SELECCIONAR**.

ELIMINAR SELECCIONADOS

Elimina de la base de datos a todos aquellos objetos que se encuentren seleccionados.

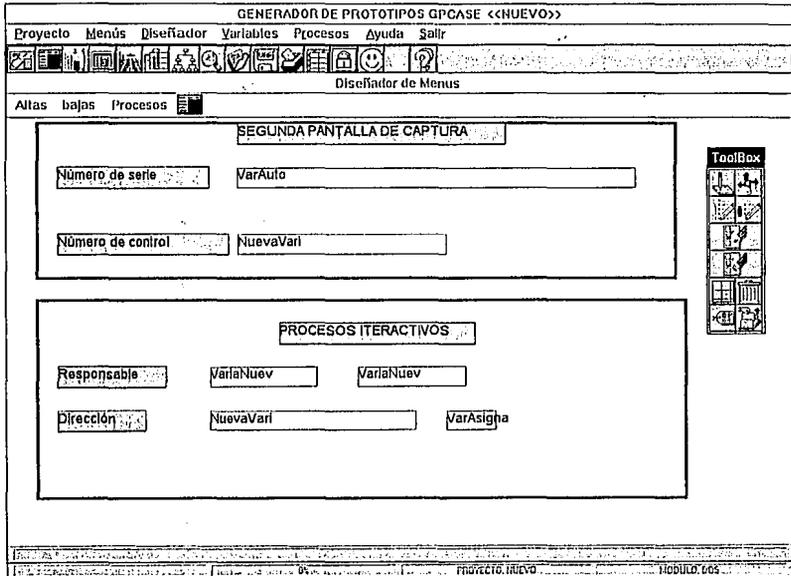
GRID

Coloca líneas que facilitan la ubicación de los objetos dentro del área de diseño, estas son colocadas en coordenadas de 80 x 24 (Columnas x Renglones, respectivamente).



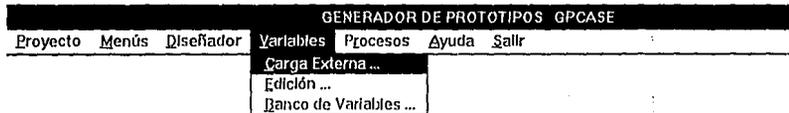
ETIQUETAS

Coloca los nombres cortos dentro de cada campo para poder facilitar su identificación, muestra la longitud exacta de las etiquetas en coordenadas de caracteres mediante el uso de sombreado (Los caracteres empleados en modos gráficos varían de los modo caracter) .



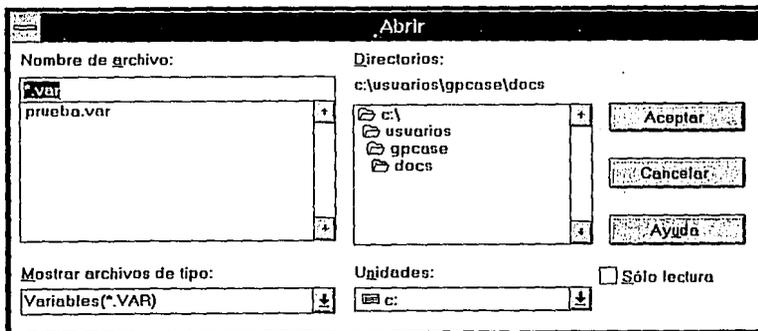
VI.1.4.- VARIABLES

Dentro de la opción de **Variables** tenemos el manejo y configuración de variables .



CARGA EXTERNA (Dialog)

Permite al usuario el poder acceder un archivo externo con la configuración de las variables que se quieren dar de alta dentro del sistema GPCASE .



Este dialog da todas las facilidades para buscar el archivo, ya que contiene una ventana en la cual se puede observar en forma gráfica el directorio, otra en la cual puede cambiar de unidad, así como una en la cual puede observar todos los archivos que fueron encontrados, para abrir el archivo basta con seleccionarlo y oprimir el botón de **Acceptar** .

EDICIÓN (Dialog)

En este dialog se pueden dar de alta nuevas variables, modificar y dar de baja.

El nombre de la variable se coloca en el *edit* que tiene como etiqueta **Identificador ID** , el cual tiene como máximo 8 caracteres y tienen la finalidad de funcionar como llave en el manejo de bases de datos (por lo cual se recomienda el uso de nombre *mnemónicos* que permitan identificar su funcionalidad en base al nombre), la variables se presentan en el *listbox* (lista de ítems) de la derecha, lo que le permite al usuario verlas en forma alfabética y seleccionarlas con solo tocarlas dentro de la lista..

El *edit* de *Nombre* permite al usuario poner una descripción de 35 caracteres , al hacer esto se asigna en *nom. corto* los primeros 10 caracteres que permiten identificar cada campo dentro de la etapa de diseño de módulos pantalla con solo activar el comando de *Etiquetas*

Configuración de variables																									
Identificador (ID) (8 Caracteres Máx.):	ABZDEF																								
Nombre :	VardeAsig																								
Nom. corto:	VarAsigna	<table border="1"> <tr> <th>Fechas</th> <th>Actualización</th> </tr> <tr> <td>Creación</td> <td>17/00/93</td> </tr> <tr> <td>Actualización</td> <td>17/00/93</td> </tr> </table>		Fechas	Actualización	Creación	17/00/93	Actualización	17/00/93																
Fechas	Actualización																								
Creación	17/00/93																								
Actualización	17/00/93																								
Tipo :	ASCII	<table border="1"> <tr> <td>ABZDEF</td> <td>↑</td> </tr> <tr> <td>AZCDEF</td> <td></td> </tr> <tr> <td>CALLE</td> <td></td> </tr> <tr> <td>CARGO</td> <td></td> </tr> <tr> <td>CODPOS</td> <td></td> </tr> <tr> <td>COLONIA</td> <td></td> </tr> <tr> <td>DOMICILI</td> <td></td> </tr> <tr> <td>EDAD</td> <td></td> </tr> <tr> <td>EMPLEADO</td> <td></td> </tr> <tr> <td>IBCDEF</td> <td></td> </tr> <tr> <td>LICENCIA</td> <td>↓</td> </tr> </table>		ABZDEF	↑	AZCDEF		CALLE		CARGO		CODPOS		COLONIA		DOMICILI		EDAD		EMPLEADO		IBCDEF		LICENCIA	↓
ABZDEF	↑																								
AZCDEF																									
CALLE																									
CARGO																									
CODPOS																									
COLONIA																									
DOMICILI																									
EDAD																									
EMPLEADO																									
IBCDEF																									
LICENCIA	↓																								
Longitud :	30	F2 = Borrar; F3 = Actual; F4 = Calendario																							
Valores Continuos																									
Valor Inicial :	<input type="text"/>	Valor Final :	<input type="text"/>																						
Línea de Ayuda (Mensaje: 80 caracteres)																									
Variable fecha																									
<table border="1"> <tr> <td>Descripción</td> <td>Valores Disc</td> <td>Añade/Mod</td> <td>Elimina</td> <td>Cancela</td> </tr> </table>				Descripción	Valores Disc	Añade/Mod	Elimina	Cancela																	
Descripción	Valores Disc	Añade/Mod	Elimina	Cancela																					

Descripción le permite al usuario poner una descripción completa de la variable (1 K byte de memoria)

Fecha : Aparecen dos edits que permiten asignar la fecha de inicio y terminación del modulo, si el usuario presiona la tecla *F3* el sistema le proporcionara la fecha actual del sistema en el edit en el cual se encuentre, si presiona *F2* se borra la fecha contenida en dicho edit, si el usuario presiona *F4* aparecerá un calendario proporcionado por el sistema de la computadora el cual le permitirá elegir una fecha adecuada a sus necesidades .

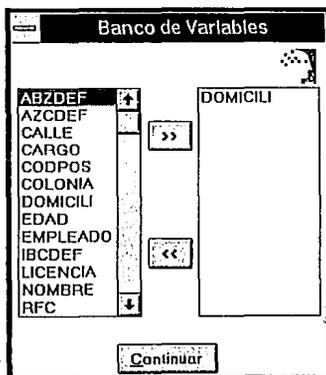
Línea de Ayuda : Este campo permite que se introduzca un mensaje de 80 caracteres el cual será desplegado en la fase de diseño y en el programa final en tiempo de ejecución, la finalidad de este campo es proporcionar una ayuda clara para el usuario final, ya que al encontrarse en un determinado campo aparecerá la información que se le haya proporcionado a este.

Para dar de alta los datos así como para futuras correcciones basta con oprimir el botón de *Añade/Mod.*

Si el usuario desea borrar una variable solo deberá de presionar el botón de *Elimina.*

BANCO DE VARIABLES (Dialog)

El Banco de variables permite al usuario elegir solo aquellas que necesite para el diseño de un módulo pantalla. En el *ListBox* de la izquierda aparecen todas las variables del proyecto mientras que en el de la derecha aparecen solo aquellas que hayan sido seleccionadas, para poder seleccionar una variable basta con elegirla dentro del *ListBox* de la izquierda y oprimir el botón de >> y para sacarla del de la derecha lo hacemos con << .



VI.1.5.- PROCESOS

Al elegir esta opción del menú principal el sistema nos presentara un menú con las siguientes opciones :

GENERADOR DE PROTOTIPOS GPCASE						
Proyecto	Menús	Diseñador	Variables	Procesos	Ayuda	Salir
				Módulos		
				Liga		
				Simulación		
				Ventana de Consulta		
				Consulta	Proyectos	
				Vista Gráfica	Menú	
				Zoom	Módulos	
				Generar Código	Variables	
				Documentación	Capas Filtros	
				Seguridad		

MÓDULOS

El nombre que caracterizara al modulo se coloca en el *edit* que tiene como etiqueta **ID** (IDentificador del modulo), el cual tiene como máximo 8 caracteres y tienen la finalidad de funcionar como llave en el manejo de bases de datos (por lo cual se recomienda el uso de nombre *mnemónicos* que permitan identificar su funcionalidad en base al nombre), así como el de permitir el manejo de varios módulos dentro de un mismo *dialog*, dichos módulos se presentan en el *listbox* (lista de ítems) de la derecha, el cual le permite al usuario ver los módulos existentes en forma alfabética, permitiéndole así mismo el cambio a otro modulo con solo elegir a otro identificador dentro del *listbox*.

Como podemos notar muchas veces un nombre *mnemónico* no es lo suficientemente claro para describir la funcionalidad de un modulo debido a que pueden existir ambigüedades, confusiones o simplemente no decir nada, es por ello que se añade un *edit* etiquetado con **Nombre** que permite al usuario poner una descripción de 50 caracteres con la finalidad de permitir dar una idea clara del modulo y que evite principalmente los problemas ya mencionados, y si esto no fuese suficiente para describir la finalidad u objetivo del modulo, el usuario tiene la facilidad de poner una descripción más amplia (1 K byte de memoria) con solo oprimir el botón de *Descripción*.

Si el usuario escoge la opción de pantalla le aparecerá el *dialog* de la siguiente forma :

Registro de Modulos																										
ID(8 Char):	ALTAS	Tipo	<input type="radio"/> Pantalla <input type="radio"/> Reporte	<table border="1"> <tr><td>ALTAS</td><td>↑</td></tr> <tr><td>ASI</td><td></td></tr> <tr><td>BAJAS</td><td></td></tr> <tr><td>BUENO</td><td></td></tr> <tr><td>CAMBIOS</td><td></td></tr> <tr><td>CAPTURA</td><td></td></tr> <tr><td>CUALQUI</td><td></td></tr> <tr><td>CUALQUIE</td><td>↓</td></tr> </table>	ALTAS	↑	ASI		BAJAS		BUENO		CAMBIOS		CAPTURA		CUALQUI		CUALQUIE	↓	<table border="1"> <tr><td>Descripción</td></tr> <tr><td>Añade/mod</td></tr> <tr><td>Elimina</td></tr> <tr><td>Conting</td></tr> <tr><td>Cancela</td></tr> </table>	Descripción	Añade/mod	Elimina	Conting	Cancela
ALTAS	↑																									
ASI																										
BAJAS																										
BUENO																										
CAMBIOS																										
CAPTURA																										
CUALQUI																										
CUALQUIE	↓																									
Descripción																										
Añade/mod																										
Elimina																										
Conting																										
Cancela																										
Nombre :	Módulo de captura de altas																									
Fechas	Creación :	13/02/93	Actualización :	13/00/93																						
F2 = Borrar, F3 = Actual, F4= Calendario																										
Línea de ayuda :																										
Pantalla de captura de altas																										

mientras que si escoge reporte le aparecerá de la siguiente forma :

Registro de Modulos																										
ID(8 Char):	ALTAS	Tipo	<input type="radio"/> Pantalla <input checked="" type="radio"/> Reporte	<table border="1"> <tr><td>ALTAS</td><td>↑</td></tr> <tr><td>ASI</td><td></td></tr> <tr><td>BAJAS</td><td></td></tr> <tr><td>BUENO</td><td></td></tr> <tr><td>CAMBIOS</td><td></td></tr> <tr><td>CAPTURA</td><td></td></tr> <tr><td>CUALQUI</td><td></td></tr> <tr><td>CUALQUIE</td><td>↓</td></tr> </table>	ALTAS	↑	ASI		BAJAS		BUENO		CAMBIOS		CAPTURA		CUALQUI		CUALQUIE	↓	<table border="1"> <tr><td>Descripción</td></tr> <tr><td>Añade/mod</td></tr> <tr><td>Elimina</td></tr> <tr><td>Conting</td></tr> <tr><td>Cancela</td></tr> </table>	Descripción	Añade/mod	Elimina	Conting	Cancela
ALTAS	↑																									
ASI																										
BAJAS																										
BUENO																										
CAMBIOS																										
CAPTURA																										
CUALQUI																										
CUALQUIE	↓																									
Descripción																										
Añade/mod																										
Elimina																										
Conting																										
Cancela																										
Nombre :	Módulo de captura de altas																									
Fechas	Creación :	13/02/93	Actualización :	13/00/93																						
F2 = Borrar, F3 = Actual, F4= Calendario																										
Línea de ayuda :																										
Pantalla de captura de altas																										
Encabezado :																										
Reporte del proceso de ALTAS del sistema de NOMINA																										
Pie de Página :																										
DISE&ADO POR EL SISTEMA GPCASE																										

Fecha : Aparecen dos edit que permiten asignar la fecha de inicio y terminación del modulo, si el usuario presiona la tecla **F3** el sistema le proporcionara la fecha actual del sistema en el edit en el cual se encuentre, si presiona **F2** se borra la fecha contenida en dicho edit, si el usuario presiona **F4** aparecerá un calendario proporcionado por el sistema de la computadora el cual le permitirá elegir una fecha adecuada a sus necesidades .

Línea de Ayuda : Este campo permite que se introduzca un mensaje de 80 caracteres el cual será desplegado en la fase de diseño y en el programa final en tiempo de ejecución., la finalidad de este campo es proporcionar una ayuda clara para el usuario final, ya que al encontrarse en un determinado campo aparecerá la información que se le haya proporcionado a este.

Encabezado : Dentro de este campo se podrá realizar la captura del mensaje que formara parte del encabezado de la forma del reporte de impresión, tiene una longitud máxima de 80 caracteres.

Pie de página : Dentro de este campo se podrá realizar la captura del mensaje que formara parte del pie de página de la forma del reporte de impresión, tiene una longitud máxima de 80 caracteres.

Para dar de alta los datos proporcionados por el usuario, así como para futuras correcciones basta con oprimir el botón de *Añade/Mod.*

Si el usuario desea borrar un modulo solo deberá de presionar el botón de *Elimina*, eliminándose con ello todo lo concerniente con dicho modulo.

Si el usuario presiona el botón de *Continuar* automáticamente se hará referencia a la llave seleccionada (IDentificador del modulo 8 caracteres) , por el contrario si el usuario elige la opción de *Cancelar* se tomara la llave del modulo que este por default.

LIGA (Dialog : Liga de Módulos)

El *dialog* de liga nos permite realizar la conexión entre los módulos previamente creados en la etapa del diseño de menú, ya que dichos módulos cuentan con la particularidad de ser 'módulos de inicio de secuencia' por lo cual todos aquellos que cuenten con dicha condición podrán ser listados dentro del *combo* del campo de **Módulo de referencia**, los módulos creados fuera de la etapa de diseño de menú y que además formaran parte de una secuencia indefinida de módulos que harán referencia a los ya mencionados se encontraran listados dentro de los *listbox*'s de **Ligas** y **Disponibles**, los primeros son aquellos que forman parte de la liga o secuencia de aparición de cada módulo, esto es , para el ejemplo de la figura de la parte inferior tenemos que el módulo de inicio de secuencia será CUENTAS, el cual se encontrara ligado con CAPTURA1 el cual a su vez con CAPTURA2 y este con el último módulo que es CAPTURA3. Dicha secuencia es establecida por el usuario por medio de la inserción y borrado de los módulos dentro de la lista de ligas, la inserción de un módulo disponible se realiza mediante la selección de este en la lista de **Disponibles** y presionado el botón con los símbolos de << , y el borrado de un módulo dentro de la secuencia de la liga se realiza mediante de la selección de este en la lista de **Ligas** y presionando la tecla de >> , los módulos que no pertenezcan a una lista de liga se encontraran dentro del *listbox* de **Disponibles**. Cuando se elige un módulo dentro del *ComboBox* de **Módulo de referencia** se presenta automáticamente la lista de ligas a la cual hace referencia .



SIMULACIÓN

Permite ver como funcionara la aplicación del usuario una vez compilada en el lenguaje que se elija .

VENTANA DE CONSULTA

Permite activar la ventana de consulta que tiene similitud en presentación como una hoja de Excel o Lotus para permitir ver la información del sistema en forma clara y rápida .

CONSULTA

Por medio de consulta se pueden elegir el tipo de información cruzada que se quiera observar :
Proyectos, menús, módulos y variables.

VISTA GRAFICA

Presenta en forma de diagrama la configuración del sistema hecha por el usuario en forma arborecente . Lo anterior es para tener una idea clara de como se encuentran las ligas entre los módulos con los menús y de los primeros con otros módulos diferentes .

ZOOM

La gráfica se presenta al 100% por lo que muchas veces puede aparecer en forma reducida que no sea clara la apreciación de los módulos, es por ello que se incluye la facilidad de un ZOOM el cual puede ser realizado al marcar el área que se quiere observar a detalle mediante el uso del mouse, esto es, se elige un punto presionando el botón izquierdo del mouse, se mueve este hasta el área elegida donde se libera el botón izquierdo y automáticamente se produce el efecto del mouse, para cambiar entre la gráfica del 100% y el área de ZOOM se elije dicho ítem dentro del menú o se presiona el botón con la lupa.

GENERAR CÓDIGO

Permite al usuario el generar el código de su diseño para que posteriormente lo compile y obtenga una aplicación idéntica a lo que diseño con el sistema GPCASE .

DOCUMENTACIÓN

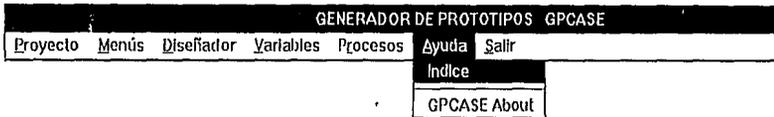
Proporciona información de los elementos que el usuario haya creado para la definición de su sistema, tales como : Menús, Variables, Módulos, etc.

El usuario puede consultar la información generada, usando la ayuda de GPCASE dentro de la opción de documentación , o checando el archivo de DOCUMENT:HLP dentro del directorio de DOCS.

SEGURIDAD

Pensando en que este sistema puede ser empleado para el desarrollo de diversas aplicaciones, lo cual implica a un responsable o supervisor y uno o varios responsables de cada aplicación, es por ello que pensamos en mantener un grado de seguridad , por ejemplo : dar la responsabilidad únicamente al supervisor de crear, modificar y eliminar proyectos , así como el de asignar y cambiar passwords a cada uno de ellos, quedando en los usuarios de cada proyecto la definición de variables, módulos, ligas , utilización de procesos, etc .. pero nunca la modificación de la definición del proyecto (Cabe hacer notar que el sistema GPCASE cuenta con un parámetro que indica el porcentaje de tiempo consumido con respecto a lo planeado (definido por el supervisor) y la fecha del sistema , esto permite tener un punto de referencia para tener una estimación del avance de lo planeado y lo real).

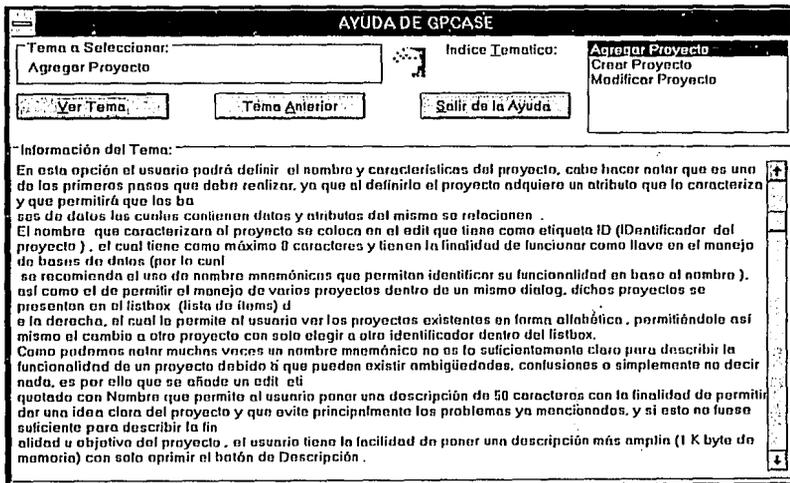
VI.1.6.- AYUDA DE GPCASE



INDICE (dialog)

Este *dialog* nos presenta la información de todas las operaciones y funcionalidad del sistema GPCASE. Está organizada de tal manera que al consultarla el diseñador pueda ir eligiendo temas relacionados y vea con mayor profundidad algún tópic. La forma de trabajar con este *dialog* es la que se describe a continuación.

Primeramente debemos mencionar que existen dos formas de invocar la Ayuda de GPCASE, una es por medio de la selección del ítem Índice del menú Ayuda de GPCASE. La segunda es por medio de su ícono respectivo, el cual se identifica por un signo de interrogación final en la barra de íconos.



Tema a seleccionar es un *edit* que indicará en cada ocasión cuál es el tema que está por analizarse si el usuario lo decide. Siempre refleja la selección actual de la *list box Índice Temático*.

Índice Temático nos lista los tópicos que podemos consultar y que tienen una relación directa con la información del tópico que actualmente se presenta en el *edit Información del Tema*.

Información del Tema presenta la información del tema previamente elegido, tiene un *scroll bar* vertical que permite al consultor revisar la información completa sin problema.

Cuando se entra por vez primera a la ayuda siempre se presenta por defecto la información de la "Ayuda de GPCASE" en donde se le refiere al consultor como está estructurada la consulta de la ayuda, menciona las partes del *dialog* AYUDA DE GPCASE y su operación, etc. Asimismo se posiciona por defecto como posible selección el tema de "Menús" y a su vez se refleja en **Tema a Seleccionar** y el *push button Ver Tema* se habilita para el usuario.

