

UNIVERSIDAD POPULAR AUTÓNOMA DEL ESTADO DE PUEBLA

SISTEMAS COMPUTACIONALES

**INGENIERIA DE SOFTWARE PARA EL DEPARTAMENTO
DE REVISION DE PROCESOS DE LA UNIVERSIDAD POPULAR
AUTONOMA DEL ESTADO DE PUEBLA**

T E S I S

Que para obtener el título:

LICENCIADO EN SISTEMAS COMPUTACIONALES

Presentan:

**ROLANDO FABRICIO PENAGOS MACAL
EFRAÍN SALOMÓN FLORES**

Director de tesis:

L.C.A. JAIME F. CASTILLO RODRÍGUEZ

H. Puebla de Zaragoza, a 11 de abril de 1997



UPAEP – Secretaría General

Dirección General de Apoyos Académicos

Dirección del Centro de Recursos para el Aprendizaje y la Investigación.

Biblioteca Central - **Karol Wojtyła**

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 textos, imágenes, gráficas, 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 de donde la obtuvo mencionando el autor o autores involucrados en el documento.

Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

A Dios...

A mi Padre...

A mi asesor Jaime y a Conny...

A mi novia Brenda ...

A mis amigos Concepción, Ana Dehely, Moisés, Rolando...

Efraín Salomón Flores

Agradecimientos

A Dios, por darme la paciencia necesaria para realizar este trabajo, y más que nada, por darme vida.

A mis padres, María Luisa y Mariano, por darme siempre su apoyo, aún cuando sea una gran distancia la que nos separe.

A Efraín, por haber sido mi compañero de tesis y sobre todo por ser mi amigo. Afortunada o desafortunadamente, logramos terminar satisfactoriamente este trabajo... ¡Buena suerte!

A Moisés, por mantener la unidad entre nosotros tres, además de brindarme su comprensión y paciencia en todo momento.

A Conchita, por brindarme su amistad y por ayudarme a sobrellevar las etapas difíciles por las que atravesé.

A Ana y Brenda, por compartir conmigo su amistad, además de momentos que hicieron más llevadera la elaboración de este trabajo.

A Jaime, indudablemente por darme la oportunidad de realizar esta tesis y por demostrar que el ser honesto con uno mismo es lo que más rinde frutos; y a Connie, ya que sin su ayuda este trabajo dejaría mucho que desear.

Y también a...

Rolando Fabricio Penagos Macal

ÍNDICE

Introducción	1
Capítulo 1. Ingeniería de Software	3
1.1 Aspectos Generales de la Ingeniería de Software.	3
1.1.1 Historia de la Ingeniería de Software.....	3
1.1.2 Elementos de la Ingeniería de Software.....	5
1.2 La crisis del software	6
1.2.1 Problemas	7
1.2.2 Causas	9
1.2.2.1 Complejidad	10
1.2.2.2 Conformidad	11
1.2.2.3 Adaptabilidad	11
1.2.2.4 Invisibilidad.....	12
1.3 Modelos para el desarrollo de software	12
1.3.1 Modelo Construir y Arreglar.....	13
1.3.2 Modelo de Cascada	14
1.3.3 Modelo de Prototipo Rápido.....	16
1.3.4 Modelo Incremental	18
1.3.5 Modelo de Espiral.....	20
1.4 Proceso de desarrollo de software	23
1.4.1 Investigación preliminar.....	24
1.4.2 Determinación de los requerimientos del sistema	24
1.4.3 Diseño del sistema	25
1.4.4 Desarrollo de software.....	25
1.4.5 Prueba de los sistemas	25
1.4.6 Implantación y evaluación	26
1.4.7 Fase de mantenimiento.....	27
1.5 Herramientas CASE.....	27
1.5.1 Características.....	28
1.5.2 Lower CASE.....	29
1.5.3 Upper CASE.....	29
1.5.4 Ventajas	30
Capítulo 2. Encuesta	31
2.1 Objetivo.....	31
2.2 Determinación de la Muestra	31
2.3 Elaboración de la Encuesta	31
2.4 Análisis e Interpretación de la Encuesta.....	32
2.4.1 Preguntas Relacionadas con la Ingeniería de Software.....	32
2.4.2 Preguntas Sobre la Experiencia al Elaborar Sistemas	35

2.4.3 Preguntas Relacionadas con Definiciones	38
2.4.4 Preguntas sobre la Utilización de Software, Metodologías y Documentación	42
2.5 Formato de Encuesta	47
2.6 Resultados.....	48
Capítulo 3. Benchmarking	50
3.1 Consideraciones a Tomar en el Estudio de Benchmarking	50
3.1.1 Sistema Operativo	51
3.1.2 Plataforma de Hardware.....	52
3.1.3 Red.....	53
3.1.4 Protocolos de Red	53
3.1.5 Acceso a Internet	54
3.1.6 Tipo de Herramienta.....	55
3.1.7 Tipo de Aplicación	56
3.1.8 Conectividad a Través de SQL.....	56
3.1.9 Conectividad a Través de ODBC.....	57
3.1.10 Interfases de Aplicación	58
3.1.11 Características Generales de las Bases de Datos	59
3.1.12 Incorporación de Código de Otros Lenguajes	60
3.1.13 Herramientas de Desarrollo.....	61
3.1.14 Herramientas de Administración	62
3.1.15 Seguridad de Acceso	63
3.1.16 Triggers	64
3.1.17 Importación y Exportación de Archivos	65
3.1.18 Indexación	65
3.1.19 Bases de Datos Soportadas en Forma Nativa	66
3.1.20 Otras Características.....	67
3.2 Resultados.....	69
Capítulo 4. Estudio de Desarrolladores.....	71
4.1 Objetivo	71
4.2 Análisis de la Capacidad de una Hora de Trabajo.....	71
4.2.1 Objetivo	71
4.2.3 Desarrollo	71
4.2.3 Resultados	73
4.2.4 Reporte.....	76
4.3 Análisis de la Capacidad de un Día de Trabajo	78
4.3.1 Objetivo	78
4.3.2 Fase Uno.....	78
4.3.3 Fase Dos	96
4.4 Resultados del Estudio de Desarrolladores	113

Capítulo 5. Aplicación Documentada.....	114
5.1 Instalación de Progress	114
5.2 Instalación de la Ingeniería de Software	114
5.3 Desarrollo de la Aplicación	117
5.3.1 Documentación Necesaria	117
5.3.2 Construcción de los Menús de Iconos.....	119
5.3.2.1 Construcción del Menú Principal (PSSC0001.W)	122
5.3.2.2 Construcción del Menú de Archivos (PSSC1000.W)	126
5.3.2.3 Construcción del Menú de Archivos Maestros (PSSC1100.W).....	127
5.3.2.4 Construcción del Menú de Archivos de Movimientos (PSSC1200.W).....	128
5.3.2.5 Construcción del Menú de Reportes (PSSC3000.W).....	129
5.3.2.6 Construcción del Menú de Reportes de Maestros (PSSC3100.W).....	130
5.3.2.7 Construcción del Menú de Reportes de Movimientos (PSSC3200.W).....	131
5.3.2.8 Construcción del Menú de Procesos Especiales (PSSC4000.W).....	132
5.3.2.9 Construcción del Menú de Consultas (PSSC4100.W)	133
5.3.2.10 Construcción del Menú de Consultas de Maestros (PSSC4110.W).....	134
5.3.2.11 Construcción del Menú de Consulta de Movimientos (PSSC4120.W)	135
5.3.2.12 Construcción del Menú de Gráficas (PSSC4200.W).....	136
5.3.2.13 Construcción del Menú de Estadísticas (PSSC4300.W).....	137
5.3.3 Construcción de la Ventana de ABC del Maestro de Actividades (PSSC1110.W)..	138
5.3.3.1 Creación del Archivo de Inclusión de Actividades (ABC-ACTV.I).....	139
5.3.3.2 Creación del SmartQuery de Actividad (Q-ACT.W)	141
5.3.3.3 Creación del SmartViewer de Actividad (V-ACT.W).....	145
5.3.3.4 Integración de los Objetos en la Ventana de ABC del Maestro de Actividades (PSSC1110.W)	147
5.3.4 Construcción de una Ventana de ABC de Movimientos (PSSC1210.W)	149
5.3.4.1 Creación del Archivo de Inclusión de Avance de Programa de Salud (ABC-AVPS.I).....	150
5.3.4.2 Creación del SmartQuery de Avance de Programa de Salud (Q-AVPS.W)...	151
5.3.4.3 Creación del SmartViewer de Avance de Programa de Salud (V-AVPS.W)..	154
5.3.4.4 Creación del Cuadro de Diálogo de Selección de Programas (BSSCPROG.W).....	159
5.3.4.4.1 Creación del Archivo de Inclusión de Programas (ABC-PROG.I)	160
5.3.4.4.2 Creación del Archivo del SmartBrowse de Programas (B1-PROG.W) .	162
5.3.4.4.3 Integración del Cuadro de Diálogo de Selección de Programas (BSSCPROG.W).....	164
5.3.4.5. Creación del Cuadro de Diálogo de Actualización de Programas (VSSCPROG.I).....	166
5.3.4.5.1. Creación del SmartViewer de Programas (V-PROG.W).....	166
5.3.4.5.2. Creación del SmartQuery de Programas (Q-PROG.I)	169
5.3.4.5.3. Integración de los Objetos en el Cuadro de Diálogo de Actualización (VSSCPROG.I)	172
5.3.4.6 Integración de los Objetos en la Ventana de ABC (PSSC1210.W)	174
5.3.5 Construcción de un Reportes de Maestros (RSSC3110)	176
Figura 5.85. Cuadro de Diálogo Open Report.	177

Figura 5.86. Plantilla de Reporte	177
Figura 5.87. Cuadro de Diálogo Master Table.....	178
Figura 5.88. Cuadro de Diálogo New Join	178
Figura 5.91. Cuadro de Diálogo Insert Ffield.....	180
Capítulo 6. Resultados del Trabajo de Ingeniería de Software.....	182
6.1 Ingeniería de Software.....	182
6.1.1 Panorama General	182
6.1.2 Resultados	188
6.2 Establecimiento de Estándares	189
6.2.1 Panorama General	190
6.3 Capacitación	194
6.3.1 Panorama General	194
6.3.2 Resultados	196
6.4 Coordinación de Grupo	196
6.4.1 Panorama General	197
6.4.2 Resultados.....	198
6.5 Soporte Técnico.....	199
6.5.1 Panorama General	199
6.5.2 Resultados	200
6.6 Control de Calidad	200
6.6.1 Panorama General	201
6.6.2 Resultados	201
Conclusión	202
Glosario	203
Bibliografía	207

INTRODUCCION

La producción de software en México se ha visto favorecida en los últimos años, debido a la apertura del mercado mexicano. Por ello muchas empresas de este ramo se encuentran en una etapa de crecimiento que no sólo exige el mercado que necesita de sus productos, sino también la competencia generada por el Tratado de Libre Comercio, que obliga a que su producción sea de alta calidad, a un bajo costo y que cumplan con los tiempos de entrega.

Desde la década de los 60, el aumento en la complejidad de los sistemas de cómputo y el crecimiento de las necesidades a satisfacer, hicieron que el software entrara en una crisis que perdura hasta ahora, ya que se siguen teniendo problemas por no usar metodologías para desarrollar, por lo que sigue habiendo desperdicio de dinero. Paralelamente a la aparición de diversas metodologías para el desarrollo, la Ingeniería de Software ha demostrado ser una disciplina, que bien implantada, puede resolver los problemas de calidad, costo y entrega a tiempo, entre otros.

Por eso, el uso de la Ingeniería de Software se hace imprescindible en cualquier empresa o institución en la que se desarrolle software y pretenda obtener productos con calidad; y más ahora con el uso de sistemas operativos amigables e intuitivos, basados en una interfase gráfica de usuario, que adquieren popularidad rápidamente, ayudando al software extenderse más rápidamente.

El objetivo principal del presente trabajo es presentar la Ingeniería de Software así como los estándares establecidos, para incrementar la productividad y disminuir el tiempo de desarrollo. Se incluye la documentación técnica, su tratamiento y utilización, además de una aplicación que demuestra el uso de la ingeniería. Este proyecto es producto del análisis e investigación sobre la situación actual en el desarrollo de software, aplicando encuestas a los alumnos de la escuela de Sistemas Computacionales en la UPAEP y realizando estudios prácticos sobre el equipo de desarrollo formado en la misma Universidad para la realización de los proyectos SINAF y SINIGA.

En el primer capítulo, "Ingeniería de Software", se dan a conocer los antecedentes de esta disciplina, con la finalidad de ofrecer una visión de la importancia que tiene en el desarrollo de sistemas.

En el segundo capítulo, "Encuestas", se presentan los resultados de la encuesta aplicada a los grupos de quinta generación y sexta generación de los alumnos de la escuela de Sistemas Computacionales y el análisis de los resultados obtenidos.

En el tercer capítulo, "Benchmarking", se presenta el estudio comparativo de los principales manejadores de bases de datos relacionales existentes en el mercado, el cual da a conocer sus diversas características y en base a esto el por qué de la elección del manejador de base de datos Progress para el desarrolló.

El cuarto capítulo, "Estudio de Desarrolladores", presenta las pruebas que se realizaron con los tesisas del proyecto UPAEP, para determinar diversos aspectos, como el nivel técnico, el uso de técnicas de programación y la rapidez de desarrollo.

En el quinto capítulo, "Aplicación Documentada", se explica como se aplicó la Ingeniería de Software desarrollada en este trabajo a un módulo de SINAF.

El sexto capítulo, "Resultados del Trabajo de Ingeniería de Software", presenta los resultados del trabajo realizado con el grupo de tesisas a lo largo del desarrollo de los proyectos SINAF y SINIGA.

Capítulo 1. Ingeniería de Software

Dado el objetivo principal de este trabajo, que es el desarrollo de Ingeniería de Software para el Departamento de Revisión de Procesos, se presentan en este capítulo los conceptos básicos y fundamentales de la Ingeniería de Software, así como la importancia de esta disciplina en el desarrollo de software.

1.1 Aspectos Generales de la Ingeniería de Software

Para hablar de la Ingeniería de Software, se debe saber qué es, cómo se aplica y cuáles son sus ventajas en el proceso de desarrollo de sistemas. Tomando como base la definición de Pressman (1991), se tiene lo siguiente:

La Ingeniería de Software es una rama de la ingeniería en la que se aplican principios robustos, orientados a lograr soluciones costo-efectivas en el desarrollo de software.

Se han propuesto muchas definiciones, pero todas resaltan la importancia de una disciplina de ingeniería para el desarrollo de software.

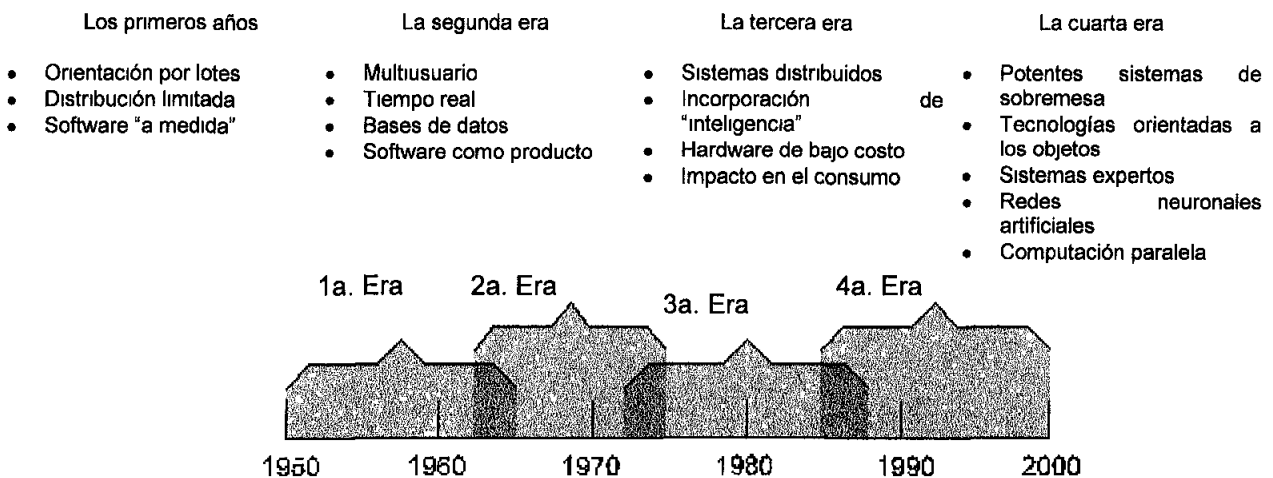
Se deben establecer y usar principios de ingeniería robustos, tal como lo menciona la definición, los cuales abarcan tres elementos: métodos, herramientas, y procedimientos. La siguiente parte de la definición describe el objetivo de la Ingeniería de Software. Como se explica más adelante, uno de los problemas más frecuentes del desarrollo de sistemas ha sido el elevado costo del mantenimiento.

1.1.1 Historia de la Ingeniería de Software

El origen de la Ingeniería de Software se remonta a una conferencia de la OTAN celebrada en Roma en 1968, cuyo objetivo principal era establecer los principios generales de ingeniería al desarrollo del software, con lo cual se pretendía resolver los problemas de productividad y calidad a los que se enfrenta todo software.

En el principio de la era de la informática, toda la atención de las personas estaba centrada en el hardware de las computadoras. Los principales puntos de atención eran la reducción del costo de procesamiento y almacenamiento de los datos. Hacia los años ochenta, se lograron importantes avances en la microelectrónica, alcanzándose a cubrir los puntos antes mencionados. Estos avances se pueden apreciar en la Figura 1.1, que muestra la evolución del software.

Figura 1.1 Evolución del Software.



Fuente: Pressman, 1993.

Hasta antes de que se establecieran los principios de la Ingeniería de Software, la producción de software era prácticamente un arte, es decir, muchos aspectos del desarrollo dependían del criterio del desarrollador, ya que existían pocos métodos sistemáticos. El desarrollo del software se hacía sin una planeación y por lo tanto los costos crecían y el mantenimiento era muy difícil; especialmente si el personal que lo desarrolló y el que lo mantuvo eran diferentes. La mayoría del software lo desarrollaba y lo utilizaba una sola persona, la cual lo escribía, lo ejecutaba y lo depuraba; por lo tanto el diseño estaba implícito y no había documentación. Además, si otra persona se involucraba al proceso de desarrollo, uso o mantenimiento, se encontraba perdida debido a esa falta de disciplina.

Hacia 1970 surgía la programación estructurada. Esta se constituía como la condición para la nueva disciplina de ingeniería. En la década de los 70's se realizaron importantes avances tanto en la industria como en las escuelas, avances que le daban nuevas características a los productos de software. Algunas de esas características eran : mayor confiabilidad, mantenimiento más fácil, mayor robustez, etc.

Conceptos tales como la programación estructurada y las metodologías estructuradas alcanzaron gran popularidad. Estas metodologías se enfocaban al

análisis y al diseño. Dichos avances fueron los que posteriormente ocasionarían la consolidación de la Ingeniería de Software.

Para 1982, el Departamento de la Defensa de Estados Unidos hizo una revisión de los distintos problemas del software, dando origen a diferentes iniciativas, iniciativas como la reanimación del programa Ada, la creación del *Software Engineering Institute* (SEI) y el programa STARTS (*Software Technology for Adaptable Reliable Systems*). La industria creó programas similares tales como *Software Productivity Consortium* y la división de software del *Micro-Electronics and Computers Consortium*. Estas iniciativas también fueron seguidas en otros países como Japón y varios en Europa.

Durante la década de los 80's se desarrollaron otros conceptos, tales como métricas de software, las herramientas CASE, el empleo de prototipos y se introdujo la reingeniería y el paradigma de los objetos, aunque también se siguieron trabajando y mejorando los métodos formales.

1.1.2 Elementos de la Ingeniería de Software

En la sección 1.1 se mencionaron los tres elementos que abarca la Ingeniería de Software, los cuales se explican brevemente a continuación:

- Los *métodos* indican cómo se va a construir el software desde un punto de vista técnico. Esto lleva a tener una notación especial, orientada por supuesto a un lenguaje, además de proporcionar un conjunto de criterios que servirán para evaluar la calidad de software.
- En cuanto a las *herramientas*, éstas ayudan a realizar las tareas de los métodos de una manera automática o semiautomática. Lo ideal sería contar con herramientas automáticas que auxiliarían durante todo el proceso de ingeniería, pero éstas no están disponibles en el mercado; por lo menos no hay una que haga todo, aunque existen diferentes herramientas que pueden auxiliar en algunos de los puntos del proceso. Si se integran las herramientas de manera que la información que proporciona una de ellas pueda ser utilizada como datos de entrada para otra herramienta, entonces se está hablando de un sistema para el soporte del desarrollo del software. Lo anterior se conoce como Ingeniería de Software Asistida por Computadora, o mejor conocida por sus siglas en ingles, CASE.
- Por otra parte los *procedimientos* ayudan a unir los métodos y las herramientas, de tal forma que facilitan un desarrollo oportuno y racional del software. En los procedimientos se definen las secuencias en las que se van a ir aplicando los

métodos, la entrega de documentación, las herramientas y los controles para el aseguramiento de la calidad.

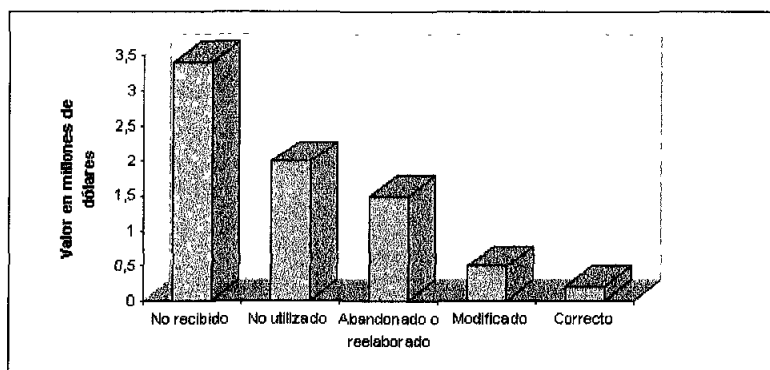
1.2 La crisis del software

Hasta ese momento el software no había tenido la atención debida, y desafortunadamente hasta hoy en día no se le sabe apreciar debidamente, ya que éste es el que permite utilizar y explotar el potencial del hardware; por lo tanto, se tiene hardware muy poderoso, y día a día, se logran avances importantes en esta área, pero se sigue descuidando el software, por lo que sin la aplicación de metodologías y herramientas adecuadas para su desarrollo no se lograrán los niveles de calidad y productividad exigidos en la actualidad.

La demanda social del software va incrementándose a pasos gigantescos conforme pasa el tiempo (Figura 1.2), al igual que los costos de éste. El comportamiento que se ha venido teniendo en la industria ha sido el de calendarios incumplidos, mala estimación del costo, problemas en cuanto a la calidad, etc. Esto en la industria es conocido como la crisis del software. Según el artículo de Soluciones Avanzadas (julio, 1994):

“En la década de 1970, el departamento de la Defensa de Estados Unidos (DoD, Department of Defense), el mayor usuario de computadoras del mundo, informó que la mayor parte de los proyectos de software no fueron terminados (aún cuando pagó por ellos), o los sistemas resultantes no se utilizaron o le fueron entregados con errores importantes de modo que tuvieron que abandonarlos o reelaborarlos o modificarlos” (Figura 1.2).

Figura 1.2 La suerte de los proyectos del DoD.



Fuente: Soluciones Avanzadas, Julio 1994.

1.2.1 Problemas

Los problemas con la producción del software están estrechamente relacionados con la evolución de los sistemas informáticos (Pressman, 1993). Según menciona Pressman (1993), existen tres principales problemas con los que se enfrenta la producción de software:

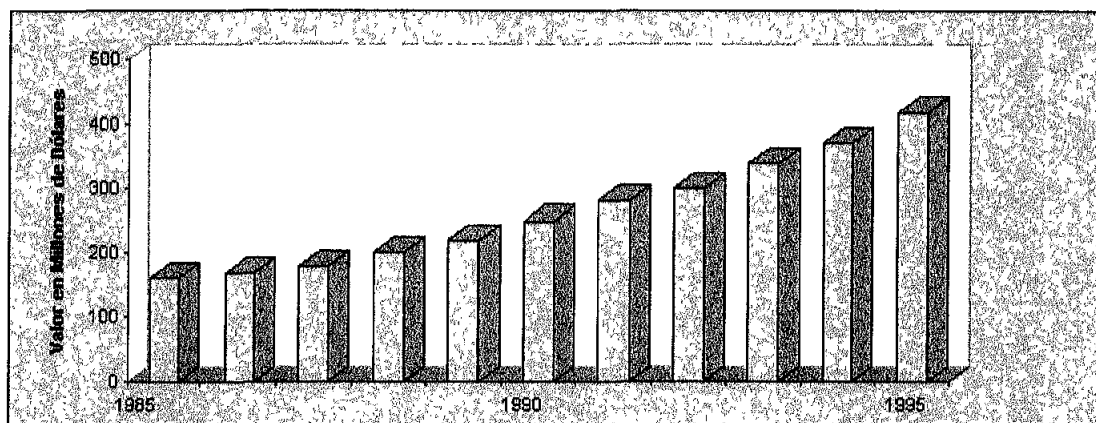
1. *La sofisticación del hardware ha dejado desfasada nuestra capacidad de construir software que pueda explotar el potencial del hardware.*
2. *Nuestra capacidad de construir nuevos programas no puede dar abasto a la demanda de nuevos programas.*
3. *Nuestra capacidad de mantener los programas existentes está amenazada por el mal diseño y el uso de recursos inadecuados.*

Sin embargo, para plantear una mejor perspectiva de los problemas del software, se reúnen en tres tipos: problemas de costo, problemas de calidad de software y problemas en la productividad.

En relación a los *problemas de costo*, el costo del software ha rebasado por mucho al costo del hardware. Un ejemplo según Soluciones Avanzadas (julio 1994):

“En 1955 el costo del primero era menor al 10% del costo del segundo; 20 años después, el costo del software era ya tres veces mayor que el del hardware y para 1985 la diferencia se incrementó nueve veces”. (Figura 1.3)

Figura 1.3 Costo Anual del Software.



Fuente: Soluciones Avanzadas, Julio 1994.

Causantes de pérdidas cuantiosas han sido los *problemas de la calidad del software*. En la revista de Soluciones Avanzadas (julio, 1994) se mencionan algunos casos tales como:

“En junio de 1962, el Mariner I se salió de curso y tuvo que ser destruido: el problema, que costó 18.5 millones de dólares, se debió a un error en uno de los programas que guiaban la nave. A pesar de que el proyecto Apollo reunió a los mejores programadores de Estados Unidos, un error borró parte de la memoria de la computadora instalada en el Apollo 8 y se detectaron 18 errores de programación en los diez días que duró el vuelo del Apollo 14. El 15 de enero de 1990, el sistema de larga distancia de la AT&T sufrió una falla catastrófica que dejó paralizada la mayor parte de la red telefónica nacional de Estados Unidos durante nueve horas. Mas tarde se encontró que la falla fue debida al software de encaminamiento.”

Las pérdidas que ha ocasionado el software no sólo han sido referentes a pérdidas materiales; también han ocasionado pérdidas humanas, según Soluciones Avanzadas (julio, 1994):

“En 1980, un paciente murió a consecuencia de un tratamiento para artritis con una máquina basada en microondas, cuyo software reprogramó inadvertidamente su marcapaso; en 1985 una falla de software en la máquina de rayos X Therac 25 dejó un paciente muerto y varios más paráliticos”.

Los *problemas de productividad* se encierran en la falta de herramientas y metodología que se dieron en los primeros días de la era de la informática. Estos problemas encuentran su solución en el uso de herramientas CASE y la implementación de una metodología de desarrollo. Sin embargo, no se ha podido solucionar enteramente el problema de la productividad, ya que el uso de nuevas herramientas y nuevas metodologías implican el abandono de ciertos paradigmas para adoptar otros nuevos.

1.2.2 Causas

Según Pressman (1993).

“No debe de extrañarnos, que el origen de muchos problemas se deban a la misma complejidad del software y por los errores de las personas encargadas del desarrollo del mismo”.

En cuanto al software, la situación es mucho más compleja, porque no se daña con el paso del tiempo; si el software tiene fallas, seguramente se introdujeron a él en alguna de las diferentes etapas del desarrollo, y no se lograrán detectar durante el proceso de pruebas; estos fallos son corregidos durante el mantenimiento, pero a veces significa la corrección o modificación del diseño.

Hasta ahora las personas que han estado como responsables frente a los proyectos de desarrollo de software han sido ejecutivos de nivel medio y alto, es decir, personas que usualmente no tienen los conocimientos necesarios de software. Se hace hincapié en que estos conocimientos necesarios de software abarcan muchos aspectos, tales como técnicas para medir el desarrollo del proyecto, técnicas que deben aplicarse a métodos efectivos de control, así como el llegar a tener un amplio conocimiento de una tecnología que cambia día a día, etc. Todo esto es para lograr una comunicación eficiente entre clientes, realizadores del software, equipo de soporte, y otros. Para que se logren comprender las características que debe reunir el software, se debe enfatizar, que si esta comunicación no es falaz, los problemas se verán multiplicados.

Además de las causas mencionadas en los párrafos anteriores, R. Scharch (1993) señala cuatro aspectos que hacen inherentemente difícil el desarrollo de software: complejidad, conformidad, adaptabilidad, invisibilidad, los cuales se explican en las siguientes secciones.

1.2.2.1 Complejidad

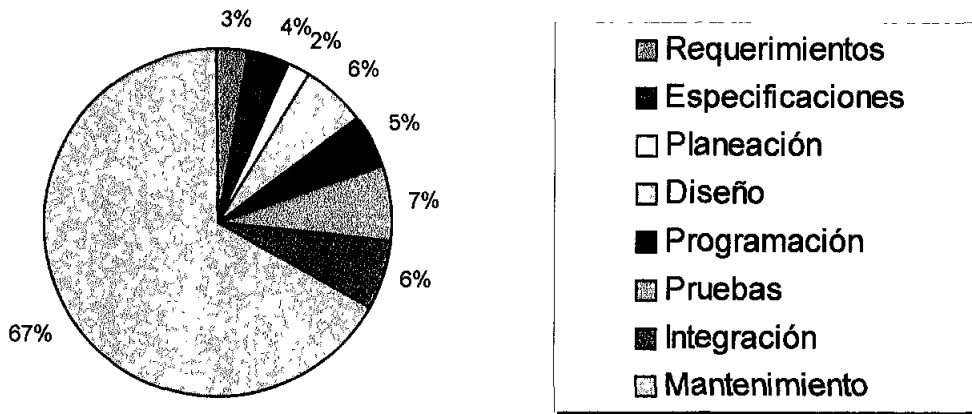
La complejidad es una propiedad esencial del software no accidental. Una consecuencia de esta complejidad es que el producto (sistema o programa) llega a ser difícil de entender. Por ejemplo, los estados de un procedimiento dependerán de los estados de sus parámetros y de las variables globales (variables que pueden ser accesadas por más de un procedimiento), los cuales afectarán el estado del producto.

Esto quiere decir que es difícil que una persona pueda entender a un producto grande como un todo. Si el producto en sí no se entiende como un todo, entonces los componentes individuales no pueden ser entendidos. Esto lleva a una comunicación imperfecta entre los miembros de un equipo, el cual resulta en el retraso en cuanto a tiempo y el aumento en los costos que caracterizan el desarrollo de software a gran escala.

Una consecuencia más de la complejidad es que dificulta el proceso de mantenimiento del software. En la Figura 1.4 se puede observar que casi dos terceras partes de la inversión en software se destina al mantenimiento. A menos que la persona o personas entiendan completamente el producto, siempre existirá el riesgo que cualquier acción correctiva o de mejoramiento tenga efectos dañinos de tal forma que

se requiera un mantenimiento posterior para reparar el daño ocasionado por el mantenimiento original. La poca documentación, la falta de documentación, o aún peor, la documentación incorrecta es una causa mayor por la que se pueda dar un mantenimiento incorrecto.

Figura 1.4 Costo del mantenimiento en el proceso de desarrollo.



Fuente: R. Scharch, 1993.

1.2.2.2 Conformidad

La conformidad se refiere a que el software debe adaptarse al problema al cual se enfrenta. Para aclarar más este concepto véase el ejemplo que menciona R. Scharch (1993):

“Se necesita automatizar una refinería en la cual todo se controla manualmente. En lugar de utilizar una serie de botones y elevadores, una computadora enviara los controles necesarios a los distintos componentes de la planta. Aunque la planta trabaja perfectamente, la gerencia piensa que un sistema de control por computadora incrementaría la producción. La tarea del equipo de desarrollo es desarrollar un software que esté conforme a la planta, y no que la planta esté conforme al software.”

Este tipo de conformidad a que se refiere R. Scharch, es donde el software tiene innecesariamente un grado de complejidad al momento de adaptarse a un sistema existente.

1.2.2.3 Adaptabilidad

Para explicar este aspecto, se observa el siguiente ejemplo que menciona R. Scharch (1994):

“Se puede considerar ilógico pedirle a un ingeniero civil que traslade un puente a 200 millas desde donde está, o que lo rote 90°, pero es perfectamente aceptable pedirle a un desarrollador de software que reescriba la mitad de un sistema operativo en un período de cinco años. El ingeniero civil sabe que rediseñar la mitad de un puente es caro y peligroso; es más barato y seguro construir un nuevo puente. Con el software puede suceder lo mismo, al punto de tener que reescribirlo, porque a veces los cambios son muy grandes”.

Existen cuatro razones por las que el software tiene que adaptarse:

1. El software es un modelo de la realidad, y como la realidad cambia, el software tiene que adaptarse o morir.
2. Si el software es bastante útil entonces hay presión por parte de los usuarios para extender su funcionalidad mas allá del diseño original.
3. Una de las características mas fuertes del software es que es mucho más fácil de cambiar que el hardware.
4. El ciclo de vida del software exitoso es mayor que el hardware en el que fue escrito. Como ciclo de vida del software entendemos la serie de etapas por la que las un producto de software atraviesa, desde su concepción hasta su reemplazo o retiro.

1.2.2.4 Invisibilidad

Este aspecto del software se refiere a lo difícil que es representar de una manera visual un producto, o una parte de él. Por ejemplo, un ingeniero civil puede auxiliarse de maquetas, para darle al cliente una idea de cómo quedará terminada la casa; durante la construcción de la misma, el ingeniero civil utiliza diferentes planos que lo ayudan a ir controlando el avance de la casa. Para un ingeniero en sistemas, esta tarea es bastante complicada, ya que no se puede tener este tipo de herramientas, aunque es cierto que puede utilizar diagramas para representar el flujo de datos, flujo de control, interrelaciones entre módulos, etc., es muy difícil lograr tener una visión global del producto.

1.3 Modelos para el desarrollo de software

Cuando surge la necesidad de un producto de software, se inicia con la idea de cómo se desea. Una vez que la necesidad de un producto de software se ha establecido, el producto pasa a través de ciertas fases de desarrollo. Las fases más comunes son la especificación de requerimientos, el análisis, el diseño, la implementación y el mantenimiento. A esta serie de pasos por los que pasa un producto se llama modelo de desarrollo (o modelo de ciclo de vida).

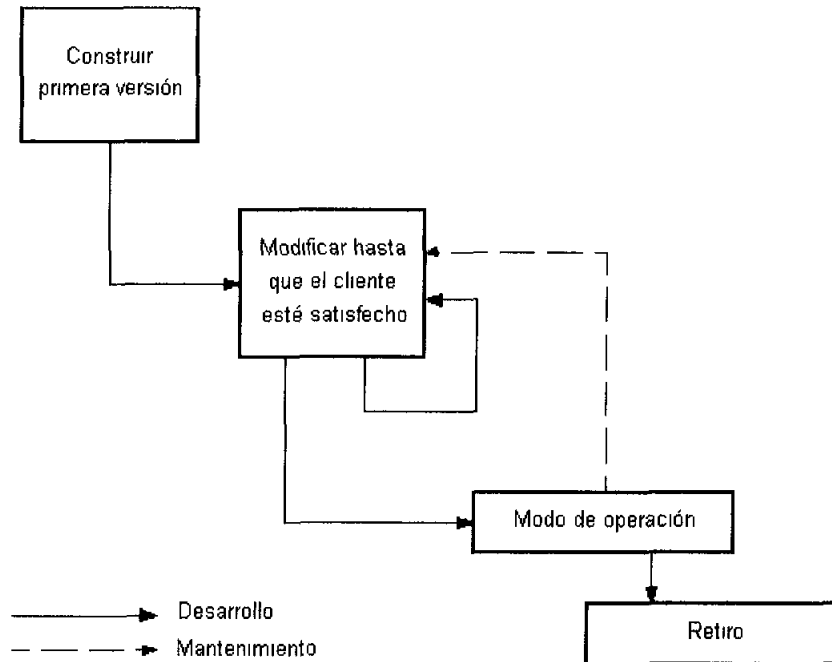
El ciclo de vida de cada producto es diferente. Algunos productos tal vez se tomen años en la etapa conceptual debido a que la plataforma de hardware que necesitan no es lo suficientemente poderosa. Otros productos serán rápidamente diseñados e implementados pero la fase de mantenimiento se prolongará mucho debido a los cambios constantes en las necesidades de los usuarios.

A continuación se explican algunos de los modelos de desarrollo más comunes (entre los que se encuentran el modelo de cascada y el de prototipo rápido), además de dar a conocer sus ventajas y desventajas.

1.3.1 Modelo Construir y Arreglar.

En este modelo, no existen especificaciones ni diseño, sino que se construye el producto y se reelabora cada vez que es necesario, hasta que el usuario queda satisfecho con el resultado. Este modelo se muestra en la Figura 1.5.

Figura 1.5 Modelo construir y arreglar.



Fuente: R. Scharch, 1993.

Según R. Scharch (1993):

“Aunque este modelo es apropiado para productos de 100 ó 200 líneas de código de largo, es totalmente inapropiado para productos de tamaño razonable.”

De este modelo se resume lo siguiente:

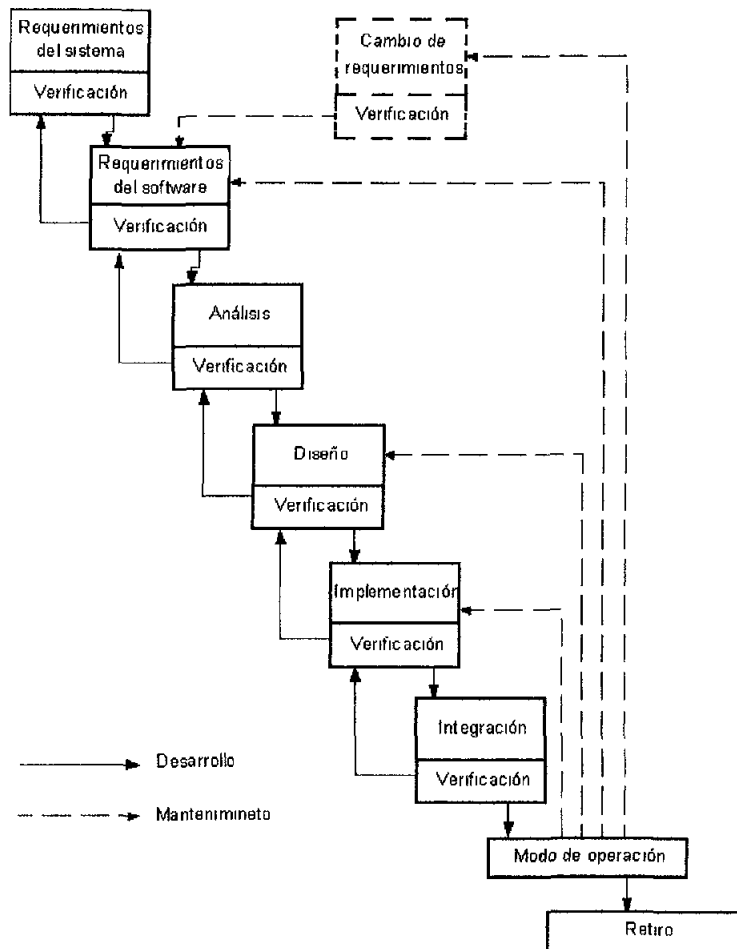
- Las Figuras 1.3 y 1.4 (costo del software) indican que el costo en que se incurre en el cambio de un producto, es relativamente menor si se hace en la fase de requerimientos, especificación o de diseño. Sin embargo, si estos cambios se realizan cuando se ha codificado o peor aún, cuando el producto está operando, los costos se elevan mucho.
- El mantenimiento del producto es extremadamente difícil si no hay documentos de diseño o especificaciones.

Para evitar caer en este tipo de desarrollo, es necesario escoger un modelo entre las partes involucradas (equipo de desarrollo y cliente), para que una vez que se acuerde un modelo, pueda iniciarse formalmente el desarrollo del producto

1.3.2 Modelo de Cascada.

El modelo de cascada fue ampliamente usado para el desarrollo de software al inicio de la década de los 80's. Este fue puesto en marcha por Royce (Royce, 1970). En la Figura 1.6 se muestra este modelo.

Figura 1.6 Modelo de Cascada.



Fuente: R. Scharch, 1993.

A continuación se presentan las fases que tiene este modelo:

- Se determinan los *requerimientos*, (lo que se desea) los cuales son verificados por el cliente y por el grupo de aseguramiento de calidad del software (SQA, Software Quality Assurance).

- Se determinan las *especificaciones* del producto, esto es, se produce un documento en el que se especifica lo que el producto debe hacer. Esta fase termina cuando el grupo SQA revisa el documento y el cliente lo aprueba.
- Una vez que el cliente firma el documento para su aceptación, se inicia con la fase de *análisis*. En esta fase se estudia la situación de la organización para observar cómo trabaja y decidir si es necesario realizar una mejora.
- Cuando el cliente aprueba el tiempo y costo estimados del proyecto, da inicio la fase de *diseño*. Esta fase produce documentos de diseño que describen cómo debe trabajar el producto.
- Terminado el diseño, se procede a la *implementación* (programación). Se integran y se prueban los diferentes módulos del producto.
- La fase de *integración* llega cuando todos los módulos han sido verificados. Se pone en marcha el producto y se realizan pruebas.

Durante el análisis y diseño, se pueden encontrar errores en las especificaciones (características que debe tener el producto). Estos errores pueden ser, entre muchos otros, que las especificaciones estén incompletas (se omitieron ciertas características que debe tener el producto), contradictorias (dos o más especificaciones definen el producto en una forma incompatible), o ambiguas (que las especificaciones se puedan interpretar en más de una forma). Estos errores hacen necesario revisar las especificaciones antes de seguir con el proceso de desarrollo.

Observando la Figura 1.6, se presenta una característica muy importante de este modelo: la retroalimentación. Los desarrolladores pueden llegar a revisar nuevamente las especificaciones e incluso los requerimientos en base a la información obtenida en el proceso de diseño.

Si es necesario uno o más cambios en las especificaciones, se pide la aprobación del cliente. Cuando se aprueban estos cambios, se integran a los documentos de especificación y planeación y se prosigue. Una vez que no se detectan más errores, y por lo tanto, no se requieren más cambios en los documentos anteriores, se procede a la implementación.

Los módulos se implementan, se documentan y se integran. En la práctica las fases de implementación e integración, se llevan a cabo en paralelo, ya que los módulos se van integrando cada vez que son implementados y probados.

Gracias a la característica de retroalimentación de este modelo, es posible que durante la integración sea necesario regresarse y hacer modificaciones al código, modificaciones precedidas tal vez, por cambios en las especificaciones y el diseño.

Cabe destacar que en este modelo, ninguna fase está completa mientras los documentos no lo estén, además de que deben ser aprobados por el grupo de SQA.

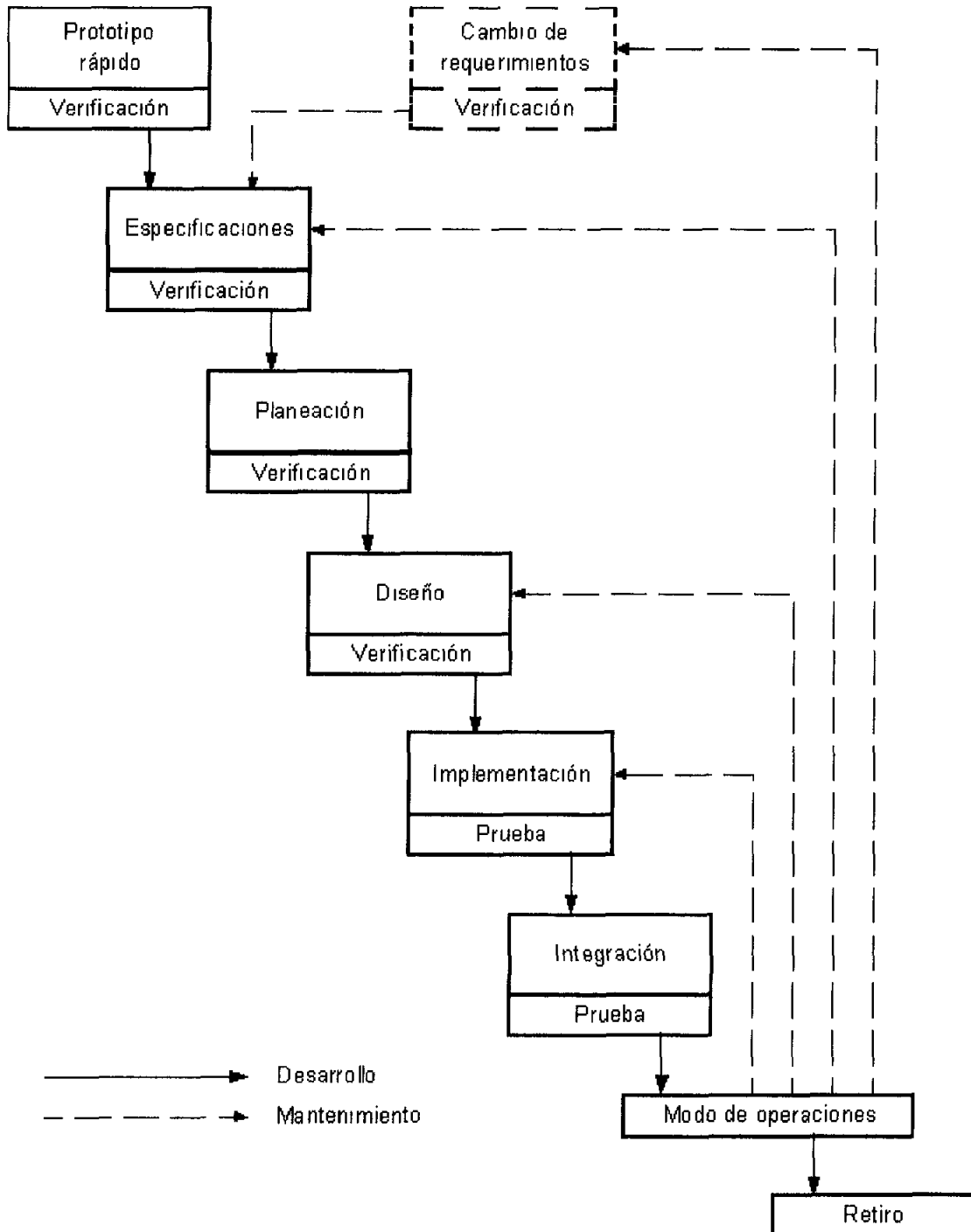
Por ejemplo, si son necesarios cambios en las especificaciones, esta fase no se puede considerar completa hasta que los documentos no reflejen estos cambios y no sean revisados y aprobados por el grupo de SQA.

Una vez que el equipo de desarrollo establece que el producto está listo, es entregado al cliente junto con la documentación especificada en el contrato (como pueden ser manuales de usuario, manual técnico, etc.), y se pone en operación. Cuando es necesario hacer algún arreglo o extender la funcionalidad de una parte del producto, esto se hace en la fase mantenimiento, la cual no sólo hace cambios en la implementación, sino en las especificaciones y el diseño.

1.3.3 Modelo de Prototipo Rápido.

La característica fundamental de este modelo es la construcción (como su nombre lo indica) de un prototipo. Este prototipo es un subconjunto funcional del producto final. Por ejemplo, si el producto es un sistema de facturación, el prototipo consistiría en la captura de datos y la impresión de reportes, pero sin actualizar alguna base de datos y sin la validación de los datos de entrada. Este modelo se muestra en la Figura 1.7.

Figura 1.7 Modelo prototipo rápido.



Fuente: R. Scharch, 1993.

La construcción del prototipo facilita al equipo de desarrollo la creación del producto final. Según Pressman (1992), el prototipo puede tomar una de las tres formas siguientes:

“1) Un prototipo en papel o un modelo basado en PC que describa la interacción hombre-máquina, de forma que facilite al cliente la comprensión de cómo se producirá dicha interacción, 2) un prototipo que implemente algunos subconjuntos de la función requerida del producto deseado, o 3) un programa existente que ejecute parte o toda la función deseada, pero que tenga otras características que deban ser mejoradas con el nuevo trabajo de desarrollo”.

Para la construcción del prototipo se comienza con la recolección de los requisitos. Luego se hace un “diseño rápido”, que se enfoca principalmente a la representación visual del producto al cliente.

Una vez construido el prototipo, se sigue con las fases del modelo de cascada. Se genera el documento de las especificaciones. Las especificaciones resultantes se presumen correctas, ya que el prototipo ha sido validado a través de la interacción con el cliente, esto hace pensar en que la retroalimentación que se da en el modelo de cascada, es menos necesario en este modelo.

Una característica fuerte de este modelo, según S. Scharch (1993), es que:

“...el desarrollo del producto es esencialmente lineal, desde el prototipo hasta el producto liberado”.

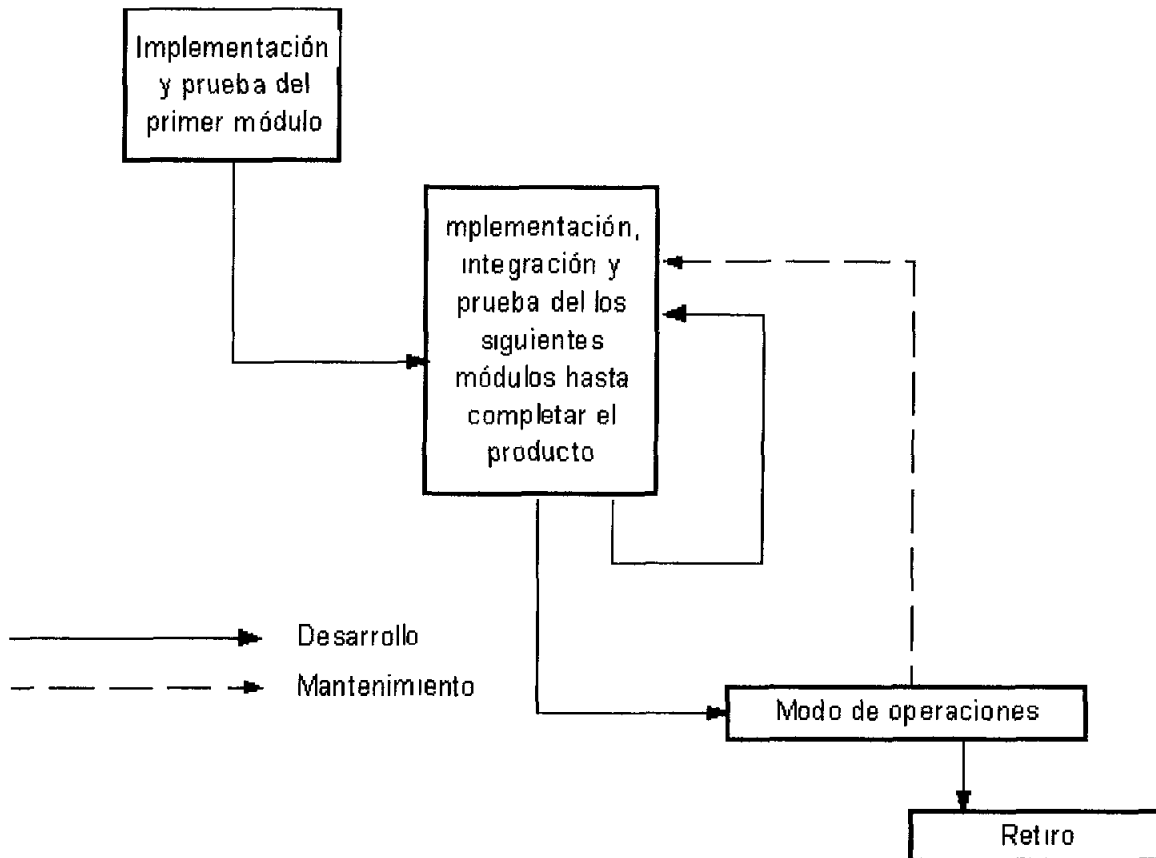
Se debe hacer notar que la construcción del prototipo debe ser lo más rápido posible, para acelerar el proceso de desarrollo, porque el propósito del prototipo es el de conocer las necesidades reales del cliente.

1.3.4 Modelo Incremental

Este modelo se basa en la construcción de piezas o partes que se van integrando paso a paso. Debe entenderse una pieza o parte del producto como un módulo. El producto se diseña, se implementa, se integra y se prueba como una serie de módulos que se va incrementando. Estos módulos interactúan entre sí para proveer una función específica. Este modelo se presenta en la Figura 1.8.

En cada etapa del modelo incremental, se codifica un nuevo módulo y se incorpora a la estructura que se tiene para probarlo como un todo. Este proceso termina cuando el producto cumple con todos los requerimientos.

Figura 1.8 Modelo Incremental.



Fuente: R. Scharch, 1993.

Para implementar con éxito este modelo, se debe tener en cuenta lo siguiente: si el producto es dividido en pocos módulos, el modelo degenera en un modelo Construir y Arreglar (Figura 1.5), y si el producto es dividido en muchos módulos, se gasta mucho tiempo en cada etapa en la prueba de integración de sólo una pequeña parte funcional.

Los modelos de cascada y de prototipo rápido, tienen como meta dar al cliente un producto completo y totalmente operativo. Por decirlo de otra forma, se le libera al cliente un producto que satisface cada uno de sus requerimientos.

En este modelo se encuentran algunas ventajas como las siguientes:

- El modelo incremental libera al cliente módulos totalmente operativos que satisfacen un subconjunto de sus requerimientos, permitiéndole trabajar completamente con el o los módulos sin tener que esperar el producto final.
- Permite que el cambio en la organización se dé gradualmente. Con la introducción gradual del producto se da tiempo a que la organización se ajuste al producto.
- El desarrollo de un producto de software requiere de una inversión. Una ventaja de este modelo, según R. Scharch (1993) es que:

“...no es necesario que el producto se complete para obtener una recuperación de la inversión, además de que el desarrollo puede interrumpirse en cualquier momento”.

En contraste con las ventajas, se observan a continuación algunas de sus desventajas:

- Cada módulo que se codifique debe incorporarse sin deshacer lo que se lleva hasta ese momento.
- El modelo incremental no distingue entre el desarrollo y el mejoramiento (mantenimiento) de un producto. Cada mejora que se le hace al producto es la adición de un nuevo módulo.
- Este modelo puede degenerar fácilmente en un modelo construir y arreglar, por las razones explicadas anteriormente.

1.3.5 Modelo de Espiral

El modelo de espiral incorpora muchos aspectos de los modelos anteriores. Cada vez que se inicia un proyecto de software, hay ciertos riesgos inherentes a él, riesgos que hay desde las primeras fases. Por ejemplo, el fabricante del hardware en el cual correrá el producto puede, en un momento dado, dejar de proveerlo, cambiando de forma dramática el proyecto. Otro ejemplo que se aprecia mencionar sería el de la empresa que planea sacar al mercado software especializado en el manejo de proyectos, pero antes de que esto suceda, otro fabricante libera un producto que hace todo lo que el software planeado puede hacer, pero a un menor precio. Estos y otros tipos de riesgos son los que se planean solucionar con este modelo.

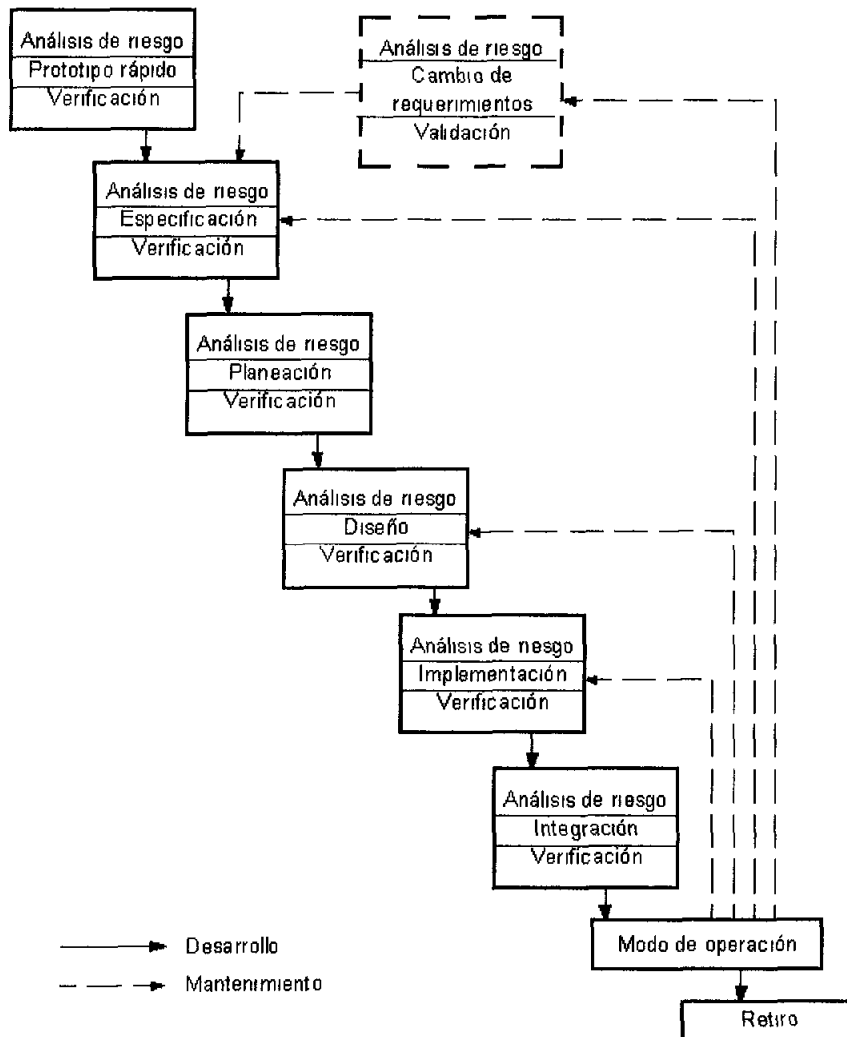
Según Scharch (1993):

“Una forma de minimizar ciertos tipos de riesgos es la construcción de un prototipo”.

La construcción de un prototipo, como se mencionó en el modelo de prototipo rápido, ayuda a los desarrolladores a asegurar que el producto liberado cumplirá con todos los requerimientos del cliente.

Como se observa en la Figura 1.9, este modelo se puede observar en su manera más simple como un modelo de cascada, en el que a cada fase le precede un análisis de riesgo. Esto lleva a tratar de controlar y resolver los riesgos que puedan darse en cada fase, y en caso de que ese riesgo no pueda eliminarse, se tiene la posibilidad de terminar el proyecto.

Figura 1.9 Versión sencilla del Modelo de Espiral.

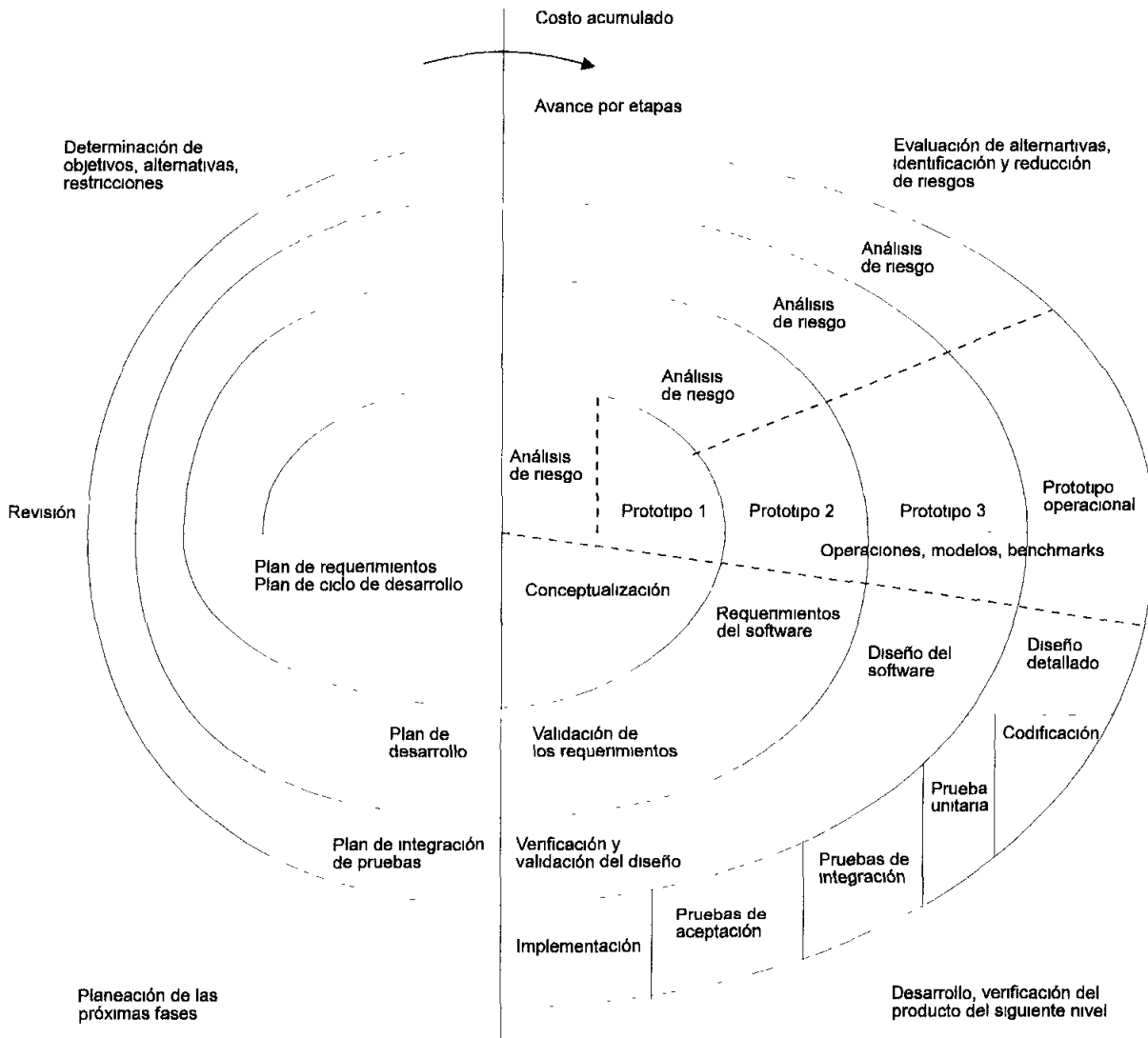


Fuente: R. Scharch, 1993.

El modelo de espiral se divide en cuatro actividades principales, representados por los cuatro cuadrantes de la Figura 1.10:

1. Determinación de objetivos, alternativas, restricciones.
2. Evaluación de alternativas, identificación y resolución de riesgos.
3. Desarrollo. Verificación del producto del siguiente nivel.
4. Planeación de las próximas fases conforme a la evaluación del cliente.

Figura 1.10 Modelo de Espiral.



Fuente: R. Scharch, 1993.

De él se señalan las siguientes características:

- La dimensión radial representa el costo acumulado a la fecha.
- La dimensión angular representa el progreso por la espiral.
- Cada ciclo de la espiral corresponde a una fase.
- En cada iteración alrededor de la espiral (desde el centro hacia el exterior), se construyen sucesivas versiones del producto, cada vez más completas.

En la primera iteración, se definen los objetivos, las alternativas y las restricciones (segundo cuadrante) para pasar al análisis e identificación de riesgos (primer cuadrante). Si el análisis determina que hay riesgos, entonces se procede a la construcción de un prototipo.

El cliente evalúa el avance y sugiere modificaciones. Esto da pauta a la siguiente iteración del modelo. Cabe hacer notar que en cada iteración del modelo, la finalización del análisis del riesgo lleva a una decisión de “seguir o no seguir”.

La construcción de varias versiones del producto (localizado en el cuarto cuadrante), se puede llevar a cabo mediante la utilización de algún otro modelo, como el modelo de cascada o de prototipo rápido. Cada vez que el producto se dirige a su terminación, el número de actividades del desarrollo (cuarto cuadrante) aumenta al alejarse del centro de la espiral.

Una de las grandes ventajas de este modelo, es que tiene un enfoque más realista para el desarrollo de software y de sistemas a gran escala [Pressman, 1993], además de que utiliza la creación de prototipos como medio para reducir los riesgos. Este modelo demanda una consideración directa de riesgos técnicos en todas las etapas del proyecto, llegando incluso a reducir los riesgos antes de que se conviertan en problemáticos.

1.4 Proceso de desarrollo de software

El proceso de desarrollo de software se puede subdividir en varias etapas, aunque existen varios enfoques, dependiendo del autor. Aquí se hace mención de la división que hace Senn (Senn, 1992), en lo que él llama “El ciclo de vida clásico del desarrollo de sistemas”. Este ciclo queda dividido en seis partes, que se describen a continuación.

1.4.1 Investigación preliminar

Cuando una persona decide que necesita ayuda de un software determinado, es cuando comienza el ciclo de vida clásico del desarrollo de sistemas, dando inicio la

primera actividad de este. Esta actividad ha sido dividida en tres partes: aclaración de la solicitud, estudio de factibilidad y aprobación de la solicitud.

1. Aclaración de la solicitud: muchas de las solicitudes que pueden recibirse no estarán formuladas de manera clara, por lo que será necesario determinar con precisión lo que el solicitante desea.
2. Estudio de Factibilidad: un documento importante que arroja la investigación preliminar es el estudio de factibilidad, el cual indicará si es factible o no el desarrollo del sistema solicitado. Este abarca tres aspectos; el primero de ellos es el aspecto técnico y se refiere al tipo de equipo que se necesitará para el software que se va a desarrollar; el segundo de ellos es el aspecto económico y se refiere a que si las utilidades que obtenga la empresa por software justifican los costos; el tercero de ellos es el aspecto operacional y se refiere a qué tan utilizado será el nuevo software, y a los problemas que encierre la resistencia al cambio.
3. Aprobación de la solicitud: no todos los proyectos que llegan a ser solicitados llegan a ser desarrollados, ya que en la parte del estudio de factibilidad pueden ser rechazados. Una vez que se ha comprobado que el software deseado es factible, se podrá pasar a las siguientes etapas.

1.4.2 Determinación de los requerimientos del sistema

Es muy importante que se comprendan todas las facetas importantes de la empresa, por esta razón es necesario que se estudien todos los procesos de una empresa junto con los empleados y los administradores, para poder determinar puntos claves como, qué es lo que hace determinado proceso, como es que se realiza dicho proceso, la frecuencia con que se realizan los procesos, determinación del volumen de transacciones, el grado de eficiencia con el que se realizan las tareas.

Esta información se obtiene por medio de entrevistas con las personas involucradas con los procesos a estudiar, aunque en algunos casos, esto no será posible, por lo que el recurrir a encuestas puede ser bastante útil, principalmente cuando se trata de un número grande de empleados a los que se quiere entrevistar. Es importante el no hacer a un lado otras fuentes de información como podrían ser manuales y reportes, la observación de las personas encargadas del desarrollo del software de las condiciones reales de las actividades de trabajo.

Conforme se va reuniendo toda esta información y se van analizando los datos sobre requerimientos con la finalidad de detectar las características que debe tener nuestro software, también se van viendo características operacionales como controles de procesamiento, tiempos de respuesta y métodos de entrada y salida.

1.4.3 Diseño del sistema

En esta etapa se tiene que establecer de qué manera el software cumplirá con los requerimientos identificados en las etapas anteriores. Se comienza haciendo una identificación de los reportes y demás salidas que debe producir el software. Una vez hecho esto es necesario especificar qué datos son necesarios para producir estos reportes, habiendo datos que tendrán que ser calculados, algunos de estos tendrán que ser guardados, otros tendrán que ser mandados a impresoras, tendrá que ser especificado el manejo que se le dará a la información.

También se llevan a cabo el diseño de las pantallas tanto de entrada como de salida, la distribución de los datos en los reportes que serán impresos y en general se especifica la apariencia del software.

En esta etapa se hacen las especificaciones de software; éstas deben ser lo más completas y claras, de manera que puedan ser usadas como guía en la etapa de programación.

1.4.4 Desarrollo de software

El software puede ser comprado a terceros; este software generalmente es de propósito general, o puede desarrollarse a la medida del solicitante. Si se opta por esta opción, será necesario transcribir todas esas especificaciones que arrojó el diseño a un lenguaje. En esta fase se genera propiamente el código del software; además en esta etapa, se desarrollará la documentación indispensable del software, la cual incluirá los manuales de procedimientos, es decir, la manera en la que el usuario operará el software.

1.4.5 Prueba de los sistemas

En esta etapa, el software será empleado de manera experimental. Se buscará el detectar el mayor número de fallas posibles, para su corrección. Antes de ser entregado al usuario, el software debe funcionar de acuerdo con las especificaciones y en la forma en que los usuarios esperan que lo haga.

El software será alimentado con conjuntos de datos que el software procesará, para ser examinados posteriormente, y confirmar si realmente está funcionando bien. También se puede pedir esto a algunos usuarios que utilicen el sistema. De esta manera se pueden encontrar fallas, además se puede ver qué tan a gusto se encuentra el usuario con el software nuevo, tomar notas sobre lo que no le gusta o lo que le parece mal, para evaluar estas observaciones y decidir si es necesario hacer algo al respecto o no.

1.4.6 Implantación y evaluación

La implantación es el proceso de verificar e instalar nuevo equipo, entrenar a los usuarios, instalar la aplicación y construir todos los archivos de datos necesarios para utilizarla. Posteriormente se deberá elegir un método para hacer la transición del sistema viejo al sistema nuevo. Se tienen varias opciones, y una de ellas consiste en que se comience a trabajar en un área determinada con el nuevo software. Algunas otras veces se trabaja simultáneamente sobre ambos, lo cual permite hacer comparaciones entre ellos. Otra opción consiste en fijar una fecha para comenzar a utilizar el software nuevo. Es importante recalcar que no importa qué método se utilice; lo importante es que el uso inicial del sistema se encuentre libre de problemas.

La aplicación una vez instalada se podrá mantener por mucho tiempo, tiempo durante el cual se darán cambios y modificaciones al software, a los archivos y a los procesos, para satisfacer las nuevas necesidades de los usuarios. La evaluación de un sistema se lleva a cabo para identificar puntos débiles y fuertes. La evaluación abarca diferentes aspectos, tal como lo menciona Senn (1992), aspectos como:

- 1. Evaluación operacional: valoración de la forma en que funciona el sistema, incluyendo su facilidad de uso, tiempo de respuesta, lo adecuado de los formatos de información, confiabilidad global y nivel de utilización.*
- 2. Impacto organizacional: identificación y medición de los beneficios para la organización en áreas tales como finanzas (costos, ingresos y ganancias), eficiencia operacional e impacto competitivo. También se incluye el impacto sobre el flujo de información interno y externo.*
- 3. Opinión de los administradores: evaluación de las actitudes de directivos y administradores dentro de la organización así como de los usuarios finales.*
- 4. Desempeño del desarrollo: la evaluación del proceso de desarrollo de acuerdo con criterios tales como tiempo y esfuerzo de desarrollo, concuerdan con presupuestos y estándares, y otros criterios de administración de proyectos. También incluye la valorización de los métodos y herramientas utilizados en el desarrollo.*

1.4.7 Fase de mantenimiento

El software va a requerir de ciertas correcciones una vez que se le entregue al cliente. Estas correcciones se pueden deber a que se encontraron errores en el producto, errores que no fueron detectados en las pruebas. Otra circunstancia por la que sería necesario corregir el software podría deberse a cambios tanto internos de la empresa o cambios externos a ellos, cambios que exijan modificaciones al software

1.5 Herramientas CASE

Las herramientas dentro de la Ingeniería de Software son muy importantes, de la misma manera que lo son dentro de otras áreas. Cualquier herramienta permite hacer las cosas mejor y mucho más rápido, así pues, las herramientas de Ingeniería de Software ayudan a incrementar la competencia profesional y reducen o incluso llegan a eliminar fuentes de error.

La Ingeniería de Software Asistida por Computadora (CASE, Computer Aided Software Engineering) está cambiando la manera de producir software; estas herramientas permiten automatizar los aspectos clave de todo el proceso de desarrollo, desde el principio hasta el final.

Según Senn (1992):

“...básicamente las herramientas CASE incluyen cinco componentes: herramientas para diagramación, un depósito de información, generadores de interfaces, generadores de código y herramientas de administración. Las herramientas de alto nivel reciben la mayor importancia, aunque ya están apareciendo generadores de código de bajo nivel”.

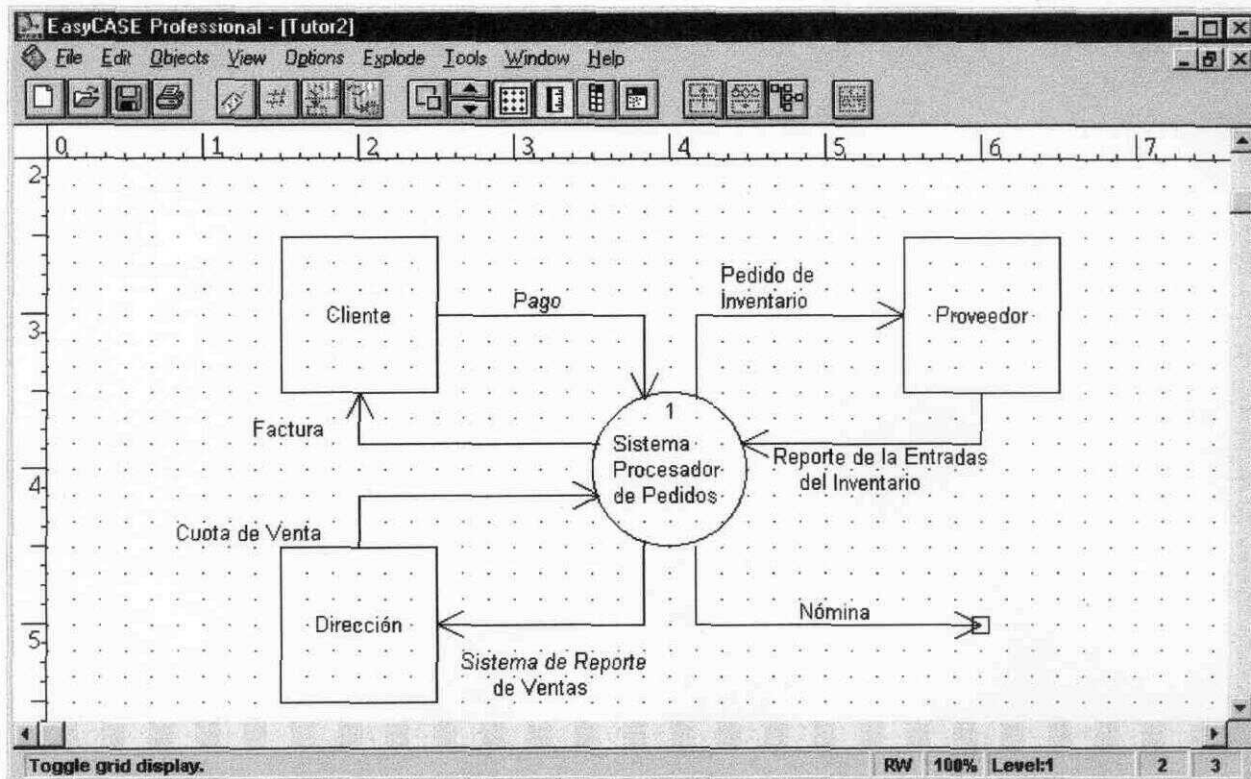
1.5.1 Características

Una herramienta CASE debe proporcionar una variedad de características para que pueda realmente ayudar en todas las áreas que involucra la creación de un producto. Tomando como base a Yourdon (1992), se establecen las siguientes características:

- Apoyo gráfico
- Manejo de gráficas para múltiples tipos de modelos.
- Características de revisión de errores.
- Comparación de modelos diferentes.
- Apoyo de Ingeniería de Software adicional.

Apoyo Gráfico: Es muy importante que se pueda crear un DFD o cualquier otro diagrama, de una manera rápida y sencilla; para esto influye mucho el apoyo gráfico que brinde la herramienta. En la Figura 1.11 se observa un DFD, que la herramienta deberá ser capaz de generar.

Figura 1.11. Diagrama de Flujo de Datos.



Fuente: EasyCASE.

Manejo de gráficas para múltiples tipos de modelos: Es importante que se puedan utilizar diferentes tipos de metodologías dentro de estas herramientas. Esta característica es deseable cuando se mezclan dos o más metodologías en el proceso de diseño.

Características de revisión de errores: Estas herramientas deben de ser capaces de asegurar que el modelo que ha sido creado éste completo e internamente consistente. La revisión del balance, de las reglas, etc. son algunos de los puntos a considerar.

Comparación de metodologías diferentes: Es muy importante que exista la posibilidad de verificar la consistencia de diversos tipos de metodologías de un sistema, ya sea que se quiera comparar modelos diferentes en una fase de un proyecto, o que se necesite comparar diferentes metodologías en diferentes fases de un proyecto.

Apoyo de Ingeniería de Software adicional: Generación automática de código para el mantenimiento de las tablas, a partir de diagramas de entidad-relación.

1.5.2 Lower CASE

El objetivo de estas herramientas es el de ayudar al analista a formular la lógica del programa, la descripción física de datos, los algoritmos de procesamiento de datos y la interacción con los dispositivos de entrada y salida.

Estas herramientas deben ser lo más automatizadas que se pueda, por tanto, una vez definidas las características antes mencionadas, este tipo de herramientas deben ser capaces de convertir estas especificaciones a código programable. Existen generadores de aplicaciones, de programas y de pantallas.

1.5.3 Upper CASE

El objetivo de estas herramientas se centra en la automatización de las primeras actividades del proceso de desarrollo de sistemas. Estas actividades son: el análisis de requerimientos y el diseño lógico, según menciona Senn (1992):

“Entre los muchos aspectos que se toman en cuenta al desarrollar herramientas para esta fase, se hayan las técnicas de soporte para ayudar al analista a preparar especificaciones formales que carezcan de ambigüedades, a validar las descripciones del sistema con el objeto de determinar su consistencia y complementación, y a seguir la evolución de los requerimientos de la aplicación en características que formen parte del sistema que finalmente será implantado”.

Estas herramientas ayudan en la creación de los diagramas de flujo de datos, diagramas estructurados, diagramas de estructura de datos, diagramas de relación entre las entidades, etc. El realizar alguno de los diagramas anteriores o todos es una tarea muy engorrosa si se hace a papel y lápiz, e intentar hacerlos en alguna herramienta de dibujo tampoco es una buena solución. Algunas de estas herramientas no sólo hacen mucho más fácil y amigable el diseño de estos diagramas, si no que además checan o validan algunas de las restricciones de estos diagramas. Además si es necesario hacer modificaciones, éstas pueden ser hechas rápidamente, y se pueden imprimir un

sin número de veces. El tratar de hacer una modificación sobre papel o en alguna herramienta de dibujo muchas veces complica el diagrama y dificulta su claridad, o peor aun, ocupa demasiado tiempo.

1.5.4 Ventajas

Dentro de las ventajas se observan las siguientes:

- Las facilidades que las herramientas brindan para realizar diagramas y cartas, así como guardar detalles en forma interna.
- Permiten la facilidad de cambiar y volver a dibujar los diagramas.
- Ayudan a guardar los componentes del sistema, tales como datos, flujo de datos, procesos, información que se guarda en el diccionario de datos, etc. Algunos pueden incluir controles y medidas de protección que preservan la exactitud y consistencia de las descripciones que en él se encuentran.
- Ayudan en la creación de prototipos para las interfaces de los usuarios.
- Auxilian también a la generación de código, aunque en este punto cabe mencionar que las más sofisticadas herramientas de este tipo solo producen el 75% del código total; el resto debe ser escrito por el programador.
- Ayudan a calendarizar las actividades de todo el proyecto, así como la asignación de recursos a estas actividades.

El ideal de estas herramientas, son aquellas con las que se puede generar cierta información, y posteriormente usarla como datos de entrada para otra herramienta, eso da como resultado una integración de herramientas CASE.

Capítulo 2. Encuesta

Dado el objetivo principal de este trabajo, que es el desarrollo de Ingeniería de Software para el Departamento de Revisión de Procesos, se presentan en este capítulo los resultados de la encuesta realizada a los alumnos de Sistemas Computacionales.

2.1 Objetivo

El objetivo de la encuesta fue determinar el nivel de conocimientos de ingeniería, de metodologías de desarrollo y de diversas herramientas y lenguajes.

2.2 Determinación de la Muestra

Se tomó como muestra representativa a los alumnos de 6º y 8º semestre de la escuela de Sistemas Computacionales. La muestra constaba de 96 alumnos de una población de 420. Se eligió a estos semestres por ser los que tenían más conocimientos sobre la Ingeniería de Software, ya que habían cursado más del 50% de los semestres que involucra la carrera, además de tener más experiencia realizando sistemas.

2.3 Elaboración de la Encuesta

Tomando en cuenta el objetivo de la encuesta, se decidió dividirla en cuatro secciones.

- Preguntas relacionadas con la Ingeniería de Software.
- Preguntas sobre la experiencia al elaborar sistemas.
- Preguntas relacionadas con definiciones.
- Preguntas sobre la utilización de software, metodologías y documentación.

Las preguntas que se plantearon no profundizaron en términos técnicos, debido a que el objetivo era conocer el nivel de conocimientos a un nivel general.

2.4 Análisis e Interpretación de la Encuesta

Una vez aplicada dicha encuesta se pasó al análisis de ella. La clasificación de los resultados se hizo en base a la división mencionada anteriormente, tal como se presenta a continuación.

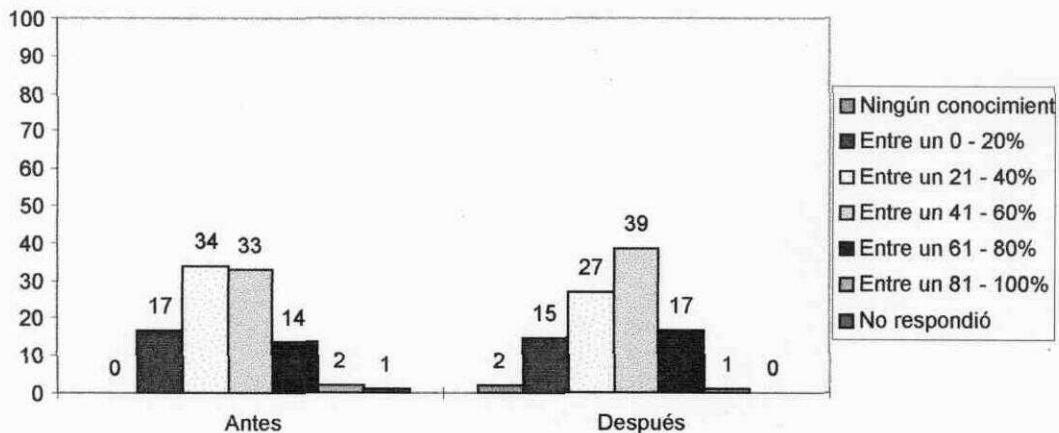
2.4.1 Preguntas Relacionadas con la Ingeniería de Software

- La primera pregunta de este bloque fue acerca de qué tanto creían saber sobre Ingeniería de Software (en una escala porcentual de 0% a 100%). Esta pregunta se realizó nuevamente al final de la encuesta con el objetivo de hacerles notar el grado de objetividad que tenían respecto a sus conocimientos.

Los resultados se vieron afectados tal como lo muestra la Figura 2.1. Es importante notar que en un principio las barras relacionadas con el rango de 21%-40% y 41%-60% casi son iguales. Después de la encuesta estas columnas se vieron afectadas, descendiendo el número de los que decían conocer entre 21%-40% y ascendiendo el número de los que creían conocer entre 41%-60%. Se puede observar que después de contestar la encuesta se observó más objetividad sobre el nivel de sus conocimientos.

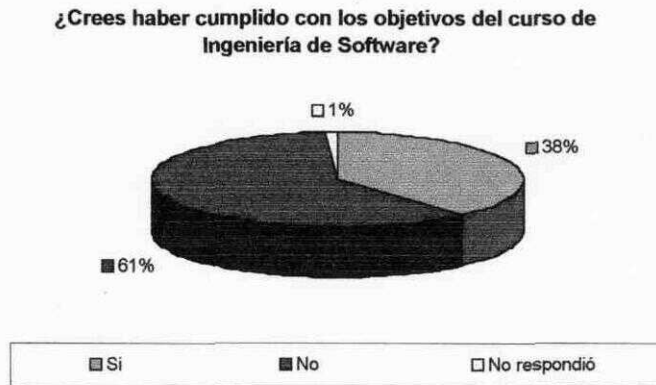
Figura 2.1

Porcentajes de conocimiento sobre Ingeniería de Software



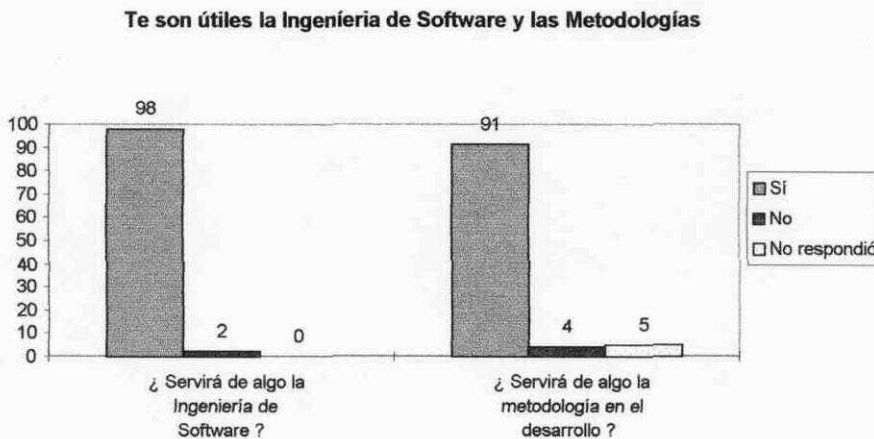
- Se les preguntó si creían haber cumplido con los objetivos del curso de Ingeniería de Software. El resultado que muestra la gráfica de la Figura 2.2 justifica los resultados de la pregunta anterior. Como puede verse, el 61% respondió que no. Los factores que influyeron sobre la respuesta están fuera del alcance de esta encuesta, por eso sólo se toma como justificación de la pregunta anterior.

Figura 2.2



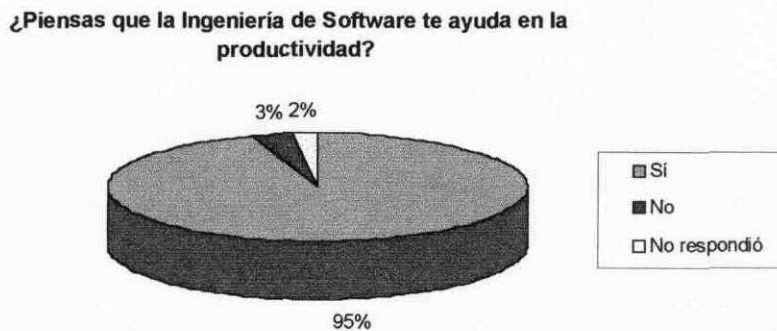
- Se preguntó si creían que la Ingeniería de Software y la Metodología tenían utilidad para el desarrollo de sistemas. Contrastando con su desempeño en el curso de Ingeniería de Software (pregunta anterior), casi el 100% respondió que sí, tal como se puede apreciar en la Figura 2.3.

Figura 2.3



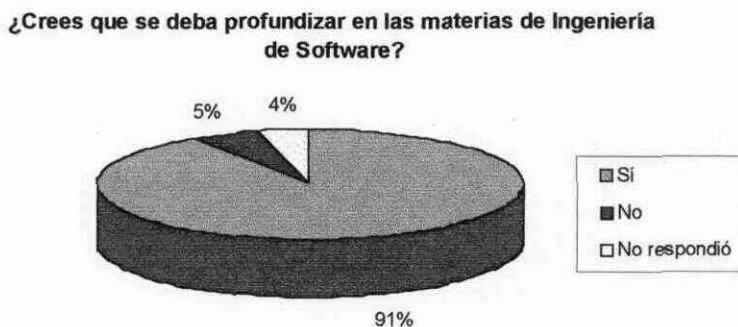
- Se preguntó si creían que la Ingeniería de Software ayuda a ser productivo. El 95% opinó que sí, mientras que sólo el 3% dijo que no. Aunque la mayoría respondió que sí, se podrá observar en las gráficas subsecuentes que no explotan esta herramienta para su provecho (ver Figura 2.4).

Figura 2.4



- Las respuestas a esta pregunta las podemos visualizar en la Figura 2.5. Al igual que en la gráfica anterior, los encuestados resaltan la importancia de la Ingeniería de Software y proponen que se profundice en estas materias.