Para consultar el **Tema a seleccionar** se presiona el *push button Ver Tema* que mostrará acto seguido la información en **Información del Tema**. **Ver Tema** sólo se habilita cuando existe un tópico que se puede consultar a mayor profundidad, de tal suerte que si la información presentada en **Información del Tema** es el nivel máximo de detalle, este *push button* no podrá presionarse y sólo se podrá regresar al tópico anterior.

Existe algo importante por mencionar, nosotros decidimos adoptar una convención para identificar los tópicos. Cuando existen temas que son factibles de profundizar, el tópico aparece en letras mayúsculas y minúsculas en la forma acostumbrada en **Tema a Seleccionar**, la diferencia viene cuando el tema es del nivel máximo de profundidad, en este caso el tópico aparece en Mayúsculas con lo que descamos hacer ver al usuario que no hay mayor profundidad de información en esta consulta, así que **Ver Tema** estará deshabilitado y sólo será factible seleccionar **Tema Anterior**. Simultáneamente cuando se dá este caso en **Índice Temático** aparecerá únicamente el tópico que se ve en **Tema a Seleccionar** y también en letras Mayúsculas.

Existe otro *push button* llamado **Tema Anterior** y este precisamente nos regresa al tópico previo. **Tema Anterior** sólo se habilita cuando se ha ingresado a consultar algún tema, por lo que en ese momento ya existe precisamente un tema anterior, definitivamente que **Tema Anterior** siempre está deshabilitado cuando se está en la ayuda general, es decir el primer nivel.

Para salir del *dialog* de ayuda se selecciona el *push button Salir de la Ayuda*, que cerrará el *dialog* y volverá a GPCASE.

VI.2.- DISEÑADOR DE MENÚ.

En el diseñador de Menú se realizará el diseño del menú que va a formar parte del sistema del usuario (Proyecto), aquí se asignará el número de opciones de nuestro menú, el número de ítems por opción, el nombre de cada opción y de cada ítem, la funcionalidad de cada ítem (es decir con que módulo de comienzo de secuencia estará conectado), se podrá renombrar ítem's, eliminar e insertar nuevos. Se recomienda antes que todo el tener un buen análisis en cuanto a la estructura y nombres del menú para poder realizar una clara y fácil conexión con módulos de comienzo de secuencia, ya que estos al generarse adquieren automáticamente el nombre de su respectivo ítem, ya que muchas veces el usuario podría generar un menú, crear ítem's conectarlos con módulos de secuencia los cuales pueden adquirir otro nombre proporcionado por el usuario el cual puede ser muy diferente al del ítem, ya que a su vez puede ser también renombrado.

VI.3.- ÁREA DE DISEÑO DE MÓDULOS PANTALLA.

El área de diseño de módulos pantalla es aquella que se encuentra entre la barra de herramientas y la línea de mensajes de ayuda, cabe hacer notar que dentro de esta se encuentra el Diseñador de menús y el ToolBox lo cual no los hace un estorbo ya que pueden ser escondidos y retornados a la pantalla de acuerdo a las necesidades del usuario, para habilitarlos basta con seleccionar Diseñador o ToolBox dentro del menú principal o presionar los botones alusivos a dichas acciones (Diseñador : Una persiana de menú, ToolBox : Un lápiz, una goma, un Sacapuntas y un clip) dentro de la barra de herramientas.

VI.4.- BARRA DE HERRAMIENTAS (Acceso rápido al menú principal).

En la barra de herramientas se encuentran pequeñas representaciones gráficas (*Iconos*) que permiten la utilización rápida de algunas funciones realizadas por el menú principal tales como :

Proyecto : Representado por un escritorio sobre del cual hay un lápiz, una goma, un bloc ,un folder y unos lentés, lo que significa el área de trabajo de una persona que se dedica al análisis, estudio y creación de nuevas ideas, en este caso el desarrollo de un sistema .

Al presionar este botón es lo mismo que entrar a la opción del menú principal y seleccionar **Proyecto** con lo cual se ejecutara un dialog en el cual se podrá realizar la consulta, inserción y borrado de un proyecto .



Menú - Diseñador : Se encuentra representado por la figura de una Opción de un menú la cual nos muestra sus ítem's contenidos en forma de persiana.

Al presionar este botón es lo mismo que entrar a la opción **Menú** del menú principal y seleccionar el ítem de **Diseñador** , con lo cual se habilitará o deshabilitara la ventana del Diseñador de menú, es decir, por default siempre se presentara esta ventana, pero basta con hacer la selección dentro del menú principal o tocar el botón para que se desaparezca, y basta con realizar el mismo procedimiento para que vuelva a aparecer.



ToolBox : Se encuentra representado por la figura de un lápiz, una goma, un Sacapuntas y un clip, al presionar este botón es lo mismo que entrar a la opción **Diseñador** del menú principal y seleccionar el ítem de **ToolBox** , con lo cual se habilitará o deshabilitara la ventana del ToolBox, es decir, por default siempre se presentara esta ventana, pero basta con hacer la selección dentro del menú principal o tocar el botón para que se desaparezca, y basta con realizar el mismo procedimiento para que vuelva a aparecer.

Cabe hacer notar que en esta ventana se encuentran las principales funciones para el diseño de módulos pantalla .



Módulos : Su representación es la figura de una pantalla en la cual se encuentran campos y etiquetas , al presionar este botón es lo mismo que entrar a la opción **Procesos** del menú principal y seleccionar el ítem de **Módulos** , con lo cual se ejecutara un dialog en el cual se podrá realizar la consulta, inserción y borrado de un módulo.



Simulación : Se encuentra representado por la figura de un avión en una pista de despegue lo que nos da una idea del Fly-Simulator (Simulador de vuelo) y debido a la popularidad que tuvo este y a su relación con el avión en pleno vuelo fue razón suficiente para integrar la figura como alusión a dicho proceso

de simulación. Al presionar este botón es lo mismo que entrar a la opción **Procesos** del menú principal y seleccionar el ítem de **Simulación** , con lo cual se ejecutara una ventana en el cual se podrá observar el comportamiento de los objetos creados durante la etapa de diseño y análisis, tales como :

El menú del sistema (creado por el usuario), los módulos ligados, los campos de captura, etc.



Liga : Se encuentra representado por la figura de una cadena lo que nos da una idea de la acción que se desea realizar, encadenar o ligar un módulo, se tiene como referencia a un módulo creado en la etapa de diseño de menú y que posee la particularidad de ser el primero para una lista indefinida de módulos los cuales pueden ser diseñados fuera de la etapa de diseño de menú. Al presionar este botón es lo mismo que entrar a la opción **Procesos** del menú principal y seleccionar el ítem de **Liga** , con lo cual se aparecerá una ventana en la cual aparecerán todos los módulos que pueden ser de referencia, todos aquellos que forman una lista o liga y todos aquellos que se encuentran libres y listos para ser ligados, en esta etapa se pueden insertar o sacar módulos de una secuencia de liga con solo oprimir los botones de << y >> respectivamente.



Vista Gráfica : Se encuentra representado por la figura de una figura arborecente que se expande de arriba hacia abajo. Al presionar este botón es lo mismo que entrar a la opción **Procesos** del menú principal y seleccionar el ítem de **Vista Gráfica** , con lo cual se habilitará o deshabilitará la ventana que contiene la gráfica del proyecto que se encuentre seleccionado. esta representación nos dará una idea clara de como se encuentra distribuido nuestro proyecto .



Zoom : Se encuentra representado por la figura de una lupa lo que nos da una idea de querer hacer un acercamiento a un área seleccionada por el usuario. Esta opción es válida para la ventana de la **Vista Gráfica** basta con tan solo seleccionar el área que se desea hacer el Zoom mediante el uso del botón izquierdo del Mouse. Al presionar el botón del **Zoom** es lo mismo que entrar a la opción **Procesos** del menú principal y seleccionar el ítem de **Zoom** , con lo cual se podrá intercambiar entre el 100% de la figura y el Zoom seleccionado.



Documentación : Se encuentra representado por la figura de un folder en el cual se coloca un documento lo que nos da una idea de llevar un control sobre un documento en específico en este caso la documentación de nuestro sistema . Al presionar el botón del **Documentación** es lo mismo que entrar a la opción **Procesos** del menú principal y seleccionar el ítem de **Documentación** , con lo cual se puede obtener la información de como esta constituido nuestro proyecto, módulos, variables, etc.



Carga externa : Se encuentra representado por la figura de un diskette lo cual nos da una idea de querer acceder información externa .Al presionar el botón del **Carga externa** es lo mismo que entrar a la opción **Variables** del menú principal y seleccionar el ítem de **Carga externa** , con lo cual parecerá un *dialog* el cual se puede buscar y abrir archivos que contengan la configuración de variables para el sistema.



Banco de variables : Se encuentra representado por la figura de un tarjetero lo que nos representa la acción de almacenar información necesaria del sistema .Al presionar el botón del **Banco de variables** es lo mismo que entrar a la opción **Variables** del menú principal y seleccionar el ítem de **Banco de variables** , con lo cual aparecerá un *dialog* el cual nos presentará del lado izquierdo todas las variables del sistema mientras que en el lado derecho se encontraran aquellas que nosotros decidamos usar en ese momento para el diseño de nuestra pantalla, con esta opción se facilita el tener solo aquellas variables que vamos a ocupar.



Ventana de Consulta : Se encuentra representado por la figura de una hoja electrónica (SpreadSheet) lo que nos trata de dar una idea de una tabla que contiene información tabular, lo que para nosotros sería la información cruzada .Al presionar el botón de **Ventana de Consulta** es lo mismo que entrar a la opción **Procesos** del menú principal y seleccionar el ítem de **Ventana de Consulta** , con lo cual se habilitará o deshabilitará la ventana del SpreadSheet, es decir, basta con hacer la selección dentro del menú principal o tocar el botón para que aparezca, y basta con realizar el mismo procedimiento para que desaparezca. La información que contenga esta información será aquella que el usuario haya elegido en el ítem de **Consulta** .



Seguridad : Se encuentra representado por la figura de un Candado lo que nos da un idea de un control en cuanto a la seguridad del sistema, restricción a ciertos procesos mediante el uso de claves de acceso (password).Al presionar el botón de **Seguridad** es lo mismo que entrar a la opción **Procesos** del menú principal y seleccionar el ítem de **Seguridad** , con lo cual parecerá un *dialog* al cual solo tiene acceso el supervisor el cual puede cambiar el password de un proyecto.



Generar Código : Se encuentra representado por la figura de una carita feliz lo que nos da a entender que el usuario ha llegado a una de las últimas etapas dentro del análisis y diseño mediante el uso de GPCASE y espera generar el código que le permita ver su aplicación en el lenguaje que escogió.

Al presionar este botón es lo mismo que entrar a la opción **Procesos** del menú principal y seleccionar el ítem de **Generar Código** , con lo cual aparecerá un *dialog* en el cual se presentará un analista de los lenguajes disponibles para dicha generación del código.

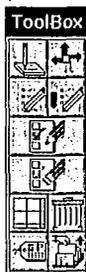


Ayuda : Se encuentra representado por la figura de un signo de interrogación lo que nos da a entender que el usuario tiene una duda o quiere consultar un tema en específico para poder hacer un uso correcto de las funciones del sistema. Al presionar este botón es lo mismo que entrar a la opción **Ayuda** del menú principal con lo cual aparecerá una ventana en la cual podrá consultar los tópicos que ahí se presentan.



VI.5.- TOOLBOX(Caja de herramientas para funciones del menú principal).

En el ToolBox (Caja de herramientas) se encuentran pequeñas representaciones gráficas (*Iconos*) que permiten la utilización rápida de algunas funciones de diseño de módulos pantalla realizadas por el menú principal, la ventana del ToolBox puede ser habilitada o deshabilitada eligiendo el ítem de ToolBox.



Por default siempre aparecerá el ToolBox pero si el usuario lo desea puede quitarlo y ponerlo en el momento en que el lo desea con solo presionar el *icono* que hace referencia a este y que tiene como figuras representativas en su interior a un clip, un lápiz, una goma y aun sacapuntas y se encuentra en la barra de iconos de la parte superior del sistema.

Cabe hacer notar que en esta ventana se encuentran las principales funciones para el diseño de módulos pantalla tales como :

Marco : Se encuentra representado por la figura de un lápiz dibujando un cuadro sobre de una hoja de papel lo cual nos representa la acción de poder dibujar un marco. Al presionar este botón es lo mismo que entrar a la opción **Diseñador** del menú principal y seleccionar el ítem de **Marco** , con lo cual se cambiara el cursor a la forma de un marco con una flecha lo que nos indicara que el sistema se encuentra listo para pintar marcos dentro del área de diseño de módulos, para dibujar un marco basta con presionar el botón izquierdo del mouse, mover el mouse sin soltar el botón izquierdo hacia una posición que el usuario considere adecuada (aunque si esta no es adecuada el sistema válida que se pinten solo aquellas figuras que se encuentren dentro de los límites de diseño) para aceptar la figura basta con solo soltar el botón izquierdo. Mientras no se cambie dentro del menú la opción o se vuelva a presionar el botón alusivo a dicha acción el sistema podrá seguir pintando marco.



Muevo : Se encuentra representado por la figura de flechas que apuntan hacia varios lados y que nos da la idea de movimiento que es precisamente lo que vamos hacer, mover objetos, tales como :

Campos, Etiquetas, Marcos .

Al presionar este botón es lo mismo que entrar a la opción **Diseñador** del menú principal y seleccionar el ítem de **Muevo** , con lo cual se cambiara el cursor a la forma de una mano lo que nos indicara que el sistema se encuentra listo para mover objetos dentro del área de diseño de módulos, para realizar esta acción basta con presionar el botón izquierdo del mouse dentro del objeto seleccionado, mover el mouse sin soltar el botón izquierdo hacia una posición que el usuario considere adecuada (aunque si esta no es adecuada el sistema valida que se muevan solo aquellas figuras que se encuentren dentro de los límites de diseño y no hace modificaciones de la posición original del objeto) para fijar el objeto basta con solo soltar el botón izquierdo. Cabe mencionar que aunque nos encontramos en un sistema que trabaja con coordenadas gráficas el sistema ajustara las figuras a las coordenadas de diseño en modo carácter (80x24).Mientras no se cambie dentro del menú la opción o se vuelva a presionar el botón alusivo a dicha acción el sistema podrá seguir moviendo objetos.



Texto : Se encuentra representado por la figura de un lápiz escribiendo sobre de una hoja de papel lo cual nos representa la acción de poder escribir un texto o etiqueta. Al presionar este botón es lo mismo que entrar a la opción **Diseñador** del menú principal y seleccionar el ítem de **Texto** , con lo cual aparecerá una ventana de dialog que nos permitirá introducir la cadena de caracteres, una vez introducida esta se presiona el botón de **Añade/Mod** para aceptarla , seguido esto el cursor del sistema cambiara a la forma de un lápiz para indicar al usuario que dentro del buffer de memoria se encuentra un texto el cual se colocara dentro del área de diseño de módulos con solo presionar el botón izquierdo del mouse.



Cambia Texto : Se encuentra representado por la figura de un lápiz escribiendo sobre de una hoja de papel y uná goma a un lado lo cual nos representa la acción de poder corregir un texto o etiqueta. Al presionar este botón es lo mismo que entrar a la opción **Diseñador** del menú principal y seleccionar el ítem de **Cambia Texto** , con lo cual cambiara el cursor a la forma de un lápiz lo que le indicara al usuario que el sistema se encuentra listo para realizar correcciones sobre el texto que se encuentre dentro del área de diseño de módulos, al realizar la selección sobre un objeto que tenga la particularidad de ser de texto se aparecerá una ventana de dialog con el texto a modificar, para aceptar la modificación basta con presionar el botón de **Añade/Mod.** Mientras no se cambie dentro del menú la opción o se vuelva a presionar el botón alusivo a dicha acción el sistema podrá seguir haciendo modificaciones.



Seleccionar : Se encuentra representado por la figura de un lápiz marcando una opción sobre de una hoja de papel lo cual nos representa la acción de poder elegir sobre opciones definidas (marcar seleccionar, etc.). Al presionar este botón es lo mismo que entrar a la opción **Diseñador** del menú principal y seleccionar el ítem de **Seleccionar** , con lo cual cambiara el cursor a la forma de una mano lo que le indicaría l usuario que el sistema seleccionara todos aquellos objetos que toque el usuario con el cursor de este, para realizar esta acción basta con que el usuario presione el botón izquierdo del mouse sobre un objeto seleccionado, al efectuarse la selección el sistema pondrá marcas en las esquinas del objeto para indicar que ha sido seleccionado. La razón de seleccionar uno o varios objetos se debe a que muchas veces se desean borrar uno o varios objetos a la vez . Cabe hacer notar que dicha selección se registra en la base de datos por lo que si el usuario apaga la máquina dichas selecciones no desaparecerán.



Deseleccionar : Se encuentra representado por la figura de un lápiz borrando marcas de selección sobre de una hoja de papel lo cual nos representa la acción de poder quitar las opciones definidas por el usuario (desmarcar). Al presionar este botón es lo mismo que entrar a la opción **Diseñador** del menú principal y seleccionar el ítem de **Deseleccionar** , con lo cual cambiara el cursor a la forma de una mano lo que le indicará al usuario que el sistema se encuentra listo para deseleccionar todos aquellos objetos que el usuario halla seleccionado previamente .



Eliminar: Se encuentra representado por la figura de un bote de basura representa la acción de tirar a la basura lo que ya no se desea. Al presionar este botón es lo mismo que entrar a la opción **Diseñador** del menú principal y seleccionar el ítem de **Eliminar**, con lo cual eliminarán todos aquellos objetos que previamente hayan sido seleccionados. Cabe hacer notar que se puede realizar la misma acción si se presiona la tecla de *Del* en teclado de la computadora.



Grid: Se encuentra representado por la figura de un fragmento de pantalla con líneas que la dividen lo cual nos representa que ha sido dividida para poder dar una ubicación exacta de esta. Al presionar este botón es lo mismo que entrar a la opción **Diseñador** del menú principal y seleccionar el ítem de **Grid**, con lo cual aparecerán divisiones dentro del área del diseño de módulos de pantalla, mismas que desaparecerán con volver a presionar el botón del grid o elegir el ítem dentro del menú principal por default no se colocara el grid.



Etiquetas : Se encuentra representado por la figura de una etiqueta lo cual nos representa la acción de colocar algo que distinga en forma clara y lo más fácil posible a un objeto de otro .Al presionar este botón es lo mismo que entrar a la opción **Diseñador** del menú principal y seleccionar el ítem de **Etiquetas** , con lo cual se pondrá distinguir los objetos tipo texto, ya que estos aparecerán con un fondo oscuro, mientras que los objetos tipo campo (variables) contendrá el nombre corto que se les haya asignado.



Asigna Campos : Se encuentra representado por la figura de una mano tomando información de un tarjetero lo que nos representa la acción de poder elegir información necesaria de un banco de datos .Al presionar este botón es lo mismo que entrar a la opción **Diseñador** del menú principal y seleccionar el ítem de **Asigna Campos** , con lo cual aparecerá un *dialog* con las variable previamente elegidas en el *dialog* de Banco de variables.

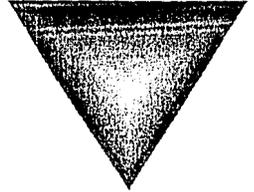
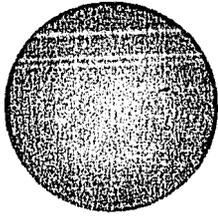


VI.6.- LÍNEA DE MENSAJES DE AYUDA(Mensajes de cada variable de diseño).

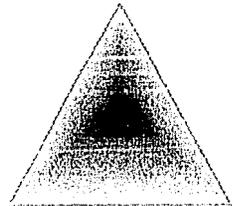
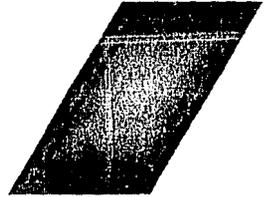
Dentro de esta línea aparecerán todos aquellos mensajes que el usuario halla colocada en cada una de las variables y que le permitirán saber el nombre de cada una de ellas, se hace mención a que estos mensajes aparecerán también dentro de la línea de ayuda que el sistema prototipo.

VI.7.- LÍNEA DE MENSAJES DE PROCESO (Avance de un proceso, proyecto, módulo, etc.).

Dentro de esta línea aparecerán todos aquellos mensajes que permitan al usuario saber en que proyecto se encuentra, en que módulo, que variable se encuentra utilizando o el avance de un proceso determinado, por ejemplo la lectura de un archivo externo, etc.



Mapa de Menús



Proyecto

Definición de Proyectos

ID (0 Correc):

Nombre (50 Caracteres máximo):

Director: Responsable Técnico:

Fecha: Inicia: Terminación:

F2 - Borrar, F3 - Actuar, F4 - CANCELAR

Porcentaje de Tiempo Consumido:

CLAVE DEL SUPERVISOR

Password:

(Solo si se trata de un nuevo Proyecto)

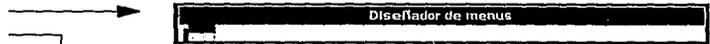
CLAVE DEL PROYECTO

Password:

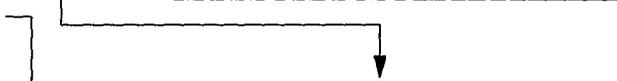
VII Mapas de Menús

Menús

Diseñador
Agregar Menu
Editar menús
Salvar Menu



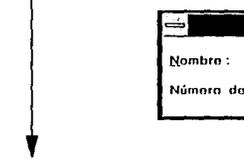
Diseñador de menús



Nuevo Menu

Nombre :

Número de ítems :

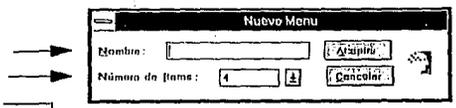


Edición de Menús

Editor :

Menú Principal :

- ALTAS
- PROCESOS
- CONSULTAS
- REPORTES



Nuevo Menu

Nombre :

Número de ítems :

Edición de ítems

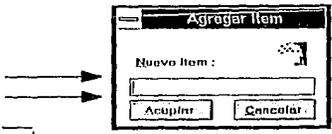
Menú :

Editor :

Ítems del Menú :

- CONSULTAS-1
- CONSULTAS-2
- CONSULTAS-3
- CONSULTAS-4
- CONSULTAS-5
- CONSULTAS-6
- CONSULTAS-7

Módulo Principal de Conexión : EMPRESAS



Agregar ítem

Nuevo ítem :

Registro de Módulos

ID(0) Ched : Tipo Pantalla Reporte

Nombre :

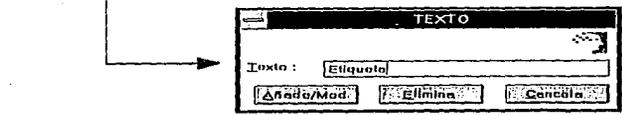
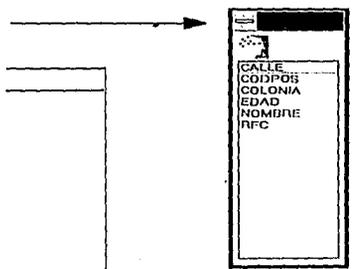
Fecha : Actualización :

F2 = Borrar, F3 = Actual, F4 = Calcular

Línea de ayuda :

Módulo de captura de datos de alta de empresa

Diseñador
Asigna Campos ...
Mueve
Texto ...
Cambia Texto ...
Marco
Seleccionar
Deseleccionar
Elimina Selec.
Grid
Etiquetas
ToolBox



VII Mapas de Menús

Variables

Carga Externa...
Edición ...
Variables ...