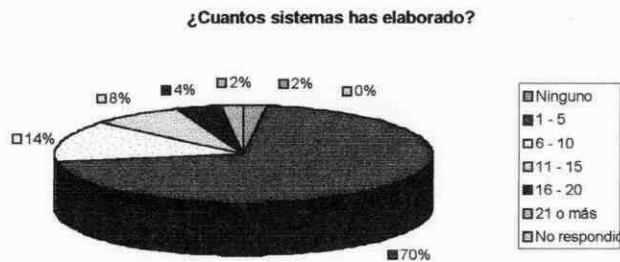
Figura 2.5



2.4.2 Preguntas Sobre la Experiencia al Elaborar Sistemas

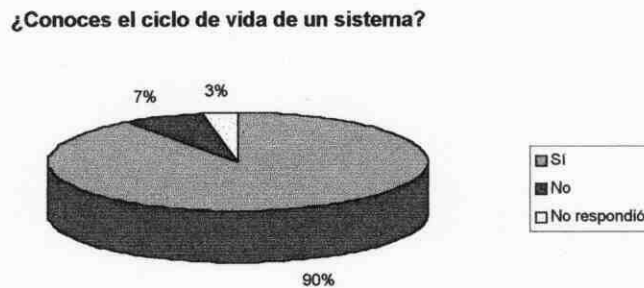
- Primero se preguntó cuántos sistemas habían realizado. En este caso, el sistema se consideraba como todos aquellos programas en los que las líneas de programación fueran ya considerables y que además haya incluido algún tipo de documentación. En la Figura 2.6 se puede apreciar que un 70% de los encuestados dijo sólo haber realizado entre uno y cinco sistemas; se resalta el hecho de que un 2% de los encuestados dijo no haber realizado ningún sistema. Para estar en sexto y octavo semestre esto es un requisito; de lo que se puede concluir que son muy pocos los sistemas que se han realizado.

Figura 2.6



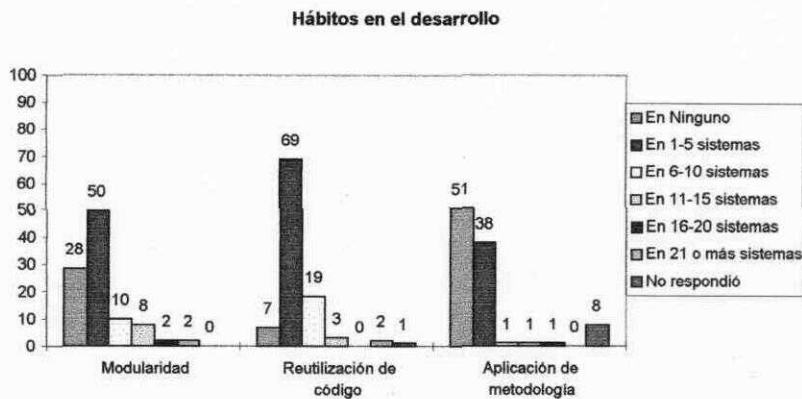
- Se preguntó si conocían el ciclo de vida de un sistema. Las respuestas a esta pregunta se pueden observar en la Figura 2.7. Se hace notar que el 90% de los encuestados respondió que sí conocía el ciclo de vida de un sistema.

Figura 2.7



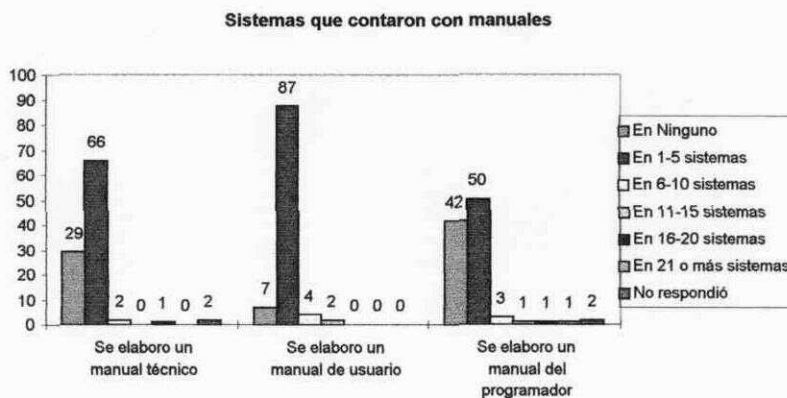
- Se preguntó también sobre hábitos de programación; en concreto si utilizaban la modularización y la reutilización de código en sus programas, además si aplicaban alguna metodología en estos. Con estos no sólo se logra una productividad alta, sino que además contribuyen a realizar productos con características de reusabilidad, de escalabilidad y de transportabilidad. En la Figura 2.8 se puede observar en cuántos de sus programas utilizaron estos conceptos. En ella se puede apreciar que la mayoría utiliza mucho la reutilización de código.

Figura 2.8



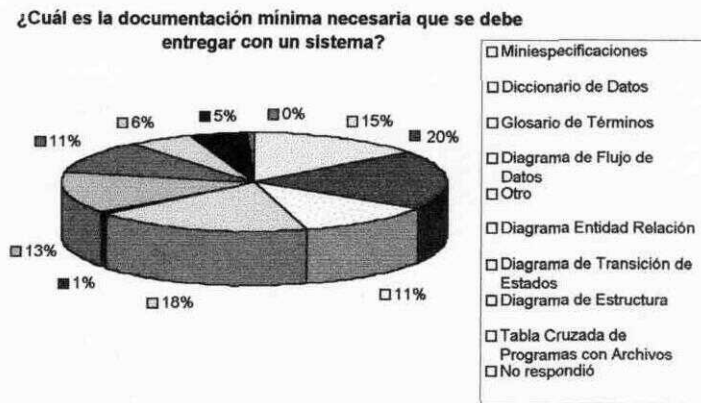
- Se preguntó también sobre el tipo de manuales que habían entregado con dichos sistemas. Ejemplos de estos manuales son: técnico, del usuario y del programador. En la Figura 2.9 se puede apreciar en cuántos de los sistemas que realizaron entregaron alguno de estos manuales. De lo anterior se puede concluir que han hecho más manuales de usuario que de programador y técnicos.

Figura 2.9



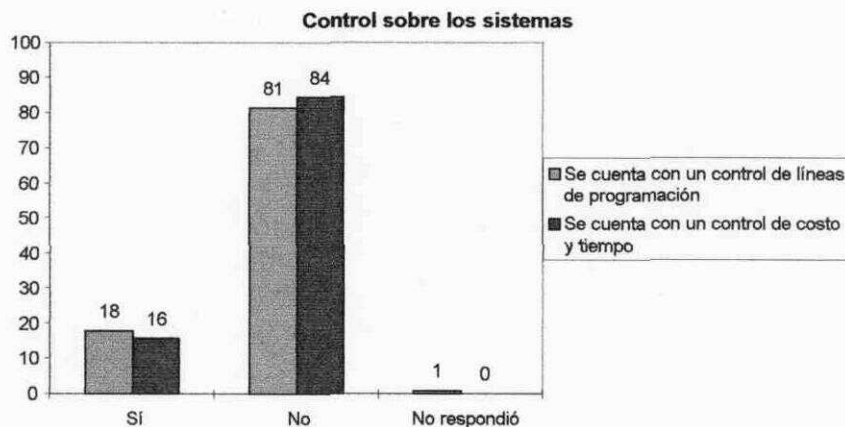
- Se preguntó cual es la documentación mínima necesaria que se debe entregar con un sistema. En la Figura 2.10 se visualizan los resultados de esta pregunta. Los cinco documentos que los encuestados consideraron más importantes son: el diccionario de datos con un 20%, el diagrama de flujo de datos con un 18%, las miniespecificaciones con un 15%, el diagrama de entidad relación con un 13% y el glosario de términos con un 11%.

Figura 2.10



- Se preguntó si llevaban algún control sobre los sistemas que habían hecho. Esto se refiere a que si llevaban un registro sobre las líneas de código que habían utilizado en sus sistemas, así como también un registro del tiempo y costo. En la Figura 2.11 se pueden ver los resultados de estas preguntas. De lo anterior se concluye que no se lleva a cabo prácticamente este tipo de control.

Figura 2.11



- Además se preguntó cuales son los problemas que han tenido en la realización de sistemas con un equipo de trabajo. Las respuestas a esta pregunta se observan en la Figura 2.12. Los cinco problemas que destacaron son: la entrega fuera de tiempo con un 20%, la falta de documentación con un 15%, el mal uso de los recursos con un 11%, los problemas técnicos con un 10% y los errores en el algoritmo con un 9%.

Figura 2.12

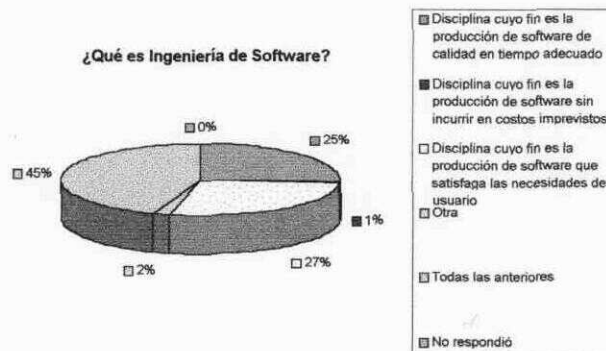


2.4.3 Preguntas Relacionadas con Definiciones

A continuación se muestran siete gráficas donde se pueden apreciar los resultados de siete preguntas en las que se les pedía a los encuestados que definieran los conceptos que se les presentó.

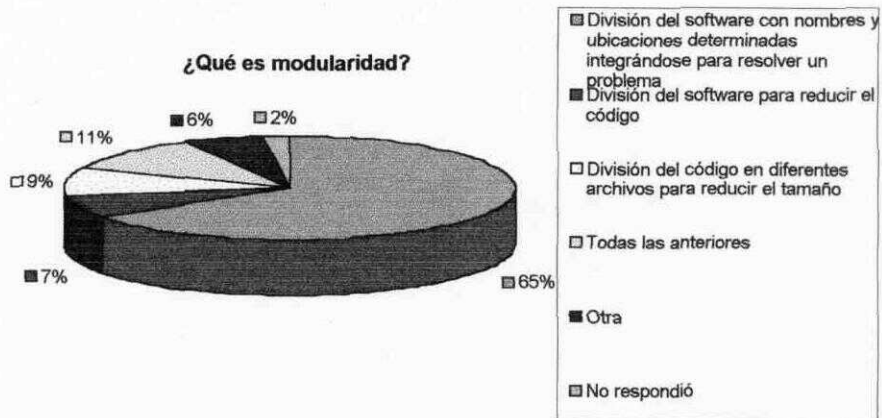
- Se preguntó primeramente qué es Ingeniería de Software. Como se puede apreciar en la Figura 2.13, para esta pregunta se seccionó la definición en tres partes a fin de que se seleccionara la opción de "Todas las anteriores". El 49% de los encuestados contestó correctamente.

Figura 2.13



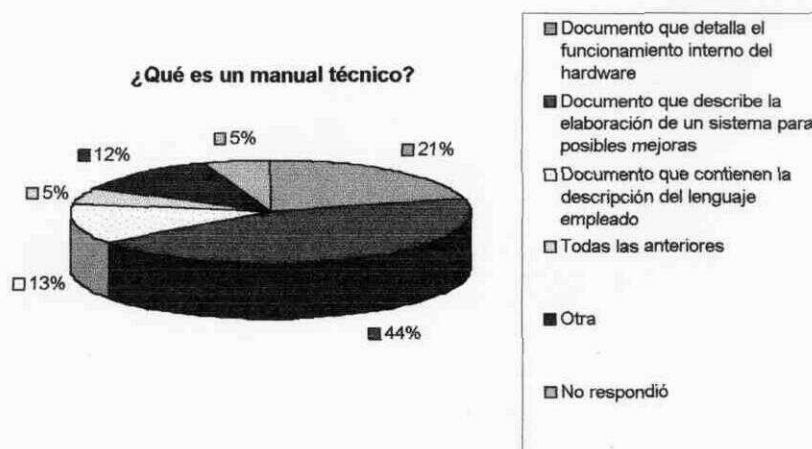
- Se preguntó qué es modularidad. En la Figura 2.14 se observan las respuestas a ésta pregunta cuya respuesta correcta es: "La división del software con nombres y ubicaciones determinadas integrándose para resolver un problema". El 65% de los encuestados respondió correctamente.

Figura 2.14



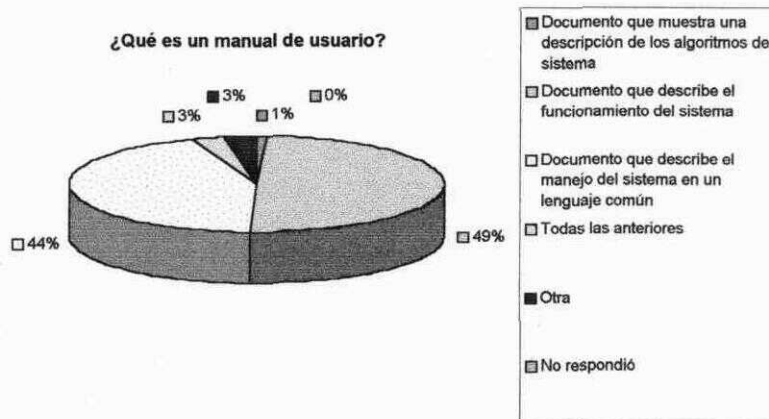
- Se preguntó qué es un manual técnico. El porcentaje de las respuestas se visualiza en la Figura 2.15. La respuesta correcta es: "Documento que describe la elaboración de un sistema para posibles mejoras". El 44% de los encuestados respondió correctamente.

Figura 2.15



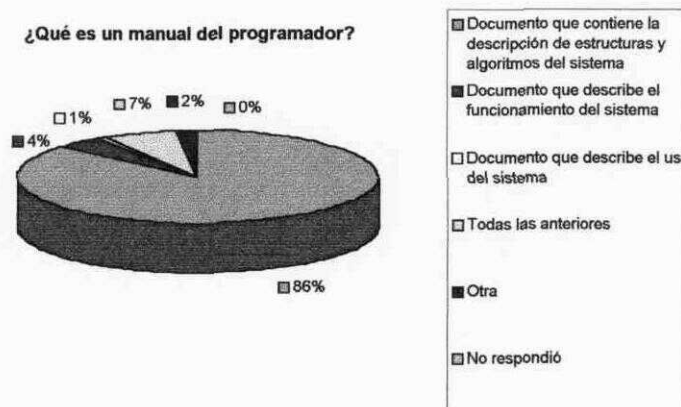
- Se preguntó qué es un manual de usuario. La Figura 2.16 muestra los porcentajes de las respuestas. La respuesta correcta es: "Documento que describe el manejo del sistema en un lenguaje común". El 44% de los encuestados contestó correctamente, pero el 49% afirmó que es un documento que describe el funcionamiento del sistema. Es importante aclarar que el documento debe estar escrito con palabras sencillas de manera que cualquier usuario con un conocimiento mínimo en computadoras lo pueda entender.

Figura 2.16



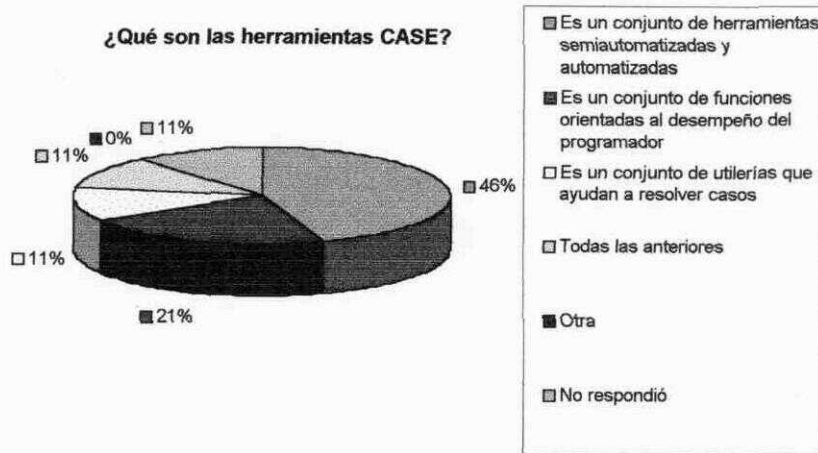
- Se preguntó qué es un manual de programador. Las respuestas a esta pregunta se pueden observar en la Figura 2.17. La respuesta correcta es: "Documento que contiene la descripción de estructuras y algoritmos del sistema". El 86% de los encuestados contestó correctamente.

Figura 2.17



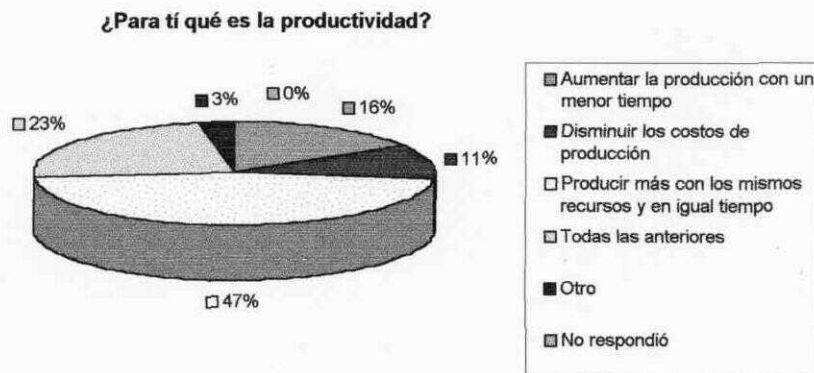
- Se preguntó qué son las herramientas CASE. En la Figura 2.18 se observan los resultados de esta pregunta cuya respuesta correcta es: "Un conjunto de herramientas semiautomatizadas y automatizadas". El 46% de los encuestados respondió correctamente.

Figura 2.18



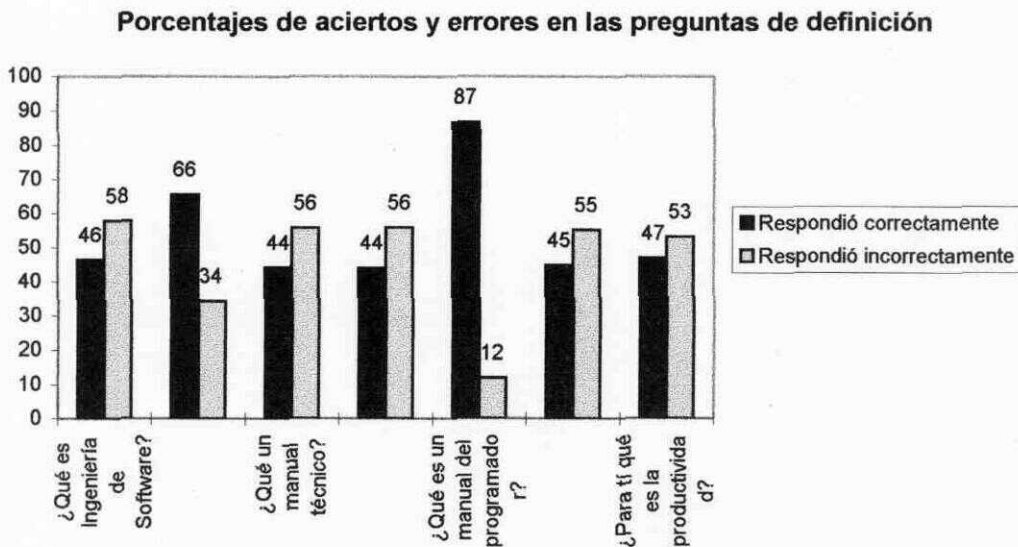
- Se preguntó qué es la productividad. El porcentaje de las respuestas se observa en la Figura 2.19. La respuesta correcta es: "Producir más con los mismos recursos y en igual tiempo". El 47% de los encuestados contestó correctamente.

Figura 2.19



- En la Figura 2.20 se concentran las respuestas de las siete gráficas anteriores. Para elaborar esta gráfica se sumaron todas las respuestas incorrectas que se despliegan bajo la etiqueta "Respondió incorrectamente". Las respuestas correctas se despliegan con la etiqueta "Respondió correctamente". Sólo dos preguntas fueron contestadas correctamente por más del 60%. Las restantes no pudieron superar el 50%. Tomando los valores de esta gráfica como valores absolutos se tiene que sólo dos preguntas fueron respondidas correctamente y las restantes erróneamente. Con lo que se concluye que menos del 50% de los encuestados maneja los conceptos correctos.

Figura 2.20

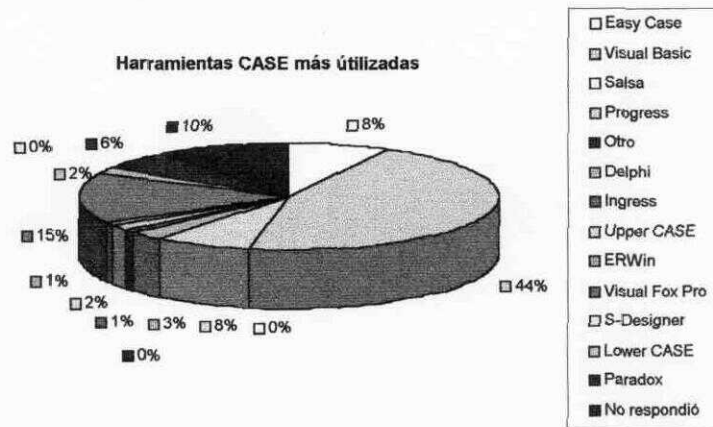


2.4.4 Preguntas sobre la Utilización de Software, Metodologías y Documentación

En las siguientes gráficas se determina el porcentaje de utilización de las herramientas CASE, los lenguajes de programación y los manejadores de bases de datos. En las opciones que escogieron los encuestados se mezclaron productos que correspondían a cualquiera de las tres categorías antes mencionadas a fin de que se escogieran las correctas.

- Se preguntó qué herramientas CASE habían empleado y cuáles conocían. En la Figura 2.21 se observan los resultados. Con esta gráfica se concluye que Visual Basic es la herramienta CASE más utilizada. Los manejadores de bases de datos y lenguajes de programación también eran utilizados como herramientas CASE. Se aclara que Visual Basic no es una herramienta CASE. Las herramientas CASE que se mencionan en las preguntas son: Easy CASE, ERWin y S-Designer.

Figura 2.21



- Se preguntó qué lenguajes de programación empleaban más. Los resultados se pueden apreciar en la Figura 2.22. Con esta gráfica se concluye que Visual Basic es el lenguaje de programación más utilizado, además de que los encuestados utilizan herramientas CASE y manejadores de bases de datos para programar. Los lenguajes de programación que se mencionan en las preguntas son : Visual Basic, RPG, Cobol, Turbo Pascal, Basic, Clipper, Fortran.

Figura 2.22



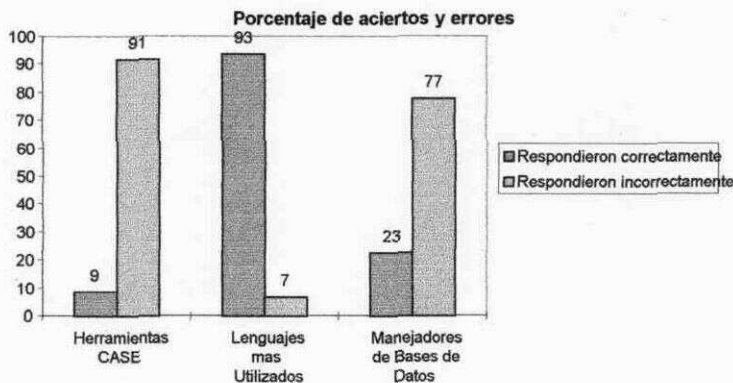
- Se preguntó qué manejadores de base de datos conocen. Las respuestas a esta pregunta se pueden observar en la Figura 2.23. Con esta gráfica se concluye que Visual Basic es conocido también como un manejador de bases de datos de otros productos que no lo son. Los manejadores mencionados en la pregunta son: Progress, Adaba, Paradox, Ingress, DB2.

Figura 2.23



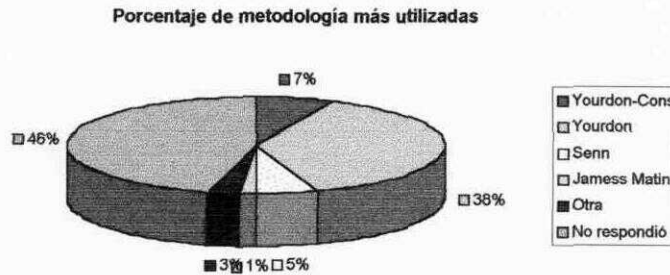
- En la Figura 2.24 se integran las respuestas de las tres gráficas anteriores. Al elaborar esta gráfica se sumaron todas las respuestas incorrectas que se despliegan bajo la etiqueta "Respondió correctamente". Las respuestas correctas se despliegan con la etiqueta "Respondió incorrectamente". Así se observa qué porcentaje de los encuestados respondió acertadamente y qué porcentaje respondió erróneamente. Se concluye que se tiene buen nivel de conocimiento de lenguajes. Pero en cuanto a manejadores de bases de datos y herramientas CASE, no se conocen siquiera los nombres de los productos. Por otro lado se maneja una buena cantidad de lenguajes, pero no así de herramientas CASE ni manejadores de bases de datos.

Figura 2.24



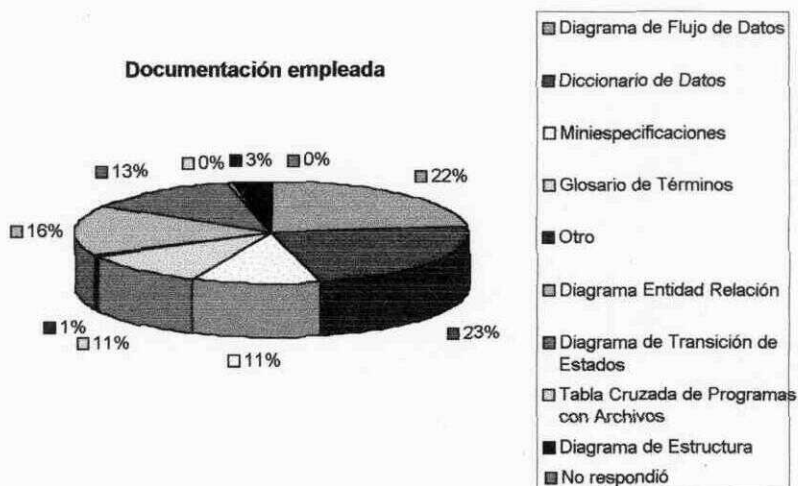
- Se preguntó que metodologías habían utilizado. Las respuestas a esta pregunta se observan en la Figura 2.25. Es importante señalar que son pocas las metodologías que son utilizadas. La que se utiliza más es la de Yourdon con un 37%. Es interesante hacer notar que un 44% se abstuvo de responder.

Figura 2.25



- Se preguntó qué documentación han llevado a cabo. Como se aprecia en la Figura 2.26, no es uniforme la elaboración de documentación. Los cinco documentos que más empleados son: el diccionario de datos, el diagrama de flujo de datos, el diagrama de entidad relación, las miniespecificaciones, glosario de términos.

Figura 2.26



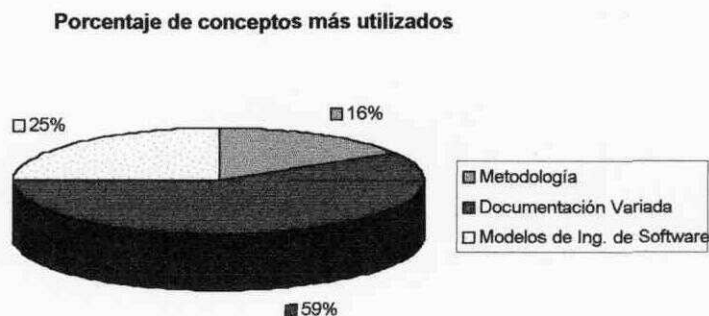
- Se preguntó qué modelos de desarrollo son los más utilizados. Las respuestas a esta pregunta se encuentran en la Figura 2.27. Se resalta el hecho de que el 41% dijo usar el modelo de Prueba y Error. Este modelo no es muy eficiente desde el punto de vista productivo; además solo el 7% usa el modelo de espiral, este es el modelo mas productivo. Un 29% no respondió esta pregunta. Es preocupante el hecho de que no supieran definir qué modelo usan al desarrollar.

Figura 2.27



- En la Figura 2.28 se unen las respuestas de las tres gráficas anteriores. Para elaborar esta gráfica se sumaron y promediaron los datos de cada una de las gráficas anteriores. Se concluye que está difundido mucho más el uso de documentación que el uso de metodologías.


Figura 2.28



2.5 Formato de Encuesta

La encuesta que se aplicó a los tesisistas se observa en la Figura 2.29a y 2.29b, dicha encuesta consta de dos hojas y está conformada por 36 preguntas.

Figura 2.29a. Primera Hoja de la Encuesta.



UNIVERSIDAD POPULAR AUTONOMA DEL ESTADO DE PUEBLA
Licenciatura en Sistemas Computacionales

ENCUESTA DE INGENIERIA DE SOFTWARE

1. ¿Qué es Ingeniería de Software?
 Disciplina cuyo fin es la producción de software de calidad en tiempo adecuado
 Disciplina cuyo fin es la producción de software sin incurrir en costos imprevistos
 Disciplina cuyo fin es la producción de software que satisfaga las necesidades del usuario
 Todas las anteriores
 Otra _____

2. ¿Qué porcentaje crees saber sobre Ingeniería de Software?
 0 - 20% 21 - 40% 41 - 60% 61 - 80% 81 - 100%

3. ¿Crees que sirva para algo la Ingeniería de Software?
 Sí No
 ¿Por qué? _____

4. ¿Crees haber cumplido con los objetivos del curso de Ingeniería de Software?
 Sí No

5. ¿Cuántos sistemas has elaborado?
 Ninguno 1 - 5 6 - 10 11 - 15 16 - 20
 21 o más

6. ¿Qué es modularidad?
 División del software con nombres y ubicaciones determinadas integrándose para resolver un problema
 División del software para reducir el código
 División del código en diferentes archivos para reducir el tamaño
 Todas las anteriores
 Otra _____

7. ¿Has empleado la modularidad en tus programas?
 Sí No

8. ¿En cuántos sistemas has usado la modularidad?
 Ninguno 1 - 5 6 - 10 11 - 15 16 - 20
 21 o más

9. ¿Te ahorra tiempo al hacer tus programas por usar modularidad?
 Sí No

10. ¿En cuántos sistemas has reutilizado código?
 Ninguno 1 - 5 6 - 10 11 - 15 16 - 20
 21 o más

11. ¿Qué metodología conoces (C) y cuál has aplicado (A)?
 C A C A C A
 Tom de Mar Yourdon James Martin
 Yourdon-Cons Senn Otra

12. ¿En cuántos sistemas has aplicado metodología?
 Ninguno 1 - 5 6 - 10 11 - 15 16 - 20
 21 o más

13. ¿Crees que usar una metodología ayuda al desarrollo de un sistema?
 Sí No
 ¿Por qué? _____

14. ¿Conoces el ciclo de Vida de un Sistema?
 Sí No

15. Asigna números ascendentes según su importancia del 1 al que creas conveniente


__Análisis	__Implementación	__Requerimientos
__Implementación	__Revisión	__Desarrollo
__Documentación	__Diseño	__Ingeniería
__Mantenimiento	__Corrección de Errores	__Planificación
__Especificaciones	__Pruebas	__Control de Calidad
__Detección de Errores	__Aseguramiento de Calidad	__Otro

16. ¿Cuál es la documentación mínima necesaria que se debe entregar con un sistema?
 Manespecificaciones Diagrama Entidad Relación
 Diccionario de Datos Diagrama de Transición de Estados
 Glosario de Términos Diagrama de Estructura
 Diagrama de Flujo de Datos Tabla Cruzada de Programas con Archivos
 Otro _____

17. De las siguientes herramientas, ¿cuales conoces? (C) y ¿cuales has empleado correctamente (E)?
 C E C E
 Diagrama de Flujo de Datos Diagrama Entidad Relación
 Diccionario de Datos Diagrama de Transición de Estados

Fuente: Ingeniería de Software.

Figura 2.29b. Segunda Hoja de la Encuesta.



UNIVERSIDAD POPULAR AUTONOMA DEL ESTADO DE PUEBLA
Licenciatura en Sistemas Computacionales

ENCUESTA DE INGENIERIA DE SOFTWARE

20. ¿Qué es un manual de usuario?
 Documento que muestra una descripción de los algoritmos del sistema
 Documento que describe el funcionamiento del sistema
 Documento que describe el manejo del sistema en un lenguaje común
 Todas las anteriores
 Otra _____

21. ¿Cuántos manuales del usuario has realizado?
 Ninguno 1 - 5 6 - 10 11 - 15 16 - 20
 21 o más

22. ¿Qué es un manual del programador?
 Documento que contiene la descripción de estructuras y algoritmos del sistema
 Documento que describe el funcionamiento del sistema
 Documento que describe el uso del sistema
 Todas las anteriores
 Otra _____

23. ¿Cuántos manuales del programador has elaborado?
 Ninguno 1 - 5 6 - 10 11 - 15 16 - 20
 21 o más

24. De los siguientes modelos de desarrollo, ¿Cuáles has utilizado?
 Prueba y Error Incremental Cascada
 Prototipos Espiral Otro _____

25. ¿Que son las herramientas CASE?
 Es un conjunto de herramientas semiautomatizadas y automatizadas
 Es un conjunto de funciones orientadas al desempeño del programador
 Es un conjunto de utilitarias que ayudan a resolver casos
 Todas las anteriores
 Otra _____

26. ¿Cuáles herramientas CASE conoces (C) y cuáles has empleado (E)?

C	E	C	E	C	E
<input type="checkbox"/> Easy Case	<input type="checkbox"/> Delphi	<input type="checkbox"/> Ingress	<input type="checkbox"/> Visual Fox P	<input type="checkbox"/> S-Designer	<input type="checkbox"/> S-Designer
<input type="checkbox"/> Visual Basic	<input type="checkbox"/> Progress	<input type="checkbox"/> Upper CASE	<input type="checkbox"/> Lower CASE	<input type="checkbox"/> ERWin	<input type="checkbox"/> Paradox
<input type="checkbox"/> Salsa	<input type="checkbox"/> Progress	<input type="checkbox"/> ERWin	<input type="checkbox"/> Paradox	<input type="checkbox"/> Paradox	<input type="checkbox"/> Paradox
<input type="checkbox"/> Otro _____					

27. ¿Qué lenguajes has utilizado?
 Visual Basic Paradox DB2
 Delphi Progress ERWin Turbo Pascal
 RPG Visual Fox P Basic
 Adaba Ingress Clipper
 Otro _____ Cobol Fortran

28. ¿Qué manejadores de bases de datos conoces?
 Visual Basic Paradox ERWin
 Turbo Pascal Visual C++ C++
 Delphi Progress Visual Fox P Basic
 RPG Cobol Clipper
 Adaba DB2 Fortran
 Otro _____

29. ¿Llevas un control de líneas de programación?
 Sí No

30. ¿Llevas un control del costo y tiempo invertido en cada programa?
 Sí No

31. ¿Para ti que es la productividad?
 Aumentar la producción con un menor tiempo
 Disminuir los costos de producción
 Producir mas con los mismos recursos y en igual tiempo
 Todas las anteriores
 Otra _____

32. ¿Pienas que la Ingeniería de Software te ayuda en la productividad?
 Sí No
 ¿Por qué? _____

33. ¿Cuáles son los problemas que has tenido en la realización de sistemas con un equipo de trabajo?
 Entrega fuera de tiempo Errores de Algoritmo Mal uso de recursos
 Falta de documentación Falta ayuda en línea Mala integración del sistema
 Duplicidad de código Problemas técnicos Incumplimiento de requerimientos
 Otra _____

Fuente: Ingeniería de Software.

2.6 Resultados

De acuerdo a las respuestas que se obtuvieron en la encuesta, se pueden hacer las siguientes conclusiones :

1. Los encuestados no demostraron mucha seguridad en cuanto a sus conocimientos de Ingeniería de Software, debido al poco aprovechamiento en sus cursos de Ingeniería de Software, aunque están plenamente convencidos de que el aplicar adecuadamente la Ingeniería de software les puede ayudar a ser más productivos.
2. Aun cuando la mayoría de los encuestados no han realizado más de 5 sistemas, dijeron conocer perfectamente el ciclo de vida de un sistema, además de tener buenas prácticas en el desarrollo de sistemas :

- Modularidad

- Reutilización de código
 - Aplicación de metodologías
 - Control del desarrollo de los sistemas
3. De las definiciones que se preguntaron a los encuestados, se concluye que tienen un nivel promedio de conocimientos básicos, y un nivel de cultura computacional bastante bajo; esta conclusión se justifica al observar los resultados de la encuesta en la Figura 2.20.
4. Dentro del estudio realizado se comprobó la existencia de un problema dentro de la Licenciatura de Sistemas Computacionales, en el cual los alumnos tenían poco conocimiento sobre herramientas CASE y manejadores de bases de datos debido al poco uso de los mismos; si estos paquetes no son utilizados se debe a que no están al alcance de los alumnos.

Se hace la aclaración de que las conclusiones a las que se llegan en esta encuesta se apegan enteramente al análisis de las respuestas de los alumnos que fueron encuestados.

Capítulo 3. Benchmarking

Dado el objetivo principal de este trabajo, que es el desarrollo de Ingeniería de Software para el Departamento de Revisión de Procesos, se presenta en este capítulo el estudio de Benchmarking de los principales manejadores de bases de datos que existen en el mercado.

3.1 Consideraciones a Tomar en el Estudio de Benchmarking

En este estudio se determinaron algunas de las características que deben poseer los manejadores de base de datos, para ser considerados como una buena opción para el desarrollo de los sistemas de SINAF y SINIGA, así como para la buena administración de la información que se maneja.

El objetivo es seleccionar a los mejores cinco productos en cada punto, aunque cabe mencionar que en algunos casos el número de productos sobresalientes era menor de cinco y en otros casos mayor, debido a que los demás productos quedaban empatados.

NOTA: La información requerida de algunos de los productos evaluados no estaba disponible en el momento de la evaluación debido a que era la primera versión o apenas estaban siendo migrados de ambientes de texto a ambientes gráficos. Ejemplos de esos productos son: CA-OpenIngres e Informix SQL.

3.1.1 Sistema Operativo

Este punto indica la cantidad de sistemas operativos para los cuales existe una versión del manejador de base de datos. El hecho de tener varias versiones del producto, asegura la transportabilidad de la base de datos y de los programas sin realizar cambios substanciales. Los siguientes productos soportan una mayor cantidad de sistemas operativos; el resto se muestra en la Tabla 3.1:

- Progress v8 12
- Oracle 12
- SQL Windows v5.0 10
- SQL Server v6.0 9
- Informix SQL 9

Tabla 3.1. Sistema Operativo.

Característica	SQL Server, V6.0	SQL Windows, V5.0	Paradox, V7.0	Sybase	Oracle	Progress, V8	Access, V2.0	Adabas	CA-Visual Objects	DB2, V2.1	FoxPro, V3.0	Informix-SQL	CA-Open Ingress
MS-DOS	■	■	■	N/D	■	■	■	□	□	■	□	■	□
Windows 3.x	■	■	■	N/D	■	■	■	□	■	■	■	■	□
Windows NT	■	■	■	■	■	■	■	■	□	■	□	■	■
Windows 95	■	■	■	■	■	■	■	■	□	□	■	□	□
NML	□	□	□	N/D	■	■	□	□	□	□	□	□	□
OS/400	□	■	□	■	N/D	■	□	□	□	□	□	□	□
CTOS	□	□	□	□	N/D	■	□	□	□	□	□	□	□
HP MPE/XL	□	N/D	□	□	■	□	□	□	□	□	□	□	□
HP MPE/ix	□	N/D	□	□	N/D	■	□	□	□	□	□	□	□
System 7 Macintosh	■	N/D	□	□	■	□	□	□	□	■	□	■	□
NetWare	■	■	N/D	■	■	■	□	□	□	□	□	■	□
OS/2	■	■	N/D	■	■	■	□	■	□	■	□	■	■
UNIX	■	■	□	■	■	■	□	■	□	■	□	■	□
UnixWare	N/D	■	□	□	N/D	□	□	□	□	□	□	□	□
UnixMotif	N/D	■	□	□	N/D	□	□	□	□	□	□	□	■
SCO Unix	N/D	N/D	□	□	N/D	□	□	□	□	□	□	■	□
VM	□	N/D	□	N/D	■	□	□	□	□	□	□	□	□
VMS	■	N/D	□	N/D	■	■	□	□	□	□	□	□	□
XENIX	□	N/D	□	N/D	N/D	□	□	□	□	□	□	■	□

Entre más sistemas operativos pueda soportar un manejador de bases de datos, más seguridad hay de que no existan problemas de compatibilidad en el futuro.

3.1.2 Plataforma de Hardware

Este aspecto se refiere a la arquitectura del hardware de un modelo en particular o familia de computadoras; esto es, la plataforma que los desarrolladores de software utilizan para escribir las aplicaciones. Los productos que soportan una cantidad mayor de plataformas son los siguientes:

- Sybase 6
- Progress v8 6
- SQL Server v6.0 5
- SQL Windows v5.0 4
- CA-Open Ingress 4
- Oracle 4

El resto de los productos se muestra en la Tabla 3.2.

Tabla 3.2. Plataformas.

Característica	SQL Server, V6.0	SQL Windows, V5.0	Paradox, V7.0	Sybase	Oracle	Progress, V8	Access, V2.0	Adabas	CA-Visual Objects	DB2, V2.1	FoxPro, V3.0	Informix-SQL	CA-Open Ingress
AS/400	■	■	□	■	■	■	□	□	□	□	□	□	□
MIPS	□	□	□	□	N/D	□	□	□	□	□	□	□	□
ALPHA	□	□	□	□	N/D	■	□	□	□	□	□	□	□
Mainframes	■	□	□	■	N/D	■	□	□	□	□	□	□	□
Vax	N/D	N/D	□	■	N/D	■	□	□	□	□	□	□	■
Unix	■	■	□	■	■	■	□	■	■	■	■	■	■
PC	■	■	■	■	■	■	■	■	■	■	■	■	■
Power PC	■	■	■	■	■	□	□	□	□	□	□	■	□
HP 3000	□	□	□	□	N/D	□	□	□	□	□	□	□	■
HP 9000	□	□	□	□	N/D	□	□	□	■	□	□	□	□

3.1.3 Red

Aquí se especifica la variedad de sistemas operativos de red que soportan estos productos. El soporte de un sistema operativo de red proporciona la opción de explotar adecuadamente la arquitectura cliente/servidor.

Los productos que proporcionan más manejo de redes son los siguientes:

- SQL Server v6.0 5
- SQL Windows v5.0 4
- Paradox v7.0 4
- Progress v8 3
- Oracle 3
- Sybase 3

El resto se observa en la Tabla 3.3.

Tabla 3.3. Red.

Característica	SQL Server, V6.0	SQL Windows, V5.0	Paradox, V7.0	Sybase	Oracle	Progress, V8	Access, V2.0	Adabas	CA-Visual Objects	DB2, V2.1	FoxPro, V3.0	Informix-SQL	CA-Open Ingress
IBM Server	■	N/D	N/D	N/D	N/D	□	□	N/D	N/D	■	N/D	N/D	N/D
MSLAN Manager	■	■	■	N/D	N/D	□	□	N/D	N/D	N/D	N/D	N/D	N/D
Windows 3.11	■	■	■	■	■	■	■	■	■	N/D	■	N/D	N/D
Windows NT Server	■	■	■	■	■	■	■	■	■	N/D	■	N/D	N/D
Novell Netware	■	■	■	■	■	■	□	N/D	N/D	■	N/D	N/D	N/D

3.1.4 Protocolos de Red

Aquí se evalúa la cantidad de protocolos de red que estos manejadores de bases de datos soportan, ya sea de forma nativa o mediante productos de terceros. Los productos que manejan un mayor número de protocolos de red son los siguientes :

- CA-Open Ingress 9
- Progress v8 7
- SQL Server v6.0 6
- DB2 v2.1 5

- Oracle

4

El resto se puede observar en la Tabla 3.4.

Tabla 3.4. Protocolos de Red.

Característica	SQL Server, V6.0	SQL Windows, V5.0	Paradox, V7.0	Sybase	Oracle	Progress, V8	Access, V2.0	Adabas	CA-Visual Objects	DB2, V2.1	FoxPro, V3.0	Informix-SQL	CA-Open Ingress
TCP / IP	■	■	■	■	■	■	□	N/D	N/D	■	□	□	■
TLI	□	□	N/D	N/D	N/D	■	□	N/D	N/D	□	□	□	□
NetBIOS	■	■	■	■	■	■	□	N/D	N/D	■	□	□	■
SPX / IPX	■	□	□	N/D	N/D	■	□	N/D	N/D	□	□	□	■
SNA	■	N/D	□	N/D	N/D	■	□	N/D	N/D	■	□	□	■
DECNet	■	N/D	□	N/D	N/D	■	□	N/D	N/D	□	□	□	■
OSI	N/D	N/D	N/D	N/D	■	■	□	N/D	N/D	□	□	□	■
APPC	N/D	N/D	□	N/D	N/D	□	□	N/D	N/D	■	□	□	□
PX	N/D	N/D	□	N/D	N/D	□	□	N/D	N/D	■	□	□	□
LU0	N/D	N/D	□	N/D	N/D	□	□	N/D	N/D	□	□	□	■
LU6.2	N/D	N/D	□	N/D	N/D	□	□	N/D	N/D	□	□	□	■
informix-Net 5.x	N/D	N/D	N/D	N/D	N/D	□	□	N/D	N/D	□	□	■	□
Informix Enterprise Gateway 7 1	N/D	N/D	N/D	N/D	N/D	□	□	N/D	N/D	□	□	■	□
Informix-DCE/NET 1.x	N/D	N/D	N/D	N/D	N/D	□	□	N/D	N/D	□	□	■	□
X.25	■	■	N/D	■	■	□	□	N/D	N/D	□	□	□	■

3.1.5 Acceso a Internet

Debido al crecimiento de Internet, el acceso a las bases de datos a través de esta red es una característica que ofrece muchas ventajas. Como se sabe, el medio de transmisión de información más popular en Internet es el Web, o “WWW”. Por eso se resalta de importancia de la utilización del HTML (Lenguaje de Marcación de Hipertexto, por sus siglas en inglés Hypertexto Markup Language) para la elaboración de páginas y Java como lenguaje de programación en Internet.