Abrir

Nombre de archivo:

Directorios:
c:\...lgpcase\plus\analisis

- c:\
- usuarios
- gpcase
- plus
- analisis
- resp0116
- respaldo
- temp

Mostrar archivos de tipo:

Unidades: Sólo lectura

Configuración de variables

Identificador (ID) (8 Caracteres Max.):

Nombre:

Nom. corto:

Tipos:

Longitud:

Valores Continuos

Valor Inicial:

Valor Final:

Fechas
Creacion:

Actualizacion:

F2 = Borrar; F3 = Actual; F4 = Calendario

Linea de Ayuda (Mensaje: 80 caracteres)

Banco de Variables

CALLE	CALLE
CARGO	CODPOS
CODPOS	COLONIA
COLONIA	EDAD
DOMICILI	NOMBRE
EDAD	RFC
RFC	
SUELDO	
TELEFONO	
ZONPOS	

VII Mapas de Menús

Procesos
Módulos
Liga
Simulación
Ventana de Consulta
Consulta
Vista Gráfica
Zoom
Generar Código
Documentación
Seguridad

Registro de Módulos

ID (8 Char): Tipo: Pantalla Repetir

Nombre:

Fecha: Actualización:

F2 = Borrar, F3 = Actual, F4 = Calendario

Línea de ayuda:

Acciones: Descripción, Añadir/mod, Eliminar, Continuar, Cancelar

Liga de módulos

Módulo Referenciado: CUENTAS

Ligas	Disponibles
CAPTURAI CAPTURAS CAPTURAJ	CAPTURAI EMPRESAS

Controles: <<, >>, Continúa

Proyectos

Menús

Módulos

Variables

Copiar Datos

Generación de Código

Generar Código

Generar, Cancelar

DOCUMENTACION

Generar Documentación

Generar, Cancelar

CLAVE DEL SUPERVISOR

Password:

Continuar, Cancelar

Cambiar Passwords

Proyectos y Supervisor

ABC
ASI
DIEN
ESCUELA
NOMINA
RUEJO
SUPERV.

Password:

Nuevo Password, Continuar

Ayuda

Índice

About



AYUDA DE GPCASE

Tema a Seleccionar: Índice Temático:

Interrogación del Tema:
AYUDA DE GPCASE (Dialog)

Este dialog nos presenta la información de todas las operaciones y funcionalidad del sistema GPCASE. Está organizada de tal manera que el consultante el consultador pueda ir siguiendo temas relacionados y va con mayor profundidad algún tópico. La forma de trabajar con este dialog es la que se describe a continuación.

Primeramente debemos mencionar que existen dos formas de invocar la Ayuda de GPCASE, una es por medio de la selección del ítem Índice del menú Ayuda de GPCASE. La segunda es por medio de su ícono respectivo, el cual se identifica por un signo de interrogación final en la barra de íconos.

Tema a seleccionar es un edit que indicará en cada ocasión cuál es el tema que está por analizarlo si el usuario lo decide. Siempre refleja la selección actual de la list box Índice Temático.

Índice Temático nos lista los tópicos que podemos consultar y que tienen una relación directa con la información del tópico que actualmente se presenta en el edit Información del Tema.

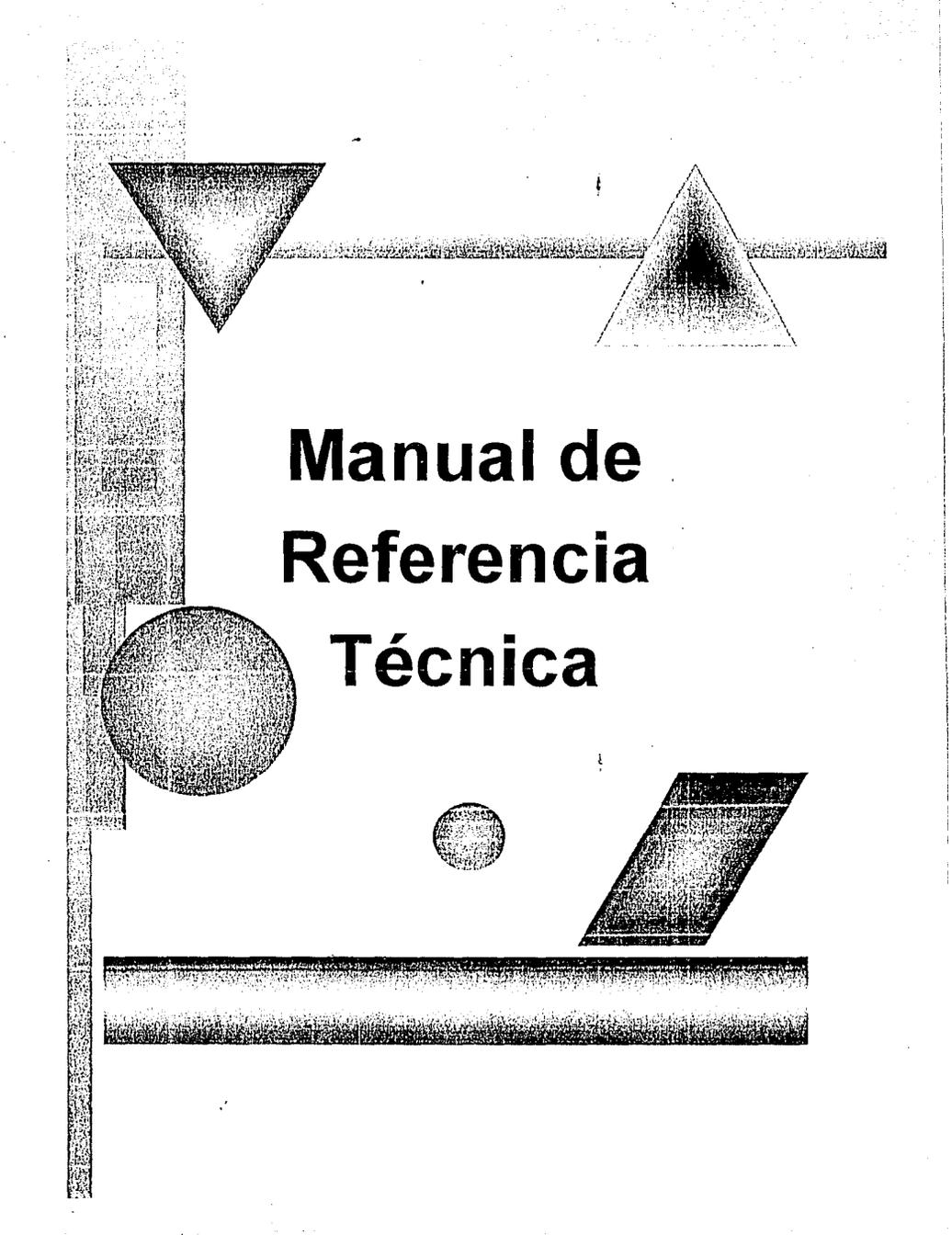
Información del Tema presenta la información del tema previamente elegido, tiene un scroll bar vertical que permite al consultor revisar la información completa sin problemas.



About GPCASE

GPCASE for Windows ver 1.0
Copyright © MSV Corporation, Dic 1993

Autores : Martin Torices Jiménez
Santiago Rubio Solís
Victor Manuel Ortiz Soriano

An abstract geometric design featuring a vertical line on the left, a horizontal line at the top, and a horizontal line at the bottom. A large inverted triangle is on the left, and a large upright triangle is on the right. A large circle is on the left, and a smaller circle is in the center. A parallelogram is on the right. The design is composed of various geometric shapes and lines, all rendered in a high-contrast, black and white style.

Manual de Referencia Técnica

VIII.1.- REFERENCIA TECNICA DE GPCASE

GPCASE es un sistema que por sus características requirió un diseño modular por los diversos tipos de funciones que se diseñaron para las distintos procesos requeridos.

La programación en Windows en especial para el Microsoft PWB C/C++ 7.0 y el Windows Maker Pro. 4.0 permite estructurar todo el código modularmente de forma natural, esto aunado al análisis y diseño respectivos resultó en la definición de los siguientes módulos:

GLOBALES.H
 GLOB_EXT.H
 GP_BTRV.C
 GP_CODE.C
 GP_DEF.H
 GP_DI.G.C
 GP_DLGM.C
 GP_MAIN.C
 GP_MSG.C
 GP_SSTBF.C
 GP_UDCD.C
 GP_UDF.C
 GP_UDFM.C
 GP_WPROC.C
 GPCASE.C
 GPCASE.DEF
 GPCASE.H
 GPCASE.RC
 SSCOMDLG.H

GLOBALES.H

(Ver listado en el apéndice)

Contiene todas las variables globales que necesita la aplicación para operar, así como las definiciones de constantes mediante declaraciones define.

GLOB_EXT.H

Este archivo es equivalente al de GLOBALES.H con una diferencia, GLOB_EXT.H tiene las mismas variables pero antecedidas por la palabra *extern* que es reservada de C, *extern* indica que esa variables ya ha sido declarada formalmente en otro sitio y se desea utilizar con su valor actual, evitando duplicarla o tener un nuevo valor.

GP_BTRV.C

(BTRiVe)

Controla y mantiene todas las funciones que se requieren para manejar las bases de datos que se generan por proyecto, con BTRV hacemos referencia a Btrieve que es el manejador de bases de datos que utilizado para desarrollar GPCASE.

GP_CODE.C

(CODE)

Es el módulo que controla todo el flujo de generación de código que realiza GPCASE una vez definido adecuadamente un proyecto.

GP_DEF.H

(DEFIniciones)

Este módulo mantiene el catálogo de prototipos de todas las funciones definidas por el equipo de desarrollo y otras más que son de soporte y es necesario declarar su prototipo para poder utilizarlas,

tal es el caso de las funciones para Btrieve. Debido a la importancia de este archivo, se describe en una sección por separado.

GP_DLG.C

(DiaLoGs)

Contiene toda la información relacionada con los *dialogs* de procesamiento general que se utilizan.

GP_DLGM.C

(DiaLoGs Menu)

Tiene las mismas características que GP_DLG pero asimismo controla otro sector de la aplicación, esta separación fué necesaria debido al tamaño y complejidad de sus respectivas funciones, esto facilitó el manejo del código y la constante actualización de las funciones mismas.

GP_MAIN.C

(Principal)

Crea y define las diferentes clases de ventanas del sistema.

GP_MSG.C

(McSsaGes)

En éste se manejan todos los mensajes de la aplicación.

GP_SSTBF.C

(Spread Sheet Tool Box Functions)

Funciones que emplean librerías del Tool Box Professional.

GP_UDCD.C

(User Defined Code Diagram)

Contiene el código que controla la presentación visual de la navegación de los prototipos, mediante diagramas.

GP_UDF.C

(User Defined Functions)

Como nos sugiere su equivalente en inglés, son las Funciones Definidas por el Usuario, el Usuario es el programador, aquí residen todas las funciones que se debieron construir para diversas tareas necesarias para el desarrollo de GPCASE y obviamente no disponibles en alguna librería de forma accesible.

GP_UDFM.C

(User Defined Functions Menu)

Este es un módulo con los mismos propósitos que GP_UDF, la única diferencia radica en que se encarga de manejar las principales funciones del diseñador de Menús de GPCASE, dada la complejidad del sistema fué necesario manejar dos módulos separados para las funciones que se construyeron, esto facilitó el control tanto de programación como de diseño.

GP_WPROC.C

(Window PROCedure)

En los programas Windows, lo recomendable es que exista un módulo separado que maneje directamente todos los Window Procedures, es decir las funciones que controlan la operación de las ventanas que pertenecen al sistema en cuestión.

GPCASE.C

Contiene a la función MainWndProc que inicia la aplicación y es la que procesa todos los mensajes de WinMain.

GPCASE.H

En éste residen todos los prototipos y definiciones que se generaron por medio de Windows Maker Pro 4.0. Al igual que GP_DEF.H se trata con detalle este archivo por separado.

GPCASE.DEF

(DEFIniciones)

Todas las aplicaciones Windows en su modo general o típico se apoyan en un archivo de definiciones de la aplicación que lleva el nombre de la aplicación y extensión DEF, en nuestro caso particular es

GPCASE.DEF.

En este archivo se incluyen los segmentos de código, datos, *stack* o pila, tipo de archivo ejecutable, se definen asimismo todos los módulos que componen la aplicación y todas las funciones que requieren ser declaradas en la sección de EXPORTS para el correcto funcionamiento de la aplicación.

GPCASE.RC

(ResourCes)

Todas las aplicaciones Windows utilizan recursos que son todos aquellos elementos de funcionalidad general para "moverse" en la aplicación tales como menús, *dialogs*, iconos, *bitmaps* y cursores. Cada uno de estos elementos tienen características muy particulares; por

ejemplo los menús llevan un nombre de cabecera y controlan a su vez un número determinado de ítems, algunos de estos ítems tienen a su vez otros menús y así sucesivamente hasta cierto nivel convencional.

Los cursores por su parte son de diversas formas y cambian su apariencia dependiendo de la posición que guarden en la aplicación para sugerir visualmente que se puede realizar una operación diferente según la forma del cursor en ese instante. Los *bitmaps* son mapas de bits y nos permiten definir figuras a cierto detalle que se manejan como una entidad para ciertas tareas muchas de ellas de animación, presentación, simulación, etc. En cuanto a los iconos mencionaremos que son elementos visualmente pequeños y que por el gráfico que muestran, el usuario puede realizar alguna operación particular que le sugiere el icono, siendo esta operación mucho más simple y veloz con un *mouse*, que su equivalente de entrar a un menú y elegir el ítem que corresponda a realizar la misma operación que con el icono. En la gran mayoría de aplicaciones Windows del mercado, siempre está disponible una barra de iconos que corresponden a operaciones equivalentes menú-ítem y el usuario es entonces libre de operar como desee o se le facilite.

Las características de los recursos se guardan por separado en los archivos correspondientes y el archivo con la extensión RC se encarga de manipularlos, si se analiza un archivo .RC podemos encontrar nombres de archivos con extensiones como estas:

BMP	Bitmaps
CUR	Cursores
DLG	Dialogs

ICO Iconos

Los menús en particular se definen directamente en código dentro del archivo .RC, analizando el archivo notaremos que se sigue el patrón:

Nombre_Recurso

Tipo_Recurso

Archivo_Recurso

Por ejemplo:

FRAMECUR

CURSOR

FIGURA

BITMAP

FIGURA.BMP

VIII.2.- DESCRIPCION DEL ARCHIVO GP_DEF.H

Primeramente se encuentra una lista de definiciones utilizando la forma estándar del Lenguaje C, mediante la directiva al compilador *define*.

El primer grupo describe características de elementos gráficos que se utilizan en GPCASE tales como movimientos del *mouse*, tipo recuadro, tipo línea, tipo campo, tipo texto, seleccionar y deseleccionar un elemento, el grupo es el siguiente:

```
#define TRACK_MOUSE      1
#define MOVE_RECT        2
#define CHANGE_RECT      3
#define TYPE_FIELD       4
#define TYPE_TEXT        5
#define TYPE_FRAME       6
#define SELECT_RECT      7
#define DESELECT_RECT    8
#define CHANGE_TEXT      9
#define SIMULACION       10
```

El segundo grupo indica los límites manejados para los campos, renglones y columnas de los elementos que define el diseñador en GPCASE, así como una longitud máxima de archivo para guardar descripciones, el grupo es el siguiente:

```
#define MAX_CAMPO        100
#define MAX_LINEA        250
#define MAX_FILE_SIZE    3096
#define MAX_ROW          24
#define MAX_COL          80
#define DESING_ROW       21
#define DESING_COL       78
```

El tercer grupo controla las características de la *barra de status* y la barra de herramientas o *toolbar*, así como las dimensiones de un *edit* para descripciones, el grupo es el siguiente:

```
#define STATUSBAR_HEIGHT 22
#define TOOLBAR_WIDTH    36
#define TOOLBAR_HEIGHT   37
#define NEW_ORG          45
#define EDIT_HEIGHT      40
```

El cuarto grupo presenta límites de valores máximos de *dialogs* en el sistema, longitud de la llave de identificación, nombre del proyecto, fechas, mensajes, encabezados, pies de página y texto, el grupo es el siguiente:

```
#define MAX_DIALOGS      40
#define GP_KEY_ID        10
#define GP_NAME_PROJECT  50
#define GP_DATE          10
#define GP_NAME          36
#define GP_MSG           80
#define GP_HEADER        80
#define GP_FOOTER        80
#define GP_TEXTO         40
```

Viene a continuación el grupo de definiciones para el control de la estructura del diseñador de menús, varias de ellas son de control interno o auxiliares, las restantes indican limitantes de la longitud del nombre de un menú y de un ítem, el máximo de caracteres en el menú que se diseña por proyecto, el máximo de proyectos, el máximo de menús por proyecto y el máximo de ítems por cada menú, la longitud estándar

para definir un archivo y la longitud máxima de una definición de ruta o *path*, el grupo es como sigue:

```
#define DB_MODULE      3
#define DB_PROJECT     4
#define DB_MENU        5
```

/****** Definitions *****/

Definitions used for control the global structure

stMENU where

the user is defining his menu prototype */

```
#define INIT_IDENTIFIERS 3000 // Here begins
the identifiers

#define MAX_ITEMS 20 // Items go from 0 to 19
#define OPTION_IDENTIFIER 20 // Opciones will
have the values 3000, 3020, 3040, ...

#define MAX_OPTIONS_MAINMENU 10 // Main
Menu Options

#define MAX_TEXT_MENU 12 // Max. of Text
in Menus

#define MAX_TEXT_ITEM 26 // Max. of Text in
Items

#define MAX_SS 56 // Max. of characters in ss[]
#define MAX_CHARS_MAINMENU 80 // Max. of
characters in MainMenu

#define MAX_PROJECTS 10 // Max. of Projects
#define MAX_LIST_NAME 40 // Max. of ListName
Length in HelpDialog

#define PATH_FILE_LENGTH 200 // Length of
Path+FileName

#define FILE_NAME_LENGTH 15 // FileName
Length 8.3
```

A continuación se encuentran los identificadores para las bases de datos, variables, diseños, módulos, proyectos y menús, que son:

/****** define's de bases de datos *****/

```
#define DB_VARIABLE      1
#define DB_DESIGN       2
```

Sigue una agrupación denominada *Functions' Prototypes* en donde residen todos los prototipos de las funciones que se generaron, así como también aquellas que requieren la definición de su prototipo para utilizarlas como las del manejador *Btrieve*. Las subagrupaciones mantienen los prototipos de las funciones según el módulo en donde se encuentre su definición (cuerpo de la función).

Para tener una idea clara del propósito de estas funciones, se describe a continuación cada una de ellas:

El primer grupo es del módulo GP_MAIN y son los prototipos requeridos para el uso de ventanas de la aplicación.

/** FUNCIONES DECLARADAS EN GP_MAIN **/

```
BOOL MyRegisterClass( HANDLE hInstance );
HWND fnCreateWindows ( HANDLE hInstance, int
nCmdShow );
HWND CreateWindowDiagram ( HANDLE hInstance );
HWND fnCreateWindowDisenoMenu( HANDLE
hInstance );
BOOL fnInitMenuBarBitmaps(HWND hWnd );
```

BOOL MyRegisterClass(HANDLE hInstance);

Registra clases de ventanas particulares.

```
HWND fnCreateWindows ( HANDLE hInstance,
int nCmdShow);
```

En el momento que se carga esta función se crean todas las ventanas que podemos apreciar en la aplicación de GPCASE.

HWND CreateWindowDiagram (HANDLE hInstance);

Esta es una ventana en la cuál se mostrará la estructura de la navegación que va creando el diseñador conforma define más y más elementos en su proyecto.

HWND fnCreateWindowDisenoMenu (HANDLE hInstance);

Como sabemos, GPCASE cuenta con una ventana especial para el diseño del menú del proyecto en cuestión, es una ventana que se crea con esta función y al inicio de GPCASE aparece en la pantalla, pero existe la posibilidad de desaparecerla o aparecerla cuando se desee mediante el icono que semeja un menú desplegado o mediante el ítem *Diseñador* del menú *Menús* de GPCASE.

BOOL fnInitMenuBarBitmaps(HWND hWnd);

Inicializa los parámetros necesarios para poder visualizar el bitmap del diseñador de menús.

Como recordarán GP_WPROC mantiene información del uso de ventanas adicionales a la ventana principal de la aplicación, en este caso sólo son dos:

/ FUNCIONES DECLARADAS EN GP_WPROC **/**

```
LONG FAR PASCAL WndProcDiagram (HWND
hWnd, UINT message, UINT wParam, LONG lParam);
LONG FAR PASCAL WndProcDisenoMenu
(HWND hWnd, UINT message, UINT wParam, LONG
lParam);
```

LONG FAR PASCAL WndProcDiagram (HWND hWnd, UINT message, UINT wParam, LONG lParam);

Esta es la ventana que mostrará el diagrama de la navegación que el diseñador va construyendo al modificar o agregar elementos a su proyecto y este prototipo se requiere para la correcta parametrización y verificación de valores en cada instancia de la misma.

LONG FAR PASCAL WndProcDisenoMenu (HWND hWnd, UINT message, UINT wParam, LONG lParam);

Esta es la ventana del diseñador de menús, en ellas van apareciendo todos los menús creados y sus respectivos ítems, este prototipo se requiere para la correcta parametrización y verificación de valores en cada instancia de la misma.

El siguiente subgrupo es del módulo GP_SSPRO para el control de las funciones de Spread Sheet de Tool Box y otras tantas creadas por nosotros para soporte de las primeras:

/ FUNCIONES DECLARADAS EN GP_SSPRO **/**

```
HWND TBCreateStatusBar(HWND hWnd);
HWND TBCreateToolBar(HWND hWnd);
void fnCreateWindowsToolBoxSS (HWND hWnd);
HWND TBCreateStatusIclp(HWND hWnd);
HWND TBCreateToolBox(HWND hWnd);
void fnInicializaSpreadSheet(HWND hSDlg);
void fnInsertaRegistroSS(HWND hWnd, int iRen);
void fnGetDateSystem (char *sDate);
```

HWND TBCreateStatusBar(HWND hWnd);

Crea la barra de estado actual de la aplicación o de la operación en proceso.

HWND TBCreateToolBar(HWND hWnd);

Crea una barra de iconos que realizan la misma función que los parámetros más frecuentemente usados en el menu principal.

void fnCreateWindowsToolBoxSS (HWND hWnd);

Esta función se diseñó para crear las ventanas SpreadSheet, las barras de iconos y las barras de mensajes de ayuda que presentará información de soporte dependiendo la operación, proceso o actividad que se realice, en otros casos aquí habrá información adicional del elemento que se encuentre en pantalla.

HWND TBCreateStatusHelp(HWND hWnd);

Esta función crea otra barra, pero en ésta se monitorea el estado actual de la ayuda.

HWND TBCreateToolBox(HWND hWnd);

Permite la creación del ToolBox que contiene los iconos que describen las funciones más empleadas en el diseño de módulos.

void fnInicializaSpreadSheet(HWND hSDlg);

Esta función la definimos para inicializar la Spread Sheet que presenta la información del proyecto o reportes, apareciendo en un formato de hoja de cálculo semejante al Microsoft Excel y con cierta funcionalidad del mismo pero con las limitantes y alcances acordes a los propósitos de GPCASE, pues la intención es presentar la información en un formato

familiar y agradable y no crear una hoja de cálculo con las facilidades ya conocidas.

void fnInsertaRegistroSS(HWND hWnd, int iRen);

Añade información de la Base de Datos al Spread Sheet.

void fnGetDateSystem (char *sDate);

Para llevar un control realmente automático de las actividades dentro de GPCASE y tener un seguimiento de las mismas se necesita conocer siempre la fecha del sistema, esta función realiza esta tarea y reporta su resultado en donde se requiera.

Bien, vayamos ahora al análisis del módulo GP_UDF, que mantiene funciones construidas por el equipo de desarrollo para diversas tareas.

```

/** FUNCIONES DECLARADAS EN GP_UDF **/
void fnFixWindow (HWND hWnd);
void fnOkMsgBox (char *szCaption, char *szFormat,
...);
void TrackMouse (HWND hwnd, POINT pt, int ixClient,
int iyClient, RECT rResult);
void TrackMouseH (HWND hwnd, POINT pt, int
ixClient, int iyClient, RECT *Rect);
void DrawSelect (HDC hdc, BOOL fDraw, float
fxProporcion, float fyProporcion, RECT
rcClip);
void PASCAL NormalizeRect (RECT *pre);
void fnMapMode (HDC hDC);
long FileLength (HANDLE hFile);
BOOL ReadFile (HANDLE hWndEdit,
POFSTRUCT pof, char *szFileName, BOOL
bAskName);

```

```

BOOL WriteFile (HANDLE hWndEdit,
int fnGetField (char *sCadena, char *sSubcadena, int
*);
void fnChargeVariables(char *sArchivo);
BOOL fnShowDataDlgVariable (HWND hDlg);
void fnGetDataDlgVariable (HWND hDlg);
void fnInitDlgVariable (HWND hDlg, int listBox, int
iCheck);
int fnOpenFile (char *sArchivo);
int fnShowFileToEdit (int iTipoArchivo, int
*iStatus);
int fnGetFileFromEdit (int iTipoArchivo, int
*iStatus);
void fnDestroyResources();
void fnCreateResouces ();
void fnMarkSelect (HDC hDC);
void fnUnchecksMarksInit (HWND hWnd);
void fnCheckMarkID (int IDM);
void fnMoveRect (HWND hWnd, LONG lParam);
void fnUnSelChange (HWND hWnd, LONG
lParam);
void fnSetFrame (HWND hWnd, LONG lParam);
void fnInvalidateRect (HDC hDC, BOOL b);
void fnSendInfoStatusBar ();
void fnGetNameID (char *sKeyID, char
*sNameID);
void fnShowDataDlgModule (HWND hDlg);
void fnGetDataDlgModule (HWND hDlg);
void fnShowDataDlgProject (HWND hDlg);
void fnGetDataDlgProject (HWND hDlg);
void fnDrawGrid (HDC hdc);
int fnVLtoPL (int iPoint, int iOrientation);
int fnPLtoVL (int iPoint, int iOrientation);
void fnHideDialogItems (HWND hDlg);
void fnCreateMenuSimulation (HWND hDlg);
void fnProcessControl();

```

```

void fnInitNameMenuControl(HWND hDlg);
void fnFindNewName(char *sName, HWND hDlg);

```

A continuación se da una breve explicación de cada función en este módulo.

void fnFixWindow (HWND hWnd);

Se ajusta la ventana al máximo permitido en el modo de video presente .

void fnOkMsgBox (char *szCaption, char *szFormat, ...);

Un usuario de aplicaciones Windows recordará que es muy frecuente la aparición de ventanas de mensajes que especifican datos adicionales, condiciones de error, avisos de problemas potenciales, etc. Generalmente estas ventanas tienen un *pushbutton* OK para cerciorarse de que el usuario tenga conocimiento de las consecuencias de cierto evento, esta función cubre las necesidades del sistema.

void TrackMouse (HWND hwnd, POINT pt, int ixClient, int iyClient, RECT *rResult);
void TrackMouseH (HWND hwnd, POINT pt, int ixClient, int iyClient, RECT *Rect);

Ambas funciones son sumamente utilizadas en GPCASE ya que reportan el movimiento del *mouse* sobre todo en la parte de diseño de pantallas y reportes, en donde el uso del *mouse* es imprescindible.

void DrawSelect(HDC hdc,BOOL fDraw, float fxProporcion, float fyProporcion, RECT rcClip);
DrawSelect dibuja en la pantalla el rectángulo seleccionado con sus dimensiones.

void PASCAL NormalizeRect (RECT *pre);
 NormalizeRect es de una gran utilidad en el diseño en conjunto con el *mouse*, en muchas ocasiones al crear marcos con el *mouse* no siempre se hace a la derecha y hacia abajo, sino al contrario a la izquierda y hacia arriba, lo cual invierte las coordenadas de visualización desde el punto de origen, esta función se encarga de detectar esta condición e invertir las coordenadas para trabajar correctamente.

void fnMapMode(HDC hDC);

Adapta el modo de video para proporcionar adecuadamente en pantalla la imagen.

long FileLength (HANDLE hFile);

Devuelve el tamaño del archivo especificado en bytes, sólo se pasa el manejador del archivo.

BOOL ReadFile (HANDLE hWndEdit, POFSTRUCT pof, char *szFileName, BOOL hAskName);

Su propósito es leer el contenido de un archivo externo y vaciar su contenido en un control *edit* para poder manejarlo dentro de Windows.

BOOL WriteFile (HANDLE hWndEdit, POFSTRUCT pof, char *szFileName, BOOL hAskName);

Al contrario que ReadFile, esta función lee el contenido que se encuentra en un control *edit* y lo guarda en el archivo que se le especificó.

int fnGetField (char *sCadena, char *sSubcadena, int *i);

fnGetField obtiene el campo especificado de una cadena, los campos deben estar separados en esa cadena mediante un caracter *pipe*.

void fnChargeVariables(char *sArchivo);

Esta es una función de transacción, su objetivo es cargar las nuevas variables de un archivo externo a la base de datos Variable.dat.

BOOL fnShowDataDlgVariable (HWND hDlg);

Despliega la información de la variable basado en la variable *dialog* respectiva.

void fnGetDataDlgVariable (HWND hDlg);

Obtiene la información de la variable del *dialog* y la actualiza en la base de datos de variable.dat.

void fnInitDlgVariable (HWND hDlg, int iListBox, int iCheck);

Inicializa la *listbox* del *dialog* especificado con la información de los ID's de las variables que se están manejando.

int fnOpenFile (char *sArchivo);

Utiliza un *Common Dialog* para obtener el nombre del archivo.

int fnShowFileToEdit (int iTipoArchivo, int *iStatus);

Despliega la información del archivo especificado en una ventana que es un *edit*.

int fnGetFileFromEdit (int iTipoArchivo, int *iStatus);

En forma inversa a `fnShowFileToEdit`, esta función toma el contenido del control `edit` y lo almacena en un archivo.

`void fnDestroyResources();`

Esta función resulta sumamente importante ya que libera la memoria que ocupan los recursos utilizados por cada aplicación Windows.

`void fnCreateResources ();`

Define todos los recursos requeridos para diseñar.

`void fnMarkSelect (HDC hDC);`

Coloca pequeños rectángulos en las esquinas del objeto seleccionado que se desea borrar y pone la bandera `status=1`, con lo que el objeto será borrado al presionar la tecla [DEL] y `status=1`.

`void fnUnchecksMarksInit (HWND hWnd);`

Quita la marca de selección por default de un item en un menu de persiana.

`void fnCheckMarkID (int IDM);`

Marca un item específico en la barra de menú.

`void fnMoveRect (HWND hWnd, LONG lParam);`

Mueve un objeto de la pantalla de diseño y actualiza las nuevas coordenadas en la base de datos `diseño.dat`.

`void fnUnSelChange (HWND hWnd, LONG lParam);`

Permite la selección, deselección o manejo del texto de un objeto.

`void fnSetFrame (HWND hWnd, LONG lParam);`

Coloca un recuadro dentro del *client area* de la ventana.

`void fnInvalidateRect (HDC hDC, BOOL B);`

Marca las áreas de dibujo dentro de una ventana.

`void fnSendInfoStatusBar ();`

Manda la información sobre el proceso a la barra del *status*.

`void fnGetNameID (char *sKeyID, char *sNameID);`

Devuelve exclusivamente el nombre del `sKeyID`.

`void fnShowDataDlgModule (HWND hDlg);`

Despliega la información que corresponde al módulo basada en el *dialog* de ese módulo.

`void fnGetDataDlgModule (HWND hDlg);`

Se obtiene la información presente en el *dialog* del módulo con lo que actualiza la base de datos `module.dat`.

`void fnShowDataDlgProject (HWND hDlg);`

Despliega la información que corresponde al proyecto basada en el *dialog* de ese proyecto.

`void fnGetDataDlgProject (HWND hDlg);`

Se obtiene la información presente en el *dialog* del proyecto con lo que actualiza la base de datos `project.dat`.

`void fnDrawGrid (HDC hdc);`

Esta función es sumamente útil ya que presenta una cuadrícula en la pantalla de diseño que facilita el trabajo al seleccionar y colocar objetos en pantalla.

int fnVLtoPL (int iPoint, int iOrientation);

Realiza una conversión de coordenadas visuales a coordenadas lógicas. Si el parámetro iOrientation=1 convierte las coordenadas del eje X, si iOrientation=0 entonces hace lo propio con el eje Y.

int fnPLtoVL (int iPoint, int iOrientation);

Hace exactamente lo contrario a fnVLtoPL, en este caso convierte de coordenadas lógicas a visuales. El parámetro iOrientation tiene el mismo propósito que en fnVLtoPL.

void fnHideDialogItems (HWND hDlg);

Esconde los *dialogs* usados dentro de la ventana de simulación para acelerar el proceso de despliegue de módulos.

void fnCreateMenuSimulation (HWND hDlg);

Crea el menú que corresponde al proyecto para su simulación.

void fnProcessControl();

Desensambla automáticamente de la llave de: DESING . (Enmascaramiento)

void fnInitNameMenuControl(HWND hDlg);

Inicializa el Dialog de módulos con el nombre del item del menú con el cual se estableciera la liga.

void fnFindNewName(char *sName, HWND hDlg);

Inicializa el Dialog de módulos con un nombre por default debido a que el nombre del menú con el cual se estableciera la liga. si éste esta repetido.

Eso fué GP_UDF, ahora el análisis del otro módulo que necesariamente se tuvo que separar para controlar la modularización de GPCASE, su nombre es GP_UDFM.

/ FUNCIONES DECLARADAS EN GP_UDFM **/**

```
void fnInitMenu(void);
void fnInitCBAddMenuDlg(HWND hWnd);
void fnAssignMenuItemToMENU(void);
void fnSaveStMENU(void);
void fnLoadStMENU(void);
void fnDisplayStMenu(void);
BOOL fnInitLBEEditMenuDlg(HWND hDlg, int iMenu);
BOOL fnInitLBEEditItemDlg(HWND hDlg);
void fnClearMainMenu(void);
void fnLoadProjectMenu(void);
int fnAvailableCharsMainMenu(HWND hDlg);
void fnCreateMenuSimulation (HWND hDlg);
void fnDestroyMenuSimulation (HWND hDlg);
void fnAssignSimulateMenuItemToMENU(HWND hDlg);
void fnEnableWindow (HWND, BOOL, ...);
void fnLoadMenu();
void fnEnableSalvarMenu ();
void fnHabilitarRenombrarMenu (HWND hDlg);
void fnRenombrarMenu(HWND hDlg);
void fnBorrarMenus(void);
void fnRenombrarItem(HWND hDlg);
void fnHabilitarRenombrarItem (HWND hDlg);
void fnBorrarItem(void);
void fnInitLBIHelpDlg(char *sArchivo,HWND hDlg);
```

```
void fnSearchStringFile(HWND hDlg, char *sArchivo,
char *sCadena);
```

A continuación una breve explicación de cada función de este módulo.

```
void fnInitMenu(void);
```

Inicializa la estructura global para los menús generados por el usuario. Asigna 0's a los ID's y \0 a las cadenas de caracteres en stMENU[].

```
void fnInitCBAddMenuDlg(HWND hWnd);
```

Inicializa la *Combo box* del dialog ADDMENU con números desde 0 hasta MAX cada vez que se va a crear o agregar un nuevo menú al proyecto.

```
void fnAssignMenuItemToStMENU( void );
```

Asigna los valores apropiados a la estructura stMENU cada vez que se realiza una modificación al menú del proyecto que diseña el usuario, de forma simultánea actualiza los ID's y los nombres de cada ítem para detectarlos correctamente en un módulo de asignación de procesos. Los ítems y los menús que no son utilizados serán nulos "\0" en las cadenas y 0's para los ID's.

```
void fnSaveStMENU(void);
```

Salva la estructura stMENU a disco con el nombre del proyecto y extensión MNU.

```
void fnLoadStMENU(void);
```

Carga del disco la estructura stMENU basándose en el ID del proyecto.

```
void fnDisplayStMenu(void);
```

Despliega en pantalla los menús que están registrados en la estructura stMENU.

```
BOOL fnInitLBEditMenuDlg(HWND hDlg, int iMenu);
```

Inicializa la *listbox* del dialog EDITMENU con los menús diseñados hasta entonces en el proyecto.

```
BOOL fnInitLBEditItemDlg(HWND hDlg);
```

Inicializa la *listbox* del dialog EDITITEM con los ítems que corresponden al menú seleccionado.

```
void fnClearMainMenu(void);
```

Libera la memoria utilizada por los menús y limpia la barra del menú principal cuando es necesario.

```
void fnLoadProjectMenu(void);
```

Inicializa las variables globales para comenzar un nuevo menú para el proyecto. Simplemente llama a la función fnInitMenu y luego llama a la función LoadStMenu.

```
int fnAvailableCharsMainMenu(HWND hDlg);
```

Nos regresa el número de caracteres de que disponemos para agregar, insertar o renombrar un menú dentro del menú principal del proyecto. Esto es con el fin de limitar la cantidad de menús y/o caracteres del menú principal y hacerlo lo más transportable posible a la mayoría de las plataformas al generar el código.

```
void fnCreateMenuSimulation (HWND hDlg);
```

Crea el menú del proyecto correspondiente para la simulación del prototipo.

void fnDestroyMenuSimulation (HWND hDlg) ;

Destruye el menú del proyecto utilizado para la simulación, logrando con ello liberar la memoria para que la ocupen otros recursos.

void fnAsignSimulateMenuItemTostMENU(HWND hDlg) ;

Funciona exactamente como **fnAsignMenuItemTostMENU** pero con la diferencia de que está diseñada con ciertas características cuando la estructura **stMENU** trabaja en la simulación del prototipo.

void fnEnableWindow (HWND, BOOL ,...);

Habilita los controles utilizados en los *dialogs* de tal forma que dependiendo el tipo de proceso y operaciones, algunos controles aparecerán habilitados o no para operar. Su versatilidad radica en que puede manejar una cantidad variable de *handles* de controles y terminar con -1 para indicar fin de parámetros.

void fnLoadMenu();

Carga el menú del proyecto pero se utiliza sólo para ciertas condiciones.

void fnEnableSalvarMenu ();

Es una función que maneja una bandera global y que dependiendo de su valor, la opción *Salvar Menú* del menu de GPCASE se habilitará o no para actualizar el menú diseñado.

void fnHabilitaRenombrarMenu(HWND hDlg) ;

Su objetivo es habilitar el *push button Renombrar* en el *dialog Edición de Menús* sólo si el dato del usuario

en el *edit* empieza con letra o número, los espacios al inicio no son válidos.

void fnRenombrarMenu(HWND hDlg) ;

En el momento que se presiona el *push button Renombrar* del *dialog Edición de Menús* se actualiza el menú tanto en la *listbox* como en el menú del proyecto.

void fnBorrarMenus(void) ;

Elimina el menú que se encuentre seleccionado en ese momento, será aquél que esté reflejado en el *edit Editor*.

void fnRenombrarItem(HWND hDlg) ;

En el momento que se presiona el *push button Renombrar* del *dialog Edición de Items* se actualiza el ítem tanto en la *listbox* como en el menú del proyecto.

void fnHabilitaRenombrarItem(HWND hDlg) ;

Su objetivo es habilitar el *push button Renombrar* en el *dialog Edición de Items* sólo si el dato del usuario en el *edit* empieza con letra o número, los espacios al inicio no son válidos.

void fnBorrarItem(void) ;

Eliminará el ítem reflejado en *Editor* del menú especificado en el *edit Menú*.

void fnInitLBHelpDlg(char *sArchivo,HWND hDlg) ;

Inicializa el list Box del *dialog* de Ayuda.

void fnSearchStringFile(HWND hDlg,char *sArchivo, char *sCadena) ;

Obtiene la información que se presentará en el *dialog* de ayuda.

Ahora pasemos a otro módulo que sólo lleva una declaración

```
/** FUNCIONES DECLARADAS EN GP_DLG **/
```

El manejo de las bases de datos es un proceso delicado y trascendente para GPCASE, por lo cual se utilizó un módulo exclusivo para el manejo de las funciones de Btrieve que es el manejador que utilizamos. A continuación el fragmento de GP_DEF.H donde se encuentran:

```
/** FUNCIONES DECLARADAS EN GP_BTRV **/
// Prototypes for Btrieve Management
int far pascal BTRCALL (int, char far *, char far *,
    unsigned far *, char far *, unsigned char, char);
int far pascal WBTRVINIT (char far *);
int far pascal WBRQSHELLINIT (char far *);
int far pascal WBTRVSTOP (void);
int BTRVSTOP (void);
int BTRV (int, char far *, char far *, unsigned far *, char
    far *, int);
void fnError (int iStatus, char *sBase);
int fnBtrv (int iBase, int iOperacion, int iLlave);
int fnBtrvBaseVariable (int iOperacion, int iLlave);
int fnBtrvBaseDesign (int iOperacion, int iLlave);
int fnBtrvBaseModule (int iOperacion, int iLlave);
int fnBtrvBaseProject (int iOperacion, int iLlave);
int fnBtrvBaseMenu (int iOperacion, int iLlave);
```

Procedamos entonces al análisis particular de cada función:

```
int far pascal BTRCALL (int, char far *, char far
*, unsigned far *, char far *, unsigned char, char);
```

```
int far pascal WBTRVINIT (char far *);
int far pascal WBRQSHELLINIT (char far *);
int far pascal WBTRVSTOP (void);
```

Estos fueron los prototipos de las funciones BTRCALL, WBTRVINIT, WBRQSHELLINIT, WBTRVSTOP que se deben declarar para poder utilizar Btrieve en Windows.

```
int BTRVSTOP (void);
```

Sólo regresa el código de Btrieve al cesar su operación.

```
int BTRV (int, char far *, char far *, unsigned far
*, char far *, int);
```

Se utiliza para realizar una interface definida del manejo de las bases de datos apoyándose a su vez en la función BTRCALL().

```
void fnError (int iStatus, char *sBase);
```

El manejo de errores es muy importante al manejar datos, esta función lo permite de tal manera que evite que la aplicación trabaje inadecuadamente mediante la visualización de mensajes de advertencia para corregir cualquier situación.

```
int fnBtrv (int iBase, int iOperacion, int iLlave);
```

Esta es la función que dependiendo los parámetros que se pasen, elegirá la función correspondiente a cada base de datos para su manejo.

```
int fnBtrvBaseVariable (int iOperacion, int iLlave
);
```

Maneja directamente los procesos sobre la base de datos Variable.dat.

```
int fnBtrvBaseDesign (int iOperacion, int iLlave);
```

Maneja directamente los procesos sobre la base de datos Design.dat.

int fnBtrvBaseModule (int iOperacion,int iLlave);

Maneja directamente los procesos sobre la base de datos Module.dat.

int fnBtrvBaseProject (int iOperacion,int iLlave);

Maneja directamente los procesos sobre la base de datos Project.dat.

int fnBtrvBaseMenu (int iOperacion, int iLlave);

Maneja directamente los procesos sobre la base de datos Menu.dat.

VIII.3.- DESCRIPCION DEL ARCHIVO GPCASE.H

A continuación se presenta la descripción de las definiciones y prototipos de funciones que se encuentran en GPCASE.H que junto con GP_DEF.H permiten el control de toda la aplicación.

```
//Filename: GPCASE.H
//"GPCASE" Generated by WindowsMAKER
Professional
//Author:
```

Estas definiciones soportan viejas versiones de compiladores de Windows.H. La generación de estas definiciones pueden desactivarse a través del Setup de generación de código de Windows Maker Pro:

```
typedef unsigned int UINT;
#ifndef WINAPI
typedef HANDLE HINSTANCE;
#endif

#ifndef WM_SYSTEMERROR
#define WM_SYSTEMERROR 0x001F
#endif
#ifndef WM_QUEUE_SYNC
#define WM_QUEUE_SYNC 0x0023
#endif
#ifndef WM_COMMNOTIFY
#define WM_COMMNOTIFY 0x0044
#endif
#ifndef WM_WINDOWPOSCHANGING
#define WM_WINDOWPOSCHANGING
0x0046
#endif
```

```
#ifndef WM_WINDOWPOSCHANGED
#define WM_WINDOWPOSCHANGED 0x0047
#endif
#ifndef WM_POWER
#define WM_POWER 0x0048
#endif
#ifndef WM_DROPFILES
#define WM_DROPFILES 0x0233
#endif
#ifndef WM_PALETTEISCHANGING
#define WM_PALETTEISCHANGING 0x0310
#endif
```

A continuación se listan todos los archivos tipo *header* que comprenden la aplicación, entre otros se encuentran:

- Archivos de definiciones y variables utilizadas como GP_DEF.H
- Todos los archivos que contienen las definiciones de los controles de los *dialogs*.
- Los archivos de definiciones para controles de *dialogs* de tipo general de Windows.

```
#include "BANCO01.H"
#include "BANCO02.H"
#include "COMMDLG.H"
#include "GP_DEF.H"
#include "SSCOMDLG.H"
#include "TOOLBOX.H"
#include "VARIABLE.H"
```

```
#include "TEXT0.H"
#include "PANTREP.H"
#include "DEFPROY.H"
#include "LINKER.H"
#include "ADDTTEM.H"
#include "ADDMENU.H"
#include "EDITTEM.H"
#include "EDITMENU.H"
#include "HELP.H"
#include "PASSWORD.H"
#include "SECURITY.H"
```

Las siguientes son las declaraciones de dos variables globales tipo *extern* que dan acceso a los manejadores de código en todos los módulos:

```
extern HINSTANCE hInst;
extern HWND MainhWnd;
```

A continuación están las definiciones de constantes para mensajes de errores:

```
#define BLD_CannotRun 4000
#define BLD_CannotCreate 4001
#define BLD_CannotLoadMenu 4002
#define BLD_CannotLoadIcon 4003
#define BLD_CannotLoadBitmap 4004
#if !defined(THISISBLDR)C
```

Ahora se incluye el archivo Windeb.H que permite el rastreo de código.

```
#define windeb_none
#include <WINDEB.H>
```

El siguiente es el listado de prototipos de funciones que se generaron por medio de Windows Maker Professional. Para simplificar la explicación aquí se

menciona un patrón de notación en el cual se basa la construcción y objetivo de las funciones:

Existen funciones definidas por el usuario, que son aquellas que los desarrolladores construyen con ayuda de Windows Maker para implantar algún algoritmo, al hacerlo de esta forma Windows Maker genera el identificador de la función bajo cierta notación insertando el nombre particular. La notación es la siguiente:

NombreDeFuncion

Es el nombre de la función que se sustituye en los patrones siguientes:

BLDNombreDeFuncion()

Función de librería controlada por Windows Maker.