Los productos que soportan estas características son: (Tabla 3.5):

- SQL Server v6.0 HTML, Java
- Progress v8 HTML
- Informix-SQL HTML

El resto de los productos no soportan ninguna de las características mencionadas como se observa en la Tabla 3.5.

Tabla 3.5. Acceso a Internet.

Característica	SQL Server, V6.0	SQL Windows, V5.0	Paradox, V7.0	Sybase	Oracle	Progress, V8	Access, V2.0	Adabas	CA-Visual Objects	DB2, V2.1	FoxPro, V3.0	Informix-SQL	CA-Open Ingress
HTML	<input checked="" type="checkbox"/>	N/D	<input type="checkbox"/>	N/D	N/D	<input checked="" type="checkbox"/>	N/D	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	N/D
Java	<input checked="" type="checkbox"/>	N/D	<input type="checkbox"/>	N/D	N/D	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

3.1.6 Tipo de Herramienta

En el desarrollo de software en un ambiente cliente/servidor, es necesario contar que los productos se orienten a la parte del cliente (front end) y del servidor (back end). La parte front end provee una interfaz con el usuario que hace transparente la administración de los datos, de la cual se encargan la parte de back end.

Todos, a excepción de SQL Server V6.0 que se utiliza como back end, se clasifican en ambas categorías (Tabla 3.6).

Tabla 3.6. Tipo de Herramienta.

Característica	SQL Server, V6.0	SQL Windows, V5.0	Paradox, V7.0	Sybase	Oracle	Progress, V8	Access, V2.0	Adabas	CA-Visual Objects	DB2, V2.1	FoxPro, V3.0	Informix-SQL	CA-Open Ingress
Front End	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Back End	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

3.1.7 Tipo de Aplicación

El tipo de aplicación puede ser de 16 o 32 bits. Esto indica la plataforma más apta para ejecutar la aplicación ya sea de hardware (procesador) o software (sistema operativo). Casi todos los productos manejan las características de los 16 y 32 bits. Oracle es el único que además aprovecha las ventajas de los 64 bits como puede verse en la Tabla 3.7.

Tabla 3.7. Tipo de Aplicación.

Característica	SQL Server, V6.0	SQL Windows, V5.0	Paradox, V7.0	Sybase	Oracle	Progress, V8	Access, V2.0	Adabas	CA-Visual Objects	DB2, V2.1	FoxPro, V3.0	Informix-SQL	CA-Open Ingress
16 Bits	■	■	■	N/D	■	■	■	■	■	■	■	■	■
32 Bits	■	■	■	■	■	■	■	■	■	■	■	■	■
64 Bits	□	□	□	□	■	□	□	□	□	□	□	□	□

3.1.8 Conectividad a Través de SQL

En esta sección se indican los manejadores de bases de datos que proveen el acceso, a través de SQL, a bases de datos que no son nativas, así como el soporte de instrucciones SQL. Los manejadores de Bases de Datos que tienen un mayor número de opciones para interconectarse son:

- SQL Windows v5.0 9
- Paradox v7.0 7
- SQL Server v6.0 6
- Oracle 6
- Progress v8 5

Como puede observarse en la Tabla 3.8 CA-Visual Objects no ofrece conectividad debido a era la primera versión al momento de realizar este estudio.

Tabla 3.8. Conectividad a través de SQL.

Característica	SQL Server, V6.0	SQL Windows, V5.0	Paradox, V7.0	Sybase	Oracle	Progress, V8	Access, V2.0	Adabas	CA-Visual Objects	DB2, V2.1	FoxPro, V3.0	Informix-SQL	CA-Open Ingress
Oracle	■	■	■	■	N/A	■	□	■	□	□	■	■	■
Sybase	■	■	■	N/A	N/D	■	□	□	□	□	■	■	■
SQL Server	N/A	■	■	■	N/D	□	■	□	□	■	■	■	■
Informix	■	■	■	N/D	■	□	□	□	□	□	□	□	□
InterBase	N/D	N/D	■	N/D	N/D	□	□	□	□	□	□	□	□
CA Open Ingress	N/D	■	■	□	N/D	□	□	□	□	□	□	□	□
SQL Network	N/D	■	N/D	N/D	■	□	□	□	□	□	□	□	□
SQL Windows	■	N/A	N/D	N/D	■	□	□	□	□	□	□	□	□
AllBase/SQL	N/D	■	N/D	N/D	N/D	■	□	□	□	□	□	□	□
Novell Btrieve	N/D	■	□	N/D	■	□	□	□	□	□	□	□	□
AS/400	■	■	□	■	■	■	□	□	□	□	□	□	□
ANSI - SQL	■	N/D	■	■	■	■	■	□	□	□	□	□	□

3.1.9 Conectividad a Través de ODBC

Aquí se indican los manejadores de base de datos que ofrecen conexión a través de controladores de ODBC (Open Data Base Connectivity) a otras bases de datos. Los productos que ofrecen mayor conectividad son:

- SQL Windows v5.0 7
- Paradox v7.0 5
- Progress v8 4
- Oracle 4
- SQL Server v6.0 4

Algunos de los productos no ofrecían conectividad en la versión evaluada, tal como se observa en la Tabla 3.9.

Tabla 3.9. Conectividad a través de ODBC.

Característica	SQL Server, V6.0	SQL Windows, V5.0	Paradox, V7.0	Sybase	Oracle	Progress, V8	Access, V2.0	Adabas	CA-Visual Objects	DB2, V2.1	FoxPro, V3.0	Informix-SQL	CA-Open Ingress
Ingress	■	■	■	N/D	N/D	□	□	□	□	□	□	□	□
Paradox	N/D	■	N/A	□	N/D	□	□	□	□	□	■	□	□
DBase	N/D	■	■	N/D	■	□	□	□	□	□	■	□	□
Oracle	■	■	N/D	■	N/A	■	□	□	□	□	□	□	□
RDB	N/D	■	N/D	N/D	N/D	■	□	□	□	□	□	□	□
Btreve	N/D	N/D	■	N/D	■	□	□	□	□	□	□	□	□
AllBase/SQL	N/D	N/D	■	N/D	N/D	■	□	□	□	□	□	□	□
AS/400	■	■	□	■	■	■	□	□	□	□	□	□	□
DB2	■	■	■	■	■	□	□	■	□	□	□	□	□

3.1.10 Interfases de Aplicación

Algunos bases de datos soportan especificaciones de intercambio de datos como DDE y OLE, así como de interfases de aplicación (Apios) y extensiones (OCX y VBX). Esto le da al producto capacidades de intercambio de información con aplicaciones como procesadores de palabras, hojas de cálculo, etc.

- SQL Server v6.0 10
- Paradox v7.0 9
- SQL Windows v5.0 8
- Progress v8 7
- Oracle 7

Algunos productos debido a su madurez, proveen una característica relativamente reciente: capacidad multimedia. Esta característica hace posible el uso de imágenes, vídeo y audio como cualquier otro tipo de datos. Estos productos se observan en la Tabla 3.10.

Tabla 3.10. Interfases de Aplicación.

Característica	SQL Server, V6.0	SQL Windows, V5.0	Paradox, V7.0	Sybase	Oracle	Progress, V8	Access, V2.0	Adabas	CA-Visual Objects	DB2, V2.1	FoxPro, V3.0	Informix-SQL	CA-Open Ingress
OLE 2.0	■	■	■	N/D	■	□	■	□	N/D	□	■	□	N/D
DDE	■	■	■	N/D	■	■	□	□	N/D	□	□	■	N/D
DLL's	■	■	■	N/D	■	■	□	□	N/D	□	□	□	N/D
VBX	■	■	■	N/D	■	■	□	□	N/D	□	□	□	N/D
OCX	■	■	■	N/D	N/D	■	□	□	N/D	□	□	□	N/D
ODBC	■	■	■	N/D	■	■	■	■	N/D	■	□	■	N/D
API	■	■	■	■	■	■	□	□	N/D	□	□	□	N/D
MAPI	■	N/D	■	N/D	N/D	□	□	□	N/D	□	■	□	N/D
DRDA	N/D	N/D	□	N/D	N/D	□	□	□	N/D	■	□	■	N/D
Unix Pipes	■	N/D	□	N/D	N/D	■	□	□	N/D	□	□	□	N/D
CLI	N/D	N/D	□	N/D	N/D	□	□	□	N/D	■	□	□	N/D
CCL/Informix	N/D	N/D	□	N/D	N/D	□	□	□	N/D	□	□	■	N/D
Manejo de Capacidades Multimedia	■	■	■	□	■	□	□	N/D	N/D	N/D	□	□	N/D

3.1.11 Características Generales de las Bases de Datos

En esta sección se listan las características generales más importantes que deben tener los manejadores de bases de datos. Estas características hacen del manejador un producto robusto y confiable. Los manejadores que cumplen con todas las características deseadas son :

- SQL Server v6.0
- SQL Windows v5.0
- Oracle
- Progress v8.0
- Adabas
- DB2 v2.1
- Informix SQL

Lo anterior se observa en la Tabla 3.11.

Tabla 3.11. Características Generales de las Bases de Datos.

Característica	SQL Server, V6.0	SQL Windows, V5.0	Paradox, V7.0	Sybase	Oracle	Progress, V8	Access, V2.0	Adabas	CA-Visual Objects	DB2, V2.1	FoxPro, V3.0	Informix-SQL	CA-Open Ingres
Manejo de Bases de Datos Relacionales	■	■	N/D	■	■	■	■	■	□	■	■	■	■
Integridad de Datos	■	■	N/D	■	■	■	■	■	N/D	■	■	■	■
Seguridad de Datos	■	■	N/D	■	■	■	■	■	■	■	■	■	■
Generación Automática de Código	■	■	N/D	N/D	■	■	□	■	□	■	N/D	■	N/D
Utilización de Lenguaje de 4a Generación	■	■	■	N/D	■	■	□	■	□	■	N/D	■	■
Arquitectura Cliente/Servidor	■	■	■	■	■	■	■	■	■	■	■	■	■

3.1.12 Incorporación de Código de Otros Lenguajes

En esta sección se examina si los manejadores de bases de datos son capaces de poder ejecutar rutinas en otros lenguajes, Los manejadores de bases de datos que manejan una mayor cantidad de opciones son:

- Oracle 5
- DB2 v2.1 4
- Adabas 3
- CA-OpenIngres 3

Como se puede observar en la Tabla 3.12, los lenguajes más soportados son C y C++

Tabla 3.12. Incorporación de Código de Otros Lenguajes.

Característica	SQL Server, V6.0	SQL Windows, V5.0	Paradox, V7.0	Sybase	Oracle	Progress, V8	Access, V2.0	Adabas	CA-Visual Objects	DB2, V2.1	FoxPro, V3.0	Informix-SQL	CA-Open Ingress
C	■	■	■	■	■	■	□	■	□	■	□	■	■
C++	■	■	■	■	■	□	□	■	■	■	□	■	□
Fortran	N/D	□	□	□	■	□	□	□	□	■	□	□	■
Cobol	N/D	□	□	□	■	□	□	■	□	□	□	□	□
Ada	N/D	□	□	□	■	□	□	□	□	□	□	□	■
ESQL/C	N/D	□	□	□	N/D	■	□	□	□	□	□	□	□
REXX	N/D	□	□	□	N/D	□	□	□	□	■	□	□	□

3.1.13 Herramientas de Desarrollo

En esta sección se listan algunas de las herramientas que viene como parte del producto, las cuales ayudan en el desarrollo de las aplicaciones. Los productos que más herramientas ofrecen son (Tabla 3.13):

- Progress v8 10
- Informix-SQL 9
- SQL Server v6.0 8
- CA-Visual Objects 8
- Paradox v7.0 7
- SQL Windows v5.0 7

Una característica muy importante que debe tener el manejador de bases de datos es la recuperación automática de caídas. Se hace indispensable en aquellos productos de nivel corporativo como Progress y SQL Server. El resto de los productos se pueden observar en la Tabla 3.13.

Tabla 3.14. Herramientas de Administración.

Característica	SQL Server, V6.0	SQL Windows, V5.0	Paradox, V7.0	Sybase	Oracle	Progress, V8	Access, V2.0	Adabas	CA-Visual Objects	DB2, V2.1	FoxPro, V3.0	Informix-SQL	CA-Open Ingress
Monitoreo Remoto	■	N/D	N/D	■	■	■	N/A	N/D	N/D	N/D	□	N/D	N/D
Mantenimiento de Base de Datos	■	N/D	■	■	■	■	□	N/D	N/D	■	N/D	■	■
Administración de Datos	■	N/D	■	■	■	■	□	■	■	■	N/D	■	■

3.1.15 Seguridad de Acceso

Esta sección se refiere a las características que ofrecen los manejadores de bases de datos, en cuanto a la seguridad de la información en sus bases de datos. Los productos que más características ofrecen son (Tabla 3.15):

- Progress v8 7
- Paradox 7
- Informix SQL 7
- SQL Server v6.0 6
- Oracle 6

El tema de seguridad es sumamente importante cuando se trata de manejadores de bases de datos grandes como Oracle, Progress y SQL Server, ya que aseguran la integridad de la información.

Tabla 3.15. Seguridad de Acceso.

Característica	SQL Server, V6.0	SQL Windows, V5.0	Paradox, V7.0	Sybase	Oracle	Progress, V8	Access, V2.0	Adabas	CA-Visual Objects	DB2, V2.1	FoxPro, V3.0	Informix-SQL	CA-Open Ingress
Bloqueo de Registros	■	■	■	■	■	■	N/D	■	N/D	N/D	■	■	■
Bloqueo de Tablas	■	■	■	N/D	■	■	N/D	■	N/D	N/D	N/D	■	■
Guarda procedimientos en la BD	N/D	N/D	■	N/D	■	■	N/D	■	N/D	N/D	■	■	N/D
Funciones incorporadas de Manejo de Registros	■	■	■	■	■	■	■	■	N/D	N/D	■	■	N/D
Soporta Manejo de Mensajes y Códigos de Error	■	■	■	■	■	■	■	N/d	N/D	N/D	■	■	N/D
Acepta Variables y Regresa Valores o Mensajes	■	■	■	■	■	■	■	N/D	N/D	N/D	N/D	■	N/D
Seguridad Multinivel	■	N/D	■	■	N/D	■	N/D	□	N/D	N/D	■	■	■

3.1.16 Triggers

Esta sección se refiere a la capacidad de manejo de Triggers por los manejadores de bases de datos. Cabe mencionar que los triggers son fragmentos de código o procedimientos que realizan una actividad en específico. Sólo siete de los productos cumplieron con las características deseadas, como puede se observar en al Tabla 3.16.

Tabla 3.16. Triggers.

Característica	SQL Server, V6.0	SQL Windows, V5.0	Paradox, V7.0	Sybase	Oracle	Progress, V8	Access, V2.0	Adabas	CA-Visual Objects	DB2, V2.1	FoxPro, V3.0	Informix-SQL	CA-Open Ingress
Ejecuta Triggers Automáticamente en Modificaciones de BD	■	■	■	N/D	■	■	N/D	■	N/D	N/D	■	■	N/D
Soporte de Múltiples Triggers	■	■	■	■	■	■	N/D	■	N/D	N/D	■	N/D	N/D

3.1.17 Importación y Exportación de Archivos

Esta sección se refiere a la capacidad que tienen los productos de leer y escribir archivos con formatos diferentes a los propios. Sólo dos productos soportan los formatos más comunes, como se observa en la Tabla 3.17:

- Progress, V8
- Paradox, V7.0

Tabla 3.17. Importación y Exportación de Archivos.

Característica	SQL Server, V6.0	SQL Windows, V5.0	Paradox, V7.0	Sybase	Oracle	Progress, V8	Access, V2.0	Adabas	CA-Visual Objects	DB2, V2.1	FoxPro, V3.0	Informix-SQL	CA-Open Ingress
ASCII	■	N/D	■	■	N/D	■	□	N/D	N/D	N/D	N/D	N/D	N/D
.DBF	N/D	N/D	■	N/D	N/D	■	■	N/D	N/D	N/D	■	□	□

3.1.18 Indexación

Esta sección se refiere a que si los manejadores de bases de datos le permiten al usuario manejar índices y valores únicos para los registros de la base de datos. Los productos que ofrecen esta flexibilidad son:

- Progress v8
- SQL Windows v5.0
- FoxPro v3.0
- CA-Visual Objects
- SQL Server v6.0
- Sybase
- Oracle

En la Tabla 3.18 se pueden observar los demás productos.

Tabla 3.18. Indexación.

Característica	SQL Server, V6.0	SQL Windows, V5.0	Paradox, V7.0	Sybase	Oracle	Progress, V8	Access, V2.0	Adabas	CA-Visual Objects	DB2, V2.1	FoxPro, V3.0	Informix-SQL	CA-Open Ingress
El Usuario Puede Agregar y Borrar Indices	■	■	N/D	■	■	■	N/D	N/D	■	N/D	■	N/D	N/D
El Usuano Puede Forzar Valores Unicos	■	■	■	■	■	■	N/D	N/D	■	N/D	■	N/D	N/D

3.1.19 Bases de Datos Soportadas en Forma Nativa

Esta sección se refiere a la capacidad de acceder otras bases de datos sin utilizar controladores especiales o extensiones. Esto nos permite transferir fácilmente datos de una base a otra. Los siguientes productos son los que dan soporte a mayor cantidad de bases de datos :

- SQL Windows v5.0 13
- Progress v8 10
- Paradox v7.0 8
- SQL Server v6.0 8
- Oracle 7

Como se puede observar en la Tabla 3.19, la mayoría de los productos sólo soportan dos o más formatos aparte del propio.

Tabla 3.19. Bases de Datos Soportadas en Forma Nativa.

Característica	SQL Server, V6.0	SQL Windows, V5.0	Paradox, V7.0	Sybase	Oracle	Progress, V8	Access, V2.0	Adabas	CA-Visual Objects	DB2, V2.1	FoxPro, V3.0	Informix-SQL	CA-Open Ingress
Progress	N/D	N/D	N/D	N/D	N/D	■	□	□	□	□	□	□	□
Sybase	■	■	N/D	N/A	N/D	■	□	□	□	□	■	■	■
OS/400	■	■	■	■	■	■	□	□	□	□	□	□	□
Oracle V7 SQL	■	■	N/D	■	N/A	■	□	■	□	□	■	■	■
RDB	N/D	■	□	■	■	■	□	□	□	□	□	□	□
RMS	N/D	N/D	□	□	N/D	■	□	□	□	□	□	□	□
C-ISAM	N/D	N/D	□	□	N/D	■	□	□	□	□	□	□	□
CT-ISAM	N/D	N/D	□	□	N/D	■	□	□	□	□	□	□	□
ODI	N/D	N/D	□	□	N/D	■	□	□	□	□	□	□	□
AllBase	N/D	■	N/D	□	N/D	■	□	□	□	□	□	□	□
SQL-Server	N/A	■	■	■	■	□	■	□	□	■	■	□	■
DB2 V3 SQL	■	■	■	■	N/D	□	□	■	□	■	■	□	■
WATCOM 32-Bit SQL	N/D	N/D	N/D	N/D	N/D	□	□	□	■	□	□	□	□
Fox Pro	■	■	N/D	N/D	■	□	□	□	■	□	□	□	□
SQL	■	■	■	■	■	□	□	□	■	□	□	□	□
dBase III y IV	N/D	■	■	N/D	■	□	□	□	□	□	■	□	□
Access	■	■	■	N/D	N/D	□	□	□	□	□	■	□	□
Paradox	N/D	■	■	N/D	N/D	□	□	□	□	□	■	□	□
Informix	■	■	■	N/D	■	□	□	□	□	□	□	■	□

3.1.20 Otras Características

Aquí se evalúan características variadas que se consideran importantes para el desarrollo de las aplicaciones así como la administración de los datos. Algunos productos incorporan características poderosas como la integración de herramientas CASE y el uso de asistentes para la programación. Los productos que tienen más características son :

- Oracle 12
- Progress v8.0 11
- SQL Server v6.0 11
- SQL Windows v5.0 10
- Paradox v7.0 8

En la Tabla 3.20 se observa el resto de los productos.

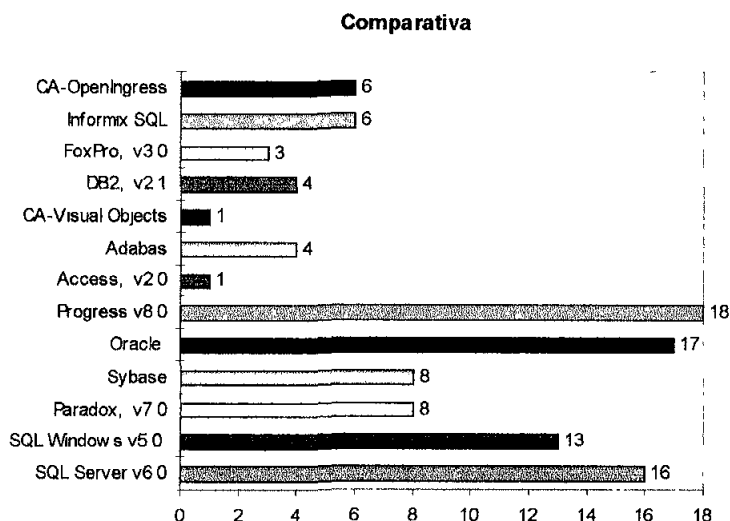
Tabla 3.20. Otras Características.

Característica	SQL Server, V6.0	SQL Windows, V5.0	Paradox, V7.0	Sybase	Oracle	Progress, V8	Access, V2.0	Adabas	CA-Visual Objects	DB2, V2.1	FoxPro, V3.0	Informix-SQL	CA-Open Ingress
Uso de Asistentes	■	■	■	N/D	■	■	■	N/D	N/D	N/D	■	N/D	N/D
Creación y Modificación de Objetos	■	■	■	N/D	■	■	N/D	□	□	■	■	■	■
Reutilización de Código	■	■	■	■	■	■	N/D	□	□	■	■	■	■
Ambiente de Desarrollo Visual	■	■	■	N/D	■	■	■	■	■	■	■	■	■
Integración de Herramientas CASE	■	■	N/D	■	■	■	□	□	□	□	□	■	□
Respaldo en Línea	■	■	N/D	■	■	■	N/D	N/D	N/D	N/D	□	■	■
Aplicaciones Ejemplo Construidas	■	■	■	N/D	■	■	N/D	N/D	N/D	N/D	N/D	N/D	N/D
Soporte de Versiones Anteriores	■	■	■	■	■	■	■	N/D	N/D	N/D	■	N/D	N/D
Programación Orientada a Eventos	■	■	■	N/D	■	■	□	□	■	■	■	■	■
Número de Nodos Soportados	N/D	N/D	N/D	N/D	∞	N/D	N/A	N/D	N/D	N/D	N/A	N/D	N/D
Recuperación Automática de Nodos	■	N/D	N/D	■	■	N/D	N/A	N/D	N/D	N/D	N/A	N/D	N/D
Integridad Referencial	N/D	N/D	N/D	N/D	■	■	N/D	N/D	■	■	N/A	■	■
Soporte Técnico	■	■	■	□	■	■	■	■	■	■	■	■	■

3.2 Resultados

En la Figura 3.1 se puede observar los productos mas sobresalientes :

Figura 3.1. Productos Más Sobresalientes.



Los cinco productos que sobresalieron, tal como se aprecia en la tabla anterior, son.

- 1) Progress v8.0
- 2) Oracle
- 3) SQL Server v6.0
- 4) SQL Windows v5.0
- 5) Sybase y Paradox, v7.0

El resultado final de este estudio es que Progress v8.0 es la mejor opción para el desarrollo de los sistemas SINAF y SINIGA. Además de las evaluaciones hechas en este documento, se tomaron en cuenta las siguientes razones :

- La UPAEP tiene como estándar el Sistema operativo Windows 95.
- Ya se había adquirido una computadora Ultra Sun que se utilizaría como servidor de la base de datos. SQL Server v6.0 corre en un servidor con Windows NT como sistema operativo. Esta adquisición se debe a que se había ya comprado una Sun para la implantación de internet.

- Progress v8.0 ofrece una herramienta para el desarrollo de aplicaciones de tipo front end (cliente). SQL Server v6.0 requería además la compra de Visual Basic para la parte del cliente.
- La razón mas importante es que la Universidad había realizado pláticas con Progress de México, por lo que la compra del producto traería beneficios extras a la Universidad.

Capítulo 4. Estudio de Desarrolladores

Dado el objetivo principal de este trabajo, que es el desarrollo de Ingeniería de Software para el Departamento de Revisión de Procesos, se presentan en este capítulo el planteamiento y desarrollo del Análisis de la capacidad de una hora de trabajo y de un día de trabajo de los desarrolladores. Estos análisis consistieron en observar qué tanto podían hacer los desarrolladores en una hora y en un día de trabajo.

4.1 Objetivo

El objetivo del estudio de desarrolladores es determinar el nivel técnico, los hábitos de programación, la productividad y el aprovechamiento de los cursos de Progress impartidos por Ingeniería de Software.

4.2 Análisis de la Capacidad de una Hora de Trabajo

4.2.1 Objetivo

El objetivo de esta actividad es el de determinar el nivel técnico de los tesisistas en un lenguaje de programación estructurado y sus hábitos de programación.

4.2.3 Desarrollo


Esta actividad consistió en elaborar un programa en un lenguaje de programación. Este programa consta de tres análisis de datos numéricos introducidos por el usuario. El planteamiento detallado del problema se presenta en la Figura 4.1.

La actividad se programó para 19 tesisistas, pero sólo se presentaron 15; para realizar este programa los 15 tesisistas contaron con una hora de tiempo máximo. Debido a la naturaleza del estudio no se podía mencionárseles el contenido de la práctica. La única información que tenían disponible era el día que tenían que presentarse: 25 de junio de 1996 en el Centro de Cómputo de Sistemas a las 10:00 horas.

Como lenguaje de programación se eligió C, ya que es un lenguaje que se imparte en la carrera de Sistemas Computacionales. El programa a realizar fue proporcionado por LCA Jaime F. Castillo Rodríguez.

Se organizaron a los tesisistas en dos grupos. A cada tesisista se le proporcionó una hoja donde se indicaba el número de máquina asignada, el directorio de trabajo donde debía quedar el programa y el programa a realizar. En la Figura 4.1 se observa dicho documento.

Figura 4.1. Planteamiento del Problema.



UNIVERSIDAD POPULAR AUTONOMA DEL ESTADO DE PUEBLA
Licenciatura en Sistemas Computacionales

ACTIVIDAD DE UNA HORA

Nombre del Alumno _____		Matrícula _____
Proyecto _____	Módulo _____	Número de Tesis _____
Número de Máquina _____	Ruta _____	Archivo _____

Un inversionista está considerando comprar acciones en la Bolsa de Valores, para esto recolecta los precios de dos tipos de acciones por 11 días consecutivos en los que estas son vendidas. Escriba un programa en 'C' para leer esta información y haga el análisis de los indicadores más adelante. Asuma que los precios de las acciones son números reales, y que los precios son introducidos por pares como a continuación se indica

Acción1 día1, Acción2 día1, Acción1 día2, Acción2 día2, Acción1 día3, Acción2 día3,

El análisis que debe hacer es el siguiente

- Determine el número de días en los que el precio de la Acción1 excede el precio de la Acción2, el número de días en los cuales los precios fueron iguales y el número de días en los que el precio de la Acción2 excedió el precio de la Acción1. Muestre en pantalla los tres valores con sus títulos apropiados
- Determine el número de días que cada Acción excedió el precio promedio. Muestre en pantalla ambos valores con sus títulos correspondientes
- Calcule los 7 promedios de movimientos de 5 días para cada Acción y muestre en pantalla esos promedios con sus respectivos encabezados. El primer promedio de movimientos es el promedio de los primeros 5 precios. El segundo promedio de movimientos es el promedio de los días 2 a 6. En general el promedio de movimientos en la posición i es el promedio de precios almacenados desde la posición i hasta la posición i+4

<p>Ejemplo</p> <p style="text-align: center;">Dados los precios</p> <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Acción 1</th> <th style="text-align: left;">Acción 2</th> </tr> </thead> <tbody> <tr><td>32 80</td><td>32 00</td></tr> <tr><td>33 10</td><td>32 65</td></tr> <tr><td>35 00</td><td>33 45</td></tr> <tr><td>38 00</td><td>35 35</td></tr> <tr><td>35 70</td><td>35 45</td></tr> <tr><td>35 70</td><td>35 70</td></tr> <tr><td>35 95</td><td>35 05</td></tr> <tr><td>34 05</td><td>34 24</td></tr> <tr><td>33 35</td><td>33 75</td></tr> <tr><td>32 85</td><td>33 05</td></tr> <tr><td>31 75</td><td>32 75</td></tr> </tbody> </table>	Acción 1	Acción 2	32 80	32 00	33 10	32 65	35 00	33 45	38 00	35 35	35 70	35 45	35 70	35 70	35 95	35 05	34 05	34 24	33 35	33 75	32 85	33 05	31 75	32 75	<p style="text-align: center;">Se debe Obtener</p> <p>Número de días en que Acción1 > Acción2 6 Número de días en que Acción1 = Acción2 1 Número de días en que Acción1 < Acción2 4</p> <p>Número de días en que Acción1 > promedio 6 Número de días en que Acción2 > promedio 6</p> <p style="text-align: center;">Promedio de los 7 movimientos de 5 días</p> <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Acción1</th> <th style="text-align: left;">Acción2</th> </tr> </thead> <tbody> <tr><td>34 52</td><td>33 78</td></tr> <tr><td>35 10</td><td>34 52</td></tr> <tr><td>35 87</td><td>35 00</td></tr> <tr><td>35 48</td><td>35 16</td></tr> <tr><td>34 95</td><td>34 84</td></tr> <tr><td>34 34</td><td>34 36</td></tr> </tbody> </table>	Acción1	Acción2	34 52	33 78	35 10	34 52	35 87	35 00	35 48	35 16	34 95	34 84	34 34	34 36
Acción 1	Acción 2																																						
32 80	32 00																																						
33 10	32 65																																						
35 00	33 45																																						
38 00	35 35																																						
35 70	35 45																																						
35 70	35 70																																						
35 95	35 05																																						
34 05	34 24																																						
33 35	33 75																																						
32 85	33 05																																						
31 75	32 75																																						
Acción1	Acción2																																						
34 52	33 78																																						
35 10	34 52																																						
35 87	35 00																																						
35 48	35 16																																						
34 95	34 84																																						
34 34	34 36																																						

Fuente: Reporte de la Capacidad de una Hora de Trabajo.

Se establecieron políticas que debían cumplirse durante el desarrollo de la práctica:

- No estaba permitido hablar entre ellos.
- Una vez terminado el tiempo disponible, se recogerían las hojas empleadas así como la del planteamiento del problema.
- Una vez que terminaran debían abandonar el salón.
- El equipo de Ingeniería no podía responder preguntas que implicaran una solución al problema. Sólo serían permitidas preguntas de tipo conceptual y técnicas.

Para almacenar los programas del primer grupo se utilizó el servidor Novell del Centro de Cómputo. Esto permitió borrar los programas del primer grupo de tesis para que no fueran utilizados por el segundo grupo.

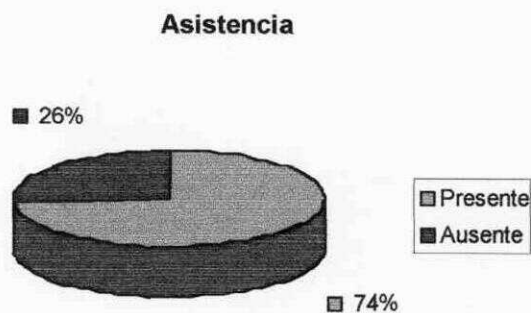
Con el segundo grupo se procedió de la misma manera y se aplicaron las mismas políticas.

4.2.3 Resultados

Las gráficas de los resultados se presentan a continuación.

- En la gráfica de la Figura 4.2 se puede observar la asistencia de los alumnos a esta actividad.

Figura 4.2. Asistencia de Tesistas.



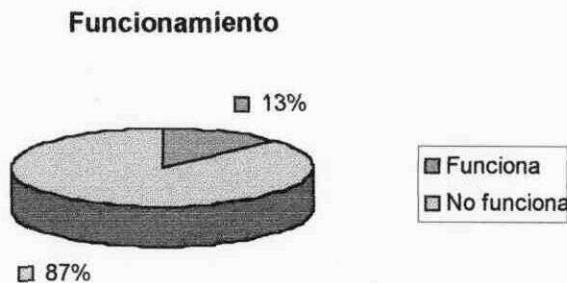
- En la gráfica de la Figura 4.3 se puede observar el porcentaje de los tesisas que intentaron realizar los diferentes análisis que plantea el problema. En este punto no se evalúa la entrega de los programas ya compilados, sino el hecho de que el código indicara que el tesisas intentó llegar a la solución.

Figura 4.3. Análisis Realizados.



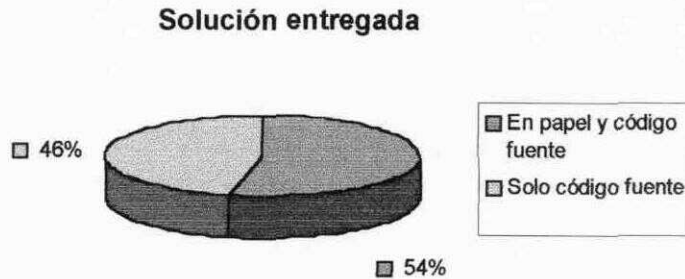
- En la gráfica de la Figura 4.4 se puede visualizar el porcentaje de programas que funcionaron. Cabe aclarar que no era necesario que el programa realizara los tres análisis; sólo bastaba que corriera. Para analizar los programas que elaboraron los tesisas, sólo se corrió el archivo ejecutable que ellos proporcionaron; en el caso de aquellos programas que sólo contaban con el archivo fuente, se utilizó el compilador de Turbo C para probarlos.

Figura 4.4. Funcionamiento.



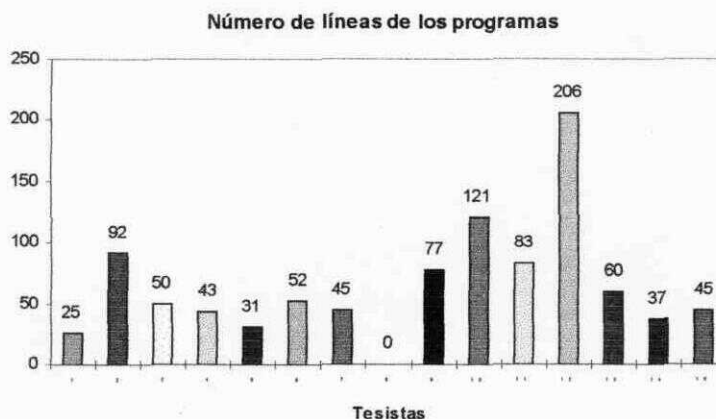
- En la gráfica de la Figura 4.5 se muestra el porcentaje de los tesisas que entregaron no sólo un archivo fuente, sino además la solución al problema en papel. Esta solución podía ir desde la utilización de pseudocódigo y diagramas hasta código en C.

Figura 4.5. Solución Entregada.



- En la gráfica de la Figura 4.6 se observa el número de líneas de código (exceptuando comentarios y encabezado) de los programas entregados. De los 15 tesisas que participaron en esta actividad, 1 no entregó ni una línea de código, ya sea en papel o en archivo.

Figura 4.6. Número de Líneas de los Programas.



- En la gráfica 4.7 se visualiza el porcentaje de tesis que entregó el archivo fuente y los que además entregaron un archivo ejecutable .

Figura 4.7. Archivos Entregados.



4.2.4 Reporte

A continuación se presenta el reporte de la Capacidad de una Hora de Trabajo. Primero se dan a conocer los puntos evaluados y a continuación la tabla comparativa.

4.2.4.1 Puntos Evaluados

Los puntos que se evaluaron de la Capacidad de una Hora de Trabajo son los siguientes :

- Asistencia.
- Alcance de análisis.
- Funcionamiento general.
- Solución entregada.
- Número de líneas de código.
- Archivos entregados.

4.2.4.2 Tabla comparativa

En la Tabla 4.1 se muestran los puntos evaluados en la actividad de una hora.

Tabla 4.1. Evaluación de la Capacidad de una Hora de Trabajo.

Nombre	Asistencia	Funciona		Análisis			Solución		No. de líneas	Archivos	
		F	NF	1	2	3	Pap	Cód		.C	.EXE
Proyecto UPAEP											
1 Campos Hernández Andrei	✓		✓					✓	25	✓	
2 Covarrubias Ponce José Antonio	✓	✓		✓	✓	✓	✓	✓	92	✓	✓
3 De Velasco Aizpuru Luis Enrique	✓		✓					✓	50	✓	
4 Díaz Gutiérrez Juan Manuel	✓		✓					✓	43	✓	✓
5 Herrera Sánchez Claudia O.	✓		✓					✓	31	✓	✓
6 Lobatón Hernandez Alejandro	✓	✓		✓	✓			✓	52	✓	✓
7 López Pérez Verónica	✓		✓					✓	45	✓	✓
8 Morales Fernández José Leopoldo	✓		✓						0	✓	
9 Navarro Cariño Daphne	x	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
10 Pérez Duarte Ma. Eugenia	✓		✓					✓	77	✓	✓
11 Robledo Ramírez Herman	✓		✓					✓	121	✓	✓
12 Xartuni Magos Bernardo	✓		✓					✓	83	✓	✓
13 Zambrano González Silverio	✓		✓					✓	206	✓	✓
Proyecto Quiz											
14 Juárez Marín Mario	x	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
15 Pérez Ahuatzi Moisés	✓		✓				✓		60	✓	✓
16 Santiago Enrque Francisco	✓		✓					✓	37	✓	✓
17 Tenahuac C. Concepción	✓		✓					✓	45	✓	
18 Uranga Díaz Aida	x	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
19 Zarrabal Santiago José Luis	x	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
19 Totales	15	2	13	2	2	1	2	13	967	15	11

- ✓ : Sí
- x : No
- n/a : No Aplica

Fuente: Reporte de la Capacidad de una Hora de Trabajo.

A continuación se explican cada uno de los puntos de la tabla anterior:

- En la columna "Asistencia", se puede observar las personas que asistieron a la actividad y cuáles no.
- En la columna "Funciona", se indica si el programa funciona (columna "F") o no (columna "NF").
- En la columna "Análisis", se indica el análisis que resolvió satisfactoriamente, de los tres que se pidieron.
- En la columna "Solución", se indica si el tesista entregó una solución en papel (columna "Pap") o en un archivo fuente (columna "Cód").

- En la columna “No. de líneas”, se registra el número de líneas de código que cada archivo fuente contiene. No se tomaron en cuenta comentarios y encabezado.
- En la columna “Archivos”, se registra la extensión de los archivos que fueron entregados por cada tesista.

4.3 Análisis de la Capacidad de un Día de Trabajo

Esta actividad en un inicio no fue planeada en dos fases, pero debido a los resultados obtenidos en la primera se decidió llevar a cabo la segunda.

4.3.1 Objetivo

El objetivo fue determinar el aprovechamiento que los tesistas tuvieron de los cursos impartidos, además de observar la cantidad de trabajo realizado por cuenta propia.

4.3.2 Fase Uno

4.3.2.1 Objetivo

El objetivo de esta fase fue el determinar la manera en que afrontan la presión de realizar programas de un día para otro.

4.3.2.2 Desarrollo

Para esta actividad se pidió a los tesistas presentarse a una junta, donde se les dijo que tenían que elaborar varios programas en Progress y entregarlos al día siguiente. Los programas requeridos fueron los siguientes.

- **Pantalla de presentación.** Esta pantalla debía servir de presentación del módulo que había realizado los programas. Como no se especificó si debía llevar un gráfico o no, se dejó a criterio de los tesistas.
- **Pantalla que pide una contraseña.** Consiste en una pantalla que pide la contraseña de acceso a las demás opciones. En caso de no ingresar la contraseña correcta el programa debía terminar.
- **Un menú de iconos.** Consiste en una pantalla con varios botones con iconos. Cada botón representa una opción que manda a los demás programas.

- **Un programa de Altas, Bajas y Cambios (ABC).** Esta pantalla debía permitir realizar altas, bajas y cambios de los registros de una tabla de la base de datos.
- **Un programa de consultas.** Consiste en una pantalla que permita realizar consultas de los registros de la base de datos.
- **Un reporte.** Consiste en una pantalla que muestra un reporte de los datos de una tabla y permita al usuario imprimirlo.
- **Un programa de “Acerca de...”.** Consiste en una pantalla que muestra los nombres de los integrantes y el nombre del módulo al que pertenecían

Se les dio un día para la realización de los programas. Ingeniería se encargó de evaluarlos así como de emitir un reporte con las observaciones hechas.

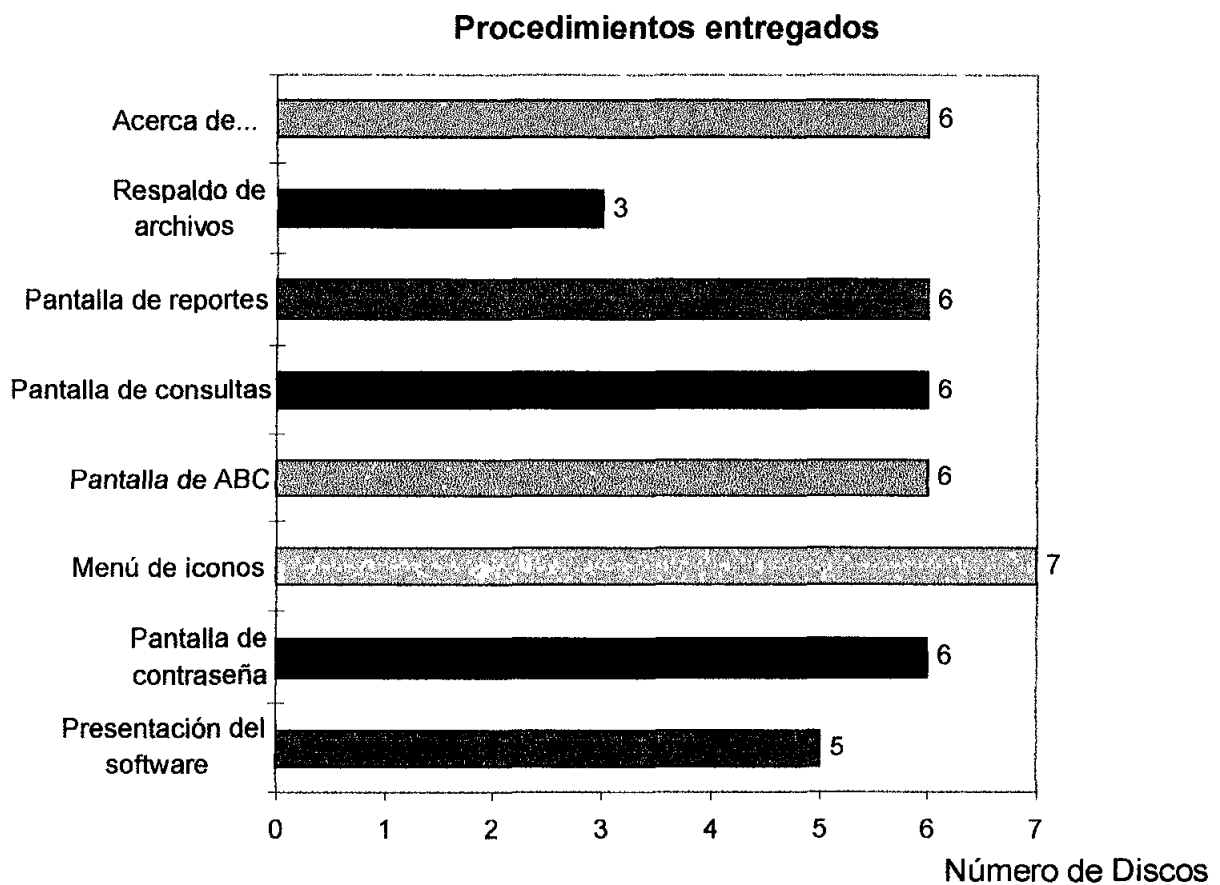
4.3.2.3 Resultados

Para el control de los programas que se entregaron para su evaluación se decidió agruparlos en grupos denominados disquetes. Esto fue en parte porque la distribución de los tesis no es uniforme, es decir, no están distribuidos en equipos con el mismo número de integrantes, además para no usar el nombre de los participantes en la evaluación.