BLD_NombreDeFuncionUDCFunc()

Función definida por el usuario (creada por un programador)

Los siguientes tres tipos de funciones trabajan en conjunto para procesar los eventos de los diálogos:

BLD_NombreDeFuncionDlgProc()

Procedimiento para proceso de *dialogs*.

BLD_NombreDeFuncionDlgFunc()

Función para procesamiento de controles de un *dialog*.

BLD_NombreDeFuncionDlgDefault()

Función que canaliza todos los mensajes recibidos que no se decidieron procesar del *dialog*.

Con estas consideraciones se describe más fácilmente cada función .

BLD_WM_MensajeMsg()

Función que procesa el *Mensaje* indicado.

```
int PASCAL WinMain( HINSTANCE hInstance,
HINSTANCE hPrevInstance, LPSTR lpCmdLine,
int nCmdShow );
```

Esta es la función principal de la aplicación, es aquella que empieza la ejecución del código, equivale en este caso a la función *main()* del lenguaje C.

```
LONG FAR PASCAL BLDMainWndProc( HWND
hWnd, UINT message, UINT wParam, LONG
lParam );
```

Esta es la función que procesa todos los mensajes que llegan a *WinMain*.

```
LONG FAR PASCAL BLDDefWindowProc(
HWND hWnd, UINT message, UINT wParam,
LONG lParam );
```

Esta función recibe los mensajes de la ventana principal y lo canaliza a una función más específica para su procesamiento particular.

```
BOOL BLDKeyTranslation( MSG *pMsg );
```

Traslada códigos de teclas oprimidas para *dialogs modeless*.

```
BOOL BLDInitApplication( HANDLE hInst,
HANDLE hPrev, int *pCmdShow, LPSTR lpCmd );
BOOL BLDExitApplication( void );
```

```
HWND BLDCreateClientControls( char
*pTemplateName, FARPROC lpNew );
```

Crea los controles del *Client area* de la ventana en cuestión.

```
BOOL BLDInitMainMenu( HWND hWnd );
```

Inicializa la estructura que maneja el menú de cada proyecto.

```
BOOL BLDMenuCommand( HWND hWnd, UINT
message, UINT wParam, LONG lParam );
```

Procesa los mensajes captados canalizándolos a una función más específica.

```
BOOL BLDRRegisterClass( HANDLE hInstance );
```

Permite el registro de clases adicionales de ventanas.

```
HWND BLDCreateWindow(HANDLE hInstance );
```

Crea la ventana que se le pase por su manejador o *handle*.

```
int BLDDisplayMessage(HWND hWnd, unsigned
nMsg, char *pContext, int iType );
```

Despliega los mensajes según el evento lo amerite.

```
BOOL BLDSwitchMenu( HWND hWnd, char
*pTemplateName );
```

Permite el cambio de un menú a otro con todos los parámetros necesarios para evitar conflictos.

```
BOOL BLDDrawBitmap (
LPDRAWITEMSTRUCT lpDrawItem, char
*pBitmapName, BOOL bStretch);
```

Dibuja los bitmaps que se utilizarán en la aplicación.

```
BOOL BLDDrawIcon( LPDRAWITEMSTRUCT
lpDrawItem, char *pIconName );
```

Dibuja los iconos que se utilizarán en la aplicación.

void BLDMoveWindow(HWND hWnd, int x, int y, int nWidth, int nHeight, BOOL bRepaint);

Controla el movimiento de las ventanas, considerando su posición, dimensiones, etc.

BOOL BLDInitMenuBitmaps(HWND);

Coloca bitmaps en menús

BOOL BLDDeleteMenuBitmaps(void);

Borra los bitmaps utilizados en los menús.

BOOL BLD_CargaExternaUDCFunc(HWND hWnd,UINT message,UINT wParam, LONG lParam);

Carga las variables de un formato distinto y se desean aprovechar para diseñar en GPCASE.

BOOL FAR PASCAL BLD_BancodeVariablesDlgProc(HWND hWnd,UINT message , UINT wParam, LONG lParam);
int BLD_BancodeVariablesDlgFunc(HWND hWnd,UINT message, UINT wParam, LONG lParam);
BOOL BLD_BancodeVariablesDlgDefault(HWND hWnd, UINT message, UINT wParam, LONG lParam);

Procesamiento de los eventos del *dialog* Banco de Variables.

LONG BLD_WM_CREATEMsg(HWND hWnd,UINT message, UINT wParam, LONG lParam);

Procesa los eventos a realizar justo antes de la creación de una ventana.

LONG BLD_WM_DESTROYMsg(HWND hWnd,UINT message, UINT wParam, LONG lParam);

Procesa los eventos a realizar justo antes de la destrucción de una ventana.

BOOL FAR PASCAL BLD_EdicionDlgProc(HWND hWnd,UINT message , UINT wParam, LONG lParam);

HWND BLD_EdicionDlgFunc(HWND hWnd, UINT message , UINT wParam, LONG lParam);

BOOL BLD_EdicionDlgDefault(HWND hWnd,UINT message, UINT wParam, LONG lParam);

Las tres funciones anteriores trabajan en conjunto para procesar los eventos del *dialog* Edición.

BOOL BLD_MarcoUDCFunc(HWND hWnd,UINT message,UINT wParam, LONG lParam);

Pinta los marcos de las pantallas de captura.

BOOL BLD_MueveUDCFunc(HWND hWnd,UINT message,UINT wParam, LONG lParam);

Mueve objetos en la ventana de diseño.

BOOL BLD_CambiaTamaoUDCFunc(HWND hWnd,UINT message,UINT wParam, LONG lParam);

Cambia el tamaño de objetos en la ventana de diseño.

LONG BLD_WM_LBUTTONDOWNMsg(HWND hWnd, UINT message, UINT wParam, LONG lParam);

Se presionó el botón izquierdo del *mouse*.

LONG BLD_WM_MOUSEMOVEMsg(HWND hWnd, UINT message, UINT wParam, LONG lParam);

Detecta movimiento del *mouse*.

LONG BLD_WM_LBUTTONDOWNMsg(HWND hWnd, UINT message, UINT wParam, LONG lParam);

Se soltó el botón izquierdo del *mouse*.

LONG BLD_WM_COMMANDMsg(HWND hWnd, UINT message, UINT wParam, LONG lParam);

Se solicita proceso de comando.

BOOL FAR PASCAL BLD_AsignaCamposDlgProc(HWND hDlg, UINT message, UINT wParam, LONG lParam);

HWND BLD_AsignaCamposDlgFunc(HWND hWnd, UINT message, UINT wParam, LONG lParam);

BOOL BLD_AsignaCamposDlgDefault(HWND hWnd, UINT message, UINT wParam, LONG lParam);

Las tres funciones anteriores trabajan en conjunto para procesar los eventos del *dialog* Asigna Campos.

LONG BLD_WM_PAINTMsg(HWND hWnd, UINT message, UINT wParam, LONG lParam);

Redibujar una ventana.

BOOL FAR PASCAL BLD_TextoDlgProc(HWND hDlg, UINT message, UINT wParam, LONG lParam);

int BLD_TextoDlgFunc(HWND hWnd, UINT message, UINT wParam, LONG lParam);

BOOL BLD_TextoDlgDefault(HWND hWnd, UINT message, UINT wParam, LONG lParam);

Las tres funciones anteriores trabajan en conjunto para procesar los eventos del *dialog* Texto.

LONG BLD_WM_SIZEMsg(HWND hWnd, UINT message, UINT wParam, LONG lParam);

Cambio de tamaño de una ventana.

LONG BLD_WM_RBUTTONDOWNMsg(HWND hWnd, UINT message, UINT wParam, LONG lParam);

Se presionó el botón derecho del *mouse*.

BOOL BLD_SeleccionarUDCFunc(HWND hWnd,UINT message,UINT wParam, LONG lParam);

Selecciona un objeto en la pantalla de diseño.

BOOL BLD_DeseleccionarUDCFunc(HWND hWnd,UINT message,UINT wParam, LONG lParam);

Quita la selección a un objeto en la pantalla de diseño.

LONG BLD_WM_KEYDOWNMsg(HWND hWnd, UINT message, UINT wParam, LONG lParam);

Procesa la tecla presionada.

BOOL BLD_CambiaTextoUDCFunc(HWND hWnd,UINT message,UINT wParam, LONG lParam);

Cambia el texto en la pantalla de diseño.

BOOL FAR PASCAL BLD_ConfiguracinDlgProc(HWND hDlg, UINT message, UINT wParam, LONG lParam);

HWND BLD_ConfiguracinDlgFunc(HWND hWnd, UINT message, UINT wParam, LONG lParam);

BOOL **BLD_ConfiguracinDlgDefault**(**HWND** hWnd, **UINT** message, **UINT** wParam, **LONG** lParam);

Las tres funciones anteriores trabajan en conjunto para procesar los eventos del *dialog* Configuración.

BOOL **FAR** **PASCAL**
BLD_ProyectoDlgProc(**HWND** hDlg, **UINT** message, **UINT** wParam, **LONG** lParam);
HWND **BLD_ProyectoDlgFunc**(**HWND** hWnd, **UINT** message, **UINT** wParam, **LONG** lParam);
BOOL **BLD_ProyectoDlgDefault**(**HWND** hWnd, **UINT** message, **UINT** wParam, **LONG** lParam);

Las tres funciones anteriores trabajan en conjunto para procesar los eventos del *dialog* Proyecto.

BOOL **BLD_GridUDCFunc**(**HWND** hWnd, **UINT** message, **UINT** wParam, **LONG** lParam);

Presenta o quita la malla de diseño.

BOOL **BLD_EtiquetasUDCFunc**(**HWND** hWnd, **UINT** message, **UINT** wParam, **LONG** lParam);

Controla las etiquetas en las pantallas.

BOOL **FAR** **PASCAL**
BLD_SimulacinDlgProc(**HWND** hDlg, **UINT** message, **UINT** wParam, **LONG** lParam);
int **BLD_SimulacinDlgFunc**(**HWND** hWnd, **UINT** message, **UINT** wParam, **LONG** lParam);
BOOL **BLD_SimulacinDlgDefault**(**HWND** hWnd, **UINT** message, **UINT** wParam, **LONG** lParam);

Las tres funciones anteriores trabajan en conjunto para procesar los eventos del *dialog* que simula el prototipo del proyecto.

BOOL **FAR** **PASCAL** **BLD_LigaDlgProc**(**HWND** hDlg, **UINT** message, **UINT** wParam, **LONG** lParam);

int **BLD_LigaDlgFunc**(**HWND** hWnd, **UINT** message, **UINT** wParam, **LONG** lParam);

BOOL **BLD_LigaDlgDefault**(**HWND** hWnd, **UINT** message, **UINT** wParam, **LONG** lParam);

Las tres funciones anteriores trabajan en conjunto para procesar los eventos del *dialog* Ligas de Módulos.

BOOL **BLD_GrficoCUDCFunc**(**HWND** hWnd, **UINT** message, **UINT** wParam, **LONG** lParam);

Genera el código C gráfico.

BOOL **FAR** **PASCAL**
BLD_AgregarMenDlgProc(**HWND** hDlg, **UINT** message, **UINT** wParam, **LONG** lParam);
int **BLD_AgregarMenDlgFunc**(**HWND** hWnd, **UINT** message, **UINT** wParam, **LONG** lParam);

BOOL **BLD_AgregarMenDlgDefault**(**HWND** hWnd, **UINT** message, **UINT** wParam, **LONG** lParam);

Las tres funciones anteriores trabajan en conjunto para procesar los eventos del *dialog* Agregar Menús al prototipo.

BOOL **FAR** **PASCAL**
BLD_EditarMensDlgProc(**HWND** hDlg, **UINT** message, **UINT** wParam, **LONG** lParam);
int **BLD_EditarMensDlgFunc**(**HWND** hWnd, **UINT** message, **UINT** wParam, **LONG** lParam);
BOOL **BLD_EditarMensDlgDefault**(**HWND** hWnd, **UINT** message, **UINT** wParam, **LONG** lParam);

Las tres funciones anteriores trabajan en conjunto para procesar los eventos del *dialog* Editar menús del prototipo.

BOOL FAR PASCAL

BLD_MenuBarDlgProc(HWND hDlg, UINT message, UINT wParam, LONG lParam);

int BLD_MenuBarDlgFunc(HWND hWnd, UINT message, UINT wParam, LONG lParam);

BOOL BLD_MenuBarDlgDefault(HWND hWnd, UINT message, UINT wParam, LONG lParam);

Las tres funciones anteriores trabajan en conjunto para procesar los eventos del *dialog* que funciona dependiendo del bitmap de generación de menús.

LONG

BLD_WM_WINDOWPOSCHANGEDMsg(HWND hWnd, UINT message, UINT wParam, LONG lParam);

La ventana ha cambiado en su forma.

BOOL FAR PASCAL

BLD_EditarItemsDlgProc(HWND hDlg, UINT message, UINT wParam, LONG lParam);

int BLD_EditarItemsDlgFunc(HWND hWnd, UINT message, UINT wParam, LONG lParam);

BOOL BLD_EditarItemsDlgDefault(HWND hWnd, UINT message, UINT wParam, LONG lParam);

Las tres funciones anteriores trabajan en conjunto para procesar los eventos del *dialog* Editar ítems del menú seleccionado.

BOOL FAR PASCAL

BLD_InsertarMenuDlgProc(HWND hDlg, UINT message, UINT wParam, LONG lParam);

int BLD_InsertarMenuDlgFunc(HWND hWnd, UINT message, UINT wParam, LONG lParam);

BOOL BLD_InsertarMenuDlgDefault(HWND hWnd, UINT message, UINT wParam, LONG lParam);

Las tres funciones anteriores trabajan en conjunto para procesar los eventos del *dialog* que sirve para la inserción de menús en el proyecto.

BOOL FAR PASCAL

BLD_AgregarItemDlgProc(HWND hDlg, UINT message, UINT wParam, LONG lParam);

int BLD_AgregarItemDlgFunc(HWND hWnd, UINT message, UINT wParam, LONG lParam);

BOOL BLD_AgregarItemDlgDefault(HWND hWnd, UINT message, UINT wParam, LONG lParam);

Las tres funciones anteriores trabajan en conjunto para procesar los eventos del *dialog* que agrega ítems a los menús.

BOOL FAR PASCAL

BLD_InsertarItemDlgProc(HWND hDlg, UINT message, UINT wParam, LONG lParam);

int BLD_InsertarItemDlgFunc(HWND hWnd, UINT message, UINT wParam, LONG lParam);

BOOL BLD_InsertarItemDlgDefault(HWND hWnd, UINT message, UINT wParam, LONG lParam);

Las tres funciones anteriores trabajan en conjunto para procesar los eventos del *dialog* que inserta ítems a los menús.

BOOL BLD_SalvarMenUDCFunc(HWND

hWnd,UINT message,UINT wParam, LONG lParam);

Verifica si se debe salvar el menú por algún cambio detectado.

```
BOOL          BLD_ToolBoxUDCFunc(HWND
hWnd,UINT message,UINT wParam,LONG
IParam);
```

Controla el ToolBox.

```
BOOL BLD_SalirUDCFunc(HWND hWnd,UINT
message,UINT wParam,LONG IParam);
```

Función definida por usuario para salir de GPCASE.

```
LONG          BLD_WM_ENTERIDLEMsg(HWND
hWnd, UINT message, UINT wParam, LONG
IParam );
```

Controla el proceso del mensaje: Aviso de previo acceso a menú o a *dialog*.

```
LONG BLD_WM_SETFOCUSMsg(HWND hWnd,
UINT message, UINT wParam, LONG IParam );
```

Una ventana ganó prioridad.

```
BOOL          FAR          PASCAL
BLD_IndiceHelpDlgProc(HWND hDlg,  UINT
message ,  UINT wParam,  LONG IParam);
```

```
int BLD_IndiceHelpDlgFunc(HWND hWnd,  UINT
message,  UINT wParam,  LONG IParam );
```

```
BOOL BLD_IndiceHelpDlgDefault(HWND hWnd,
UINT message,  UINT wParam,  LONG IParam);
```

Las tres funciones anteriores trabajan en conjunto para procesar los eventos del *dialog* de Ayuda.

```
BOOL          FAR          PASCAL
BLD_GPCASEAboutDlgProc(HWND hDlg,  UINT
message ,  UINT wParam,  LONG IParam);
```

```
int BLD_GPCASEAboutDlgFunc(HWND hWnd,
UINT message,  UINT wParam,  LONG IParam);
```

```
BOOL          BLD_GPCASEAboutDlgDefault(HWND
hWnd,  UINT message,  UINT wParam,  LONG
IParam);
```

Las tres funciones anteriores trabajan en conjunto para presentar el about de GPCASE.

```
BOOL BLD_CartaEstructuradaUDCFunc(HWND
hWnd,UINT message,UINT wParam,LONG
IParam);
```

Activa la Ventana de la Vista Gráfica. (*****)

```
BOOL          BLD_ANSICUDCFunc(HWND
hWnd,UINT message,UINT wParam,LONG
IParam);
```

Genera el código C ANSI.

```
BOOL          BLD_DisenadorUDCFunc(HWND
hWnd,UINT message,UINT wParam,LONG
IParam);
```

Habilita y deshabilita los Items del menú Diseñador.

```
BOOL          BLD_EliminaSelecUDCFunc(HWND
hWnd,UINT message,UINT wParam,LONG
IParam);
```

Elimina selecciones previas.

```
BOOL          BLD_NavegaciónUDCFunc(HWND
hWnd,UINT message,UINT wParam,LONG
IParam);
```

Presenta el record de la navegación.

```
BOOL          FAR          PASCAL
BLD_DocumentaciónDlgProc(HWND hDlg,  UINT
message ,  UINT wParam,  LONG IParam);
```

```
int BLD_DocumentacinDlgFunc(HWND hWnd,
UINT message, UINT wParam, LONG lParam);
```

```
BOOL BLD_DocumentacinDlgDefault(HWND
hWnd, UINT message, UINT wParam, LONG
lParam);
```

Las tres funciones anteriores trabajan en conjunto para procesar los eventos del *dialog* Documentación.

```
BOOL BLD_VentanadeConsultaUDCFunc(HWND
hWnd, UINT message, UINT wParam, LONG
lParam);
```

Presenta la ventana de consulta.

```
BOOL BLD_ZoomUDCFunc(HWND hWnd,UINT
message,UINT wParam, LONG lParam);
```

Procesa el zoom.

```
BOOL FAR PASCAL BLD_PassWordDlgProc
(HWND hDlg, UINT message , UINT wParam,
LONG lParam);
```

```
int BLD_PassWordDlgFunc(HWND hWnd, UINT
message, UINT wParam, LONG lParam );
```

```
BOOL BLD_PassWordDlgDefault(HWND hWnd,
UINT message, UINT wParam, LONG lParam);
```

Las tres funciones anteriores trabajan en conjunto para procesar los eventos del *dialog* de PassWords.

BOOL

```
BLD_SupervisorPasswordUDCFunc(HWND
hWnd,UINT message,UINT wParam, LONG
lParam);
```

Controla el PassWord del supervisor.

```
BOOL FAR PASCAL BLD_SeguridadDlgProc
(HWND hDlg, UINT message , UINT wParam,
LONG lParam);
```

```
int BLD_SeguridadDlgFunc(HWND hWnd, UINT
message, UINT wParam, LONG lParam );
```

```
BOOL BLD_SeguridadDlgDefault(HWND hWnd,
UINT message, UINT wParam, LONG lParam);
```

Las tres funciones anteriores trabajan en conjunto para procesar los eventos del *dialog* de Seguridad.

```
BOOL BLD_VerifySUPasswdUDCFunc(HWND
hWnd, UINT message, UINT wParam, LONG
lParam);
```

Verificar el PassWord del supervisor.

A continuación se presentan las variables de apoyo para el diseño declaradas como tipo *extern*:

```
extern HWND EdicinhDlg;
extern FARPROC EdicinpProc;
extern HWND AsignaCamposhDlg;
extern FARPROC AsignaCamposlpProc;
extern HWND ConfiguracinhDlg;
extern FARPROC ConfiguracinpProc;
extern HWND ProyectohDlg;
extern FARPROC ProyectolpProc;
```

Ahora tenemos variables, tipos y constantes para los controles de la ventana principal:

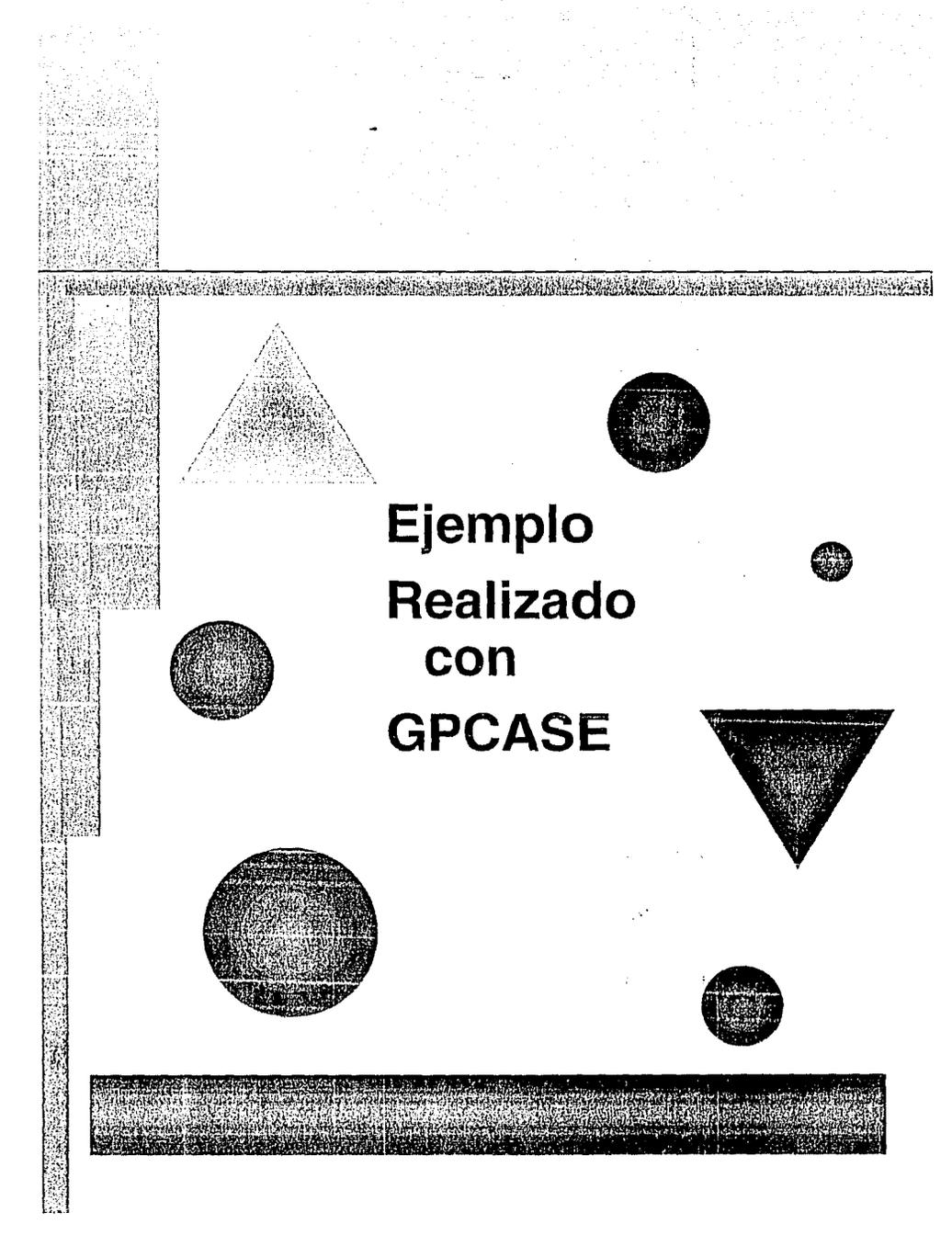
```
extern HWND hClient; //Handle to window in
client area.
extern FARPROC lpClient; //Function for
window in client area.
#define CLIENTSTRIP WS_MINIMIZE |
WS_MAXIMIZE| WS_CAPTION | WS_BORDER |
WS_DLGFRAME| WS_SYSTEMMENU |
WS_POPUP | WS_THICKFRAME |
DS_MODALFRAME
typedef struct
```

```

{
    unsigned long dtStyle;
    BYTE dtItemCount;
    int dtX;
    int dtY;
    int dtCX;
    int dtCY;
} BLD_DLTEMPLATE;
typedef BLD_DLTEMPLATE far
*LPBLD_DLTEMPLATE;
#endif
#define WMPDEBUG
// User Defined ID Values
#define IDM_Mdulos 4019
#define IDM_Liga 4020
#define IDM_Simulacin 4021
#define IDM_Ventana de Consulta 4022
#define IDM_Proyectos 4023
#define IDM_Men 4024
#define IDM_Mdulos4025 4025
#define IDM_Variables 4026
#define IDM_Carta Estructurada 4027
#define IDM_Zoom 4028
#define IDM_Cdigo CANSI 4029
#define IDM_Cdigo CGRfiteo 4030
#define IDM_Documentacin 4031
#define IDM_Seguridad 4032
#define IDM_Indice 4033
#define IDM_GPCASEAbout 4034
#define IDM_Salir 4035
#define IDM_simulacion1 4036
#define IDM_simulacion2 4037
#define IDM_simulacion3 4038
#define IDM_simulacion4 4039
#define IDM_MenuBar 4040
#define IDM_Proyecto 4000
#define IDM_Diseador 4001
#define IDM_AgregarMenu 4002
#define IDM_EditarMens 4003
#define IDM_SalvarMen 4004
#define IDM_AsignaCampos 4005
#define IDM_Mueve 4006
#define IDM_Texto 4007
#define IDM_CambiaTexto 4008
#define IDM_Marco 4009
#define IDM_Seleccionar 4010
#define IDM_Deseleccionar 4011
#define IDM_EliminaSelec 4012
#define IDM_Grid 4013
#define IDM_Etiquetas 4014
#define IDM_ToolBox 4015
#define IDM_CargaExterna 4016
#define IDM_Edificin 4017
#define IDM_BmcodeVariables 4018

```

Este es el listado de los ID's generados por Windows Maker para controlar los elementos como menús e ítems del menú principal de GPCASE:

The image features a complex abstract design. A prominent vertical grid of thin lines runs down the left side. A thick horizontal bar spans the top, and another thick horizontal bar is located at the bottom. The central text is surrounded by several geometric shapes: a light triangle at the top left, a dark circle at the top right, a small dark circle to the right of the text, a dark circle at the bottom left, a large dark circle at the bottom left, a dark inverted triangle at the bottom right, and a small dark circle at the bottom right.

**Ejemplo
Realizado
con
GPCASE**

EJEMPLO REALIZADO CON GPCASE

Este ejemplo servirá de guía, para mostrar de manera sencilla y rápida el uso de GPCASE.

Antes de iniciar GPCASE para realizar el ejemplo , si el usuario desea desarrollar su propia aplicación debe tener definidas todas las variables a utilizar para la aplicación deseada así como también su menú, las pantallas y reportes diseñados.

GPCASE es un completo sistema para generar prototipos, provee al usuario herramientas de diseño que ayudan al usuario a generar una aplicación en forma sencilla, permite la visión por fases del sistema según el avance del desarrollo y modificaciones de cualquier tipo en cualquier momento, además de que al final se puede generar la aplicación diseñada en código C estandar y "C" Gráfico.

Ejemplo Realizado con GPCASE.

Paso 1 CREACION DE UN NUEVO PROYECTO.

Elegir el menú Proyecto

Nombre del Proyecto

Definición de Proyectos

ID (8 Caracteres):

Nombre (50 Caracteres máxima):

Sistema de soporte a usuarios:

Dirección: Responsable Técnica:

Fecha Inicia: Fecha Terminación:

F2 - Borrado, F3 - Actual, F4 - Calendario

Porcentaje de Tiempo Consumido:

ABC
ASI
DEN
ESCUELA
MOTOROLA
ROMPA
NUEVO
PIUEBA

Llenar todos los campos de la ventana proyecto y después dar "Click" en el Boton que indica Añade/Mod.

El supervisor es el unico que puede definir las características, cambiar, definir claves de acceso o eliminar un proyecto. Las características

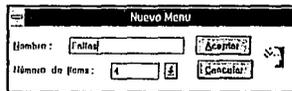
Ejemplo Realizado con GPCASE .

Paso 2

CREACIÓN DEL MENÚ

Verificar en el menú Menús que la opción Diseñador este seleccionada, Seleccionar del menú Menús, Agregar menú ó dar "Click" en el icono localizado en la parte superior izquierda de la pantalla principal de GPCASE.

Icono acceso a
menús

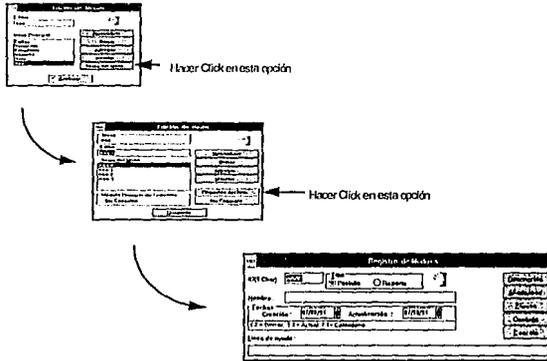
Un cuadro de diálogo con el título 'Nuevo Menú'. Contiene dos campos de texto: 'Nombre:' con el valor 'Ejemplos' y 'Número de ítems:' con el valor '4'. Hay botones 'Aceptar?' y 'Cancelar?'.

Teclar en el campo Nombre el Nombre del Menú e indicar el número de ítems que contendrá el menú.

Después con la opción Editar Menú del menú Menús modificar los ítems del menú que se acaba de crear para renombrarlos con los nombres de las opciones correctas.

Ejemplo Realizado con GPCASE.

Paso 3. Definición de Conexiones



Paso 3.

Definición de Conexiones

La conexión es la pantalla que se ejecuta cuando se selecciona una opción. Para definir la conexión se editan los ítems del menú y se da "Click" en el Botón que indica conexión y en el Dialog Box que se muestra teclear los datos que se piden y dar "Click" en botón Continuar. Este paso se debe Repetir para todas y cada una de las opciones creadas en el menú.

Ejemplo Realizado con GPCASE .

Paso 4. CREACIÓN DE VARIABLES

Nombre de la variable
que se mostrará en el diseño

Configurar datos de variables

Identificador (M) (8 Caracteres Mx) PZZZZZ

Nombre: Acción Tomada

ACCIONES: CONTAR, CLAVE, D. INCIDIO, DELEGAR, DIRIGIR, FALLA1, FALLA2, FALLA3, FALLA4

Nom. corto: Acción Tomada Ejemplo: Contado

Actuador(s): H1/H1/1 2

Tipo: Contador 1

Longitud: 18 F2 + Valor F3 + Actual F4 + Calcular

Valores Continuos:

Valor Inicial: Valor Final:

Línea de Ayuda (Máximo: 80 caracteres):
Acción Tomada (Una de sus opciones)

Descripción: Valores Desc. Añadir/Mod. Eliminar Cancelar

Para agregar dar Click

Conjunto de variables creadas con anterioridad

Una vez definidos los nombres correctos de las opciones y sus conexiones se generan y crean las variables que se utilizarán en el sistema. También se tiene la opción de haber generado las variables con un sistema externo en cuyo caso se pueden cargar dichas variables al sistema. Para mayor detalle consultar el Manual del Usuario.

Para la creación de variables, en el menú Variables seleccionar la opción Edición, la cual mostrará el Dialog Box **Configuración de variables**

- En Nombre se describe la variable
- En nombre corto se teclea el nombre que se utilizará para la generación de código etc.

Para adicionar esta variable dar "Click" en el Botón Añade/Mod

Ejemplo Realizado con GPCASE.



Icono de Banco de variables

Variables disponibles por agregar

Banco de Variables

ACCION	CAANTIDAD	ACCION	CAANTIDAD
CLAVE	1	CLAVE	1
D_INGUMO	1	D_INGUMO	1
DESCRIPC	1	DESCRIPC	1
DDFICHA	1	DDFICHA	1
FALLA1	1	FALLA1	1
FALLA2	1	FALLA2	1
FALLA3	1	FALLA3	1
FALLA4	1	FALLA4	1
FALLA5	1	FALLA5	1
FECI_ING	1	FECI_ING	1
FECI_PED	1	FECI_PED	1

Continuar

Hacer Click en este botón para adicionarlas

variables Seleccionadas

Repetir el mismo procedimiento anterior para cada una de las variables del sistema y para adicionar las variables al Banco de variables seleccionar del menú Variables la opción Banco de variables o dar "Click" en el icono banco de variables.

Ejemplo Realizado con GPCASE.

Paso 5.

DISEÑO DE PANTALLAS Y REPORTES



Icono de Módulos

Lenar los datos
como se piden

Registro de Módulos	
ID (8 Char):	<input type="text" value="AREASERV"/> Tipo: <input checked="" type="radio"/> Pantalla <input type="radio"/> Reporte
Nombre:	<input type="text" value="servicios por Area promedio"/>
Enchós:	
Creación:	<input type="text" value="24/07/94"/>
Actualización:	<input type="text" value="05/07/94"/>
F2 = Borrar, F3 = Actual, F4 = Calendario	
Línea de ayuda:	
<input type="text" value="Servicios por Area atendidos semanalmente"/>	

AREASERV	2	Definición
ATENDIDO	3	Añade/mod
CAMBIO	4	Elimina
CONSULTA	5	Configura
DEVOLUCI	6	Resalta
ENTREGA	7	
GENERAL	8	
PEPIDO	9	

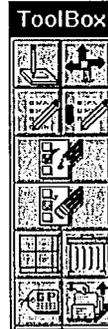
Para el diseño de pantallas y reportes seleccionar del menú Procesos la opción Módulos o dar "Click" en el icono de módulos.

Ejemplo Realizado con GPCASE.

Paso 6.

DISEÑANDO

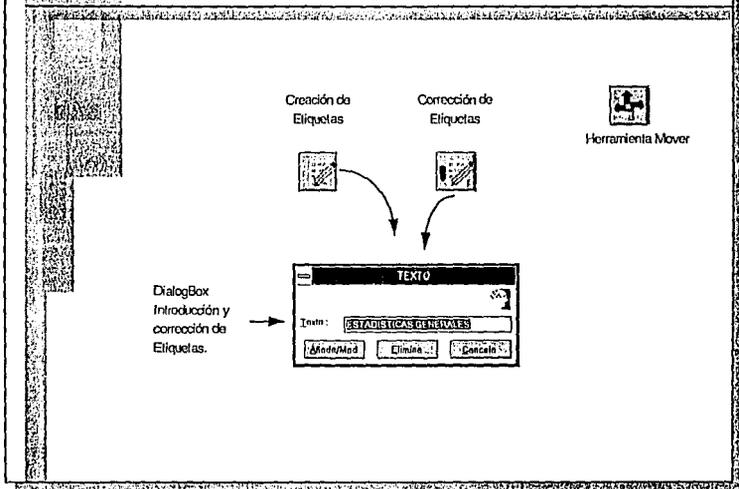
Herramientas
de ayuda en el
diseño.



Después del paso anterior se mostrará en la parte inferior del video el nombre del módulo que se iniciará a diseñar.

Para ayudar a un mejor diseño en forma rápida y sencilla se cuenta con una caja de herramientas la cual puede ser colocada en cualquier parte de la pantalla para que no obstruya el área de diseño.

Ejemplo Realizado con GPCASE.



Orden de diseño

- Iniciar con títulos y etiquetas
- Crear y colocar las etiquetas en su lugar correspondiente según el prediseño o según se desee, se recomienda hacer uso de la herramienta de Grid y etiquetas, para una mayor facilidad en el diseño.

Si la etiqueta se coloca en un lugar no deseado es posible moverla para reubicarla en el lugar deseado, haciendo uso de la herramienta Mover y si la etiqueta no es la correcta o simplemente se desea cambiar es posible modificarla seleccionando del menú Diseñador la opción modificar o dando un "Click" en el icono "Modificar etiqueta" mostrará el DialogBox de corrección en el cual se tecldea la etiqueta correcta y dar un "Click" en el Botón Añade/Mod.

Ejemplo Realizado con GPCASE .

Asignación de Campos



ASIGNAR	1
CAMBIAR	1
CLAVE	1
D. INICIO	1
DESIGNAR	1
INFORMA	1
FALLA1	1
FALLA2	1
FALLA3	1
FALLA4	1
FALLA5	1
FALLA6	1
FALLA7	1
FALLA8	1
FALLA9	1
FALLA10	1
FALLA11	1
FALLA12	1
FALLA13	1
FALLA14	1
FALLA15	1
FALLA16	1
FALLA17	1
FALLA18	1
FALLA19	1
FALLA20	1
FALLA21	1
FALLA22	1
FALLA23	1
FALLA24	1
FALLA25	1
FALLA26	1
FALLA27	1
FALLA28	1
FALLA29	1
FALLA30	1
FALLA31	1
FALLA32	1
FALLA33	1
FALLA34	1
FALLA35	1
FALLA36	1
FALLA37	1
FALLA38	1
FALLA39	1
FALLA40	1
FALLA41	1
FALLA42	1
FALLA43	1
FALLA44	1
FALLA45	1
FALLA46	1
FALLA47	1
FALLA48	1
FALLA49	1
FALLA50	1
FALLA51	1
FALLA52	1
FALLA53	1
FALLA54	1
FALLA55	1
FALLA56	1
FALLA57	1
FALLA58	1
FALLA59	1
FALLA60	1
FALLA61	1
FALLA62	1
FALLA63	1
FALLA64	1
FALLA65	1
FALLA66	1
FALLA67	1
FALLA68	1
FALLA69	1
FALLA70	1
FALLA71	1
FALLA72	1
FALLA73	1
FALLA74	1
FALLA75	1
FALLA76	1
FALLA77	1
FALLA78	1
FALLA79	1
FALLA80	1
FALLA81	1
FALLA82	1
FALLA83	1
FALLA84	1
FALLA85	1
FALLA86	1
FALLA87	1
FALLA88	1
FALLA89	1
FALLA90	1
FALLA91	1
FALLA92	1
FALLA93	1
FALLA94	1
FALLA95	1
FALLA96	1
FALLA97	1
FALLA98	1
FALLA99	1
FALLA100	1

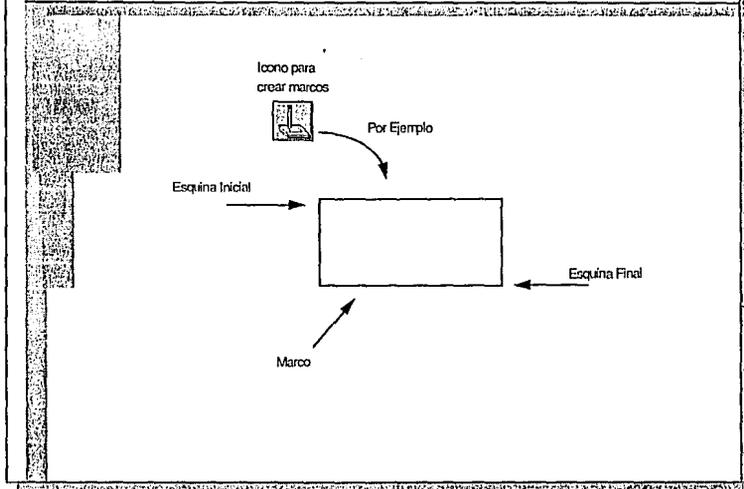
Elegir la Variable
y colocarla en el
lugar correspondiente
según prediseño o
según se desee

- Continuar con Asignación de campos.

Seleccionar del menú Diseñador la opción Asigna Campos o dar un "Click" en el icono asigna campos y elegir la variable desea de las opciones mostradas.

Este proceso debe de hacer con todas y cada una de los campos a asignar.

Ejemplo Realizado con GPCASE .



Para terminar el diseño es posible agregar marcos a la pantalla. que se esta diseñando según se desee o requiera.

Seleccionar del menú Diseñador Seleccionar Marco o dar "Click" en el icono de Marcos.

Dar un "Click" en donde se desea inicie la primer esquina del marco (Esquina Superior Izquierda POR Ejemplo) y sin soltar el botón Iquirdo, Mover el Mouse Hasta donde se desea termine el marco (Esquina Inferior Derecha POR Ejemplo).

Ejemplo Realizado con GPCASE.

Selección y Eliminación de Objetos



Icono Para Selección



Icono Para Des-Selección



Icono Para Eliminar

Si en el transcurso del diseño se decide eliminar algunos objetos creados,

se deben seleccionar primero y después borrar del menú Diseñador seleccionar la opción **Seleccionar**

y después dar un "Click" sobre el objeto que se desea eliminar y después del menú Diseñador Elimina

Selecc. o con un "Click" en los equivalentes iconos.

Ejemplo Realizado con GPCASE .

Al Terminar el Prototipo



Simulación del Prototipo



Generación de Código



Generación de Documentación



Vista Gráfica

Para un acercamiento de la vista gráfica Seleccionar del menú Procesos la opción Zoom o hacer "Click" en el icono de Zoom Representado por Una Lupa.



Una vez terminado el diseño se puede ejecutar una simulación. Seleccionar del menú Procesos la opción simulación o dar un "Click" en el icono de Simulación.

Para una vista gráfica del prototipo creado Seleccionar del menú procesos la opción vista gráfica o hacer "Click" en el icono de Vista Gráfica.

Adicionalmente al prototipo, es posible generar código y documentación del prototipo para ello basta con seleccionar del menú Procesos la opción Generar Código o hacer "Click" en el ícono Generación de Código. Y para la documentación seleccionar del menú Procesos la opción Documentación o hacer "Click" en el ícono Genera Documentación.

El código generado puede editarse y compilarse con un compilador de "C" estándar o "C" Gráfico con el fin de generar un programa ejecutable. Con un poco más de programación adicional es posible generar un sistema completo.