A partir de los resultados que se obtuvieron de la evaluación de los programas (para más detalle de esta evaluación consultar la sección 4.3.2.4 titulada Reporte), se elaboraron algunas gráficas; dichas gráficas se presentan a continuación.

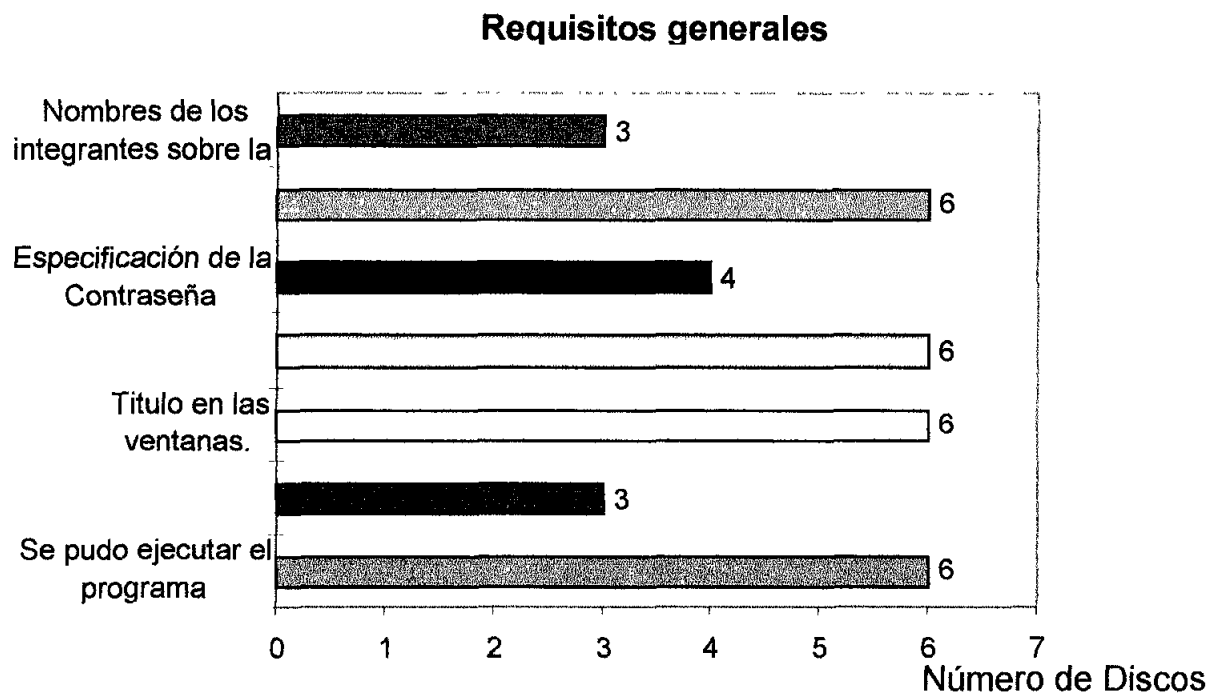
- En la gráfica de la figura 4.8, se puede observar el número de procedimientos que los tesistas entregaron. Se hace notar que la mayoría de los tesistas entregaron seis de los siete programas que se les pidieron, solamente un tesista entregó tres y además solamente uno entregó todos.

Figura 4.8. Procedimientos Entregados.



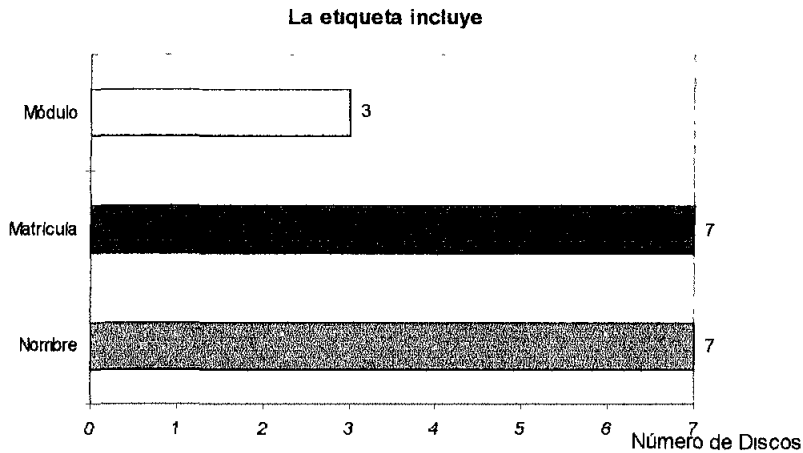
- Para estos programas se les pidió a los tesisistas que los programas cumplieran con un cierto número de requisitos. Dichos requisitos y su cumplimiento se puede observar en la Figura 4.9. Se hace notar que ninguno de los disquetes entregados cumplió con todos los requisitos y que solamente tres de los disquetes que se entregaron deshabilitaban las pantallas que no se estaban en uso.

Figura 4.9. Requisitos Generales.



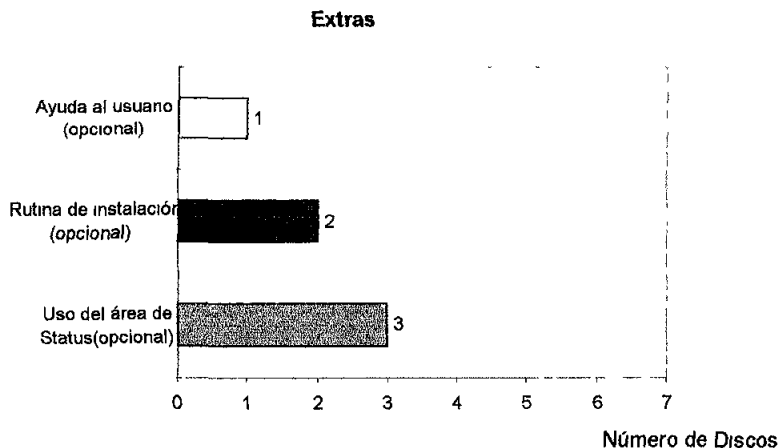
- En la Figura 4.10 se puede observar qué tanto cumplieron con los datos que debían ingresar en la etiqueta de sus discos. Menos de la mitad cumplió con colocar el nombre de su módulo en la etiqueta, aunque todos colocaron su matrícula y su nombre.

Figura 4.10. Etiqueta.



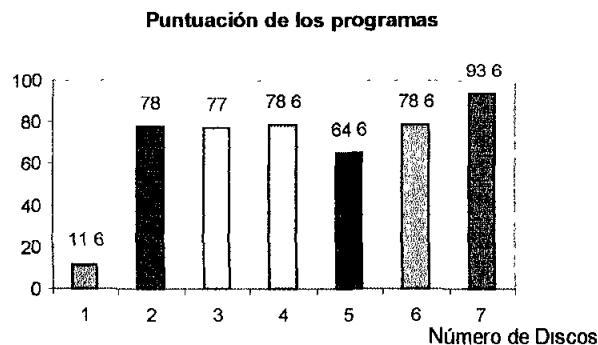
- En la figura 4.11 se pueden observar cuántos de los disquetes que se entregaron. Incluyeron algunos de los puntos extras que se evaluaron, y como se puede observar, casi no incluyeron estas características en sus programas.

Figura 4.11. Extras.



- La gráfica 4.12 muestra la puntuación de los programas que se entregaron para su evaluación. Como se puede observar, seis de los siete grupos de programas obtuvieron una puntuación que osciló entre 64.6 y 78.6 puntos. La puntuación más baja fue de 11.6 puntos y la más alta de 93.6 puntos. Como se puede observar en esta gráfica, la mayoría alcanzó una puntuación bastante aceptable.

Figura 4.12. Puntuación de los Programas.



4.3.2.4 Reporte

Como se menciona en la sección anterior, para un mejor control de la evaluación, se utilizó el número de disco en lugar de los nombres de los tesistas. A continuación se lista cada punto evaluado de los programas entregados.

“Procedimientos entregados”. Se listan a continuación los programas que se pidieron a los tesistas :

- Presentación del software
- Pantalla de Contraseña
- Menú de iconos
- Pantalla de ABC
- Pantalla de consultas
- Pantalla de reportes
- Respaldo de archivos
- Acerca de...

“Requisitos generales”. Los siguientes son los requisitos mínimos que deben cumplir los programas entregados :

- Se pudo ejecutar el programa
- Deshabilita las pantallas que no están en uso
- Título en las ventanas

- Mensajes al usuario (de error, de avisos)
- Nombres de los integrantes sobre la etiqueta del disco
- Nombre del archivo principal
- Nombre del directorio de trabajo
- Especificación del Contraseña
- Procedimientos integrados

“La etiqueta incluye”. Se listan los puntos que deben tener la etiqueta del disco, los puntos que se evaluaron son:

- Nombre
- *Matrícula*
- Módulo

“Extras” : Son características que no fueron obligatorias pero que ayudaron a mejorar el aspecto del programa entregado. Los puntos que se evaluaron son:

- Uso del área de Status
- Rutina de instalación
- Ayuda al usuario


El reporte quedó dividido de la siguiente manera:

- Lista de los programas entregados
- Puntos evaluados en la tabla general
- Tabla General
- Puntos evaluados en la tabla general de programas y en la tabla de detalles de programas.
- Evaluación del disco 1
- Evaluación del disco 2
- Evaluación del disco 3
- Evaluación del disco 4
- Evaluación del disco 5
- Evaluación del disco 6
- Evaluación del disco 7

En las siguientes páginas se presenta el reporte con la evaluación de la Fase Uno.

En la Figura 4.13 se presenta la lista de los tesisistas que entregaron los programas para su evaluación, y el número de disco que se les fue asignado, este número de disco fue utilizado a lo largo del reporte.

Figura 4.13. Lista de Programas Entregados.



UNIVERSIDAD POPULAR AUTONOMA DEL ESTADO DE PUEBLA
Licenciatura en Sistemas Computacionales

EVALUACION DE PROGRAMAS

Lista de Programas Entregados

A continuación se listan los programas que fueron entregados a Ingeniería de Software para su evaluación

- Disco 1 SINAF Contabilidad y Presupuestos, Integrantes .
 Robledo Ramírez Herman
 Xartuni Magos Bernardo
 Zambrano González Silverio

- Disco 2 : SINAF, Cuentas por Cobrar, Integrantes
 De Velasco Aizpuru Luis Enrique
 Díaz Gutiérrez Juan Manuel

- Disco 3 . SINAF, Abastecimiento y Almacén, Integrantes .
 Navarro Cariffo Daphne
 Herrera Sánchez Claudia O

- Disco 4 : SINIGA Biblioteca, Integrantes .
 López Pérez Verónica
 Pérez Duarte Ma Eugenia

- Disco 5 SINIGA Servicio Social, Integrantes .
 Morales Fernández José Leopoldo


- Disco 6 : SINIGA Becas y Exalumnos, Integrantes
 Lobatón Hernandez Alejandro

- Disco 7 . SINAF, Activos Fijos , Integrantes .
 Campos Hernández Andrei
 Covarrubias Ponce José Antonio

Fuente: Reporte de la Capacidad de un Día de Trabajo (Fase Uno).

En la Figura 4.14 se observa la hoja que acompañó al reporte para que los tesisas estuvieran enterados de los puntos que se evaluaron y que se presentan en la Tabla General.

Figura 4.14. Puntos Evaluados en la Tabla General de la Fase Uno.



UNIVERSIDAD POPULAR AUTONOMA DEL ESTADO DE PUEBLA
Licenciatura en Sistemas Computacionales

EVALUACION DE PROGRAMAS

Puntos Evaluados en la Tabla General

Procedimientos Entregados

- Se refiere a cuáles de los programas fueron entregados por los tesisas para su evaluación

Requisitos Generales

- **Se pudo ejecutar el programa:** Se refiere a que si se pudieron correr todos los programas que fueron entregados por los tesisas
- **Deshabilita las pantallas que no están en uso:** Se evalúa si el programador deshabilita u oculta una ventana cuando ésta ha llamado a otra ventana o a un cuadro de dialogo
- **Título en las ventanas:** Se evalúa que por lo menos a alguna ventana se le haya cambiado el título, obviamente es necesario darles título a todas las ventanas pero para esta aplicación no se exigirá así
- **Mensajes al usuario:** Se evalúa que por lo menos cuando se pide el Password, se le mande un mensaje al usuario avisándole que ha ingresado la clave incorrecta
- **Especificación de la contraseña:** Se evalúa si se incluye la contraseña del módulo
- **Procedimientos Integrados:** Se evalúa que los programas se ejecuten desde un programa principal o por separado
- **Nombres de los integrantes en la etiqueta del disco:** Se evalúa si la información aparece escrita a mano o con impresora
- **Nombre del archivo principal:** Se evalúa si se incluye el nombre del archivo principal Algunos lo especificaron en el disquete o en el programa de instalación, sin embargo se tomó en cuenta para la evaluación
- **Nombre del directorio de trabajo:** Se evalúa si se utilizó un directorio de trabajo para la aplicación

La etiqueta incluye

- **Nombre:** Se refiere a que si los tesisas incluyeron su nombre en la etiqueta del disco
- **Matrícula:** Se refiere a que si los tesisas incluyeron la matrícula en la etiqueta del disco
- **Módulo:** Se refiere a que si los tesisas incluyeron su módulo en la etiqueta del disco

Extras

- **Uso del área de Status (opcional):** Este punto no es obligatorio, pero es bueno si por lo menos en alguna ventana se utilizó esta área
- **Rutina de instalación (opcional):** Este punto no es obligatorio, ya que no se especificó en ningún momento que la aplicación pudiera tener una rutina para instalarse, sin embargo ayuda a mejorar la puntuación
- **Ayuda al usuario (opcional):** Al igual que los dos puntos anteriores, este punto no es obligatorio Indica si se proporciona algún de ayuda al usuario como mensajes de alerta, de información, etc

Fuente: Reporte de la Capacidad de un Día de Trabajo (Fase Uno).

En la Figura 4.15 se observa la Tabla General, la cual agrupa las evaluaciones de los siete disquetes, en esta tabla se presentan todos los puntos evaluados sin la valoración que alcanzaron los programas entregados.


Figura 4.15. Tabla General de la Fase Uno.

UNIVERSIDAD POPULAR AUTONOMA DEL ESTADO DE PUEBLA							
Licenciatura en Sistemas Computacionales							
EVALUACION DE PROGRAMAS							
Tabla General							
Puntos a evaluar	Disco 1	Disco 2	Disco 3	Disco 4	Disco 5	Disco 6	Disco 7
Procedimientos entregados							
Presentación del software	n/a	✓	x	✓	✓	✓	✓
Pantalla de password	n/a	✓	✓	✓	✓	✓	✓
Menú de iconos	✓	✓	✓	✓	✓	✓	✓
Pantalla de ABC	n/a	✓	✓	✓	✓	✓	✓
Pantalla de consultas	n/a	✓	✓	✓	✓	✓	✓
Pantalla de reportes	n/a	✓	✓	✓	✓	✓	✓
Respaldo de archivos	n/a	x	x	✓	x	x	✓
Acerca de	n/a	✓	✓	✓	✓	✓	✓
Requisitos generales							
Se pudo ejecutar el programa	x	✓	✓	✓	✓	✓	✓
Deshabilita las pantallas que no están en uso	n/a	x	x	✓	✓	✓	x
Título en las ventanas	n/a	✓	✓	✓	✓	✓	✓
Mensajes al usuario (de error, de avisos)	n/a	✓	✓	✓	✓	✓	✓
Especificación de la Contraseña	n/a	✓	✓	✓	✓	x	✓
Procedimientos Integrados	n/a	✓	✓	✓	x	✓	✓
Nombres de los integrantes sobre la etiqueta	A mano	A mano	Impresos	Impresos	A mano	Impresos	A mano
Nombre del archivo principal	ALTAS W	PRESENTA W	EMPEZAR W	No especificado	Wn/aPASS W	No especificado	ACT FIJO W
Nombre del Directorio de trabajo	No especificado	C\VAL	C\AGENDA	C\IPROG	C\IPOLO	C\PROGWORK	C\ACTFIJO
La etiqueta incluye							
Nombre	✓	✓	✓	✓	✓	✓	✓
Matrícula	✓	✓	✓	✓	✓	✓	✓
Módulo	x	✓	✓	x	x	x	✓
Extras							
Uso del área de Status (opcional)	n/a	✓	✓	x	x	✓	x
Rutina de instalación (opcional)	n/a	x	✓	✓	x	x	x
Ayuda al usuario (opcional)	n/a	x	✓	x	x	x	x

Fuente: Reporte de la Capacidad de un Día de Trabajo (Fase Uno).

En la Figura 4.16 se pueden apreciar los puntos evaluados en las tablas generales y de detalle, así como la máxima calificación que se podía obtener en cada punto que se evaluó.

Figura 4.16. Puntos Evaluados en las Tablas por Disco.



UNIVERSIDAD POPULAR AUTONOMA DEL ESTADO DE PUEBLA
Licenciatura en Sistemas Computacionales

EVALUACION DE PROGRAMAS

Puntos evaluados en la Tabla General De Programas

Los puntos que se evalúan en esta tabla, son los mismos que se utilizan en la tabla general, por lo que ya no se vuelven a definir en esta sección

Puntos evaluados	Disco 1	Puntuación
Puntuación en esta tabla		
- Nombre	SI / NO	3
- Matricula	SI / NO	3
- Módulo	SI / NO	3
Nombre de los integrantes sobre la etiqueta del disco	Impreso / A mano	1
Nombre del archivo principal	Nombre Archivo / No especificado	1
Nombre del Directorio de trabajo	Nombre del Directorio / No especificado	1
Especificación del Contraseña	Si especificado / No especificado	1
Procedimientos Integrados	SI / NO	1
Problemas para ejecutar el programa o en los programas	Si corrió / No corrió	1
Deshabilita las pantallas que no están en uso	SI / NO	1
Título en las ventanas	SI / NO	1
Mensajes al usuario	SI / NO	1
Suma		10

Puntos evaluados en la Tabla de Detalles de los Programas

- **Puntuación** En esta columna se coloca la puntuación que cada programa alcanza según las demás columnas
- **Entregado** . En esta columna se especificó si este programa fue entregado en el disquete
- **Teclas rápidas** . En esta columna se especifica si utiliza teclas rápidas para hacer mas amigable el ambiente para el usuario
- **Uso de Templates** : En esta columna se especifica si los programas que nos presentaron fueron contruidos con los templates de Progress
- **Ayuda** . En esta columna se especifica si se incluyó ayuda para el usuario
- **Comentarios** En esta columna se escriben los comentarios que Ingeniería de Software hace respecto al programa que se revisa


Puntos a Evaluar	Puntuación	Entregado	Teclas rápidas	Uso de Templates	Ayuda	Comentarios
Presentación del software	4	SI / NO	SI / NO	SI / NO	SI / NO	
Pantalla del Contraseña	10	SI / NO	SI / NO	SI / NO	SI / NO	
Módulo de sesiones	10	SI / NO	SI / NO	SI / NO	SI / NO	
Pantalla de ABC	14	SI / NO	SI / NO	SI / NO	SI / NO	
Pantalla de Consultas	14	SI / NO	SI / NO	SI / NO	SI / NO	
Pantalla de Reportes	14	SI / NO	SI / NO	SI / NO	SI / NO	
Respaldo de archivos	14	SI / NO	SI / NO	SI / NO	SI / NO	
Acercas de	10	SI / NO	SI / NO	SI / NO	SI / NO	
Suma	90					

Fuente: Reporte de la Capacidad de un Día de Trabajo (Fase Uno).

A partir de la Figura 4.17 a la Figura 4.23 se muestran las hojas con las evaluaciones detalladas por cada disco, además de incluir comentarios y observaciones y la valoración que cada punto obtuvo.

En la Figura 4.17 se puede visualizar la evaluación del disco 1.

Figura 4.17. Evaluación del Disco 1.



UNIVERSIDAD POPULAR AUTONOMA DEL ESTADO DE PUEBLA
Licenciatura en Sistemas Computacionales

EVALUACION DE PROGRAMAS

No DE TESIS	FECHA	HOJA	DE
DESCRIPCIÓN DE LA TESIS			
MATRICULA	NOMBRE	No LISTA	GRUPO
200481	Herman Robledo		
200528	Bernardo Xaruni		
200483	Silvia Zembrano		

Tabla General de Programas del Disco 1

Puntos evaluados	Disco 1	Puntuación
<u>Puntos incluye</u>		
Nombre	SI	3
Matricula	SI	3
Módulo	NO	0
Nombres de los integrantes en la etiqueta del disco	A usuario	0
Nombre del archivo principal	AL-TAS.W	1
Nombre del directorio de trabajo	No especificado	0
Especificación de la contraseña	No especificado	0
Procedimientos integrados	-	0
Problemas para ejecutar el o los programas	No corrió	0
Deshabilita las pantallas que no están en uso	-	0
Título en las ventanas	-	0
Mensajes al usuario	-	0
Suma		16

Tabla de Detalles de los Programas del Disco 1

Puntos a Evaluar	Puntuación	Entregado	Teclas Rápidas	Uso de Templates	Ayuda	Comentarios
Presentación del Programa	0					
Pantalla de Contraseña	0					
Menú de iconos	10	SI			NO	
Pantalla de ABC	0					
Pantalla de Consultas	0					
Pantalla de Reportes	0					
Respaldo de archivos	0					
Accesa de	0					
Suma	10					

Total de Puntos acumulados 11.6


Observaciones:

- Se encontró un error en el nombre de un campo de la base de datos. En el programa se llamaba de otra manera.
- Program indica un error en el tipo de dato. Se corrigieron 5 de estos errores.
- El primer icono despliega dentro de la misma pantalla otros objetos. Sin embargo estos no hacen nada.
- Los otros dos iconos seleccionaban archivos de la unidad A.

Fuente: Reporte de la Capacidad de un Día de Trabajo (Fase Uno).

En la Figura 4.18 se puede visualizar la evaluación del disco 2.

Figura 4.18. Evaluación del Disco 2.



UNIVERSIDAD POPULAR AUTONOMA DEL ESTADO DE PUEBLA
Licenciatura en Sistemas Computacionales

EVALUACION DE PROGRAMAS

No. DE FLSIS	FECHA	HOJA	DE
DESCRIPCIÓN DE LA FLSIS		Módulo de Cuentas por Cobrar	
MATRICULA	NOMBRE	NO. LISTA	GRUPO
200491	Luis de Velasco		8° "B"
200521	Manuel Diaz		8° "B"

Tabla General De Programas, del Disco 2

Puntos evaluados	Disco 1	Puntuación
<i>Etiqueta incluye,</i>		
- Nombre	SI	3
- Matricula	SI	3
- Módulo	SI	3
Nombre de los integrantes sobre la etiqueta del disco	A mano	0
Nombre del archivo principal	PRESENTA W	1
Nombre del Directorio de trabajo	CVA\	1
Especificación del Password	SI	1
Procedimientos integrados	SI	1
Problemas para ejecutar el programa o en los programas	Algunos	0
Deshabilita las pantallas que no están en uso	NO	0
Título en las ventanas	SI	1
Mensajes al usuario	SI	1
Suma		7

Tabla de Detalles de los Programas del Disco 2

Puntos a Evaluar	Puntuación	Entregado	Letras fijadas	Uso de Templates	Ayuda	Comentarios
Presentación del software	4	SI	-	NO	-	
Pantalla del Password	10	SI	-	NO	NO	
Menú de Iconos	10	SI	NO	SI	NO	
Pantalla de ABC	10	SI	SI	SI	NO	<ul style="list-style-type: none"> • Tiene una barra de desplazamiento horizontal, que no tiene un uso aparente • La opción de actualizar despliega una pantalla que no tiene un botón para finalizar o regresar • Si se crea un registro, y después se oprime actualizar o guardar (sin ingresar datos), Programa despliega varios avisos <ul style="list-style-type: none"> • Combina palabras en español y en inglés • Permite al usuario escribir sobre los widgets
<i>Pantalla de Consultas</i>	13	SI	NO	NO	NO	
<i>Pantalla de Reportes</i>	14	SI	SI	NO	NO	
<i>Respaldo de archivos</i>	0	NO	-	NO	-	
<i>Acerca de</i>	10	SI	-	NO	-	
Suma	71					

Total de Puntos acumulados: 78


Problemas generales para correr el programa

- No se detectaron problemas

Fuente: Reporte de la Capacidad de un Día de Trabajo (Fase Uno).

En la Figura 4.19 se puede visualizar la evaluación del disco 3.

Figura 4.19. Evaluación del Disco 3.



UNIVERSIDAD POPULAR AUTONOMA DEL ESTADO DE PUEBLA
Licenciatura en Sistemas Computacionales

EVALUACION DE PROGRAMAS

No DE TESIS	IT CIA	HOJA	DE
DESCRIPCIÓN DE LA TESIS	Abastecimiento y Almacén		
MA TRICULA	NOMBRE	Ne LISIA	GRUPO
200511	Claudia O Herrera Sanchez		8º
200495	Daphne J Navarro Cantfo		8º

Tabla General De Programas, del Disco 3

Puntos evaluados	Disco 1	Puntuación
Etiqueta intlige		
Nombre	SI	5
Manuales	SI	3
- Módulo	SI	3
Nombre de los integrantes sobre la etiqueta del disco	Impresos	1
Nombre del archivo principal	IMPZAR W	1
Nombre del Directorio de trabajo	C:\AGENDA	1
Especificación del Password	SI	1
Procedimientos Integrados	SI	1
Problemas para ejecutar el programa o en los programas	Aparios	0
Deshabilita las pantallas que no están en uso	NO	0
Título en las ventanas	SI	1
Mensajes al usuario	SI	1
Suma		8

Tabla de Detalles de los Programas del Disco 3

Puntos a Evaluar	Puntuación	Pregado	Teclas rápidas	Uso de Templates	Ayuda	Comentarios
Presentación del software	4	NO			-	
Pantalla del Password	10	SI		NO	NO	• En el área de Status, colocaron una etiqueta, el texto tiene errores
Menú de iconos	9	SI	NO	SI	NO	• Combina palabras en español y en inglés • Si se oprime la tecla "B" se termina el programa.
Pantalla de ABC	10	SI	NO	SI	SI	• La opción de modificar no tiene un botón para regresar a la ventana de ABC • Si se crea un registro, y después se oprime modificar o guardar (sin ingresar datos), Progres despliega varios avisos • La opción de modificar despliega una pantalla que no tiene un botón para finalizar o regresar
Pantalla de Consultas	13	SI	NO	NO	NO	• Combina palabras en español y en inglés.
Pantalla de Reportes	13	SI	NO	NO	NO	• Combina palabras en español y en inglés.
Respaldo de archivos	0	NO		NO		• Combina palabras en español y en inglés.
Accesa de	10	SI		NO	-	
Suma	69					

Totol de Puntos acumulados 77

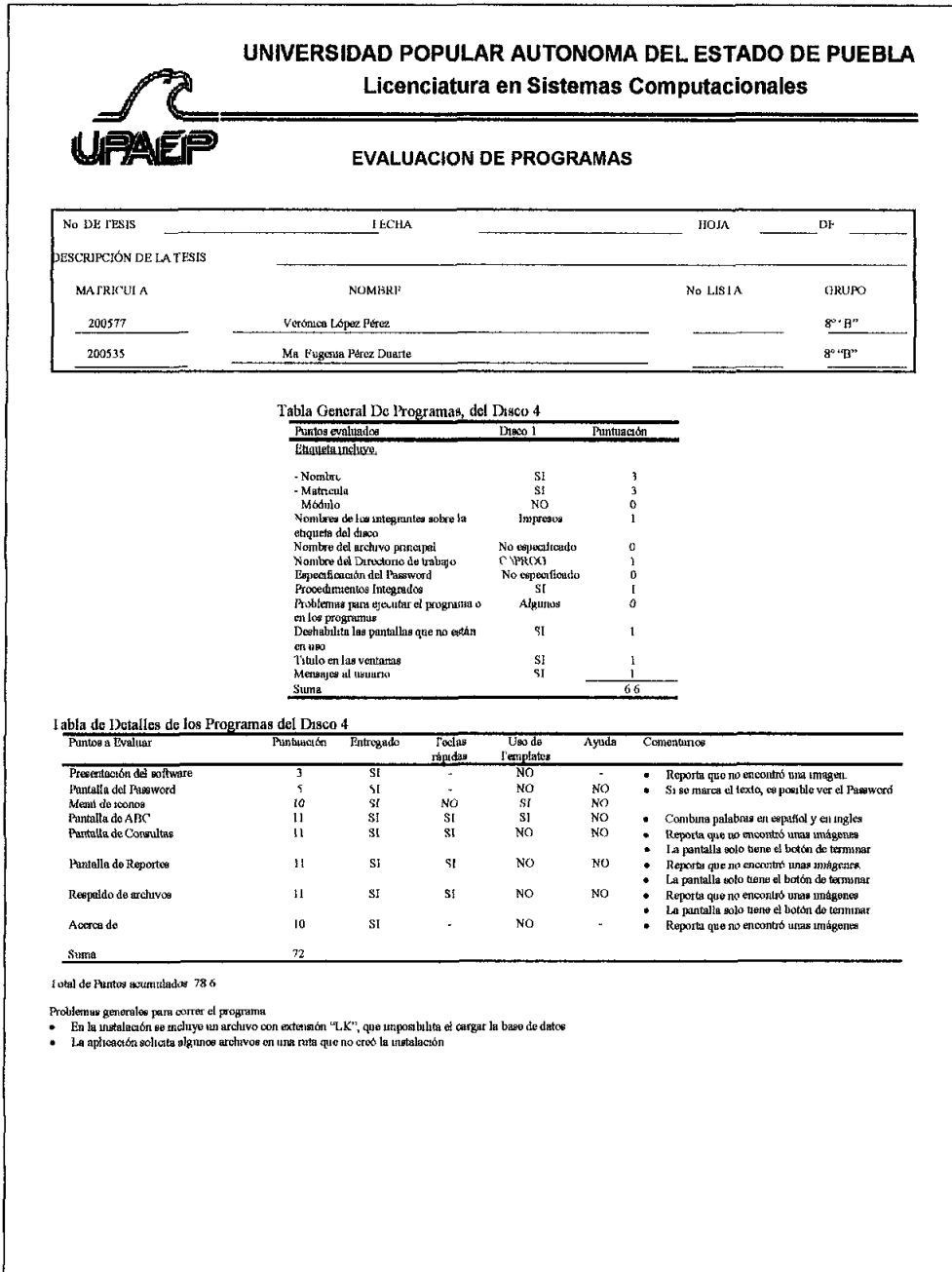
Problemas generales para correr el programa

- Los comandos de la instalación fallaron

Fuente: Reporte de la Capacidad de un Día de Trabajo (Fase Uno).

En la Figura 4.20 se puede visualizar la evaluación del disco 4.


Figura 4.20. Evaluación del Disco 4.



Fuente: Reporte de la Capacidad de un Día de Trabajo (Fase Uno).

En la Figura 4.21 se puede visualizar la evaluación del disco 5.

Figura 4.21. Evaluación del Disco 5.



UNIVERSIDAD POPULAR AUTONOMA DEL ESTADO DE PUEBLA
Licenciatura en Sistemas Computacionales

EVALUACION DE PROGRAMAS

No DE ILSIS	FECHA	HOJA	DE
DESCRIPCIÓN DE LA ILSIS			
MATRICULA	NOMBRE	No LISTA	GRUPO
200504	Polo Morales		8° B"

Tabla General De Programas, del Disco 5

Puntos evaluados	Disco 1	Puntuación
<i>Etiqueta incluye:</i>		
Nombre	SI	3
Matricula	SI	3
Módulo	NO	0
Nombre de los integrantes sobre la etiqueta del disco	A mano	0
Nombre del archivo principal	W-PASS W	1
Nombre del Directorio de trabajo	C:\POLO	1
Ejemplificación del Password	SI	1
Procedimientos fitegrados	SI	1
Problemas para ejecutar el programa o en los programas	No se encuentran	1
Deshabilita las pantallas que no están en uso	SI	1
Titulo en las ventanas	SI	1
Mensajes al usuario	SI	1
Suma		8 6

Tabla de Detalles de los Programas del Disco 5

Puntos a Evaluar	Puntuación	Entregado	Teclas rápidas	Uso de templates	Ayuda	Comentarios
Presentación del software	4	SI	-	NO	-	
Pantalla del Password	5	SI	-	NO	NO	• Si se marca el texto, es posible ver el Password
Menú de comos	10	SI	NO	NO	NO	
Pantalla de ABC	13	SI	SI	SI	NO	• Combina palabras en español y en inglés
Pantalla de Consultas	14	SI	SI	NO	NO	
Pantalla de Reportes	0	SI	-	-	-	• No hace nada después de que se hace clic sobre este objeto
Respaldo de archivos	0	NO	-	-	-	
Acorde o	10	SI	-	NO	-	
Suma	56					

Total de Puntos acumulados 64 6


Problemas generales para correr el programa

- No se detectaron problemas

Fuente: Reporte de la Capacidad de un Día de Trabajo (Fase Uno).

En la Figura 4.22 se puede visualizar la evaluación del disco 6.

Figura 4.22. Evaluación del Disco 6.



UNIVERSIDAD POPULAR AUTONOMA DEL ESTADO DE PUEBLA
Licenciatura en Sistemas Computacionales

EVALUACION DE PROGRAMAS

No DE IESIS	FECHA	HOJA	DE
DESCRIPCIÓN DE LA FESIS			
MAI RICULA	NOMBRE	No LISIA	GRUPO
200495	Alejandro Lobato Hernández		8° "B"

Tabla General De Programas, del Disco 6

Puntos evaluados	Disco 1	Puntuación
Etiqueta incluye:		
- Nombre	SI	3
- Matricula	SI	3
- Módulo	NO	0
Nombres de los integrantes sobre la etiqueta del disco	Impresos	1
Nombre del archivo principal	No especificado	0
Nombre del Directorio de trabajo	C:\PROGWOR K	1
Especificación del Password	NO	0
Procedimientos integrados	SI	1
Problemas para ejecutar el programa o en los programas	Algunos	0
Deshabilita las pantallas que no están en uso	SI	1
Título en las ventanas.	SI	1
Mensajes al usuario	SI	1
Suma		66

Tabla de Detalles de los Programas del Disco 6

Puntos a Evaluar	Puntuación	Entregado	1 o las réplicas	Uso de templates	Ayuda	Comentarios
Presentación del software	4	SI	-	NO	-	
Pantalla del Password	10	SI	-	NO	NO	
Menú de iconos	10	SI	NO	NO	NO	
Pantalla de ABC	14	SI	SI	SI	NO	• Los cambios para las misterias, no es tan buena como el de la tabla alumnos
Pantalla de Consultas	14	SI	SI	SI	NO	
Pantalla de Reportes	10	SI	SI	SI	NO	
Respaldo de archivos	0	NO	-	-	-	
Acceso de	10	SI	-	NO	-	
Suma	72					

Todal de Puntos acumulados 78 6

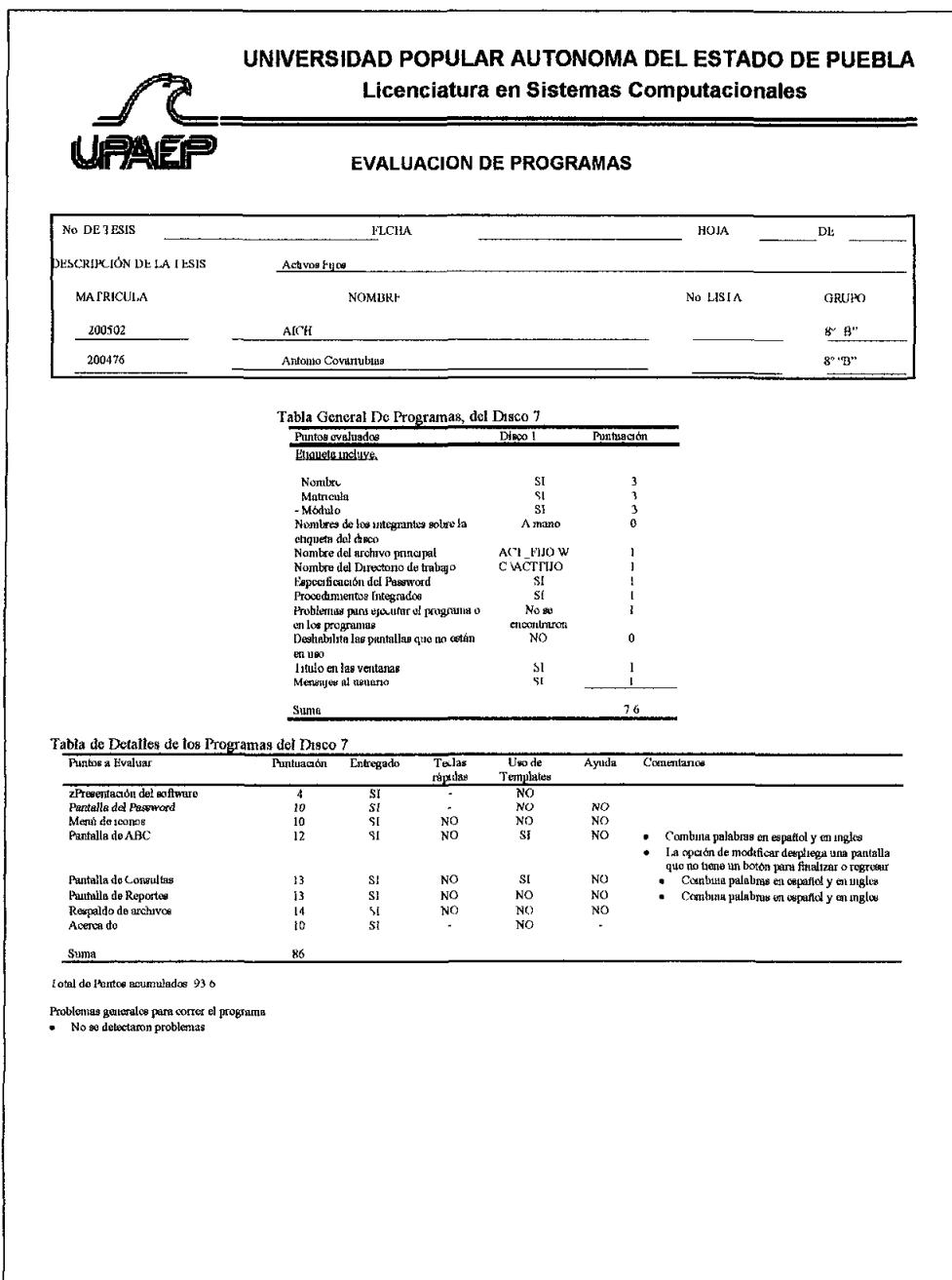
Problemas generales para correr el programa

- Se escribió mal una palabra reservada, y no permitía correr el programa

Fuente: Reporte de la Capacidad de un Día de Trabajo (Fase Uno).

En la Figura 4.23 se puede visualizar la evaluación del disco 7.

Figura 4.23. Evaluación del Disco 7.



Fuente: Reporte de la Capacidad de un Día de Trabajo (Fase Uno).

4.3.3 Fase Dos

4.3.3.1 Objetivo

El objetivo de esta fase fue determinar qué tan perceptivos podrían ser para hacer correcciones sobre los programas que entregaron en la Fase Uno.

4.3.3.2 Desarrollo

Para esta segunda fase se les entregó el reporte de la Fase Uno (sección 4.3.2.4) que especifica los puntos evaluados y los problemas que se detectaron. A continuación se les pidió que estos problemas fueran corregidos, dándoles cuatro días más para entregar las correcciones.

Además de las correcciones, se les pidió que hicieran algunos programas extras a la Fase Uno; estos consistían en hacer dos tipos de consultas, poner cortes de control y ordenar los datos en los reportes. Se determinó pedir estas características extras después de un curso rápido impartido al terminar la entrega de los programas de la Fase Uno. En ese curso se enseñaron algunas características avanzadas del Report Builder y otros conceptos de los SmartObjects.

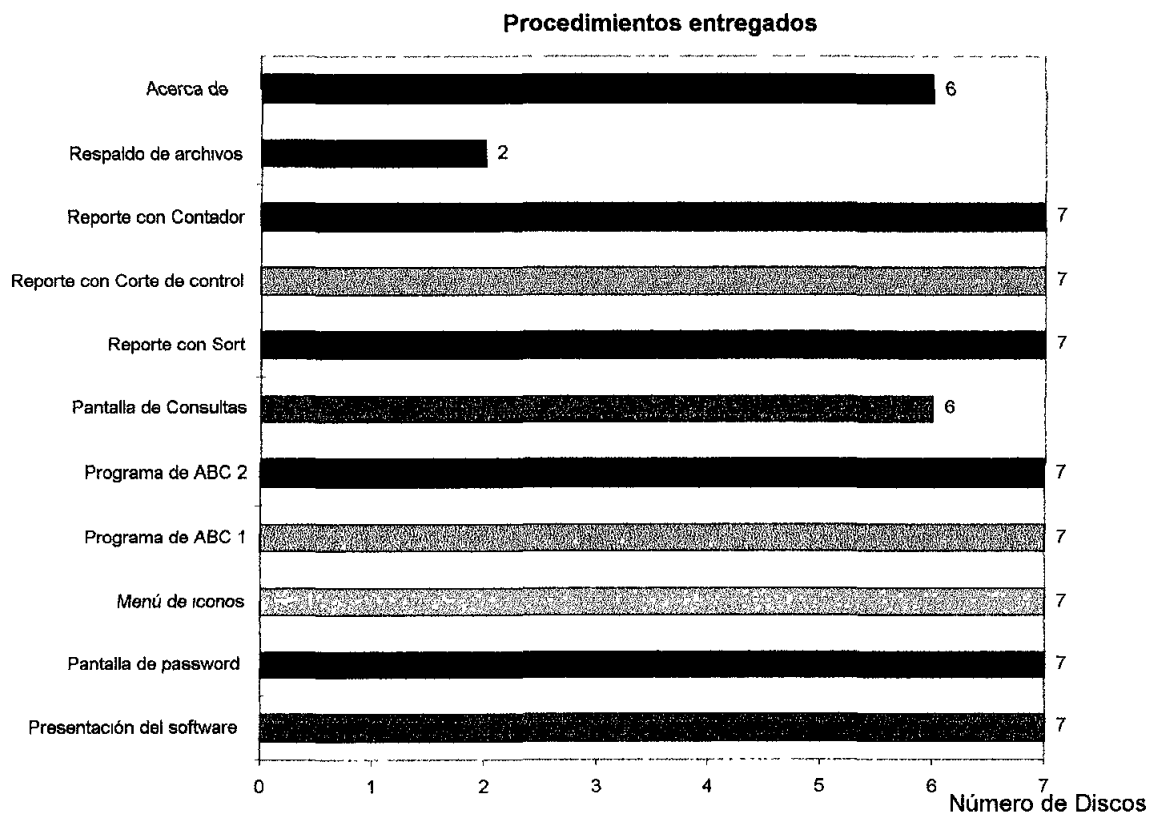
4.3.3.3 Resultados

Fueron varios los problemas que se detectaron en la Fase Uno. Para la segunda Fase, se detectó que algunos de los problemas que se les había pedido que resolvieran, no fueron resueltos de manera satisfactoria. Aparte de las correcciones que se les indicaron, hubo varios aspectos dentro de los programas que podían ser mejorados; con esto se refiere a la consistencia entre las pantallas, el diseño, la distribución y la funcionalidad de estas.

Al igual que en la Fase Uno, a partir de los resultados que se obtuvieron de la evaluación de los programas se elaboraron algunas gráficas, las cuales se presentan a continuación.

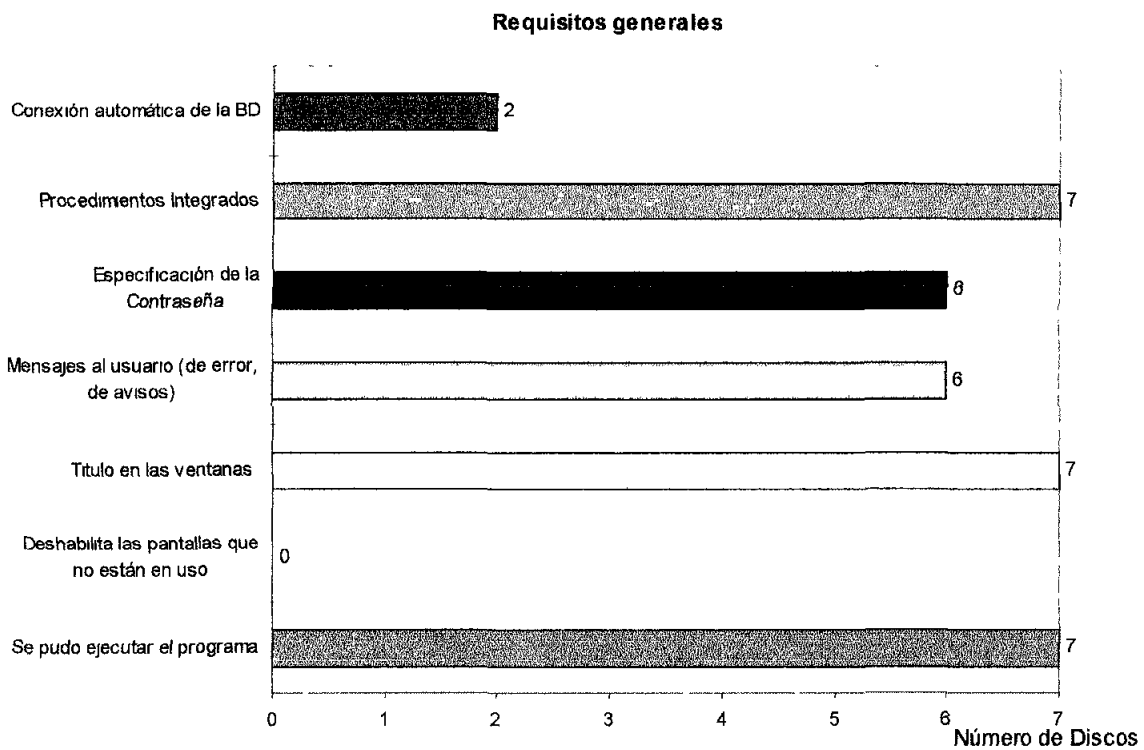
- La gran mayoría de los programas que se pidieron fueron entregados; esto se puede visualizar en la gráfica de la Figura 4.24.