Pantallas Generadas Con El Sistema GPCASE

SISTEMA GPCASE

SISTEMA DE SOPORTE A USUARIOS

Clave:[_____]

Nombre:[_____]

FALLAS

Red[_]Hardware[_]Software[_]Impresión[_]Otros[_]

Descripción[_____]

Introduzca la clave del usuario iniciando con letra

SISTEMA GPCASE

Pedido

SISTEMA GPCASE

SOLICITUD DE SERVICIOS

Clave:[_____]

Nombre:[_____]

TIPO DE SERVICIO

Red[_]Hardware[_]Software[_]Comunicación[_]Otros[_]

Descripción[_____]

Introduzca la clave del usuario iniciando con letra

SISTEMA GPCASE

ATENCIÓN DE SERVICIOS

Fecha de Reporte:[_____]

Responsable:[_____]

Fecha De Atención:[_____]

Acción Tomada:[_____]

Fecha del Reporte

PANTALLA DE SERVICIOS DE ESTADÍSTICAS DE SERVICIO
SISTEMA C/CONS

[Consulta de Servicios]

Fecha de Reporte: []

Responsable: []

Fecha de Atención: []

Acción Tomada: []

Mostrar el tipo de problema

PANTALLA DE ESTADÍSTICAS DE SERVICIO
SISTEMA C/CONS

PANTALLA DE ESTADÍSTICAS DE SERVICIO

Total de Servicios atendidos	[]
Servicios de RED	[]
Servicios de Hardware:	[]
Servicios de Software:	[]
Servicios de Impresión:	[]
Otros Servicios	[]

Mostrar el tipo de problema

PANTALLA DE SERVICIOS DE ESTADÍSTICAS DE SERVICIO
SISTEMA C/CONS

SERVICIOS POR AREA

Oficina: [] = []

Total de Reportes:	[]
Reportes de RED	[]
Reportes de Hardware:	[]
Reportes de Software:	[]
Reportes de Impresión:	[]
Otros Reportes:	[]
Total de Reportes Atendidos:	[]

Mostrar el tipo de problema

CODIGO GENERADO CON GPCASE

```
#include "header.h"
#include "MOTOROLA.h"
#include "gpcod.c"
main()
{
    fnInitStructMenu ("MOTOROLA.mnu");
    fnInIt();
}

fnEjecuta(int iModulo)
{
    switch(iModulo)
    {
        case ID_REPORTAR :
            fnREPORTAR();
            break;
        case ID_CONSULTA :
            fnCONSULTA();
            break;
        case ID_PEDIDO :
            fnPEDIDO();
            break;
        case ID_ATENDIDO :
            fnATENDIDO();
            break;
        case ID_SERCONSU :
            fnSERCONSU();
            break;
        case ID_SERVICIO :
            fnSERVICIO();
            break;
        case ID_AREASERV :
            fnAREASERV();
            break;
        case ID_GENERAL :
            fnGENERAL();
            break;
        case ID_ENTREGA :
            fnENTREGA();
            break;
        case ID_DEVOLUCI :
            fnDEVOLUCI();
            break;
        case ID_CAMBIOS :
            fnCAMBIOS();
            break;
        case ID_REPORTES :
            fnREPORTES();
            break;
    }
}
```

```

        case ID_REP_ESTA :
            fnREP_ESTA();
            break;
    } // fin de switch
}

void fnREPORTAR()
{
    int i;
    char ss[50];
    strcpy (ss, "");
    for (i=0; i < IDM_REPORTAR; i++)
        fnMarco(stmREPORTAR[i].ix,
stmREPORTAR[i].iy,stmREPORTAR[i].ix+stmREPORTAR[i].iAncho,stmREPORTAR[i].iy+stmREPORTAR[i].iAlto);
    for (i=0; i < IDT_REPORTAR; i++)
        fnTexto(sttREPORTAR[i].ix, sttREPORTAR[i].iy,sttREPORTAR[i].sTexto);
    for (i=0; i < IDP_REPORTAR; i++)
        fnMarcaCampo(stpREPORTAR[i].ix,
stpREPORTAR[i].iy,stpREPORTAR[i].ix+stpREPORTAR[i].iAncho,stpREPORTAR[i].iy+stpREPORTAR[i].iAlto);
    stLIGA_DEF = stLIGA_REPORTAR;
    for (i=0;fnLeeCampo(stpREPORTAR[i], ss, IDP_REPORTAR, &i) != GP_ESC;);
}

void fnCONSULTA()
{
    int i;
    char ss[50];
    strcpy (ss, "");
    for (i=0; i < IDM_CONSULTA; i++)
        fnMarco(stmCONSULTA[i].ix,
stmCONSULTA[i].iy,stmCONSULTA[i].ix+stmCONSULTA[i].iAncho,stmCONSULTA[i].iy+stmCONSULTA[i].iAlto);
    for (i=0; i < IDT_CONSULTA; i++)
        fnTexto(sttCONSULTA[i].ix, sttCONSULTA[i].iy,sttCONSULTA[i].sTexto);
    for (i=0; i < IDP_CONSULTA; i++)
        fnMarcaCampo(stpCONSULTA[i].ix,
stpCONSULTA[i].iy,stpCONSULTA[i].ix+stpCONSULTA[i].iAncho,stpCONSULTA[i].iy+stpCONSULTA[i].iAlto);
    stLIGA_DEF = stLIGA_CONSULTA;
    for (i=0;fnLeeCampo(stpCONSULTA[i], ss, IDP_CONSULTA, &i) != GP_ESC;);
}

void fnPEDIDO()
{
    int i;
    char ss[50];
    strcpy (ss, "");
    for (i=0; i < IDM_PEDIDO; i++)
        fnMarco(stmPEDIDO[i].ix,
stmPEDIDO[i].iy,stmPEDIDO[i].ix+stmPEDIDO[i].iAncho,stmPEDIDO[i].iy+stmPEDIDO[i].iAlto);
    for (i=0; i < IDT_PEDIDO; i++)
        fnTexto(sttPEDIDO[i].ix, sttPEDIDO[i].iy,sttPEDIDO[i].sTexto);
}

```

```

    for (i=0; i < IDP_PEDIDO; i++)
        fnMarcaCampo(stpPEDIDO[i].ix,
stpPEDIDO[i].iy,stpPEDIDO[i].ix+stpPEDIDO[i].iAncho,stpPEDIDO[i].iy+stpPEDIDO[i].iAlto);
    stLIGA_DEF = stLIGA_PEDIDO;
    for (i=0;fnLeeCampo(stpPEDIDO[i], ss, IDP_PEDIDO, &i) != GP_ESC);
}

```

```

void fnATENDIDO()

```

```

{
    int i;
    char ss[50];
    strcpy(ss,"");
    for (i=0; i < IDM_ATENDIDO; i++)
        fnMarco(stmATENDIDO[i].ix,
stmATENDIDO[i].iy,stmATENDIDO[i].ix+stmATENDIDO[i].iAncho,stmATENDIDO[i].iy+stmATENDID
O[i].iAlto);
    for (i=0; i < IDT_ATENDIDO; i++)
        fnTexto(stATENDIDO[i].ix, stATENDIDO[i].sTexto);
    for (i=0; i < IDP_ATENDIDO; i++)
        fnMarcaCampo(stpATENDIDO[i].ix,
stpATENDIDO[i].iy,stpATENDIDO[i].ix+stpATENDIDO[i].iAncho,stpATENDIDO[i].iy+stpATENDIDO[i
].iAlto);
    stLIGA_DEF = stLIGA_ATENDIDO;
    for (i=0;fnLeeCampo(stpATENDIDO[i], ss, IDP_ATENDIDO, &i) != GP_ESC);
}

```

```

void fnSERCONSUMU()

```

```

{
    int i;
    char ss[50];
    strcpy(ss,"");
    for (i=0; i < IDM_SERCONSUMU; i++)
        fnMarco(stmSERCONSUMU[i].ix,
stmSERCONSUMU[i].iy,stmSERCONSUMU[i].ix+stmSERCONSUMU[i].iAncho,stmSERCONSUMU[i].iy+stmSERCONS
U[i].iAlto);
    for (i=0; i < IDT_SERCONSUMU; i++)
        fnTexto(stSERCONSUMU[i].ix, stSERCONSUMU[i].sTexto);
    for (i=0; i < IDP_SERCONSUMU; i++)
        fnMarcaCampo(stpSERCONSUMU[i].ix,
stpSERCONSUMU[i].iy,stpSERCONSUMU[i].ix+stpSERCONSUMU[i].iAncho,stpSERCONSUMU[i].iy+stpSERCONSUMU[i
].iAlto);
    stLIGA_DEF = stLIGA_SERCONSUMU;
    for (i=0;fnLeeCampo(stpSERCONSUMU[i], ss, IDP_SERCONSUMU, &i) != GP_ESC);
}

```

```

void fnSERVICIO()

```

```

{
    int i;
    char ss[50];
    strcpy(ss,"");
    for (i=0; i < IDM_SERVICIO; i++)
        fnMarco(stmSERVICIO[i].ix,
stmSERVICIO[i].iy,stmSERVICIO[i].ix+stmSERVICIO[i].iAncho,stmSERVICIO[i].iy+stmSERVICIO[i].iAl
to);
}

```

```

    for (i=0; i < IDT_SERVICIO; i++)
        fnTexto(sttSERVICIO[i].ix, sttSERVICIO[i].iy, sttSERVICIO[i].sTexto);
    for (i=0; i < IDP_SERVICIO; i++)
        fnMarcaCampo(stpSERVICIO[i].ix,
stpSERVICIO[i].iy, stpSERVICIO[i].ix+stpSERVICIO[i].iAncho, stpSERVICIO[i].iy+stpSERVICIO[i].iAlto)
;
    stLIGA_DEF = stLIGA_SERVICIO;
    for (i=0; fnLeeCampo(stpSERVICIO[i], ss, IDP_SERVICIO, &i) != GP_ESC);
}

```

```

void fnAREASERV()
{
    ir. ";
    char ss[50];
    strcpy (ss, "");
    for (i=0; i < IDM_AREASERV; i++)
        fnMarco(stmAREASERV[i].ix,
stmAREASERV[i].iy, stmAREASERV[i].ix+stmAREASERV[i].iAncho, stmAREASERV[i].iy+stmAREASERV[i].iAlto);
    for (i=0; i < IDT_AREASERV; i++)
        fnTexto(sttAREASERV[i].ix, sttAREASERV[i].iy, sttAREASERV[i].sTexto);
    for (i=0; i < IDP_AREASERV; i++)
        fnMarcaCampo(stpAREASERV[i].ix,
stpAREASERV[i].iy, stpAREASERV[i].ix+stpAREASERV[i].iAncho, stpAREASERV[i].iy+stpAREASERV[i].iAlto);
    stLIGA_DEF = stLIGA_AREASERV;
    for (i=0; fnLeeCampo(stpAREASERV[i], ss, IDP_AREASERV, &i) != GP_ESC);
}

```

```

void fnGENERAL()
{
    int i;
    char ss[50];
    strcpy (ss, "");
    for (i=0; i < IDM_GENERAL; i++)
        fnMarco(stmGENERAL[i].ix,
stmGENERAL[i].iy, stmGENERAL[i].ix+stmGENERAL[i].iAncho, stmGENERAL[i].iy+stmGENERAL[i].iAlto);
    for (i=0; i < IDT_GENERAL; i++)
        fnTexto(sttGENERAL[i].ix, sttGENERAL[i].iy, sttGENERAL[i].sTexto);
    for (i=0; i < IDP_GENERAL; i++)
        fnMarcaCampo(stpGENERAL[i].ix,
stpGENERAL[i].iy, stpGENERAL[i].ix+stpGENERAL[i].iAncho, stpGENERAL[i].iy+stpGENERAL[i].iAlto);
    stLIGA_DEF = stLIGA_GENERAL;
    for (i=0; fnLeeCampo(stpGENERAL[i], ss, IDP_GENERAL, &i) != GP_ESC);
}

```

```

void fnENTREGA()
{
    int i;
    char ss[50];
    strcpy (ss, "");
    for (i=0; i < IDM_ENTREGA; i++)

```

```

    strcpy (ss, "");
    stLIGA_DEF = stLIGA_REPÖRTES;
}

void fnREP_ESTA()
{
    int i;
    char ss[50];
    strcpy (ss, "");
    for (i=0; i < IDM_REP_ESTA; i++)
        fnMarco(stmREP_ESTA[i].ix,
stmREP_ESTA[i].iy,stmREP_ESTA[i].ix+stmREP_ESTA[i].iAncho,stmREP_ESTA[i].iy+stmREP_ESTA[i]
.iAlto);
    for (i=0; i < IDT_REP_ESTA; i++)
        fnTexto(stREP_ESTA[i].ix, stREP_ESTA[i].iy,stREP_ESTA[i].sTexto);
    for (i=0; i < IDP_REP_ESTA; i++)
        fnMarcaCampo(stpREP_ESTA[i].ix,
stpREP_ESTA[i].iy,stpREP_ESTA[i].ix+stpREP_ESTA[i].iAncho,stpREP_ESTA[i].iy+stpREP_ESTA[i].iAl
to);
    stLIGA_DEF = stLIGA_REP_ESTA;
    for (i=0; fnLeeCampo(stpREP_ESTA[i], ss, IDP_REP_ESTA, &i) != GP_ESC;);
}

```

```

        fnMarco(stmENTREGA[i].ix,
stmENTREGA[i].iy,stmENTREGA[i].ix+stmENTREGA[i].iAncho,stmENTREGA[i].iy+stmENTREGA[i].iAlto);
        for (i=0; i < IDT_ENTREGA; i++)
            fnTexto(stENTREGA[i].ix, stENTREGA[i].iy,stENTREGA[i].sTexto);
        for (i=0; i < IDP_ENTREGA; i++)
            fnMarcaCampo(stpENTREGA[i].ix,
stpENTREGA[i].iy,stpENTREGA[i].ix+stpENTREGA[i].iAncho,stpENTREGA[i].iy+stpENTREGA[i].iAlto
);
        stLIGA_DEF = stLIGA_ENTREGA;
        for (i=0;fnLeeCampo(stpENTREGA[i], ss, IDP_ENTREGA, &i) != GP_ESC);
    }

void fnD.VOLUNT()
{
    int i;
    char ss[50];
    strcpy(ss,"");
    for (i=0; i < IDM_DEVOLUCI; i++)
        fnMarco(stmDEVOLUCI[i].ix,
stmDEVOLUCI[i].iy,stmDEVOLUCI[i].ix+stmDEVOLUCI[i].iAncho,stmDEVOLUCI[i].iy+stmDEVOLUCI
I[i].iAlto);
    for (i=0; i < IDT_DEVOLUCI; i++)
        fnTexto(stDEVOLUCI[i].ix, stDEVOLUCI[i].iy,stDEVOLUCI[i].sTexto);
    for (i=0; i < IDP_DEVOLUCI; i++)
        fnMarcaCampo(stpDEVOLUCI[i].ix,
stpDEVOLUCI[i].iy,stpDEVOLUCI[i].ix+stpDEVOLUCI[i].iAncho,stpDEVOLUCI[i].iy+stpDEVOLUCI[i]
.iAlto);
    stLIGA_DEF = stLIGA_DEVOLUCI;
    for (i=0;fnLeeCampo(stpDEVOLUCI[i], ss, IDP_DEVOLUCI, &i) != GP_ESC);
}

void fnCAMBIOS()
{
    int i;
    char ss[50];
    strcpy(ss,"");
    for (i=0; i < IDM_CAMBIOS; i++)
        fnMarco(stmCAMBIOS[i].ix,
stmCAMBIOS[i].iy,stmCAMBIOS[i].ix+stmCAMBIOS[i].iAncho,stmCAMBIOS[i].iy+stmCAMBIOS[i].iAl
to);
    for (i=0; i < IDT_CAMBIOS; i++)
        fnTexto(stCAMBIOS[i].ix, stCAMBIOS[i].iy,stCAMBIOS[i].sTexto);
    for (i=0; i < IDP_CAMBIOS; i++)
        fnMarcaCampo(stpCAMBIOS[i].ix,
stpCAMBIOS[i].iy,stpCAMBIOS[i].ix+stpCAMBIOS[i].iAncho,stpCAMBIOS[i].iy+stpCAMBIOS[i].iAlto);
    stLIGA_DEF = stLIGA_CAMBIOS;
    for (i=0;fnLeeCampo(stpCAMBIOS[i], ss, IDP_CAMBIOS, &i) != GP_ESC);
}

void fnREPORTES()
{
    int i;
    char ss[50];

```

ARCHIVO MOTOROLA.H GENERADO POR GPCASE

```
#define ID_REPORTAR 57
#define ID_CONSULTA 58
#define ID_PEDIDO 59
#define ID_ATENDIDO 60
#define ID_SERCONSU 61
#define ID_SERVICIO 62
#define ID_AREASERV 63
#define ID_GENERAL 64
#define ID_ENTREGA 66
#define ID_DEVOLUCI 67
#define ID_CAMBIOS 68
#define ID_REPORTES 69
#define ID_REP_ESTA 70

#define MAX_TEXT_MENU 26
#define MAX_ITEM_PERSIANA 21
#define MAX_PERSIANAS 10
#define ID_MENU OPCIONES 5
```

```
fnEjecuta(int iModulo);
void fnREPORTAR();
void fnCONSULTA();
void fnPEDIDO();
void fnATENDIDO();
void fnSERCONSU();
void fnSERVICIO();
void fnAREASERV();
void fnGENERAL();
void fnENTREGA();
void fnDEVOLUCI();
void fnCAMBIOS();
void fnREPORTES();
void fnREP_ESTA();
#define IDM_REPORTAR 2
#define IDT_REPORTAR 10
#define IDP_REPORTAR 8
#define IDM_CONSULTA 2
#define IDT_CONSULTA 10
#define IDP_CONSULTA 8
#define IDM_PEDIDO 3
#define IDT_PEDIDO 10
#define IDP_PEDIDO 8
#define IDM_ATENDIDO 1
#define IDT_ATENDIDO 5
#define IDP_ATENDIDO 4
#define IDM_SERCONSU 2
#define IDT_SERCONSU 5
#define IDP_SERCONSU 4
#define IDM_SERVICIO 1
#define IDT_SERVICIO 8
#define IDP_SERVICIO 6
#define IDM_AREASERV 2
#define IDT_AREASERV 9
#define IDP_AREASERV 9
#define IDM_GENERAL 7
#define IDT_GENERAL 17
#define IDP_GENERAL 17
#define IDM_ENTREGA 3
#define IDT_ENTREGA 6
#define IDP_ENTREGA 8
#define IDM_DEVOLUCI 2
#define IDT_DEVOLUCI 6
```

```
#define IDP_DEVOLUCI 8
#define IDM_CAMBIOS 2
#define IDT_CAMBIOS 6
#define IDP_CAMBIOS 8
#define IDM_REP_ESTA 8
#define IDT_REP_ESTA 11
#define IDP_REP_ESTA 8
```

```
stLIGA stLIGA_REPORTAR = {0, 0};
```

```
stCAMPO stPREPORTAR [] = {
    19, 9, 6, 1, 4, "Introduzca la clave del usuario iniciando con letra",
    20, 11, 30, 1, 4, "Introduzca el Nombre del Usuario Iniciando con el apellido",
    17, 15, 1, 1, 4, "Elija este campo si su falla es de red",
    28, 15, 1, 1, 4, "Elija este campo si su falla es de Hardware",
    39, 15, 1, 1, 4, "Elija este campo si su falla es de Software",
    51, 15, 1, 1, 4, "Elija este campo si su falla es de Impresión",
    59, 15, 1, 1, 4, "Elija este campo para fallas particulares (describirla en el campo descripción)",
    21, 19, 50, 1, 4, "escriba una descripción de hasta 50 caracteres",
    0, 0, 0, 0, 0, ""};
```

```
stTEXTO stREPORTAR [] = {
    26, 5, "SISTEMA DE SOPORTE A USUARIOS",
    12, 9, "Clave:",
    12, 11, "Nombre:",
    32, 13, "FALLA",
    13, 15, "RED",
    19, 15, "Hardware",
    30, 15, "Software",
    41, 15, "Impresión",
    53, 15, "Otros",
    9, 19, "Descripción",
    0, 0, ""};
```

```
stMARCO stMREPORTAR [] = {
    6, 6, 69, 15,
    11, 14, 50, 3,
    0, 0, 0, 0};
```

```
stLIGA stLIGA_CONSULTA = {0, 0};
```

```
stCAMPO stPCONSULTA [] = {
    23, 7, 6, 1, 4, "Introduzca la clave del usuario iniciando con letra",
    24, 9, 30, 1, 4, "Introduzca el Nombre del Usuario Iniciando con el apellido",
    21, 14, 1, 1, 4, "Elija este campo si su falla es de red",
    32, 14, 1, 1, 4, "Elija este campo si su falla es de Hardware",
    43, 14, 1, 1, 4, "Elija este campo si su falla es de Software",
    55, 14, 1, 1, 4, "Elija este campo si su falla es de Impresión",
    63, 14, 1, 1, 4, "Elija este campo para fallas particulares (describirla en el campo descripción)",
    22, 18, 50, 1, 4, "escriba una descripción de hasta 50 caracteres",
    0, 0, 0, 0, 0, ""};
```

```
stTEXTO stICONSULTA [] = {
    21, 3, "SISTEMA DE SOPORTE A USUARIOS",
    16, 7, "Clave:",
    16, 9, "Nombre:",
    38, 12, "FALLAS",
    17, 14, "RED",
    23, 14, "Hardware",
    34, 14, "Software",
    45, 14, "Impresión",
    57, 14, "Otros",
    11, 18, "Descripción",
    0, 0, ""};
```

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

stMARCO stmCONSULTA [] = {

6, 4, 69, 16,
14, 13, 51, 3,
0, 0, 0, 0};

stLIGA stLIGA_PEDIDO = {0, 0};

stCAMPO stpPEDIDO [] = {

23, 6, 6, 1, 4, "Introduzca la clave del usuario iniciando con letra",
24, 8, 30, 1, 4, "Introduzca el Nombre del Usuario Iniciando con el apellido",
20, 13, 1, 1, 4, "Elija este campo si su falla es de red",
31, 13, 1, 1, 4, "Elija este campo si su falla es de Hardware",
42, 13, 1, 1, 4, "Elija este campo si su falla es de Software",
57, 13, 1, 1, 4, "Elija este campo si su falla es de Impresión",
65, 13, 1, 1, 4, "Elija este campo para fallas particulares (describirla en el campo descripción)",
20, 18, 50, 1, 4, "Escriba una descripción de hasta 50 caracteres",
0, 0, 0, 0, 0, ""};

stTEXTO sttPEDIDO [] = {

30, 3, "SOLICITUD DE SERVICIOS",
16, 6, "Clave:",
16, 8, "Nombre:",
33, 11, "TIPO DE SERVICIO",
16, 13, "Red",
22, 13, "Hardware",
33, 13, "Software",
44, 13, "Comunicación",
59, 13, "Otros",
9, 18, "Descripción",
0, 0, ""};

stMARCO stmPEDIDO [] = {

6, 4, 68, 12,
13, 12, 55, 3,
6, 16, 68, 5,
0, 0, 0, 0};

stLIGA stLIGA_ATENDIDO = {0, 0};

stCAMPO stpATENDIDO [] = {

34, 7, 8, 1, 4, "Fecha del reporte",
29, 9, 30, 1, 4, "Introduzca el Nombre del Usuario Iniciando con el apellido",
35, 11, 8, 1, 4, "Fecha Propuesta para solución del problema",
31, 13, 30, 1, 4, "Acción Tomada Para dar una solución",
0, 0, 0, 0, 0, ""};

stTEXTO sttATENDIDO [] = {

29, 4, "ATENCION DE SERVICIOS",
16, 7, "Fecha de Reporte:",
16, 9, "Responsable:",
16, 11, "Fecha De Atención:",
16, 13, "Acción Tomada:",
0, 0, ""};

stMARCO stmATENDIDO [] = {

11, 5, 57, 11,
0, 0, 0, 0};

stLIGA stLIGA_SERCONSU = {0, 0};

stCAMPO stpSERCONSU [] = {

43, 7, 8, 1, 4, "Elija el tipo del problema",
38, 9, 25, 1, 4, "Acción Tomada Para dar una solución",
44, 11, 8, 1, 4, "Fecha Propuesta para solución del problema",
40, 13, 25, 1, 4, "Acción Tomada Para dar una solución",

```
0, 0, 0, 0, 0, "" );

stTEXTO stSERCONSU [] = {
  31, 4, "Consulta de Servicios",
  25, 7, "Fecha de Reporte:",
  25, 9, "Responsable:",
  25, 11, "Fecha de Atención:",
  25, 13, "Acción Tomada:",
  0, 0, ""};

stMARCO stmSERCONSU [] = {
  30, 4, 23, 1,
  14, 6, 58, 10,
  0, 0, 0, 0};

stLIGA stLIGA_SERVICIO = {0, 0};

stCAMPO stPSERVICIO [] = {
  56, 7, 4, 1, 4, "Muestra Valor Total de servicios atendidos calculo en Run Time",
  56, 9, 4, 1, 4, "Total de Servicios de Red",
  56, 11, 4, 1, 4, "Total de Servicios de Hardware",
  56, 13, 4, 1, 4, "Total de Servicios de Software",
  56, 15, 4, 1, 4, "Total de Servicios de Impresión",
  56, 17, 4, 1, 4, "Total de Otros Servicios ",
  0, 0, 0, 0, 0, "" };

stTEXTO stSERVICIO [] = {
  27, 3, "PANTALLA DE ESTADISTICAS",
  33, 4, "DE SERVICIO",
  16, 7, "Total de Servicios atendidos",
  16, 9, "Servicios de RED",
  16, 11, "Servicios de Hardware:",
  16, 13, "Servicios de Software:",
  16, 15, "Servicios de Impresión:",
  16, 17, "Otros Servicios",
  0, 0, "" };

stMARCO stmSERVICIO [] = {
  6, 6, 68, 14,
  0, 0, 0, 0};

stLIGA stLIGA_AREASERV = {0, 0};

stCAMPO stPAREASERV [] = {
  27, 5, 3, 1, 4, "Elija el tipo del problema",
  33, 5, 25, 1, 4, "Nombre de la Oficina",
  51, 8, 4, 1, 4, "Total de Reportes",
  51, 10, 4, 1, 4, "Total de Reportes de RED",
  51, 12, 4, 1, 4, "Total de Reportes de Hardware",
  51, 14, 4, 1, 4, "Total de Reportes de Software",
  51, 16, 4, 1, 4, "Total de Reportes de Impresión",
  51, 18, 4, 1, 4, "Total de Otros reportes",
  51, 20, 4, 1, 4, "Muestra Valor Total de servicios atendidos, calculo en Run Time",
  0, 0, 0, 0, 0, "" };

stTEXTO stAREASERV [] = {
  31, 3, "SERVICIOS POR AREA",
  17, 5, "Oficina:",
  21, 8, "Total de Reportes:",
  21, 10, "Reportes de RED",
  21, 12, "Reportes de Hardware:",
  21, 14, "Reportes de Software:",
  21, 16, "Reportes de Impresión:",
  21, 18, "Otros Reportes:",
  21, 20, "Total de Reportes Atendidos:"};
```

0, 0, "");

stMARCO stmAREASERV [] = {

15, 5, 49, 1,
9, 6, 60, 16,
0, 0, 0, 0);

stLIGA stLIGA_GENERAL = {0, 0};

stCAMPO stpGENERAL [] = {

46, 6, 4, 1, 4, "Total de Reportes",
51, 8, 4, 1, 4, "Muestra Valor Total de servicios atendidos calculo en Run Time",
28, 13, 4, 1, 4, "Total de Reportes de RED",
40, 13, 4, 1, 4, "Total de Servicios de Red",
58, 13, 2, 1, 4, "Frecuencia de Reportes de RED",
28, 15, 4, 1, 4, "Total de Reportes de Hardware",
40, 15, 4, 1, 4, "Total de Servicios de Hardware",
58, 15, 2, 1, 4, "Frecuencia de Reportes de Hardware",
28, 17, 4, 1, 4, "Total de Reportes de Software",
40, 17, 4, 1, 4, "Total de Servicios de Software",
58, 17, 2, 1, 4, "Frecuencia de Reportes de Software",
28, 19, 4, 1, 4, "Total de Reportes de Impresión",
40, 19, 4, 1, 4, "Total de Servicios de Impresión",
58, 19, 2, 1, 4, "Frecuencia de Reportes de Impresión",
28, 21, 4, 1, 4, "Total de Otros reportes",
40, 21, 4, 1, 4, "Total de Otros Servicios",
58, 21, 2, 1, 4, "Frecuencia de Otros Reportes",
0, 0, 0, 0, 0, "");

stTEXTO sttGENERAL [] = {

28, 3, "ESTADISTICAS GENERALES",
21, 6, "Total de Reportes:",
21, 8, "Total de Reportes Atendidos:",
15, 11, "FALLA",
26, 11, "REPORTES",
38, 11, "SERVICIOS",
52, 11, "FRECUENCIA (%)",
13, 13, "RED",
60, 13, "%",
13, 15, "Hardware:",
60, 15, "%",
13, 17, "Software:",
60, 17, "%",
13, 19, "Impresión:",
60, 19, "%",
13, 21, "Otros:",
60, 21, "%",
0, 0, "");

stMARCO stmGENERAL [] = {

11, 3, 58, 1,
11, 5, 58, 5,
11, 11, 58, 1,
11, 12, 58, 11,
24, 12, 12, 11,
36, 12, 14, 11,
50, 12, 19, 11,
0, 0, 0, 0);

stLIGA stLIGA_ENTREGA = {0, 0};

stCAMPO stpENTREGA [] = {

21, 8, 8, 1, 4, "Frecuencia de Otros Reportes",
21, 11, 3, 1, 4, "Elija el tipo del problema",
26, 11, 50, 1, 4, "escriba una descripción de hasta 50 caracteres",

21, 14, 3, 1, 4, "Elija el tipo del problema",
30, 14, 25, 1, 4, "Elija el tipo del problema",
21, 17, 3, 1, 4, "Cantidad de insumos a entregar",
21, 20, 6, 1, 4, "Introduzca la clave del usuario iniciando con letra",
30, 20, 30, 1, 4, "Introduzca el Nombre del Usuario iniciando con el apellido",
0, 0, 0, 0, 0, ""};

stTEXT0 stlENTREGA [] = {
36, 5, "ENTREGA DE INSUMOS",
13, 8, "Fecha:",
13, 11, "Oficina",
13, 14, "Insumos:",
13, 17, "Cantidad",
13, 20, "Usuario",
0, 0, ""};

stMARCO stmENTREGA [] = {
35, 5, 20, 1,
12, 7, 65, 6,
12, 13, 65, 9,
0, 0, 0, 0};

stLIGA stlLIGA_DEVOLUCI = {0, 0};

stCAMPO stpDEVOLUCI [] = {
31, 8, 8, 1, 4, "Fecha del pedido de insumos",
31, 10, 3, 1, 4, "Elija el tipo del problema",
35, 10, 25, 1, 4, "Nombre de la Oficina",
31, 14, 3, 1, 4, "Elija el tipo del problema",
35, 14, 25, 1, 4, "Elija el tipo del problema",
31, 16, 3, 1, 4, "Cantidad de insumos a entregar",
31, 18, 6, 1, 4, "Introduzca la clave del usuario iniciando con letra",
39, 18, 30, 1, 4, "Introduzca el Nombre del Usuario iniciando con el apellido",
0, 0, 0, 0, 0, ""};

stTEXT0 stlDEVOLUCI [] = {
29, 3, "DEVOLUCION DE INSUMOS",
21, 8, "Fecha:",
21, 10, "Oficina:",
21, 14, "Insumos:",
21, 16, "Cantidad:",
21, 18, "Usuario:",
0, 0, ""};

stMARCO stmDEVOLUCI [] = {
15, 6, 59, 6,
15, 12, 59, 9,
0, 0, 0, 0};

stLIGA stlLIGA_CAMBIOS = {0, 0};

stCAMPO stpCAMBIOS [] = {
26, 6, 8, 1, 4, "Fecha del pedido de insumos",
26, 8, 3, 1, 4, "Elija el tipo del problema",
31, 8, 25, 1, 4, "Nombre de la Oficina",
26, 12, 3, 1, 4, "Elija el tipo del problema",
31, 12, 25, 1, 4, "Elija el tipo del problema",
26, 14, 3, 1, 4, "Cantidad de insumos a entregar",
26, 16, 6, 1, 4, "Introduzca la clave del usuario iniciando con letra",
33, 16, 30, 1, 4, "Introduzca el Nombre del Usuario iniciando con el apellido",
0, 0, 0, 0, 0, ""};

stTEXT0 stlCAMBIOS [] = {
31, 3, "CAMBIOS DE INSUMOS",
16, 6, "Fecha:",

16, 8, "Oficina:",
16, 12, "Insumo:",
16, 14, "Cantidad:",
16, 16, "Usuario:",
0, 0, ""};

sIMARCO stmCAMBIOS [] = {

11, 5, 56, 5,
11, 10, 56, 9,
0, 0, 0, 0};

sILIGA stLIGA_REPORTES = {0, 0};

sICAMPO stpREPORTES [] = {

0, 0, 0, 0, 0, ""};

stTEXTO stIREPORTES [] = {

0, 0, ""};

sIMARCO stmREPORTES [] = {

0, 0, 0, 0};

sILIGA stLIGA_REP_ESTA = {0, 0};

sICAMPO stpREP_ESTA [] = {

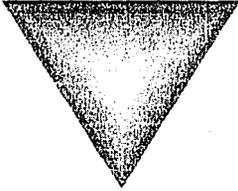
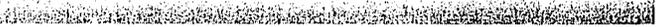
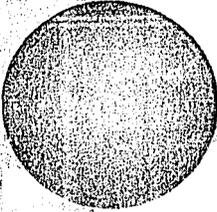
4, 8, 2, 1, 4, "Folio de Recepción de Reportes",
8, 8, 25, 1, 4, "Nombre de la Oficina",
40, 8, 2, 1, 4, "Elija el tipo del problema",
45, 8, 8, 1, 4, "Prioridad de Atención de Reportes",
58, 8, 8, 1, 4, "Elija el tipo del problema",
69, 8, 8, 1, 4, "Fecha Propuesta para solución del problema",
8, 9, 30, 1, 4, "Introduzca el Nombre del Usuario Iniciando con el apellido",
45, 9, 25, 1, 4, "Acción Tomada Para dar una solución",
0, 0, 0, 0, 0, ""};

sITEXTO stIREP_ESTA [] = {

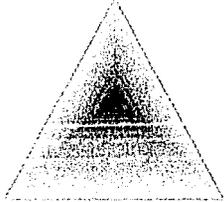
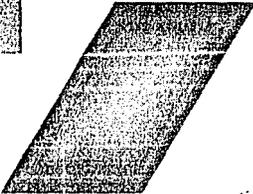
30, 3, "REPORTE DE SERVICIOS",
17, 5, "OFICINA",
45, 5, "PRIORIDAD",
59, 5, "FECHA",
70, 5, "FECHA",
3, 6, "FOLIO",
17, 6, "USUARIO",
39, 6, "FALLA",
45, 6, "ATENDIO",
58, 6, "REPORTE",
69, 6, "ATENCION",
0, 0, ""};

sIMARCO stmREP_ESTA [] = {

27, 3, 26, 1,
2, 5, 75, 2,
2, 7, 6, 15,
8, 7, 30, 15,
38, 7, 7, 15,
45, 7, 11, 2,
56, 7, 10, 2,
66, 7, 11, 2,
0, 0, 0, 0};



Conclusiones



CONCLUSIONES

La Tecnología CASE contempla dos partes fundamentales e íntimamente relacionadas que son la Metodología CASE y las Herramientas CASE:

- La Metodología CASE es una serie de pasos bien delineados que al ejecutarse correctamente permiten desarrollar un producto de software robusto y eficiente que cubre las necesidades para lo que fué diseñado, con la ventaja de optimizar los recursos humanos y materiales que participaron en su desarrollo.
- Las Herramientas CASE aceleran, facilitan y optimizan el proceso del desarrollo de sistemas de software siguiendo la Metodología CASE.

Precisamente para acelerar y facilitar el presente sistema de software se utilizó la Tecnología CASE para las fases de análisis y de diseño, que son las que llevan el mayor peso cuando se utiliza ésta.

En cuanto a los productos CASE comerciales, aún vemos desafortunadamente que el costo de adquisición es aún alto, sin embargo la relación costo-beneficio alcanzada es realmente atractiva y lo hace más que costeable.

El sistema GPCASE cubrió el objetivo propuesto que fué liberar un sistema que apoyara al diseñador para desarrollar la estructura de sus aplicaciones o *Front End*, liberándole de la carga de este diseño y proporcionándole una base a la cual sólo falta agregar los procedimientos que son particulares del sistema final.

GPCASE es un conjunto de herramientas de desarrollo gráfico para diseñar, crear, depurar y mantener una gran variedad de aplicaciones también gráficas con las cuales interactúan los usuarios. Esta característica permite a los diseñadores seleccionar los iconos correspondientes para entrar datos, actualizar información, manejar varios proyectos, cambiar rápidamente su diseño, etc. Esto representa una gran flexibilidad para poder construir o hacer prototipos con rapidez. La facilidad de utilización del sistema GPCASE también lo hace una herramienta atractiva para los usuarios más avanzados, que pueden diseñar complejas aplicaciones con un mínimo de entrenamiento.

Durante el diseño existen refinamientos progresivos de la estructura de los datos, de la estructura del programa, de los procedimientos, revisiones y documentaciones. El resultado de tener un sistema dinámico se aprecia por la rapidez de su desarrollo y calidad en un sistema final.

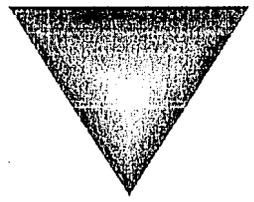
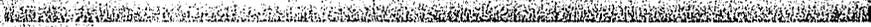
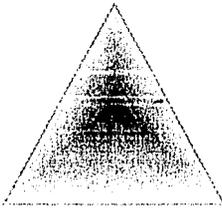
El prototipaje es la forma de crear el modelo del software que se debe construir. El modelo puede tomar una de las tres formas siguientes:

1. Un sistema creado directamente con la PC que ayuda tanto al diseñador como al usuario final a comprender el funcionamiento del futuro sistema.
2. Un prototipo implementado con algunos de los subsistemas del sistema final.
3. Un programa existente con algunas características deseadas pero con una finalidad distinta que ahorra tiempo.

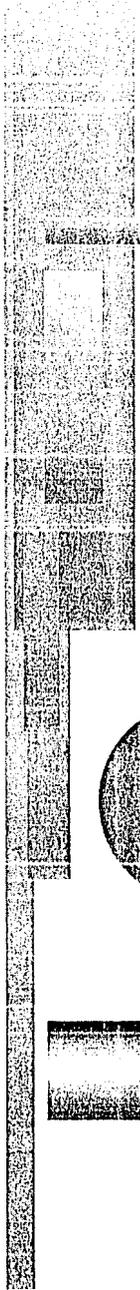
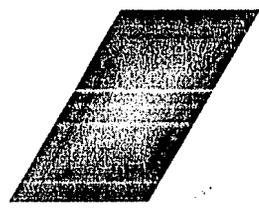
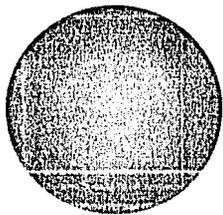
El prototipo es evaluado por el usuario final y el diseñador con la finalidad de refinar requerimientos para el desarrollo del sistema final.

Las características de GPCASE lo hacen ideal para acelerar el aprendizaje de la Metodología CASE utilizando una herramienta que no requiere grandes recursos. Además cuenta con un sistema de ayuda que lleva al usuario a la fácil consulta de los temas y en algunas secciones recibe también recomendaciones de utilización del software, prescindiendo en lo posible del manual de usuario en forma escrita y aplicando en forma inmediata sus conocimientos teóricos.

Es factible agregar otras características a GPCASE para hacerlo crecer ya que está diseñado de tal forma que las modificaciones requeridas se constituyan en un nuevo módulo, claro que esto será resultado del análisis, tiempo y esfuerzo correspondientes.



Glosario



GLOSARIO

CASE

Es el acrónimo de *Computer Aided Software Design* (Diseño de Software Asistido por Computadora).

BTRIEVE

Manejador de bases de datos de Novell.

Client area

Área de trabajo de una ventana.

NOVELL

Compañía que desarrolló el sistema operativo NetWare para redes locales.

Bitmap

Un *Bitmap* es un elemento gráfico que está constituido por un patrón de bits almacenados en memoria.

Barra de status

Es la barra que está en el tope inferior de GPCASE e indica el estado del proceso actual en GPCASE.

Caption

Es la barra superior horizontal de cualquier ventana en donde se aprecia por lo general el nombre de esa ventana.

Check Box

Es un pequeño recuadro que aparece por lo general acompañado de un texto que define su función. Si dentro del recuadro aparece una cruz en forma de X entonces esa función está activa, así un recuadro vacío indica que esa función no tiene efecto.

Check Mark

Esta es una marca generalmente en menús, que aparece en forma de "palomita" a la izquierda de un *menú* indicando con esto que esa opción está activa.

Child Window

Es una ventana que depende de otra y todos los mensajes que reciba generalmente afectan a la cual pertenecen.

Client area

Es el área de trabajo que encierra el contorno de una ventana.

ComboBox

Es la unión de dos elementos: una *ListBox* y un *Edit*, en el *edit* aparece la selección que está activa y cambia reflejando la selección del cursor actual en la *ListBox*.

CommonDialogs

Son elementos en todos los productos del ambiente Windows que permiten realizar tareas, comunes de procesamiento como manejo de archivos, impresiones, etc.

Cursor

Es una imagen *bitmap* pequeña que indica donde apunta el dispositivo de interacción, generalmente en Windows un mouse.

Dialog

Los *Dialogs* son un tipo especial de ventanas de Windows que tienen controles para la comunicación con el usuario.

Double Word

Es un tipo de dato propio de Windows equivalente a un *int* (entero) en lenguaje C.

Edit

Es una *child window* que permite ingresar información por teclado.

Extern

En Lenguaje C *extern* declara que una variable ya ha sido definida y se desea utilizar su estado actual en el nuevo archivo.

Font

Los *fonts* son todos los tipos de letras que se pueden manejar en una aplicación Windows.

Front End

Es la parte visual de un sistema de software.

GroupBox

Es un rectángulo que enmarca una serie de controles que tienen cierta relación, al elegir alguno de los controles los demás automáticamente se deshabilitan.

Handles

Son manejadores de los recursos del sistema.

Header

Significa encabezado, en el Lenguaje C se coloca al inicio de otro archivo con el fin de considerar su contenido en tiempo de compilación.

Iconos

Es una pequeña imagen utilizada para representar un componente de la aplicación.

ListBox

Es un elemento en forma de rectángulo en la cual se presenta una lista de elementos en columna susceptibles de ser seleccionados por medio del cursor.

Long Pointer String

Es un tipo de dato equivalente a un far double long en lenguaje C.

Modal Dialog

Al desplegarse este *dialog*, la aplicación se detiene hasta que el usuario cierre el *dialog*.

Modeless Dialog

El despliegue de este tipo de *dialog* no detiene la ejecución de la aplicación, pudiendo permanecer mientras dure la misma.

Mouse

Accesorio de la computadora que controla el cursor y permite una fácil interfase usuario-máquina, agilizando muchos procedimientos de pantalla.

Path

Significa ruta, ésta describe la posición exacta de un archivo dentro de la estructura del sistema de archivos.

Pipe

Es un símbolo de la tabla ASCII (124 decimal), utilizado en GPCASE para la separación de campos.

Push Button

Es una ventana rectangular pequeña que semeja un botón que lleva dentro de sí una etiqueta que indica la operación que se realizará si se presiona.

Radio Button

Son elementos que se reconocen porque son pequeños círculos que corresponden a un texto que en la forma común está a su derecha. Cuando se activan, los *radiobuttons* presentarán un punto en el centro del círculo respectivo, de otra forma aparecen los círculos vacíos.

Scroll Bar

Es una barra que aparece en los márgenes de una ventana. Es una barra delgada que tiene por extremos dos flechas en sentido opuesto divergente, dentro de la barra hay un cuadrado resaltado que indica la posición del cursor en la ventana.

SpreadSheet

Hoja de cálculo electrónica.

Stack

Es un área de trabajo para almacenar direcciones de procesos.

Sys Menu

Es un recuadro que aparece en la esquina superior izquierda de una ventana y que tiene un *bitmap* en forma de guión. Al elegirse éste se presenta un *menu* de persiana en donde las opciones corresponden a las operaciones de manejo de la ventana a la que pertenecen, como maximizarla, cerrarla, cambiarse a otra ventana, etc.

ToolBox

Toolbox es una barra de herramientas, un *toolbox* puede presentarse de dos formas principalmente, la primera es como un renglón de iconos bajo el *menú* de la aplicación y la segunda es como un *dialog* flotante que se puede cambiar de posición en la pantalla para apreciar ciertos objetos que están atrás de ésta.

Menú

En GPCASE un *menú* es cualquiera de las opciones que aparecen en el *menú principal* de GPCASE o en el *menú principal* de cada proyecto.

Item

Un *item* es una de las opciones que aparecen en forma vertical al seleccionar un *menú*.

Pantalla

Una *pantalla* es un módulo que permitirá al usuario comunicarse con la computadora, mediante la presentación de una serie de campos que presentan información.

Reporte

Un *reporte* es un módulo cuya función es presentar el resultado del procesamiento de cierta información.

Proyecto

El *Proyecto* es aquél que engloba todo el diseño de un sistema en particular, esto incluye pantallas, reportes, *menús*, etc.

Navegación

Es la forma en que se pasa de un proceso a otro, en este caso podemos hablar de que *ítem* se va a cierto proceso y como se desprende un proceso de otro proceso o de un *menú principal* o de un *ítem*.

BIBLIOGRAFIA

Documenting the Software Development

Steve Ayer y Frank Patrinostro
McGraw Hill, 1992

Software Engineering, Methods & Management

Anneliese von Mayrhauser
Academic Press, 1990

Fundamentals of Software Engineering

Carlo Ghezzi, Mehdi Jazayeri and Dino Mandrioli
Prentice Hall Press, 1991

Computer-Aided Software Design

Max Schindler
John Wiley & Sons Press, 1990

Software Specification & Design

Marilyn Keller and Ken Shumate
John Wiley & Sons Press, 1992

Análisis y Diseño de Sistemas

Kendall y Kendall
Prentice Hall Press, 1991

Software Engineering

Roger s. Pressman
A PRACTITIONER'S APPROACH
McGraw-Hill International Editions
Thrid Edition, 1992

El lenguaje de Programación C

Brian W. Kernighan & Dennis M. Ritchie
Segunda Edición, Ed Prentice Hall 1991 .

Guide to Programming,

Microsoft Windows Ver. 3.1,
Programmer.s Reference Library
Microsoft Press, 1992

Programmer's Reference Volume 1: Overview,

Microsoft Windows Ver. 3.1
For the Microsoft Windows Operating
System
Microsoft Press, 1992

Programmer's Reference Volume 2: Functions,

Microsoft Windows Ver. 3.1
For the Microsoft Windows Operating
System
Microsoft Press, 1992

Programmer's Reference Volume 3: Messages, Structures, and Macros,

Microsoft Windows Ver. 3.1 ,
For the Microsoft Windows Operating
System
Microsoft Press, 1992

Drover's Professional ToolBox for Windows,

Library Reference Versión 2.0 August
17, 1992
FarPoint

User's Guide For the Windows Graphical Environment, Ms DOS Operating System,

Microsoft Windows Ver. 3.1 , Gateway
200, 1992

Btrieve

NetWare , Novell
1991