Figura 4.24. Procedimientos Entregados.



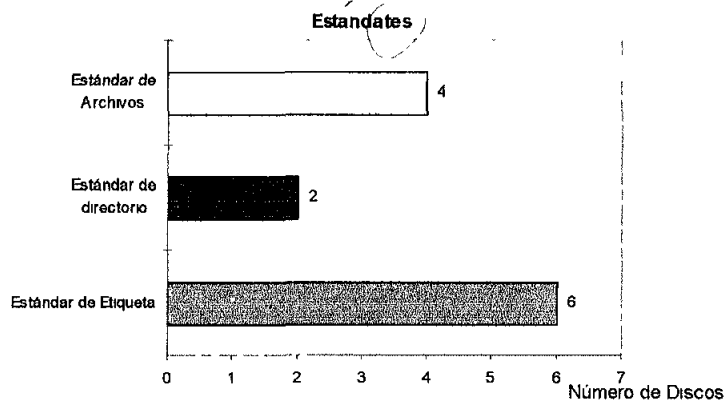
- Pese a que la gran mayoría de los programas fueron entregados, esto no significa que cumplieran con los requisitos generales que se les pidieron. Esto se visualiza en la gráfica de la Figura 4.25, de lo que se puede decir que cuatro de siete puntos fueron cubiertos de manera satisfactoria, pero los restantes puntos no fueron cubiertos de manera satisfactoria.

Figura 4.25. Requisitos Generales.



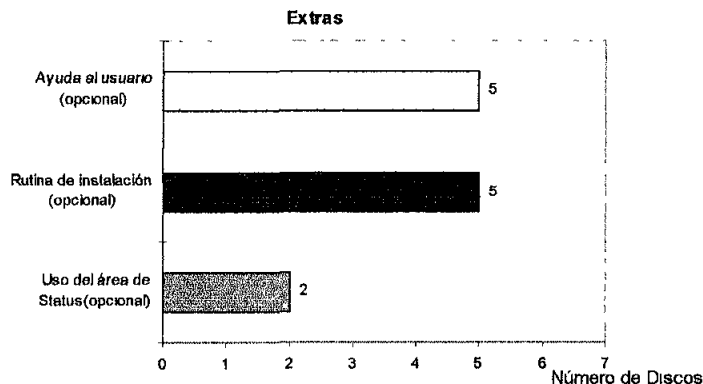
- Para esta segunda fase se desarrollaron estándares para homogeneizar los programas que fueron evaluados. En la gráfica del Figura 4.26 se puede observar cuántos de los discos que se entregaron cumplían con los estándares; se hace notar que los programas que se entregaron para su evaluación no cubrían el 100% de los estándares.

Figura 4.26. Estándares.



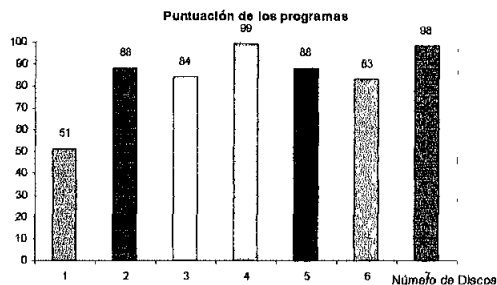
- Algunos de los programas que se entregaron para su evaluación contaban con características que no se les pidieron a los tesisistas. Estas características tampoco fueron usadas por todos, por lo que los programas no lucían estandarizados. En la gráfica de la Figura 4.27 se puede observar qué tanto se utilizaron estas características.

Figura 4.27. Extras.



- En la Figura 4.28 se observan las puntuaciones obtenidas por los programas que se entregaron para su evaluación en la Fase Dos. Como se observa, un grupo de programas obtuvo una puntuación muy baja de 51 puntos, cuatro grupos de programas obtuvieron una puntuación que osciló entre 83 y 88 puntos, y los restantes dos obtuvieron una puntuación que osciló entre 98 y 99 puntos, siendo éstas las más altas.

Figura 4.28. Puntuación de los Programas.



4.3.3.4 Reporte

Al igual que el reporte de Fase Uno, se utilizó un número de disco en lugar de los nombre de los tesisas. En este reporte sólo se presenta una tabla comparativa donde se resumen los puntos evaluados, y a continuación de esta tabla, no se detalla la evaluación de cada disco, sólo se presentan los comentarios y observaciones hechas. A continuación se lista cada punto evaluado de lo programas se evaluaron en la Fase Dos del Estudio de Desarrolladores.

Puntos Evaluados en la Tabla General

“**Procedimientos entregados**”. Se listan a continuación los programas que se pidieron a los tesisas :

- Presentación del software
- Pantalla de Contraseña
- Menú de iconos
- Programa de ABC 1
- Programa de ABC 2
- Pantalla de Consultas
- Reporte con Sort
- Reporte con Corte de Control
- Reporte con Contador

- Respaldo de archivos
- Acerca de...

“Requisitos generales”. Los siguientes son los requisitos mínimos que debían cumplir los programas entregados :

- Se pudo ejecutar el programa
- Deshabilita las pantallas que no están en uso
- Título en las ventanas.
- Mensajes al usuario (de error, de avisos)
- Especificación de la Contraseña
- Procedimientos Integrados
- Conexión automática de la BD

“Estándares”. Se elaboraron algunos estándares para normar algunos de los aspectos de desarrollo de los programas; a continuación se listan estos estándares:

- Estándar de Etiqueta
- Estándar de Directorio
- Estándar de Archivos

“Extras” : Son características que no fueron obligatorias pero que ayudaron a mejorar el aspecto del programa entregado.

- Uso del área de Status (opcional)
- Rutina de instalación (opcional)
- Ayuda al usuario (opcional)


El reporte quedó dividido de la siguiente manera:

- Lista de los programas entregados
- Puntos Evaluados en la Tabla General
- Tabla General
- Evaluación del disco 1
- Evaluación del disco 2
- Evaluación del disco 3
- Evaluación del disco 4
- Evaluación del disco 5
- Evaluación del disco 6
- Evaluación del disco 7

En las siguientes páginas se presenta el reporte con la evaluación de la Fase Dos.

En la Figura 4.29 se presenta la lista de los tesisistas que entregaron los programas para su evaluación, y el número de disco que se les fue asignado. Este número de disco es utilizado a lo largo del reporte.

Figura 4.29. Lista de Programas Entregados.



UNIVERSIDAD POPULAR AUTONOMA DEL ESTADO DE PUEBLA
Licenciatura en Sistemas Computacionales

EVALUACION DE PROGRAMAS

Lista de Programas Entregados

A continuación se listan los programas que fueron entregados a Ingeniería de Software para su evaluación

- Disco 1 SINAF Contabilidad y Presupuestos, Integrantes
 Robledo Ramírez Herman
 Xartuni Magos Bernardo
 Zambrano González Silverio

- Disco 2 SINAF, Cuentas por Cobrar, Integrantes
 De Velasco Aizpuru Luis Enrique
 Díaz Gutiérrez Juan Manuel

- Disco 3 SINAF, Abastecimiento y Almacén, Integrantes
 Navarro Carño Daphne
 Herrera Sánchez Claudia O

- Disco 4 SINIGA Servicio Social y Exalumnos, Integrantes
 López Pérez Verónica
 Pérez Duarte Ma Eugenia

- Disco 5 SINIGA Servicio Social, Integrantes
 Morales Fernández José Leopoldo


- Disco 6 SINIGA Becas, Integrantes
 Lobatón Hernandez Alejandro

- Disco 7 SINAF, Activos Fijos , Integrantes
 Campos Hernández Andrei
 Covarrubias Ponce José Antonio

Fuente: Reporte de la Capacidad de un Día de Trabajo (Fase DOS).

En la Figura 4.30 se observa la hoja que acompañó al reporte para que los tesisas estuvieran enterados de los puntos que se evaluaron y que se presentan en la Tabla General.

Figura 4.30. Puntos Evaluados en la Tabla General de la Fase Dos.



UNIVERSIDAD POPULAR AUTONOMA DEL ESTADO DE PUEBLA
Licenciatura en Sistemas Computacionales

EVALUACION DE PROGRAMAS

Puntos Evaluados en la Tabla General

Procedimientos Entregados

- Se refiere a cuáles de los programas fueron entregados por los tesisas para su evaluación

Requisitos Generales

- **Se pudo ejecutar el programa:** Se refiere a que si se pudieron correr todos los programas que fueron entregados por los tesisas
- **Deshabilita las pantallas que no están en uso:** Se evalúa si el programador deshabilita u oculta una ventana cuanto esta ha llamado a otra ventana o a un cuadro de dialogo
- **Título en las ventanas:** Se evalúa que por lo menos a alguna ventana se le haya cambiado el título, obviamente es necesario darles título a todas las ventanas pero para esta aplicación no se exigirá así
- **Mensajes al usuario:** Se evalúa que por lo menos cuando se pide el Password, se le mande un mensaje al usuario avisándole que ha ingresado la clave incorrecta
- **Especificación de la contraseña:** Se evalúa si se incluye el la contraseña del módulo
- **Procedimientos integrados:** Se evalúa que los programas se ejecuten desde un programa principal o por separado
- **Conexión de la BD:** Aquí se indica si los programas conectan automáticamente su base de datos

Estándares

- **Estándar de Etiqueta:** En este punto se indica si la etiqueta del disco cumple con los estándares definidos en el Anexo A 'Manual de Estándares de Progress'
- **Estándar de Directorio:** Se indica si cumplió con el estándar de directorios. Los estándares de directorios se especifican en el Anexo A 'Manual de Estándares de Progress'
- **Estándar de Archivos:** Este se refiere al estándar con el cual se deben nombrar los archivos. Los estándares de archivos se especifican en el Anexo A 'Manual de Estándares de Progress'

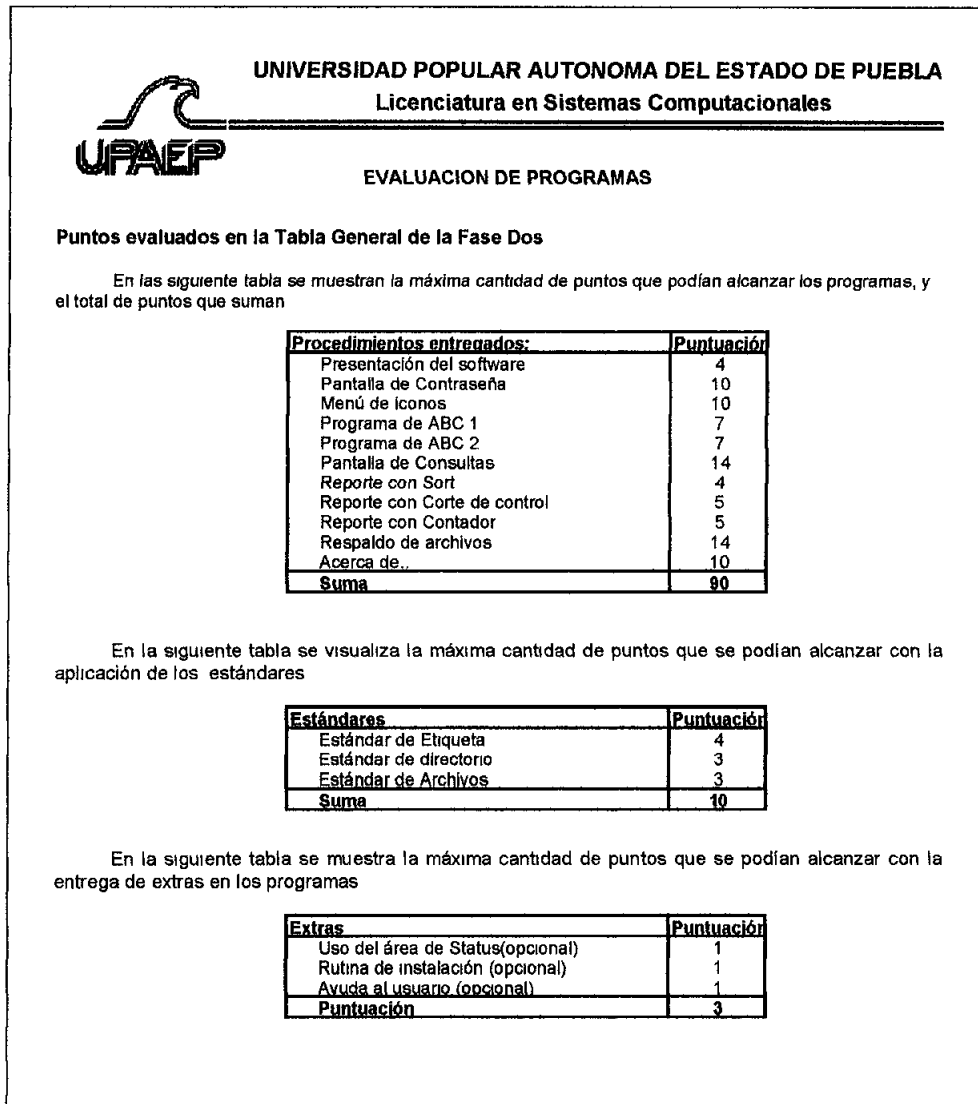
Extras

- **Uso del área de Status (opcional):** Este punto no es obligatorio, pero es bueno si por lo menos en alguna ventana se utilizó esta área
- **Rutina de Instalación (opcional):** Este punto no es obligatorio, ya que no se especificó en ningún momento que la aplicación pudiera tener una rutina para instalarse, sin embargo ayuda a mejorar la puntuación
- **Ayuda al usuario (opcional):** Al igual que los dos puntos anteriores este punto no es obligatorio. Indica si se proporciona algún de ayuda al usuario como mensajes de alerta, de información, etc

Fuente: Reporte de la Capacidad de un Día de Trabajo (Fase DOS).

En la Figura 4.31 se pueden apreciar los puntos evaluados en la tabla general de la fase dos, así como la máxima puntuación que se podía obtener en cada punto que se evaluó.


Figura 4.31. Puntos Evaluados en la Tabla General de la Fase Dos.



Fuente: Reporte de la Capacidad de un Día de Trabajo (Fase DOS).

En la Figura 4.32 se observa la Tabla General, la cual agrupa las evaluaciones de los siete disquetes. En esta tabla se presentan todos los puntos evaluados sin la valoración que alcanzaron los programas entregados.

Figura 4.32. Tabla General de la Fase Dos.



**UNIVERSIDAD POPULAR AUTONOMA DEL ESTADO DE
Licenciatura en Sistemas Computacionales**

EVALUACION DE PROGRAMAS

Tabla General


Puntos a evaluar	Disco 1	Disco 2	Disco 3	Disco 4	Disco 5	Disco 6	Disco 7
Procesamiento entregado:							
Presentación del software	✓	✓	✓	✓	✓	✓	✓
Pantalla de Contraseña	✓	✓	✓	✓	✓	✓	✓
Menú de Iconos	✓	✓	✓	✓	✓	✓	✓
Programa de ABC 1	✓	✓	✓	✓	✓	✓	✓
Programa de ABC 2	✓	✓	✓	✓	✓	✓	✓
Pantalla de Consultas	x	✓	✓	✓	✓	✓	✓
Reporte con Sort	✓	✓	✓	✓	✓	✓	✓
Reporte con Corte de control	✓	✓	✓	✓	✓	✓	✓
Reporte con Contador	✓	✓	✓	✓	✓	✓	✓
Respaldo de archivos	x	x	x	✓	✓	x	✓
Acercos de	x	✓	✓	✓	✓	✓	✓
Requisitos generales							
Se pudo ejecutar el programa	✓	✓	✓	✓	✓	✓	✓
Deshabilita las pantallas que no están en uso	x	x	x	x	x	x	x
Título en las ventanas	✓	✓	✓	✓	✓	✓	✓
Mensajes al usuario (de error, de avisos)	x	✓	✓	✓	✓	✓	✓
Especificación de la Contraseña	✓	✓	✓	✓	✓	x	✓
Procedimientos Integrados	✓	✓	✓	✓	✓	✓	✓
Conexión automática de la BD		x	✓	✓	x	x	x
Estándares							
Estándar de Estructura	✓	✓	✓	✓	✓	✓	x
Estándar de directorio	✓	✓	x	x	✓	x	x
Estándar de Archivos	x	✓	✓	✓	✓	x	x
Otras							
Uso del Área de Status (opcional)	x	x	x	x	✓	✓	x
Rutina de instalación (opcional)	x	✓	✓	✓	x	✓	✓
Ayuda al usuario (opcional)	x	✓	x	✓	✓	✓	✓

Fuente: Reporte de la Capacidad de un Día de Trabajo (Fase DOS).

A partir de la Figura 4.33 a la Figura 4.39 se muestran las hojas con las evaluaciones detalladas por cada disco, además de incluir comentarios y observaciones y la cantidad de puntos obtenidos.

En la Figura 4.33 se puede visualizar la evaluación del disco 1.

Figura 4.33. Evaluación del Disco 1.



UNIVERSIDAD POPULAR AUTONOMA DEL ESTADO DE PUEBLA
Licenciatura en Sistemas Computacionales

EVALUACION DE PROGRAMAS

No DE TESIS	FECHA	HOJA	DE
DESCRIPCION DE LA TESIS			
MAI RICULA	NOMBRE	No I I S T A	GRUPO
200495	Alejandro Lobatón Hernández		8 ^o B

Evaluación de Disco Uno

Lista de programas

- Presentación del software
- Pantalla de Contraseña
- Menu de iconos
- Programa de ABC 1
- Programa de ABC 2
- Pantalla de Consultas
- Reporte con Sort
- Reporte con Corte de control
- Reporte con Contador
- Respaldo de archivos
- Acerca de

Puntos a evaluar	Prog 1	Prog 2	Prog 3	Prog 4	Prog 5	Prog 6	Prog 7	Prog 8	Prog 9	Prog 10	Prog 11
Requisitos generales	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Se puede visualizar el programa	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Título en la ventana	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Atenuación al usuario (por error de mensajes)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Puntuación	3	0	0	0	0	0	0	0	0	0	0
SubTotal	44										

Puntos a evaluar	Puntuación
Estándar	4
Estándar de Estructura	3
Estándar de Archivos	0
Estilos	0
Uso del ícono de Status (opcional)	0
Rutina de instalación (opcional)	0
Ayuda al usuario (opcional)	0
SubTotal	7
Total	51

Notas de la instalación

- No especifica que el programa de instalación debe correrse desde windows
- El directorio de trabajo es REV_PROG y el correcto es REVPROG


Notas generales sobre el programa

- No se desactivan las pantallas que no están en uso
- No conectan su base de datos

Fuente: Reporte de la Capacidad de un Día de Trabajo (Fase DOS).

En la Figura 4.32 se observa la Tabla General, la cual agrupa las evaluaciones de los siete disquetes. En esta tabla se presentan todos los puntos evaluados sin la valoración que alcanzaron los programas entregados.

Figura 4.32. Tabla General de la Fase Dos.



UNIVERSIDAD POPULAR AUTONOMA DEL ESTADO DE
Licenciatura en Sistemas Computacionales

EVALUACION DE PROGRAMAS

Tabla General


Evolutivos a evaluar	Disco 1	Disco 2	Disco 3	Disco 4	Disco 5	Disco 6	Disco 7
Procedimientos entregados.							
Presentación del software	✓	✓	✓	✓	✓	✓	✓
Pantalla de Contraseña	✓	✓	✓	✓	✓	✓	✓
Menú de Iconos	✓	✓	✓	✓	✓	✓	✓
Programa de ABC 1	✓	✓	✓	✓	✓	✓	✓
Programa de ABC 2	✓	✓	✓	✓	✓	✓	✓
Pantalla de Consultas	x	✓	✓	✓	✓	✓	✓
Reporte con Sort	✓	✓	✓	✓	✓	✓	✓
Reporte con Corte de control	✓	✓	✓	✓	✓	✓	✓
Reporte con Contador	✓	✓	✓	✓	✓	✓	✓
Respaldo de archivos	x	x	x	✓	x	x	✓
Acerca de	x	✓	✓	✓	✓	✓	✓
Requisitos generales							
Se pudo ejecutar el programa	✓	✓	✓	✓	✓	✓	✓
Deshabilita las pantallas que no están en uso	x	x	x	x	x	x	x
Título en las ventanas	✓	✓	✓	✓	✓	✓	✓
Mensajes al usuario (de error, de avisos)	x	✓	✓	✓	✓	✓	✓
Especificación de la Contraseña	✓	✓	✓	✓	✓	x	✓
Procedimientos integrados	✓	✓	✓	✓	✓	✓	✓
Conservación automática de la BD	✓	x	✓	✓	x	x	x
Estándares							
Estándar de Etiqueta	✓	✓	✓	✓	✓	✓	x
Estándar de directorio	✓	✓	x	x	✓	x	x
Estándar de Archivos	x	✓	✓	✓	✓	x	x
Extras							
Uso del Área de Status(opcional)	x	x	x	x	✓	✓	x
Rutina de instalación (opcional)	x	✓	✓	✓	x	✓	✓
Ayuda al usuario (opcional)	x	✓	x	✓	✓	✓	✓

Fuente: Reporte de la Capacidad de un Día de Trabajo (Fase DOS).

A partir de la Figura 4.33 a la Figura 4.39 se muestran las hojas con las evaluaciones detalladas por cada disco, además de incluir comentarios y observaciones y la cantidad de puntos obtenidos.

En la Figura 4.33 se puede visualizar la evaluación del disco 1.

Figura 4.33. Evaluación del Disco 1.



UNIVERSIDAD POPULAR AUTONOMA DEL ESTADO DE PUEBLA
Licenciatura en Sistemas Computacionales

EVALUACION DE PROGRAMAS

No DE TESIS	FECHA	HOJA	DE
DESCRIPCIÓN DE LA TESIS			
MATRICULA	NCMBRE	No LISTA	GRUPO
200495	Alejandro Lolesón Hernández		8°"B"

Evaluación de Disco Uno

Lista de programas

- Presentacion del software
- Pantalla de Contraseña
- Menu de iconos
- Programa de ABC 1
- Programa de ABC 2
- Pantalla de Consultas
- Reporte con Sort
- Reporte con Corte de control
- Reporte con Contador
- Respaldo de archivos
- Acerca de

Puntos a evaluar	Prog 1	Prog 2	Prog 3	Prog 4	Prog 5	Prog 6	Prog 7	Prog 8	Prog 9	Prog 10	Prog 11
Requisitos mínimos	✓	✓	✓	✓	✓	X	✓	✓	✓	X	X
Se pudo ejecutar el programa	✓	✓	✓	✓	✓	X	✓	✓	✓	X	X
Título en la ventana	X	X	X	X	X	X	X	X	X	X	X
Mensajes al usuario (de error de discos)	X	X	X	X	X	X	X	X	X	X	X
Puntuación	3	0	9	0	0	0	3	4	4	0	0
SubTotal:	44										

Puntos a evaluar	Puntuación
Etiquetas	4
Estándar de Etiqueta	3
Estándar de Directorio	0
Estándar de Archivos	0
Extras	0
Uso del área de Status (opcional)	0
Rutina de instalación (opcional)	0
Ayuda al usuario (opcional)	0
SubTotal	7
Total	51

Notas de la instalación

- No especifica que el programa de instalación debe correrse desde windows
- El directorio de trabajo es REV_PROG y el correcto es REVPROG


Notas generales sobre el programa

- No se desactivan las pantallas que no están en uso
- No conectan su base de datos

Fuente: Reporte de la Capacidad de un Día de Trabajo (Fase DOS).

En la Figura 4.34 se puede visualizar la evaluación del disco 2.

Figura 4.34. Evaluación del Disco 2.



**UNIVERSIDAD POPULAR AUTONOMA DEL ESTADO DE
Licenciatura en Sistemas Computacionales**

EVALUACION DE PROGRAMAS

No DE TESIS _____	FECHA _____	HOJA _____	DE _____
DESCRIPCION DE LA TESIS _____			
MATRICULA _____	NOMBRE _____	No LISTA _____	GRUPO _____
200495	Alcandro I elvstón Hernández		8° "B"

Evaluación de Disco Uno

Lista de programas:

- Presentación del software
- Pantalla de Contraseña
- Menú de iconos
- Programa de ABC 1
- Programa de ABC 2
- Pantalla de Consultas
- Reporte con Sort
- Reporte con Corte de control
- Reporte con Contador
- Respaldo de archivos
- Acerca de

Puntos a evaluar	Prog. 1	Prog. 2	Prog. 3	Prog. 4	Prog. 5	Prog. 6	Prog. 7	Prog. 8	Prog. 9	Prog. 10	Prog. 11
Requisitos generales											
Se pudo ejecutar el programa	✓	✓	✓	✓	✓	✓	x	✓	✓	x	x
Título es lo correcto	✓	x	x	x	x	x	x	x	x	x	x
Mensajes al usuario (de error, de aviso)	x	x	x	x	x	x	x	x	x	x	x
Puntuación	3	9	9	6	6	0	3	4	4	0	0
SubTotal	44										

Puntos a evaluar	Puntuación
Estándares	
Estándar de Etiqueta	4
Estándar de directorio	3
Estándar de Archivos	0
Extrae	
Uso del área de Status (opcional)	0
Rutina de instalación (opcional)	0
Ayuda al usuario (opcional)	0
SubTotal	7

Total	51
--------------	----

Notas de la instalación

- No especifica que el programa de instalación debe correrse desde
- El directorio de trabajo es REV_PROG y el correcto es


Notas generales sobre el programa

- No se desactivan las pantallas que no están en
- No conectan su base de

Fuente: Reporte de la Capacidad de un Día de Trabajo (Fase DOS).

En la Figura 4.35 se puede visualizar la evaluación del disco 3.

Figura 4.35. Evaluación del Disco 3.



UNIVERSIDAD POPULAR AUTONOMA DEL ESTADO DE
Licenciatura en Sistemas Computacionales

EVALUACION DE PROGRAMAS

No DF TESIS _____	FECHA _____	HOJA _____	DI _____
DESCRIPCION DE LA TESIS _____			
MATRICULA	NOMBRE	No LISTA	GRUPO
200577	Verónica López Pérez		8º "B"
200535	Ma. Eugenia Pérez Duarte		8º "B"

Evaluación de Disco Tres

Lista de programas

- 1 Presentación del software
- 2 Pantalla de Contraseña
- 3 Menú de Iconos
- 4 Programa de ABC 1
- 5 Programa de ABC 2
- 6 Pantalla de Consultas
- 7 Reporte con Sort
- 8 Reporte con Cinta de control
- 9 Reporte con Contador
- 10 Respaldo de archivos
- 11 Acerca de

Puntos a evaluar	Pron. 1	Pron. 2	Pron. 3	Pron. 4	Pron. 5	Pron. 6	Pron. 7	Pron. 8	Pron. 9	Pron. 10	Pron. 11
Requisitos generales											
Se pudo ejecutar el programa	✓	✓	✓	✓	✓	✓	✓	✓	✓	x	✓
Título en la ventana	✓	✓	✓	✓	✓	✓	✓	✓	✓	x	✓
Manuales al usuario (de error de avisos)	✓	✓	✓	✓	✓	✓	✓	✓	✓	x	✓
Puntuación	4	10	10	7	7	14	4	5	5	0	10
SubTotal	76										

Puntos a evaluar	Puntuación
Estándares	
Estándar de Etiqueta	4
Estándar de directorio	0
Estándar de Archivos	3
Rextras	
Uso del Área de Status (opcional)	0
Rutina de instalación (opcional)	1
Ayuda al usuario (opcional)	0
SubTotal	8
Total	84

Notas generales sobre el programa

- No se desactivan las pantallas que no están en

Fuente: Reporte de la Capacidad de un Día de Trabajo (Fase DOS).

En la Figura 4.36 se puede visualizar la evaluación del disco 4.

Figura 4.36. Evaluación del Disco 4.

UNIVERSIDAD POPULAR AUTONOMA DEL ESTADO DE PUEBLA
Licenciatura en Sistemas Computacionales

EVALUACION DE PROGRAMAS

No DE TESIS _____	FECHA _____	HOJA _____	DE _____
DESCRIPCIÓN DE LA TESIS <u>Abastecimiento y Almacén</u>			
MATRICULA	NOMBRE	No LISTA	GRUPO
<u>200511</u>	<u>Claudia O. Herrera Sánchez</u>	_____	<u>8º</u>
<u>200495</u>	<u>Daphne J. Navarro Carriño</u>	_____	<u>8º</u>

Lista de programas

- 1 Presentación del software
- 2 Pantalla de Contraseña
- 3 Menú de iconos
- 4 Programa de ABC 1
- 5 Programa de ABC 2
- 6 Pantalla de Consultas
- 7 Reporte con Sort
- 8 Reporte con Corte de control
- 9 Reporte con Contador
- 10 Respaldo de archivos
- 11 Acerca de

Puntos a evaluar	Prog. 1	Prog. 2	Prog. 3	Prog. 4	Prog. 5	Prog. 6	Prog. 7	Prog. 8	Prog. 9	Prog. 10	Prog. 11
Requisitos generales											
Se pudo ejecutar el	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Título en la ventana	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Mensaje al usuario (de error, de	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Puntuación	4	10	10	7	7	14	4	5	5	14	10
SubTotal	60										

Puntos a evaluar	Puntuación
Requisitos	4
Estándar de	0
Estándar de directorio	3
Estándar de Archivos	
Otros	0
Uso del área de	1
Rutina de instalación	1
Ayuda al usuario (opcional)	
SubTotal	6
Total	66


Notas generales sobre el programa:

- No se desactivan las pantallas que no están en uso

Fuente: Reporte de la Capacidad de un Día de Trabajo (Fase DOS).

En la Figura 4.37 se puede visualizar la evaluación del disco 5.

Figura 4.37. Evaluación del Disco 5.



UNIVERSIDAD POPULAR AUTONOMA DEL ESTADO DE
Licenciatura en Sistemas Computacionales

EVALUACION DE PROGRAMAS

No DE TESIS _____	FECHA _____	HOJA _____	DE _____
DESCRIPCIÓN DE LA TESIS _____			
MATRICULA	NOMBRE	No LISTA	GRUPO
200481	Hernan Robledo		
200526	Bernardo Xartum		
200463	Silverio Zambrano		

Evaluación de Disco Cinco

Lista de programas

- 1 Presentación del software
- 2 Pantalla de Contraseña
- 3 Menú de Iconos
- 4 Programa de ABC 1
- 5 Programa de ABC 2
- 6 Pantalla de Consultas
- 7 Reporte con Sort
- 8 Reporte con Conto de control
- 9 Reporte con Contador
- 10 Respaldo de archivos
- 11 Acerca de

Puntos a evaluar	Prog. 1	Prog. 2	Prog. 3	Prog. 4	Prog. 5	Prog. 6	Prog. 7	Prog. 8	Prog. 9	Prog. 10	Prog. 11
Requisitos generales											
Se pudo ejecutar el programa	✓	✓	✓	✓	✓	✓	✓	✓	✓	x	✓
Título en la ventana	✓	✓	✓	✓	✓	✓	✓	✓	✓	x	✓
Menús al usuario (de error, de ayuda)	✓	✓	✓	✓	✓	✓	✓	✓	✓	x	✓
Puntuación	4	10	10	7	7	14	4	6	6	0	10
SubTotal	78										

Puntos a evaluar	Puntuación
Estándares	
Estándar de Etiqueta	4
Estándar de directorio	3
Estándar de Archivos	3
Extras	
Uso del área de Status (opcional)	1
Rutina de instalación (opcional)	0
Ayuda al usuario (opcional)	1
SubTotal	12
Total	90

Notas de la instalación

- No se incluyó un programa de instalación.


Notas generales sobre el programa

- No se desactivan las pantallas que no están en uso
- Colocaron un reporte por archivo, y en otro archivo incluyeron los tres reportes
- No conectan su base de datos

Fuente: Reporte de la Capacidad de un Día de Trabajo (Fase DOS).

En la Figura 4.38 se puede visualizar la evaluación del disco 6.

Figura 4.38. Evaluación del Disco 6.



UNIVERSIDAD POPULAR AUTONOMA DEL ESTADO DE
Licenciatura en Sistemas Computacionales

EVALUACION DE PROGRAMAS

No DF TESIS _____	FECHA _____	HOJA _____	DF _____
DESCRIPCION DE LA TESIS <u>Modulo de Cuentas por Cobrar</u>			
MATRICULA	NOMBRE	No LISTA	GRUPO
200491	Luis de Velasco		8° "B"
200521	Manuel Diaz		8° "B"

Evaluación de Disco Seis

Lista de programas

- 1 Presentación del software
- 2 Pantalla de Contraseña
- 3 Menu de Iconos
- 4 Programa de ABC 1
- 5 Programa de ABC 2
- 6 Pantalla de Consultas
- 7 Reporte con Sort
- 8 Reporte con Corte de control
- 9 Reporte con Contador
- 10 Respaldo de archivos
- 11 Acerca de

Puntos a evaluar	Pron. 1	Pron. 2	Pron. 3	Pron. 4	Pron. 5	Pron. 6	Pron. 7	Pron. 8	Pron. 9	Pron. 10	Pron. 11
Requisitos generales											
Se pudo ejecutar el programa	✓	✓	✓	✓	✓	✓	✓	✓	✓	x	✓
Título en la ventana	✓	✓	✓	✓	✓	✓	✓	✓	✓	x	✓
Mensajes al usuario (de error, de avisos)	✓	✓	✓	✓	✓	✓	✓	✓	✓	x	✓
Puntuación	4	10	10	7	7	14	4	5	5	0	10
SubTotal	76										

Puntos a evaluar	Puntuación
Estándares	
Estándar de Etiqueta	4
Estándar de directorio	0
Estándar de Archivos	0
Extras	
Uso del área de Status(opcional)	1
Rutina de instalación (opcional)	1
Ayuda al usuario (opcional)	1
SubTotal	3
Total	83


Notas generales sobre el programa:

- Manda un mensaje de error al no poder encontrar una imagen, aunque esto no afecto el desempeño de su programa
- No conectan su base de datos
- No se desactivan las pantallas que no están en uso
- La ruta de la base de datos para los reportes es incorrecta

Fuente: Reporte de la Capacidad de un Día de Trabajo (Fase DOS).

En la Figura 4.39 se puede visualizar la evaluación del disco 7.

Figura 4.39. Evaluación del Disco 7.



UNIVERSIDAD POPULAR AUTONOMA DEL ESTADO DE
Licenciatura en Sistemas Computacionales

EVALUACION DE PROGRAMAS

No DE TESIS	FECHA	HOJA	DF
DESCRIPCION DE LA TESIS <u>Activos Fijos</u>			
MATRICULA	NOMBRE	No LISTA	GRUPO
200502	Andrei Campos		8° "B"
200476	Antonio Covarrubias		8° "B"

Evaluación de Disco Siete

*Lista de programas**

- 1 Presentación del software
- 2 Pantalla de Contraseña
- 3 Menú de Iconos
- 4 Programa de ABC 1
- 5 Programa de ABC 2
- 6 Pantalla de Consultas
- 7 Reporte con Sort
- 8 Reporte con Corte de control
- 9 Reporte con Contador
- 10 Respaldo de archivos
- 11 Acerca de

Puntos a evaluar	Prop. 1	Prop. 2	Prop. 3	Prop. 4	Prop. 5	Prop. 6	Prop. 7	Prop. 8	Prop. 9	Prop. 10	Prop. 11
Requisitos generales											
Se pudo ejecutar el programa	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Tiempo en la ventana	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Mensajes al usuario (de error, de avisos)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Puntuación	4	10	10	7	7	14	4	5	5	14	10
Sub Total	88										

Puntos a evaluar	Puntuación
Estándares	
Estándar de Etiqueta	4
Estándar de directorio	4
Estándar de Arquivos	0
Otras	
Uso del área de Status (opcional)	0
Rutina de instalación (opcional)	0
Aviso al usuario (opcional)	0
Sub Total	8
Total	88

Notas generales sobre el programa

- Manda un mensaje de error al no poder encontrar una imagen, aunque esto no afecto el desempeño de su programa
- No conecta su base de datos
- No se desachvan las pantallas que no están en uso

Fuente: Reporte de la Capacidad de un Día de Trabajo (Fase DOS).

4.4 Resultados del Estudio de Desarrolladores

El de Análisis de la Capacidad de una Hora de Trabajo de los desarrolladores demostró que el nivel técnico de los tesisas era bajo en cuanto a la programación en el lenguaje estructurado, ya que sólo el 13% de los programas que se entregaron funcionaron. Es importante recalcar que hubo tesisas que no se presentaron, pese a que se les dijo que era muy importante su asistencia. Además, de que sólo el 54% de los desarrolladores planteó la solución del problema en papel para después programarla, con lo que se concluye que sólo ese porcentaje tenía el hábito de diseñar en papel antes de programar.

El estudio del Análisis de la Capacidad de una Hora de Trabajo de los desarrolladores demostró también en su primera fase que los cursos que se impartieron de Progress no fueron aprovechados al 100%. Esto se concluye después de revisar los resultados de esta fase, en donde se observa entre otras cosas, que hubo personas que no entregaron todos los programas, que no se cumplieron con los requisitos generales que se pidieron, y que el promedio general fue de 68.86.

En cuanto al Análisis de la Capacidad de un día de Trabajo, en la primera fase destaca el hecho de las aplicaciones dejaban mucho que desear, ya que no fueron entregados todos los procedimientos, los requisitos generales no se cubrieron en su totalidad, y sus puntuaciones no fueron muy altas.

En cuanto a la segunda fase se observó que algunos de los desarrolladores destacaron, al mejorar bastante sus aplicaciones, pero hubieron otros que no mostraron mayor esfuerzo que el que se les pidió. Para esta fase se desarrollaron estándares, que no fueron seguidos por en su 100% por los desarrolladores.

Para el mejor aprovechamiento del lenguaje de Progress y de las demás herramientas, se impartieron cursos y se dió soporte técnico. Estos cursos y el soporte técnico, solucionaron necesidades como el manejo del ambiente de desarrollo visual de Progress y los conceptos que esto implica: objetos, mensajes, ligas, lenguaje de 4a. generación; además de dar el soporte que requerían los tesisas al utilizar estas herramientas.

Capítulo 5. Aplicación Documentada

Dado el objetivo principal de este trabajo, que es el desarrollo de Ingeniería de Software para el Departamento de Revisión de Procesos, se presenta en este capítulo la elaboración de la aplicación documentada con la ingeniería desarrollada.

Todos los módulos usaron la ingeniería desarrollada, ya sea completa o parcialmente. Tomando en cuenta lo anterior, se eligió el módulo de Servicio Social para presentarlo en este capítulo, ya que hace uso de toda la ingeniería. Se debe hacer la aclaración que Ingeniería de Software no desarrolló el módulo, pero sí ayudo, en su elaboración.

5.1 Instalación de Progress

Hay que consultar el manual de instalación de Progress para la correcta instalación del paquete.

5.2 Instalación de la Ingeniería de Software

Se necesita instalar los archivos de Ingeniería de Software. Estos se encuentran comprimidos en el archivo `STD-ISW3.ARJ`. Una vez hecho esto se debe hacer lo siguiente:

1. Abrir el archivo `SMART.CST` que se encuentra en el directorio `DLC\SRC\TEMPLATE` con cualquier editor de texto y buscar la sección de `#Modulo (MOD)`. El código de esta sección se puede apreciar en la Figura 5.1

Figura 5.1. Sección del Archivo SMART.CST.

```
#Modulo (MOD)      &Default
UP-IMAGE-FILE     C:\REVPROC\IMG\BMP\ICO-SSCU.BMP
DOWN-IMAGE-FILE   C:\REVPROC\IMG\BMP\ICO-SSCD.BMP
LABEL             Aarchivos de Modulo
DIRECTORY-LIST    C:\REVPROC\SINAF\MOD\PRG\W,C:\REVPROC\SINAF\MOD\PRG\I, .
FILTER            *.* ,v*.* ,b*.* ,q*.* ,*.i
TITLE             Escoge un archivo de Modulo
```

Fuente: Ingeniería de Software.

2. En la Figura 5.1 las palabras en negrita indican que deben sustituirse por la cadena correspondiente al módulo como se muestra en la Figura 5.2.

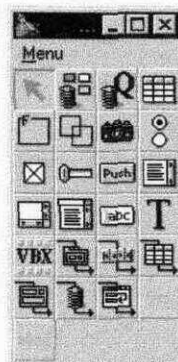
Figura 5.2. Sección Adaptada al Módulo de Servicio Social.

```
#Servicio Social &Default
UP-IMAGE-FILE     C:\REVPROC\IMG\BMP\ICO-SSCU.BMP
DOWN-IMAGE-FILE   C:\REVPROC\IMG\BMP\ICO-SSCD.BMP
LABEL             Aarchivos de Servicio Social
DIRECTORY-LIST    C:\REVPROC\SINIGA\SSC\PRG\W,C:\REVPROC\SINIGA\SSC\PRG\I, .
FILTER            *.* ,v*.* ,b*.* ,q*.* ,*.i
TITLE             Escoge un archivo de Servicio Social
```

Fuente: Ingeniería de Software.

3. Después de efectuar los cambios necesarios, guardar el archivo.
4. Ejecutar el User Interface Builder (UIB). La Paleta de Objetos que se presenta es semejante a la que se observa en la Figura 5.3.

Figura 5.3. Paleta de Objetos del UIB.

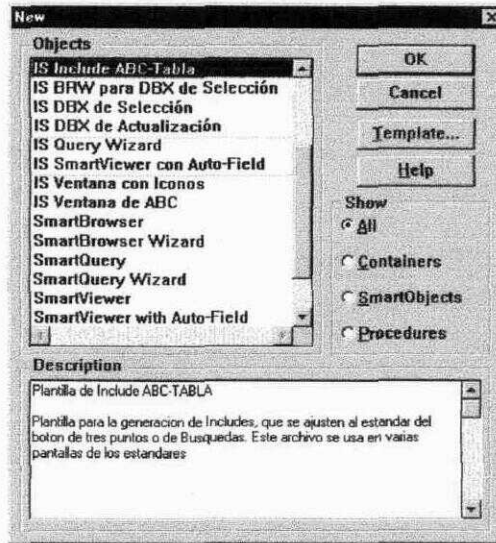


Fuente: Progress v8.

Un cambio que se observa consiste en dos iconos nuevos en la Paleta de Objetos, uno etiquetado con las letras Mod y el otro con las letras Gen.

Otro cambio son las nuevas plantillas que se tienen disponibles. Para acceder a estas plantillas se hace click en el botón **New** de la barra de herramientas del UIB. A continuación se despliega el cuadro de diálogo **New**, que se muestra en la Figura 5.4.

Figura 5.4. Cuadro de Diálogo New.



Fuente: Progress v8.

5.3 Desarrollo de la Aplicación

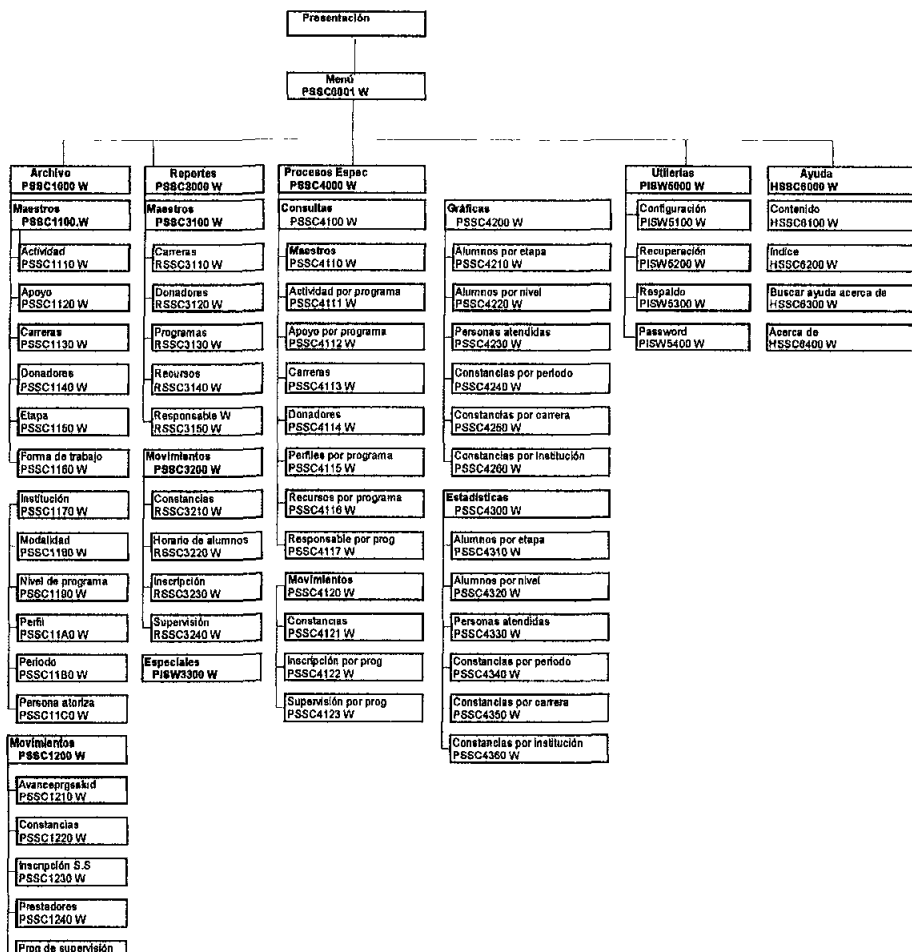
A partir de esta sección se explica la forma en que se construyó la aplicación utilizando las nuevas plantillas, los manuales de estándares, etc.

5.3.1 Documentación Necesaria

La documentación que se necesitó para la creación de esta aplicación fue:

- El Diagrama de Estructura, que se muestra en la Figura 5.5.

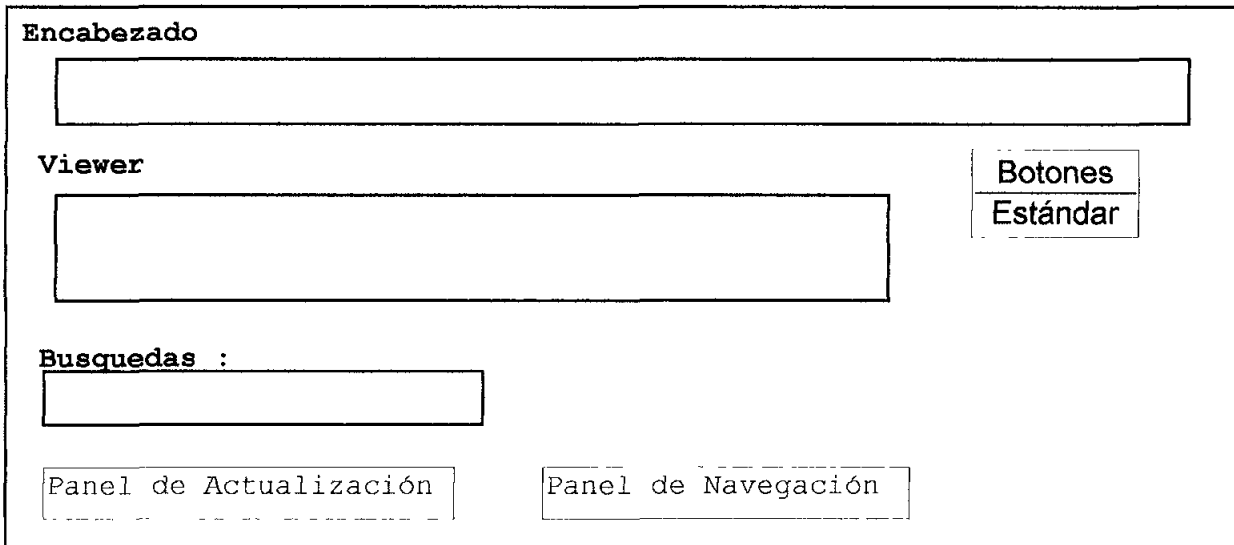
Figura 5.5. Diagrama de Estructura de Servicio Social.



Fuente: Manual Técnico del Módulo de Servicio Social.

- También fue necesario el Catálogo de Objetos para las ventanas que van a conformar la aplicación. En la Figura 5.6 se muestra un ejemplo de esta documentación.

Figura 5.6. Ejemplo del Catálogo de Objetos.



Fuente: Catálogo de Objetos del Módulo de Servicio Social.

- **Manuales de estándares.** Los manuales que se utilizaron fueron los siguientes:
 - Manual de Uso de la Ingeniería
 - Manual para Documentar el Catálogo de Objetos
 - Manual de Estándares de Progress v8
 - Manual de Estándares de VGA

En las siguientes secciones se explica a detalle la creación de cada uno de los componentes de la aplicación:

- Creación de los menús de iconos
- Creación de los archivos de Inclusión
- Creación de los SmartQueries
- Creación de los SmartViewers
- Creación de las ventanas
- Creación de los cuadros de diálogo

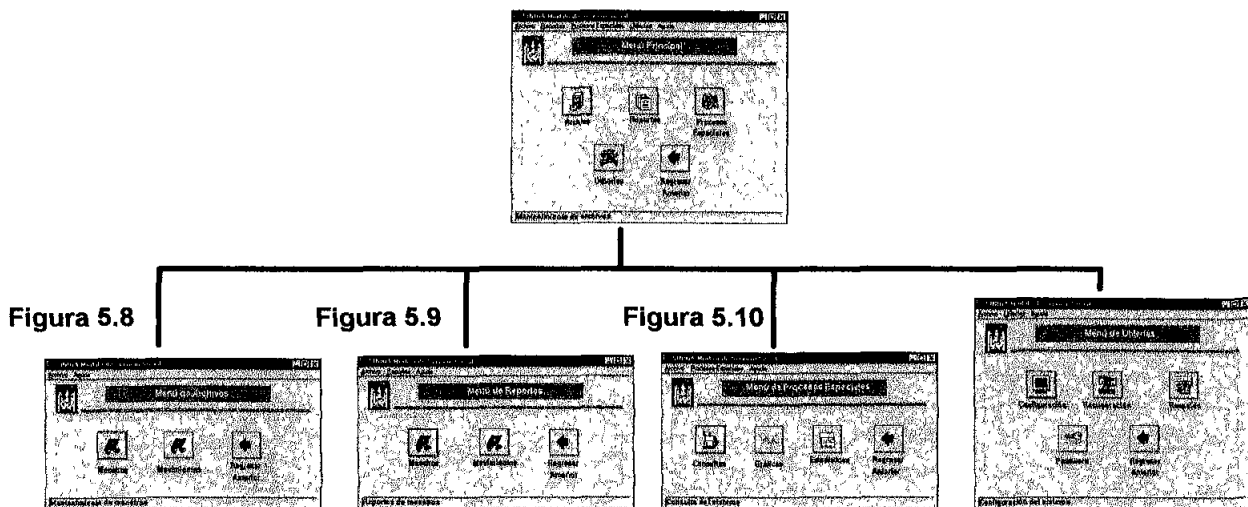
5.3.2 Construcción de los Menús de Iconos

De acuerdo con el diagrama de estructura se construye el menú de iconos. En la Figura 5.5 se visualizan los nombres de las ventanas que conforman el menú de iconos. Se resalta el hecho de que no se encuentre el programa PSSC2000 (correspondiente al menú de Cálculos) ni sus respectivas ramas; esto es porque el módulo no tiene cálculos.

A partir del Diagrama de Estructura se construyeron todas las ventanas y se obtuvo el Diagrama de Transición de Estados (DTE). Debido a su tamaño, el diagrama se dividió en varias partes. Cada una de estas partes se puede observar en las Figuras 5.7, 5.8, 5.9 y 5.10.

En la Figura 5.7 se puede observar la Ventana Principal y las otras cuatro que son llamadas desde los botones.

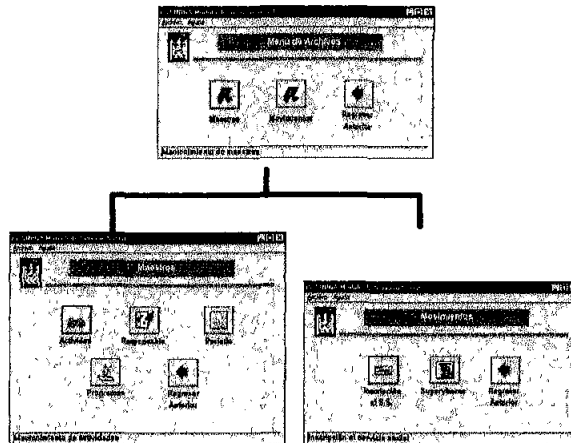
Figura 5.7. Ventana del Menú Principal y Submenús de Servicio Social.



Fuente: Módulo de Servicio Social.

La ventana de Archivos manda a llamar a otras ventanas a través de sus botones, esto se puede visualizar en la Figura 5.8.

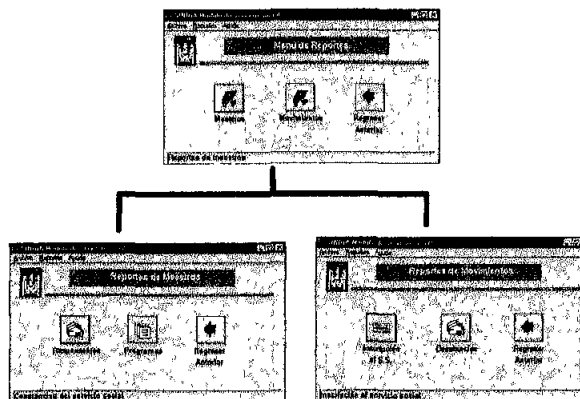
Figura 5.8. Ventana del Menú de Archivos y Submenús de Servicio Social.



Fuente: Módulo de Servicio Social.

La ventana de Reportes manda a llamar a otras ventanas a través de sus botones; esto se puede visualizar en la Figura 5.9.

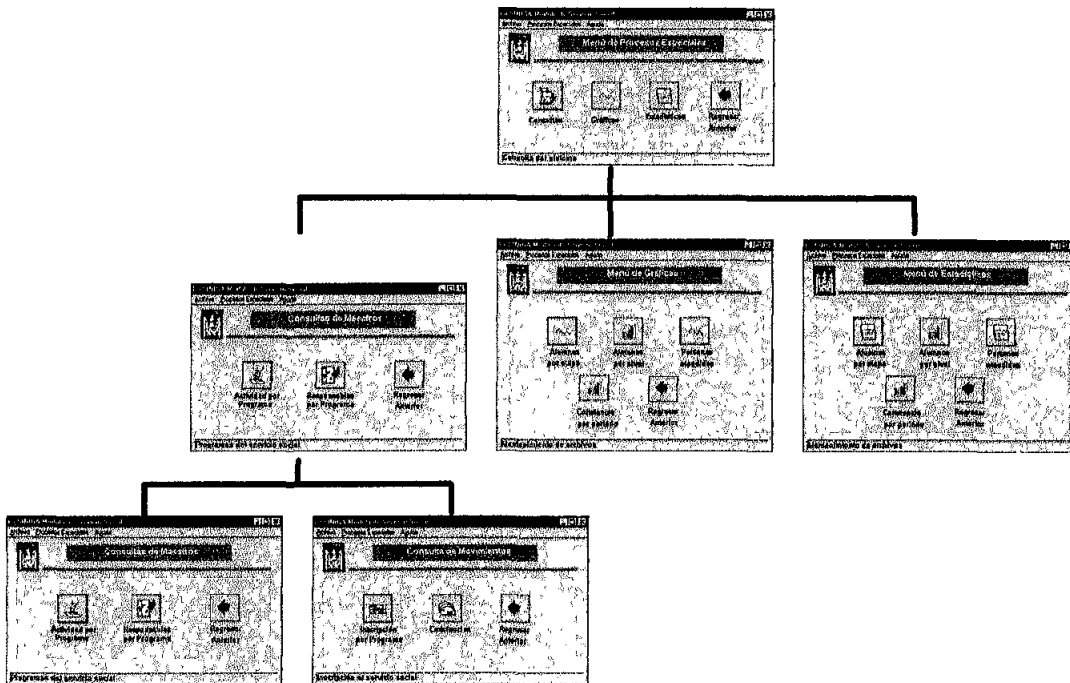
Figura 5.9. Ventana del Menú de Reportes y Submenús de Servicio Social.



Fuente: Módulo de Servicio Social.

La ventana de Procesos Especiales manda a llamar a otras ventanas a través de sus botones, esto se puede visualizar en la Figura 5.10.

Figura 5.10. Ventana de Procesos Especiales y Submenús de Servicio Social.

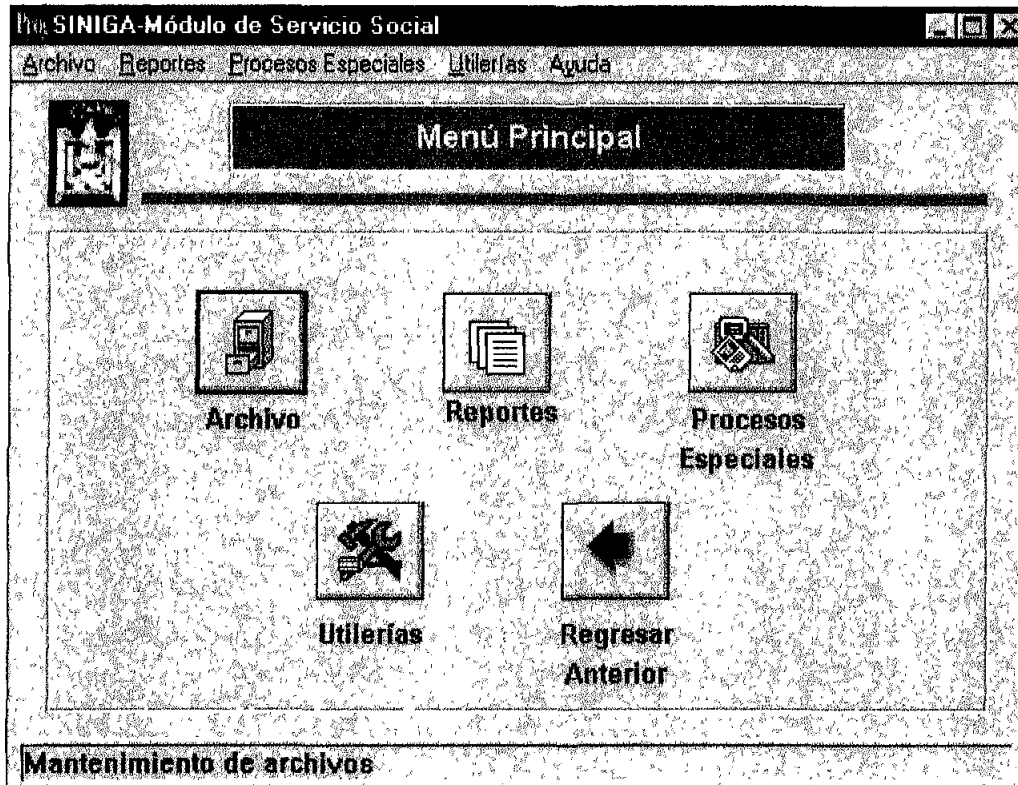


Fuente: Módulo de Servicio Social.

5.3.2.1 Construcción del Menú Principal (PSSC0001.W)

Se explica a continuación la ventana del Menú Principal. Esta ventana es la primera que se presenta al iniciar el módulo y es también, la que permite el acceso a todas las opciones del módulo. La ventana terminada se muestra en la Figura 5.11.

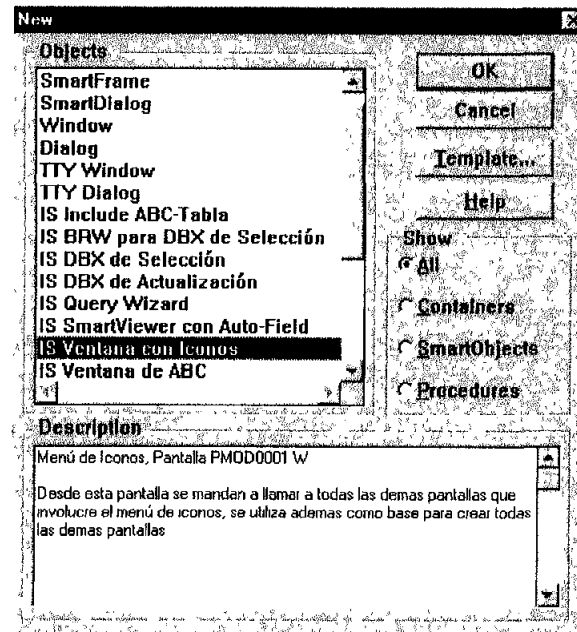
Figura 5.11. Ventana del Menú Principal de Servicio Social.



Fuente: Módulo de Servicio Social.

Para construir esta ventana se utilizó la plantilla IS Ventana con Iconos. Esta plantilla se puede elegir del cuadro de diálogo **New**, después de hacer click en el botón **New** de la Barra de Herramientas, dicho cuadro de dialogo se puede observar en la Figura 5.12.

Figura 5.12. Selección de la Plantilla IS Ventana con Iconos.

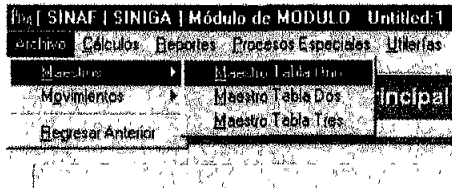


Fuente: Progress v8.

A la plantilla, una vez abierta, se le hicieron las siguientes modificaciones:

1. Se cambió el título de la ventana a SINIGA-Módulo de Servicio Social.
2. Se cambió el título del encabezado a Menú Principal.
3. La plantilla ya incluye el código para el llamado de los programas tanto en los botones como en las opciones de los menús. Este cambio se realizó accedando el evento **CHOOSE** de cada opción en el Editor de Sección. Las modificaciones son las siguientes:
4. El menú y el botón de **Cálculos** se eliminaron, ya que este módulo no cuenta con cálculos.
5. Para los botones y los menús de **Archivo, Reportes y Procesos Especiales** se cambiaron las siglas MOD por SSC.
6. Para el menú de **Reportes** se cambiaron las siglas MOD por SSC.
7. Los menús de **Utilerías y Ayuda** no sufrieron modificaciones ya que no hay que adaptarlos.
8. Se cambiaron las etiquetas de las opciones de los menús por la etiqueta correspondiente. La plantilla tiene tres opciones por cada menú, por eso se agregaron y quitaron opciones para adaptarlo al Diagrama de Estructura. Por ejemplo, en la plantilla se encuentran definidas tres opciones del submenú de **Maestros** del Menú de **Archivo** como se muestra en la Figura 5.13.

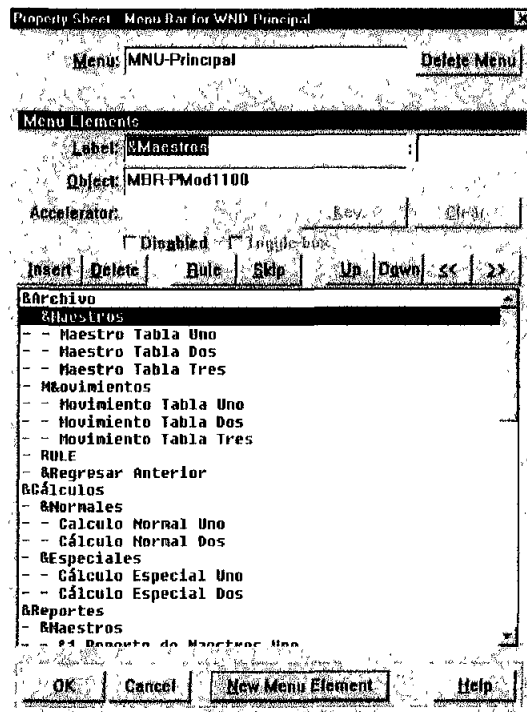
Figura 5.13. Opciones de la Plantilla del Submenú de Maestros.



Fuente: Ingeniería de Software.

- Además de las etiquetas de las opciones se cambiaron los nombres de las opciones de cada menú conforme al estándar para nombres de los elementos de los menús definido en el Anexo A 'Estándares de Progress'. Para hacer esto se acceso el cuadro de diálogo **Property Sheet** que se observa en la Figura 5.14 donde se realizaron los cambios.

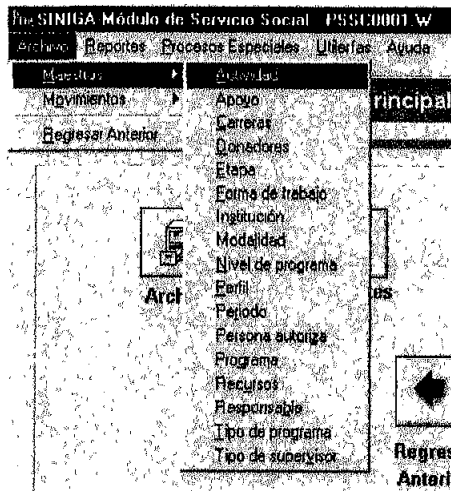
Figura 5.14. Cuadro de Diálogo Property Sheet.



Fuente: Progress v8.

Las opciones cambiadas se muestran en la Figura 5.15.

Figura 5.15. Opciones Cambiados del Submenú de Maestros.



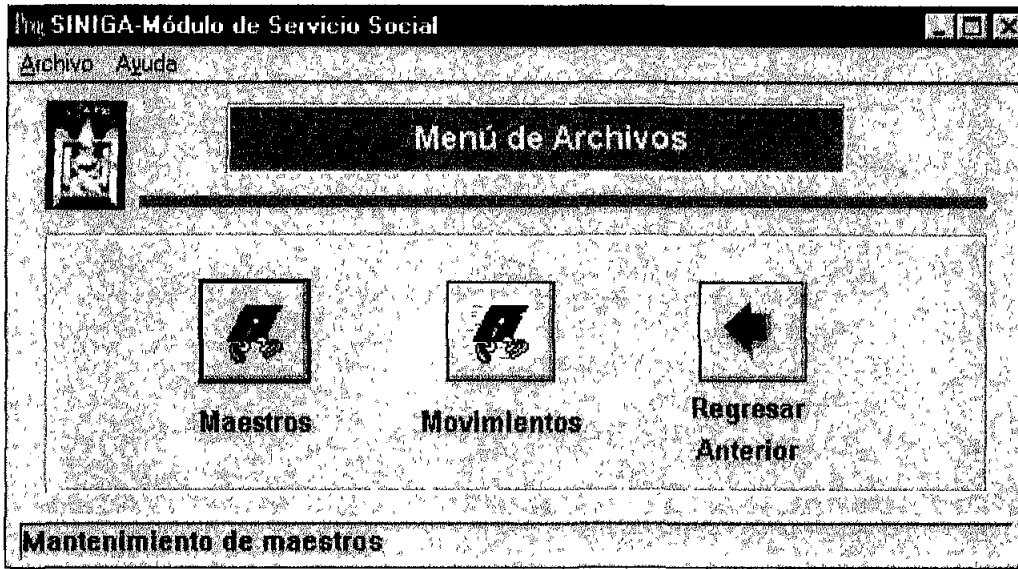
Fuente: Módulo de Servicio Social.

- 10.El código que define el nombre del archivo de ayuda que se va a desplegar para éste módulo se encuentra en la sección de **Definitions** (Definiciones) del Section Editor. Se cambió la ruta y el nombre del archivo de ayuda a C:\REVPROC\SINIGA\SSC\HLP\HSSC0001.HLP.
- 11.Por último se guardó la ventana con el nombre de PSSC0001.W.

5.3.2.2 Construcción del Menú de Archivos (PSSC1000.W)

La ventana terminada "Menú de Archivos" se puede observar en la Figura 5.16.

Figura 5.16. Ventana del Menú de Archivos.



Fuente: Módulo de Servicio Social.

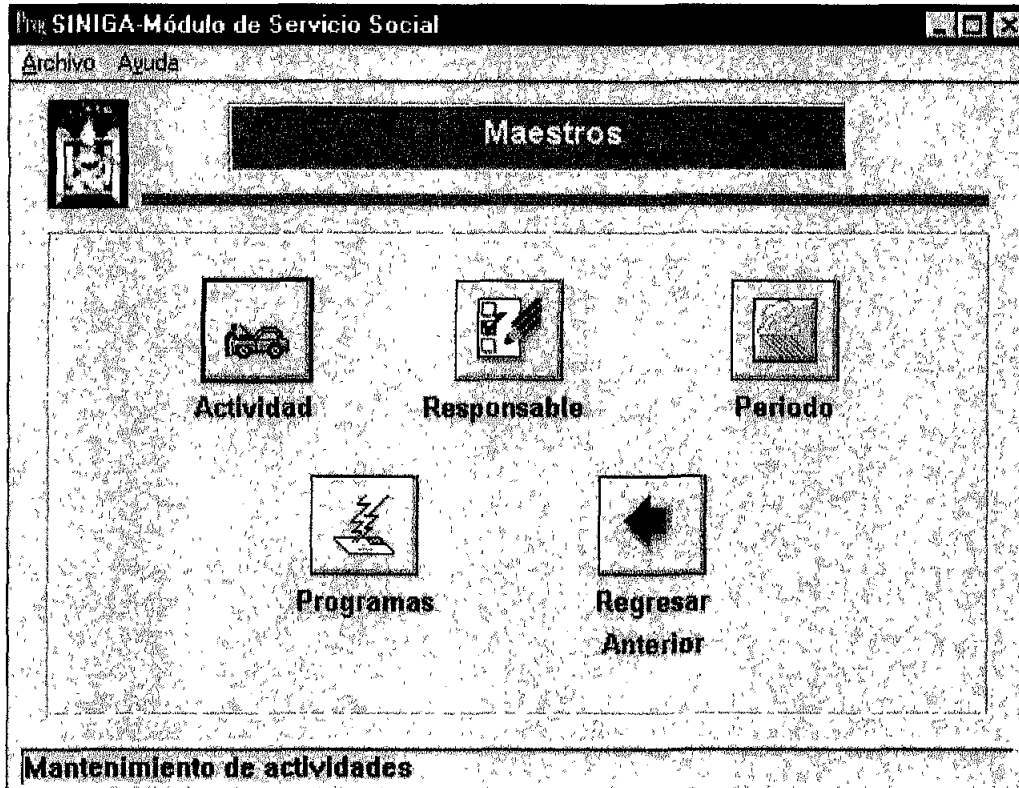
Para construir esta ventana se utilizó como plantilla el archivo PSSC0001, al cual se le hicieron las siguientes modificaciones:

1. Se cambió el título del encabezado a *Menú de Archivos*.
2. Se borrarón los siguientes botones:
 - **Procesos Especiales**
 - **Utilerías**
3. Se borrarón las siguientes opciones de la barra de menú:
 - **Reportes**
 - **Procesos Especiales**
 - **Utilerías**
4. Se cambiaron las etiquetas, los iconos y el código del llamado a los programas correspondientes de los siguientes botones:
 - **Archivo**
 - **Reportes**
5. Por último se guardó la ventana con el nombre de *PSSC1000.W*.

5.3.2.3 Construcción del Menú de Archivos Maestros (PSSC1100.W)

La ventana terminada "Maestros" se puede observar en la Figura 5.17.

Figura 5.17. Menú de Archivos Maestros.



Fuente: Módulo de Servicio Social.

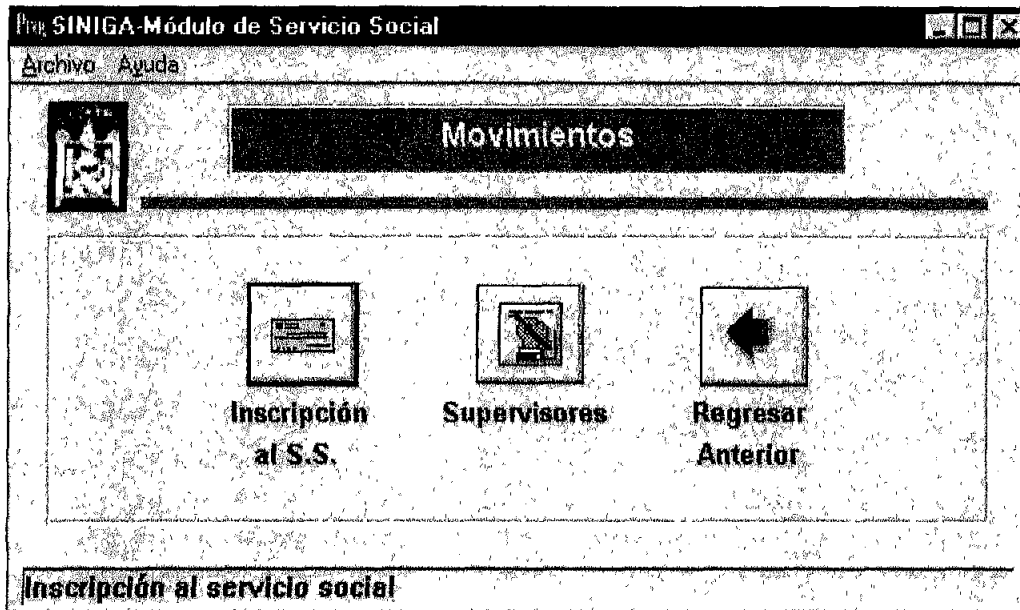
Para construir esta ventana se utilizó como plantilla el archivo PSSC1000, a la cual se le hicieron las siguientes modificaciones:

1. Se cambió el título del encabezado a **Maestros**.
2. Debido a la cantidad de archivos maestros, se eligieron los cuatro archivos más utilizados para ser llamados desde los botones. Para esto se copió dos veces el botón de **Maestros** cambiándoles después las etiquetas, los iconos y el código del llamado a los programas correspondientes.
3. Se deshabilitó la opción de **Movimientos** del menú de **Archivo**.
4. Por último se guardó la ventana con el nombre de **PSSC1100.W**.

5.3.2.4 Construcción del Menú de Archivos de Movimientos (PSSC1200.W)

La ventana terminada "Movimientos" se puede observar en la Figura 5 18.

Figura 5.18. Archivos de Movimientos.



Fuente: Módulo de Servicio Social.

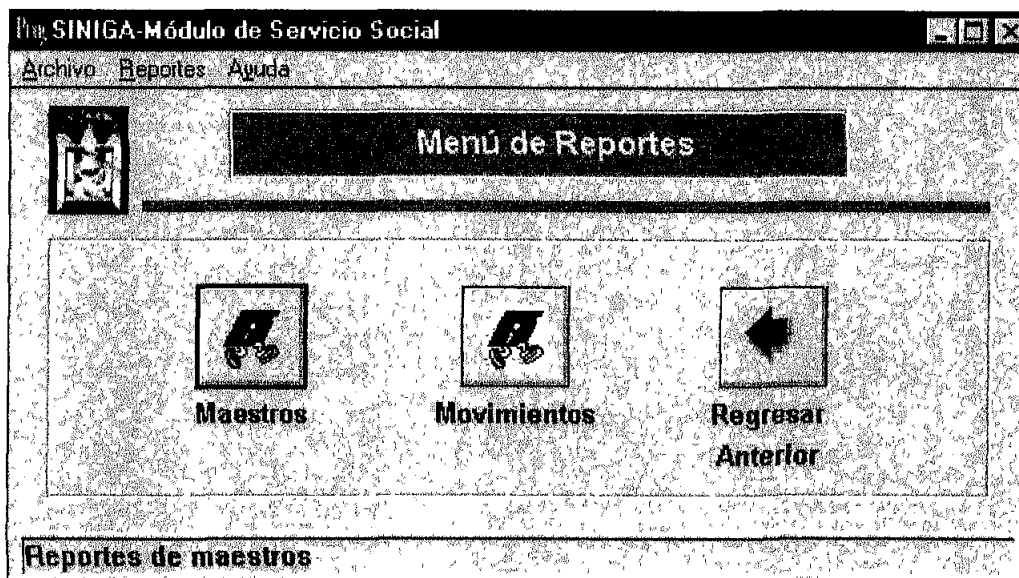
Para construir esta ventana se utilizó como plantilla el archivo PSSC1000, a la cual se le hicieron las siguientes modificaciones:

1. Se cambió el título del encabezado a **Movimientos**.
2. Se cambiaron las etiquetas, los iconos y el código del llamado a los programas correspondientes de los siguientes botones:
 - **Maestros**
 - **Movimientos**
3. Se deshabilitó la opción de **Maestros** del menú de **Archivo**.
4. Por último se guardó la ventana con el nombre de PSSC1200.W.

5.3.2.5 Construcción del Menú de Reportes (PSSC3000.W)

La ventana terminada "Menú de Reportes" se puede observar en la Figura 5.19.

Figura 5.19. Ventana del Menú de Reportes.



Fuente: Módulo de Servicio Social.

Para construir esta ventana, se utilizó como plantilla el archivo PSSC0001, al cual se le hicieron las siguientes modificaciones:

1. Se cambió el título del encabezado a **Menú de Archivos**.
2. Se borraron los siguientes botones:
 - **Procesos Especiales**
 - **Utilerías**
3. Se borraron las siguientes opciones de la barra de menú:
 - **Maestros y Movimientos del menú Archivo**
 - **Procesos Especiales**
 - **Utilerías**
4. Se cambiaron las etiquetas, los iconos y el código del llamado a los programas correspondientes de los siguientes botones:
 - **Archivo**
 - **Reportes**
5. Por último se guardó la ventana con el nombre de **PSSC3000.W**.

5.3.2.6 Construcción del Menú de Reportes de Maestros (PSSC3100.W)

La ventana terminada "Reportes de Maestros" se puede observar en la Figura 5.20.

Figura 5.20. Ventana del Menú de Reportes de Maestros.



Fuente: Módulo de Servicio Social.

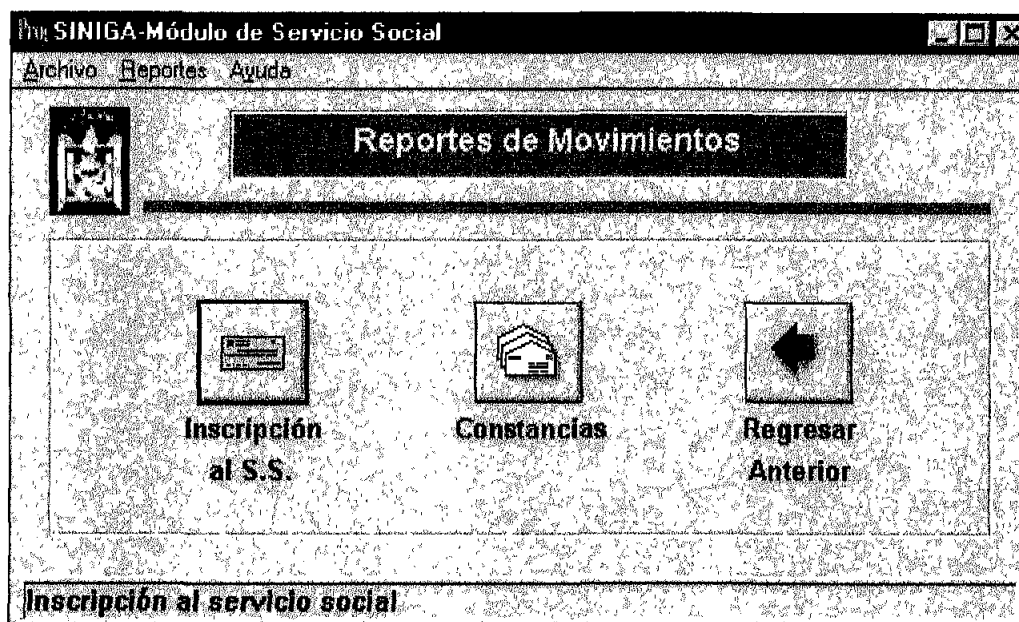
Para construir esta ventana, se utilizó como plantilla el archivo PSSC3000, a la cual se le hicieron las siguientes modificaciones:

1. Se cambió el título del encabezado a **Reportes de Maestros**.
2. Se cambiaron las etiquetas, los iconos y el código del llamado a los programas correspondientes de los siguientes botones:
 - **Maestros**
 - **Movimientos**
3. Se deshabilitó la opción de **Movimientos** del menú de **Reportes**.
4. Por último se guardó la ventana con el nombre de PSSC3100.W.

5.3.2.7 Construcción del Menú de Reportes de Movimientos (PSSC3200.W)

La ventana terminada "Reportes de Movimientos" se puede observar en la Figura 5.21.

Figura 5.21. Ventana del Menú de Reportes de Movimientos.



Fuente: Módulo de Servicio Social.

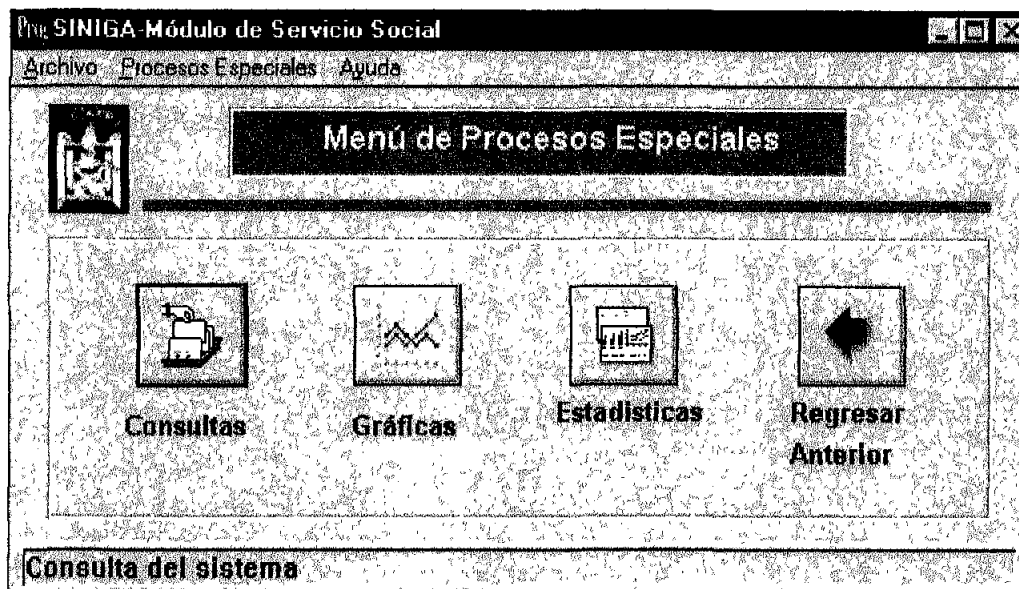
Para construir esta ventana, se utilizó como plantilla el archivo PSSC3000, a la cual se le hicieron las siguientes modificaciones:

1. Se cambió el título del encabezado a Reportes de Movimientos.
2. Se cambiaron las etiquetas, los iconos y el código del llamado a los programas correspondientes de los siguientes botones:
 - Maestros
 - Movimientos
3. Se deshabilitó la opción de Maestros del menú de Reportes.
4. Por último se guardó la ventana con el nombre de PSSC3200.W.

5.3.2.8 Construcción del Menú de Procesos Especiales (PSSC4000.W)

La ventana terminada "Menú de Procesos Especiales" se puede observar en la Figura 5.22.

Figura 5.22. Ventana del Menú de Procesos Especiales.



Fuente: Módulo de Servicio Social.

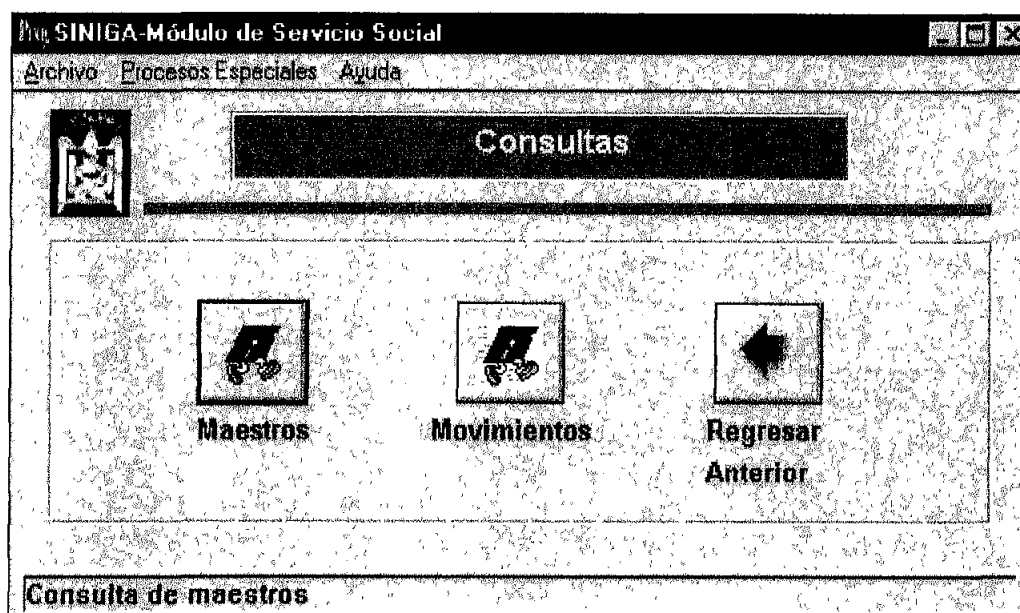
Para construir esta ventana, se utilizó como plantilla el archivo PSSC0001, al cual se le hicieron las siguientes modificaciones:

1. Se cambió el título del encabezado a Menú de Procesos Especiales.
2. Se borró el botón de **Utilerías**.
3. Se borraron las siguientes opciones de la barra de menú:
 - **Maestros y Movimientos** del menú **Archivo**
 - **Reportes**
 - **Utilerías**
4. Se cambiaron las etiquetas, los iconos y el código del llamado a los programas correspondientes de los siguientes botones:
 - **Archivos**
 - **Reportes**
 - **Procesos Especiales**
5. Por último se guardó la ventana con el nombre de PSSC4000.W.

5.3.2.9 Construcción del Menú de Consultas (PSSC4100.W)

La ventana terminada "Consultas" se puede observar en la Figura 5.23.

Figura 5.23. Ventana del Menú de Consultas.



Fuente: Módulo de Servicio Social.

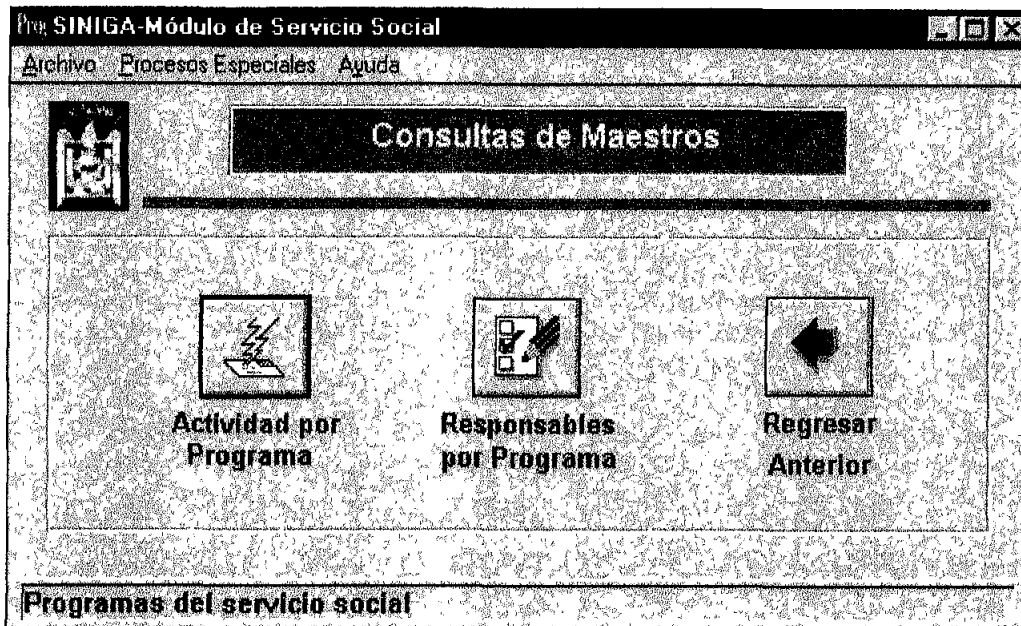
Para construir esta ventana, se utilizó como plantilla el archivo PSSC4000, a la cual se le hicieron las siguientes modificaciones:

1. Se cambió el título del encabezado a *Consultas*.
2. Se borró el botón de **Estadísticas**.
3. Se cambiaron las etiquetas, los iconos y el código del llamado a los programas correspondientes de los siguientes botones:
 - **Consultas**
 - **Gráficas**
4. Se deshabilitaron las opciones de **Gráficas** y **Estadísticas** del menú **Procesos Especiales**.
5. Por último se guardó la ventana con el nombre de *PSSC4100.W*.

5.3.2.10 Construcción del Menú de Consultas de Maestros (PSSC4110.W)

La ventana terminada "Consultas de Maestros" se puede observar en la Figura 5.24.

Figura 5.24. Ventana de Menú de Consultas de Maestros.



Fuente: Módulo de Servicio Social.

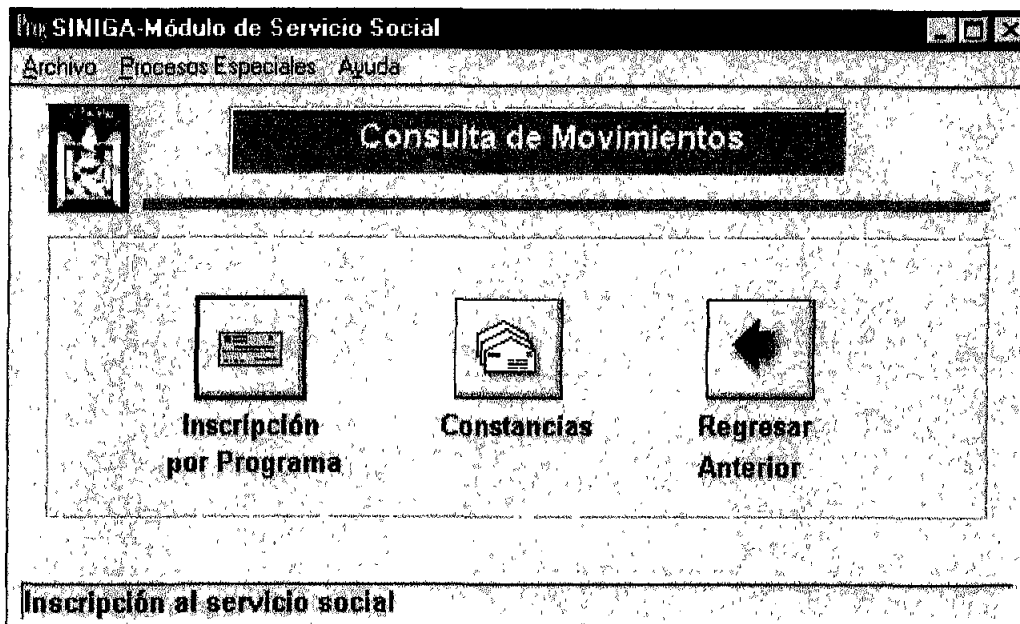
Para construir esta ventana, se utilizó como plantilla el archivo PSSC4100, a la cual se le hicieron las siguientes modificaciones:

1. Se cambió el título del encabezado a *Consultas de Maestros*.
2. Se cambiaron las etiquetas, los iconos y el código del llamado a los programas correspondientes de los siguientes botones:
 - **Maestros**
 - **Movimientos**
3. Se deshabilitó la opción **Movimientos** del submenú **Consultas** del menú **Procesos Especiales**.
4. Por último se guardó la ventana con el nombre de PSSC4110.W.

5.3.2.11 Construcción del Menú de Consulta de Movimientos (PSSC4120.W)

La ventana terminada "Consulta de Movimientos" se puede observar en la Figura 5 25

Figura 5.25. Ventana de Menú de Consulta de Movimientos.



Fuente: Módulo de Servicio Social.

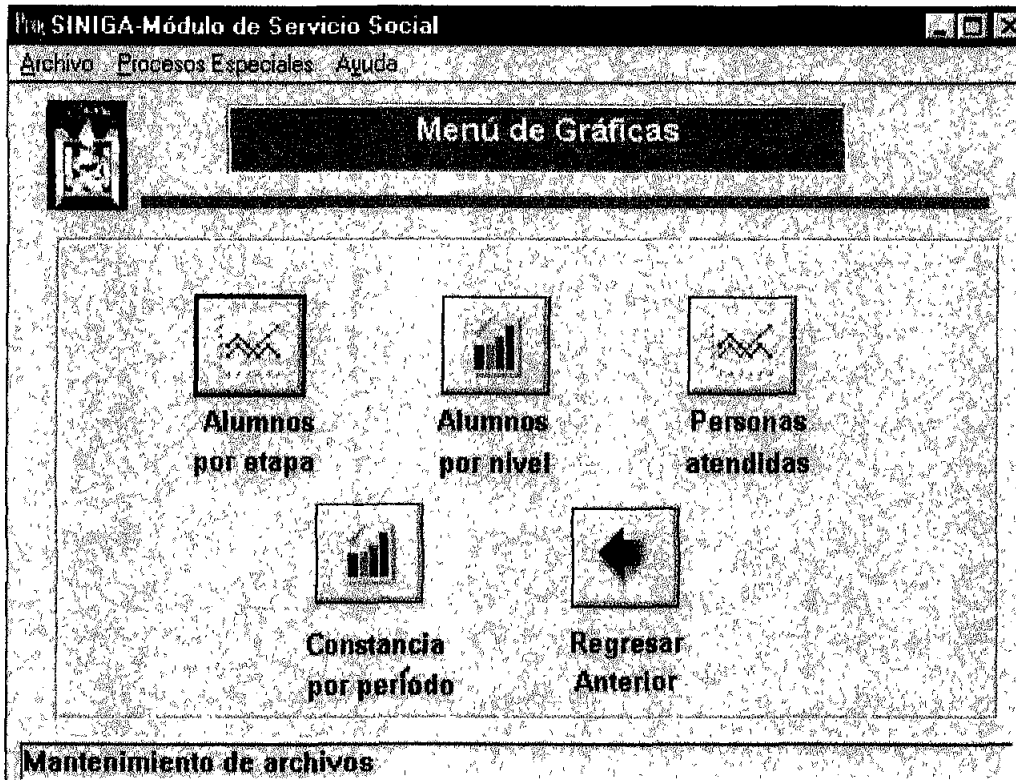
Para construir esta ventana, se utilizó como plantilla el archivo PSSC4100, a la cual se le hicieron las siguientes modificaciones:

1. Se cambió el título del encabezado a Consultas.
2. Se cambiaron las etiquetas, los iconos y el código del llamado a los programas correspondientes de los siguientes botones:
 - Maestros
 - Movimientos
3. Se deshabilitó la opción de Maestros del submenú Consultas del menú Procesos Especiales.
4. Por último se guardó la ventana con el nombre de PSSC4120.W.

5.3.2.12 Construcción del Menú de Gráficas (PSSC4200.W)

La ventana terminada "Menú de Gráficas" se puede observar en la Figura 5.26.

Figura 5.26. Ventana de Menú de Gráficas.



Fuente: Módulo de Servicio Social.

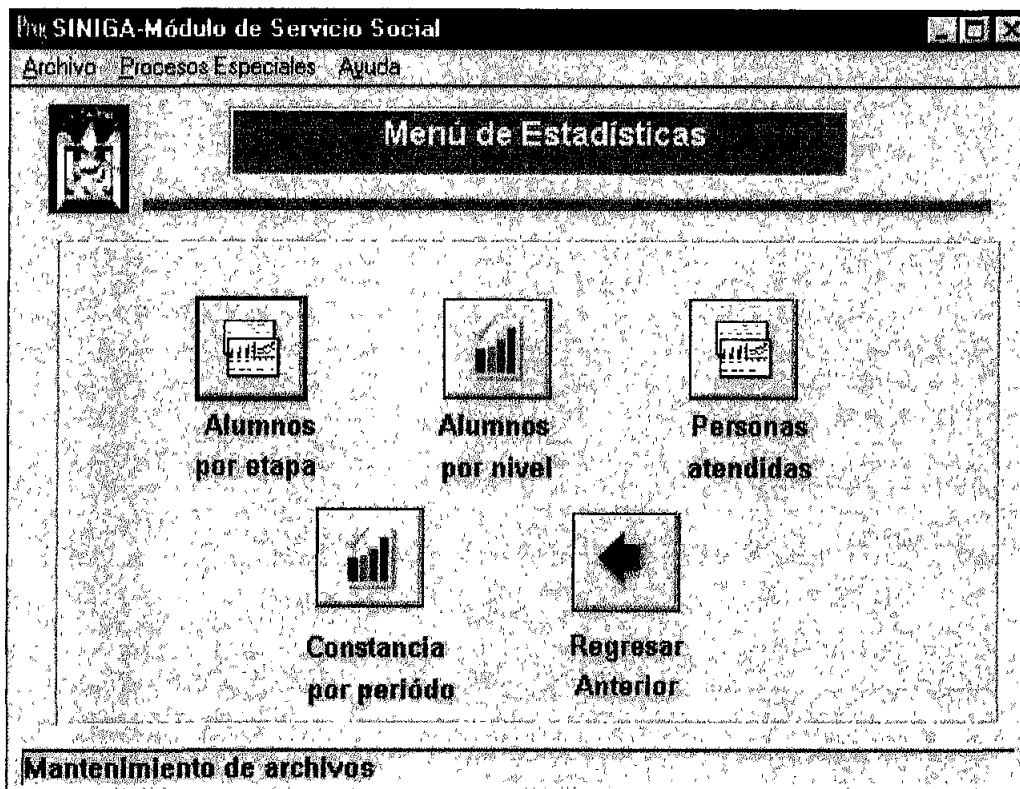
Para construir esta ventana, se utilizó como plantilla el archivo PSSC4000, a la cual se le hicieron las siguientes modificaciones:

1. Se cambió el título del encabezado a Menú de Gráficas.
2. Se borró el botón de **Estadísticas**.
3. Se cambiaron las etiquetas, los iconos y el código del llamado a los programas correspondientes de los siguientes botones:
 - **Maestros**
 - **Movimientos**
4. Se deshabilitaron las opciones de **Consultas** y **Estadísticas** de la barra de menú.
5. Por último se guardó la ventana con el nombre de PSSC4200.W.

5.3.2.13 Construcción del Menú de Estadísticas (PSSC4300.W)

La ventana terminada "Menú de Estadísticas" se puede observar en la Figura 5.27

Figura 5.27. Ventana del Menú de Estadísticas.



Fuente: Módulo de Servicio Social.

Para construir esta ventana, se utilizó como plantilla el archivo PSSC4000, a la cual se le hicieron las siguientes modificaciones:

1. Se cambió el título del encabezado a Menú de Estadísticas.
2. Se borró el botón de **Estadísticas**.
3. Se cambiaron las etiquetas, los iconos y el código del llamado a los programas correspondientes de los siguientes botones:
 - **Consultas**
 - **Gráficas**
4. Se deshabilitaron las opciones de **Consultas** y **Gráficas** del menú **Procesos Especiales**.
5. Por último se guardó la ventana con el nombre de PSSC4300.W.

5.3.3 Construcción de la Ventana de ABC del Maestro de Actividades (PSSC1110.W)

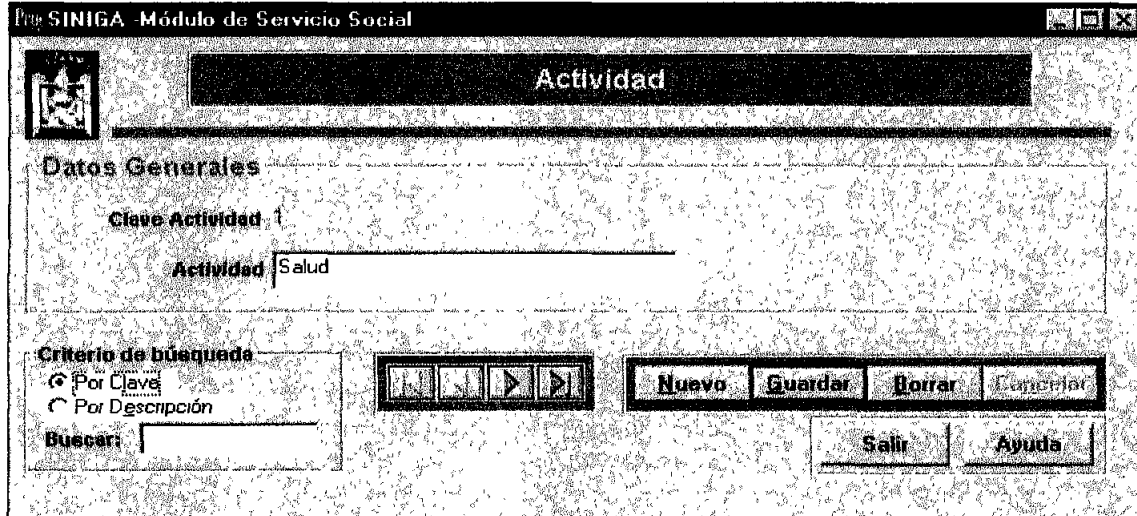
La mayoría de las ventanas de Altas, Bajas y Cambios (ABC) se componen de los siguientes elementos:

- Un SmartViewer, el cual permite visualizar y actualizar los campos.
- Un SmartQuery que provee los registros al SmartViewer
- Un archivo de inclusión (include) que contiene información necesaria para el funcionamiento de la ventana.
- Un SmartPanel de Navegación para recorrer los registros.
- Un SmartPanel de Actualización para actualizar los registros.

Las ventanas de ABC de Maestros dan mantenimiento a los archivos maestros de la base de datos. Estas ventanas son las más sencillas ya que rara vez el archivo al que se da mantenimiento necesita validarse contra otro archivo maestro.

La ventana terminada de ABC de Actividad se muestra en la Figura 5.28.

Figura 5.28. Ventana de ABC de Actividad.



Fuente: Módulo de Servicio Social.

Antes de empezar a construir la ventana fue necesario crear los siguientes archivos:

- El archivo de inclusión, que contiene las definiciones de variables utilizadas y procedimientos de uso general (ABC-ACTV . I).

- El SmartQuery de la tabla Actividad (Q-ACTV.W)
- El SmartViewer de la tabla Actividad (V-ACTV.W).

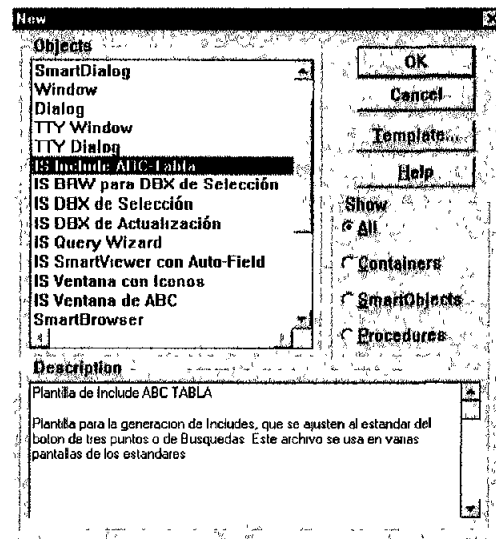
Después de crear estos archivos se procedió a la integración de todos ellos en la ventana de ABC. A continuación se explica la construcción de cada uno de ellos.

5.3.3.1 Creación del Archivo de Inclusión de Actividades (ABC-ACTV.I)

El archivo de inclusión contiene la información que necesitan las ventanas que utilizan la Ingeniería de Software desarrollada. Para crear este archivo se llevaron a cabo los siguientes pasos:

1. Se hizo click en el botón **New** de la barra de botones del UIB, lo cual despliega el cuadro de diálogo **New** que se observa en la Figura 5.29, y de éste se eligió la plantilla IS Include ABC-Tabla

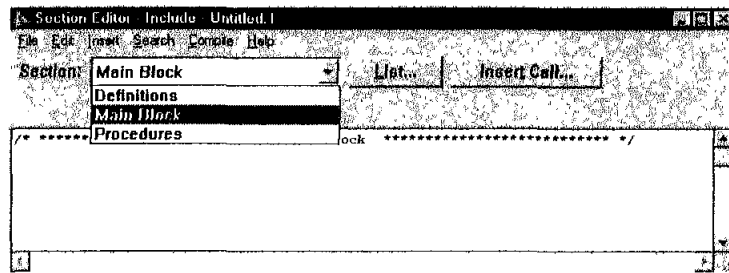
Figura 5.29. Selección de la Plantilla IS Include ABC-Tabla.



Fuente: Progress v8.

2. Inmediatamente después, aparece el Section Editor (Editor de Secciones) que se visualiza en la Figura 5.30. Ahí se selecciona la sección de **Definitions** (Definiciones).

Figura 5.30. Ventana del Editor de Sección.



Fuente: Progress v8.

3. La sección de definiciones contiene las sentencias de código que se muestran en la Figura 5.31.

Figura 5.31. Código de la Sección de Definiciones.

```

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

DEFINE BUFFER Tabla FOR NomTabla.
DEFINE VARIABLE VIWArchivo AS CHAR INITIAL "VModTabla".

&GLOBAL-DEFINE ClaveCampo CampoClave
&GLOBAL-DEFINE DsconCampo CampoDescripcion
&GLOBAL-DEFINE VIWHandle h_V-Tabla
&GLOBAL-DEFINE BRWHandle h_Bn-Tabla
&GLOBAL-DEFINE QRYHandle h_Q-Tabla

DEFINE VARIABLE BRWArchivo AS CHAR INITIAL "BModTabla".
&GLOBAL-DEFINE FifClave FIFCampoClave
&GLOBAL-DEFINE FifDscon FIFCampoDescripcion
    
```

Fuente: Ingeniería de Software.

4. Las palabras que aparecen en negritas se sustituyeron por la información relacionada con el módulo, tal como se muestra en la Figura 5.32.

Figura 5.32. Código Adaptado del Include ABC-ACT.I.

```

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

DEFINE BUFFER Tabla FOR SSCActividad.
DEFINE VARIABLE VIWArchivo AS CHAR INITIAL "VSSCACT".

&GLOBAL-DEFINE ClaveCampo Cve-Actividad
&GLOBAL-DEFINE DsconCampo Dscon-Actividad
&GLOBAL-DEFINE VIWHandle h_V-ACT
&GLOBAL-DEFINE BRWHandle h_B-ACT
&GLOBAL-DEFINE QRYHandle h_Q-ACT

DEFINE VARIABLE BRWArchivo AS CHAR INITIAL "BSSCACT".
&GLOBAL-DEFINE FifClave FIFCampoClave
&GLOBAL-DEFINE FifDscon FIFCampoDescripcion

```

Fuente: Módulo de Servicio Social.

5. Por último se guardó el archivo como ABC-ACTV, I.

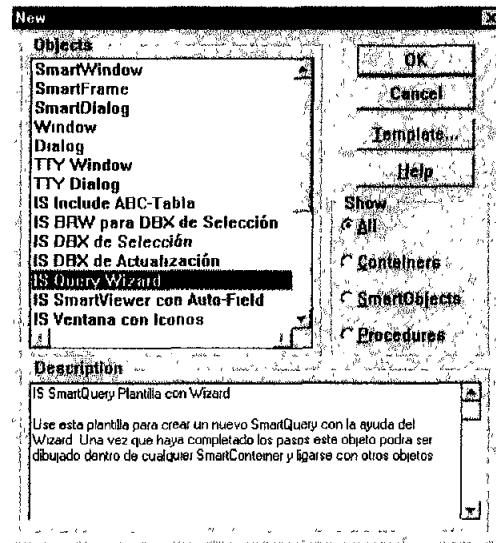
5.3.3.2 Creación del SmartQuery de Actividad (Q-ACT.W)

Un SmartQuery es un SmartObject que obtiene los registros que cumplen el criterio de selección. El SmartQuery interactúa con el SmartViewer para proveer a este los registros y con el SmartPanel de Navegación para recorrerlos. A continuación se explica la construcción del SmartQuery Q-ACTV.W.

Los pasos que se llevaron a cabo son los siguientes:

1. Se hizo click en el botón **New** de la barra de herramientas del UIB. Al hacer esto se despliega el cuadro de diálogo **New** que se observa en la Figura 5.33. En este cuadro diálogo se eligió la plantilla IS Query Wizard.

Figura 5.33. Selección de la Plantilla IS Query Wizard.

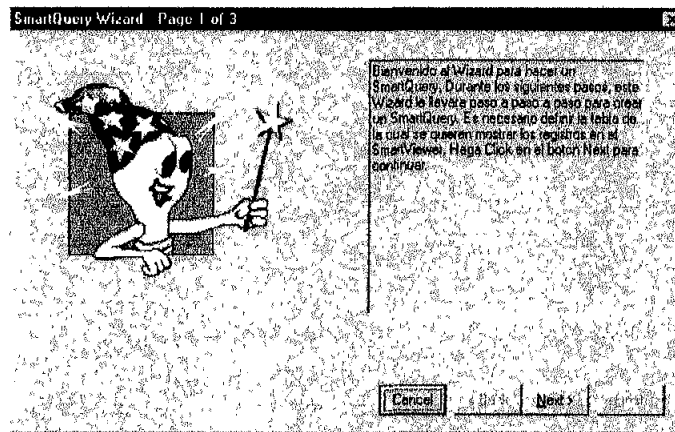


Fuente: Progress v8.

2. A continuación aparece el primer cuadro de diálogo del SmartQuery Wizard, el cual se observa en la Figura 5.34.

Los Wizards son un herramientas de Progress que facilitan la creación de SmartObjects y que pueden personalizarse como en este caso. El SmartQuery Wizard le guía por una serie de pasos involucrados en la creación del SmartQuery; ofrece información vital acerca de cada paso y lo mantiene informado del lugar en el proceso donde se encuentra. Una característica especialmente útil del SmartQuery Wizard es que le permite, en cualquier punto, regresar y modificar el trabajo hecho en los pasos anteriores.

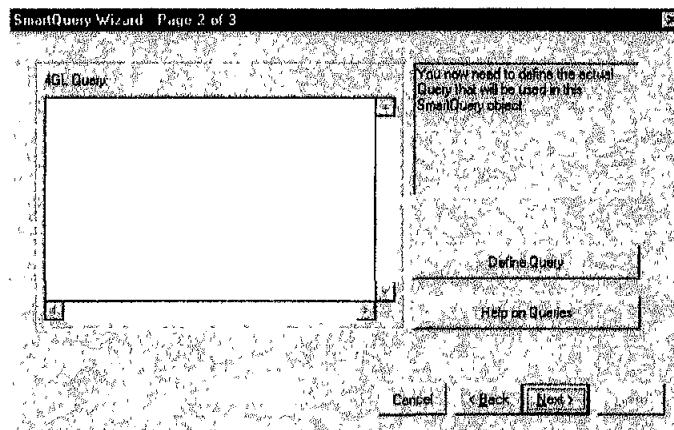
Figura 5.34. SmartQuery Wizard de Actividad. Página 1 de 3.



Fuente: Ingeniería de Software.

3. Se continua con el siguiente cuadro de diálogo del Wizard que se observa en la Figura 5.35, oprimiendo el botón **Next**. En este cuadro de diálogo se pasa a definir el Query haciendo click en el botón **Define Query**.

Figura 5.35. SmartQuery Wizard de Actividad. Página 2 de 3.

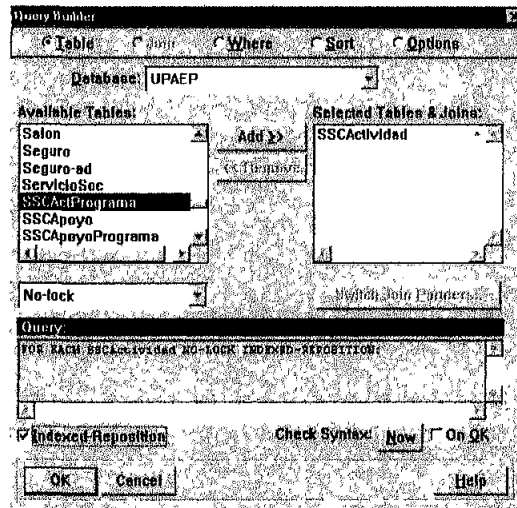


Fuente: Ingeniería de Software.

2. Los anterior despliega el cuadro de diálogo **Query Builder** que se observa en la Figura 5.36 donde se especificó la tabla *SSCActividad*, a la cual se le daría mantenimiento. Para mejorar la velocidad de búsqueda de los registros, se seleccionó la casilla **Indexed-Reposition**.

3. Por último se oprimió el botón de **OK**. Esta última acción regresa al usuario al cuadro de diálogo de la Figura 5.35.

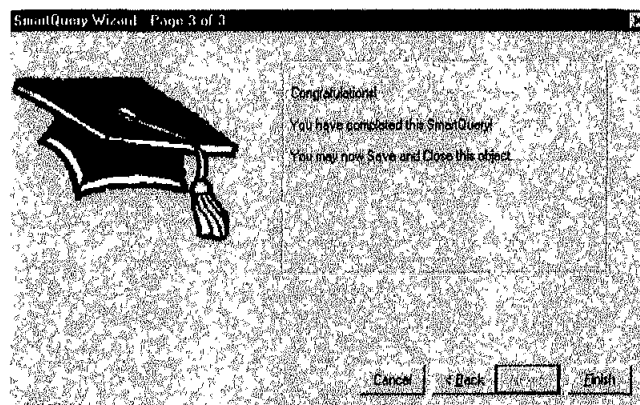
Figura 5.36. Cuadro de Diálogo Query Builder de Actividad.



Fuente: Progress v8.

4. Se continuó con el siguiente cuadro de diálogo del Wizard que se observa en la Figura 5.37 oprimiendo el botón **Next**. Este cuadro de diálogo notifica la terminación de la construcción del SmartQuery.

Figura 5.37. SmartQuery Wizard de Actividad. Página 3 de 3.



Fuente: Ingeniería de Software.

5. Por último se guardó el archivo como Q-ACTV.W.

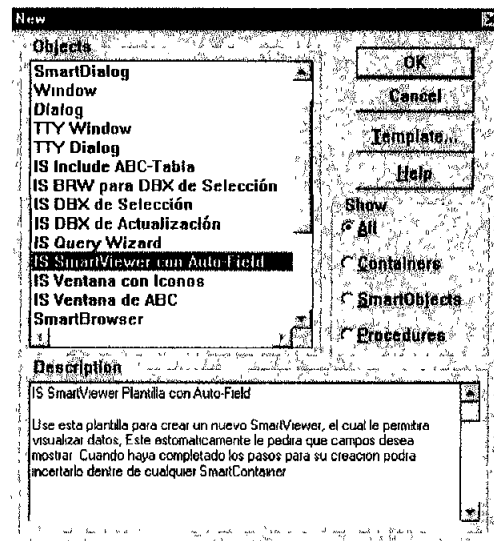
5.3.3.3 Creación del SmartViewer de Actividad (V-ACT.W)

Los SmartViewers son SmartObjects que despliegan los campos de las tablas y permiten modificar los registros a través de la interacción con los SmartPanels de Actualización..

Para construir el SmartViewer de Actividad se llevaron a cabo los siguientes pasos:

1. Se hizo click en el botón **New** de la barra de herramientas del UIB. Al continuación se despliega el cuadro de diálogo **New** que se observa en la Figura 5.38. En este cuadro de diálogo se eligió la plantilla IS SmartViewer con Auto-Field.

Figura 5.38. Selección de la Plantilla IS SmartViewer con Auto-Field.

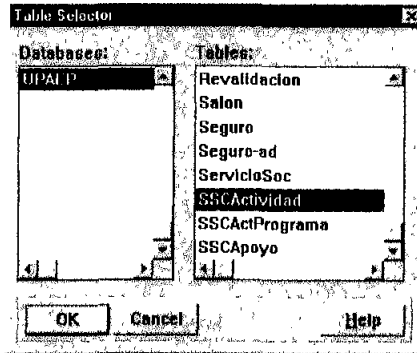


Fuente: Progress v8.

La característica de esta plantilla es que permite seleccionar la tabla y los campos desde el momento en que se abre; para esto se despliegan varios cuadros de diálogo como se muestran a continuación.

- Se despliega el cuadro de diálogo **Table Selector** donde se seleccionó la tabla **SSCActividad** que se observa en la Figura 5.39. A continuación se oprimió el botón **OK**.

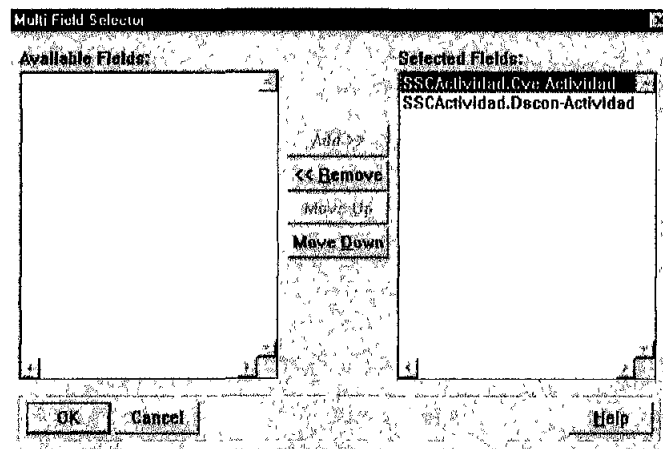
Figura 5.39. Table Selector, Tabla SSCActividad.



Fuente: Progress v8.

- Inmediatamente después aparece el cuadro de diálogo **Multi-Field Selector** donde se seleccionaron los campos que tendría el SmartViewer: **Cve-Actividad** y **Dscon-Actividad**. Esto se observa en la Figura 5.40. Para terminar se hizo click en el botón **OK**.

Figura 5.40. Multi-Field Selector de la Tabla SSCActividad.



Fuente: Progress v8.

- Una vez que los campos quedaron colocados en el SmartViewer se procedió a aplicar los estándares definidos en el Anexo B 'Estándares de Ventana VGA'.

5. Por último se agregó el Archivo de Inclusión para el incremento de la secuencia: IS_SEC.I. Este cambio se realizó en el evento **ENTRY** del fill-in **UPAEP.SSCActividad.Dscon-Actividad**. En la Figura 5.41 se observa el llamado al archivo de inclusión con los parámetros necesarios.

Figura 5.41. Llamado al Archivo de Inclusión.

```
{c:\revproc\gnr\i\is_sec.i &Tabla = SSCActividad
                        &Secuencia =SSSC-Cve-Actividad
                        &Campo = Cve-Actividad
}
```

Fuente: Módulo de Servicio Social.

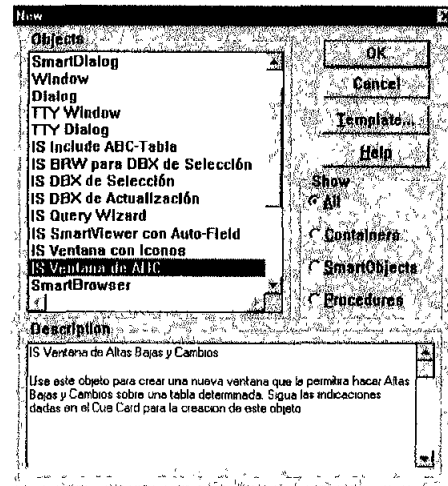
6. Por último se guardó el archivo como V-ACTV.W.

5.3.3.4 Integración de los Objetos en la Ventana de ABC del Maestro de Actividades (PSSC1110.W)

Una vez construidos los archivos anteriores ya se puede elaborar la ventana de ABC de Actividades. Esta ventana se generó con una de las plantillas especialmente desarrolladas para tal propósito. A continuación se detallan los pasos que se llevaron a cabo para su creación:

1. Se hizo click en botón **New** de la barra de herramientas del UIB, lo cual despliega el cuadro de diálogo **New** que se observa en la Figura 5.42 y de este se eligió la plantilla IS Ventana de ABC.

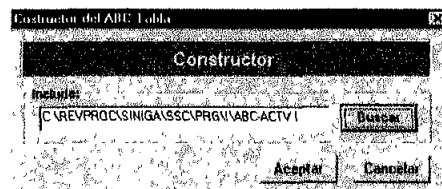
Figura 5.42. Selección de la Plantilla IS Ventana de ABC.



Fuente: Progress v8.

2. A continuación se despliega el cuadro de dialogo **Constructor Del ABC-Tabla** que se observa en la Figura 5.43, donde se ingresó el nombre y la ruta del Archivo de Inclusión: C:\REVPROC\SINIGA\SSC\PRG\I\ABC-ACTV.I.

Figura 5.43. Constructor del Include ABC-ACTV.I



Fuente: Ingeniería de Software.

3. Como los SmartObjects que contenía la plantilla no eran los que se necesitaban, se cambiaron sus instancias por la de los SmartObjects correctos:
 - Se cambió la instancia del SmartViewer por V-ACTV.W.
 - Se cambió la instancia del SmartQuery por Q-ACTV.W.
4. Se cambió el título de la ventana a SINIGA-Módulo de Servicio Social.
5. Se cambió el título del encabezado a Actividades.
6. Por último se guardó el archivo con el nombre de PSSC1110.W.

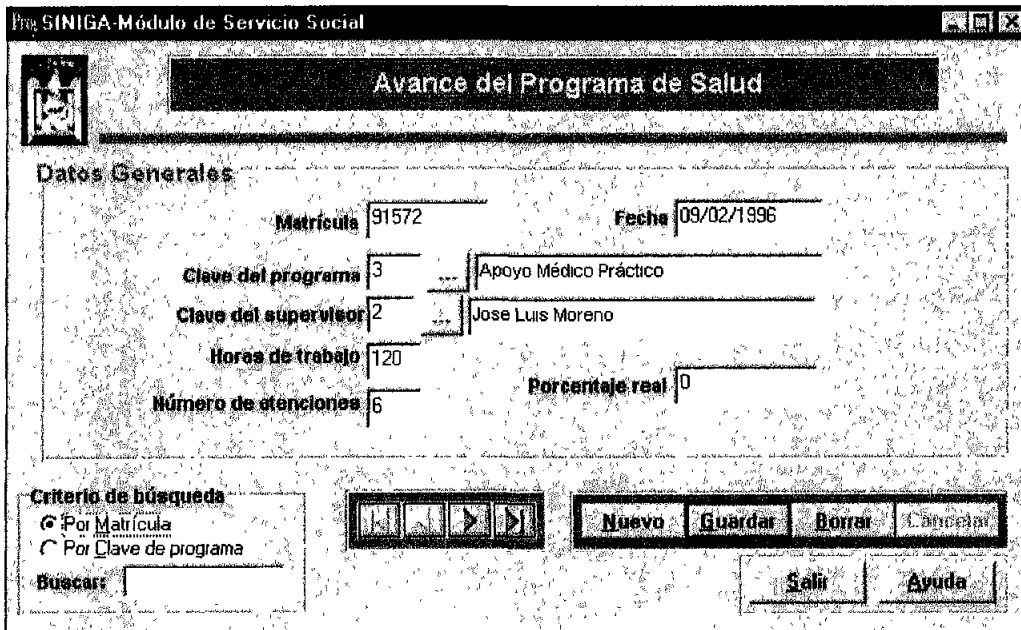
5.3.4 Construcción de una Ventana de ABC de Movimientos (PSSC1210.W)

Las ventanas de ABC de Movimientos dan mantenimiento a archivos de movimientos, es decir, archivos que se actualizan frecuentemente. Estos archivos de movimientos necesitan datos de los archivos maestros, lo cual significa que en una ventana de ABC de Movimientos se tenga la opción de acceder a ventanas que presenten la información de un archivo maestro. En esas ventanas, además de visualizar, se puede seleccionar el registro y que éste sea regresado a la ventana de Movimientos.

Para lograr lo anterior se llevó a cabo lo siguiente: se colocó un botón de tres puntos entre los campos de clave y descripción. Al momento de hacer click en él, se llama a un cuadro de diálogo de selección del archivo maestro en cuestión. La creación del cuadro de diálogo de selección se detalla en la Sección 5.3.4.5.

Como esta característica se repetía en las ventanas de ABC de Movimientos, se generaron los archivos de inclusión con el código necesario el debido funcionamiento de dichas ventanas. También se modificó la Paleta de Objetos para que incluyera el botón de 3 puntos con todos los estándares necesarios, y así pudiera seleccionarse y colocarse en la ventana; evitando al desarrollador su creación por cada nueva ventana de ABC. La ventana terminada se puede observar en la Figura 5.44.

Figura 5.44. Ventana de ABC de Movimientos.



Fuente: Módulo de Servicio Social.

Antes de empezar a construir la ventana fue necesario crear los siguientes archivos:

- El archivo de inclusión que contiene las definiciones y el código necesario para el funcionamiento de la ventana.
- El SmartQuery de la tabla SSCAvancePrgSalud.
- El SmartViewer de la tabla SSCAvancePrgSalud.

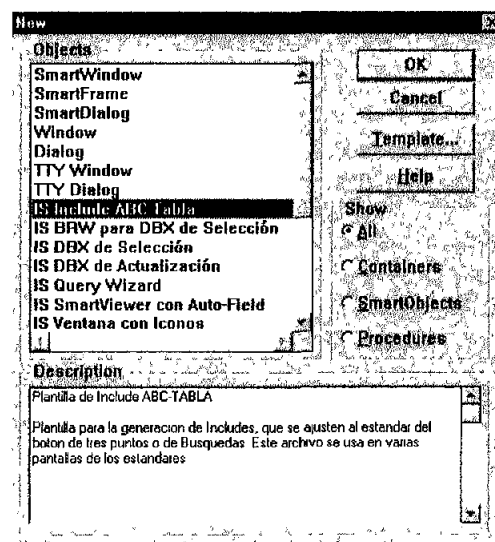
A continuación se detalla la creación de cada uno de los archivos anteriores.

5.3.4.1 Creación del Archivo de Inclusión de Avance de Programa de Salud (ABC-AVPS.I)

El archivo de inclusión contiene la información que necesitan las ventanas desarrolladas por Ingeniería de Software. Para crear este archivo se llevaron a cabo los siguientes pasos:

1. Se hizo click en el botón **New** de la barra de botones del UIB. Al hacer esto se despliega el cuadro de diálogo **New** que se observa en la Figura 5.45 y de este se eligió la plantilla IS Include ABC-Tabla.

Figura 5.45. Selección de la Plantilla IS Include ABC-Tabla.



Fuente: Progress v8.

2. Se accesó la sección de Definitions del Section Editor.
3. La sección de definiciones contiene las sentencias de código que se pueden observar en la Figura 5.27. Las palabras que aparecen en negritas se sustituyeron por la información relacionada con el módulo, tal como se muestra en la Figura 5.46.

Figura 5.46. Código Adaptado del Include ABC-AVPS.I.

```

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

DEFINE BUFFER Tabla FOR SSCAvancePrgSalud.
DEFINE VARIABLE VIWArchivo AS CHAR INITIAL "VSSAVPS".

&GLOBAL-DEFINE ClaveCampo Matricula.
&GLOBAL-DEFINE DsconCampo Dscon-Actividad
&GLOBAL-DEFINE VIWHandle h_V-AVPS
&GLOBAL-DEFINE BRWHandle h_B-AVPS
&GLOBAL-DEFINE QRYHandle h_Q-AVPS

DEFINE VARIABLE BRWArchivo AS CHAR INITIAL "BSSCAVPS".
&GLOBAL-DEFINE FifClave UPAEP.SSCAvancePrgSalud.Matricula.
&GLOBAL-DEFINE FifDscon FIF-Cve-Programa.

```

Fuente: Módulo de Servicio Social.

4. Por último se guardó el archivo como ABC-AVPS.I.

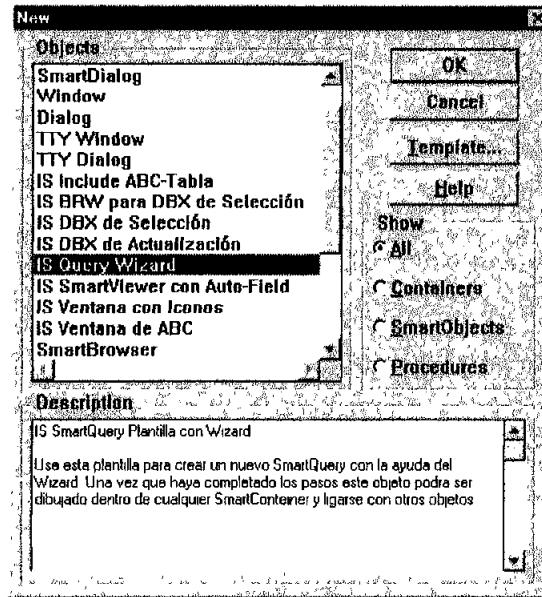
5.3.4.2 Creación del SmartQuery de Avance de Programa de Salud (Q-AVPS.W)

Al igual que el SmartQuery de Actividad, el SmartQuery de Avance de Programa de Salud interactúa con las ventanas desarrolladas por Ingeniería de Software. A continuación se explica la construcción del SmartQuery.

Los pasos que se llevaron a cabo fueron los siguientes:

1. Se hizo click en el botón **New** de la barra de herramientas del UIB. Al hacer esto se despliega el cuadro de diálogo **New** que se observa en la Figura 5.47 y de éste se eligió la plantilla IS Query Wizard.

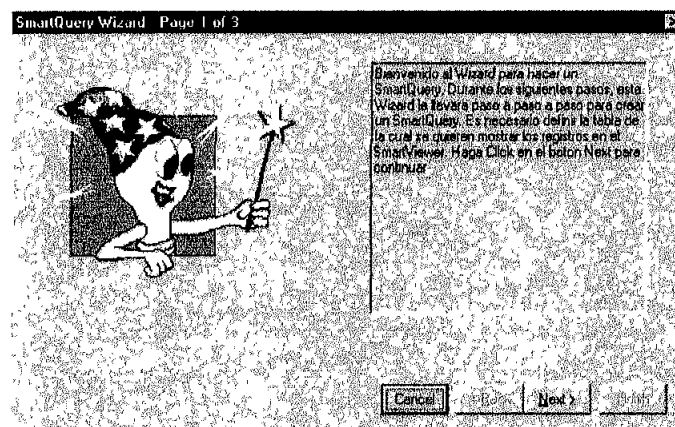
Figura 5.47. Selección de la Plantilla IS Query Wizard.



Fuente: Progress v8.

2. A continuación aparece el primer cuadro de diálogo del **Smartquery Wizard Página 1 de 3**, el cual se observa en la Figura 5.48.

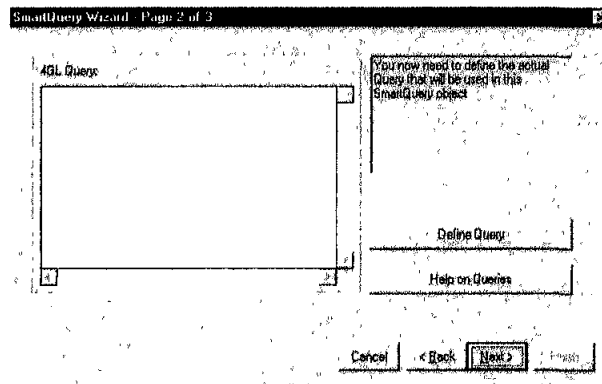
Figura 5.48. SmartQuery Wizard de Avance de Programa de Salud. Página 1 de 3.



Fuente: Ingeniería de Software.

- Se continuó con el siguiente cuadro de diálogo del Wizard (Figura 5.49) oprimiendo el botón **Next**. En este cuadro de diálogo se pasó a definir el Query, haciendo click en el botón **Define Query**.

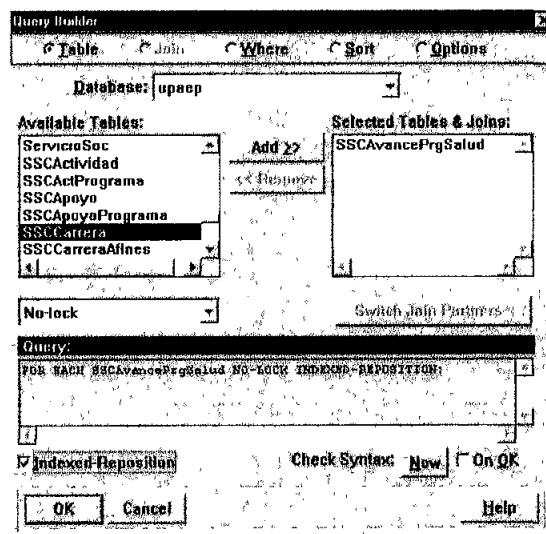
Figura 5.49. SmartQuery Wizard de Avance de Programa de Salud. Página 2 de 3



Fuente: Ingeniería de Software.

- Lo anterior despliega el cuadro de diálogo **Query Builder** que se observa en la Figura 5.50 donde se eligió la tabla *SSCAvancePrgSalud*, a la cual se le daría mantenimiento. Para mejorar la velocidad de búsqueda de los registros, se seleccionó la casilla **Indexed-Reposition**.

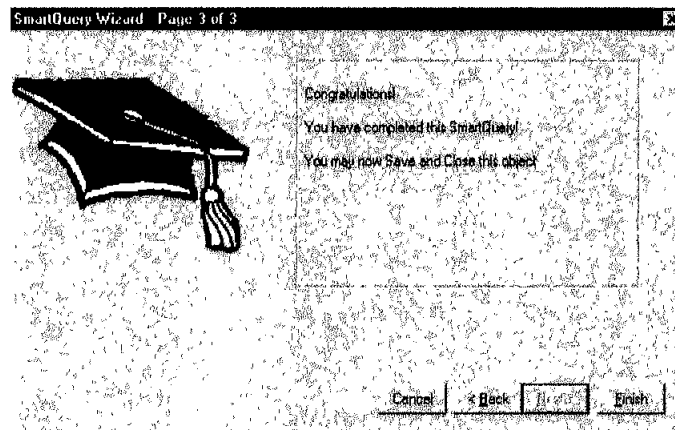
Figura 5.50. Cuadro de Diálogo Query Builder de Avance de Programa de Salud.



Fuente: Progress v8.

5. Por último se oprimió el botón de **OK**. Con esta última acción, se regresa al cuadro de diálogo de la Figura 5.49.
6. Se continuó con el siguiente cuadro de diálogo del Wizard que se observa en la Figura 5.51 oprimiendo el botón **Next**, el cual notifica la terminación de la construcción del SmartQuery.

Figura 5.51. SmartQuery Wizard de Avance de Programa de Salud. Página 3 de 3.



Fuente: Ingeniería de Software.

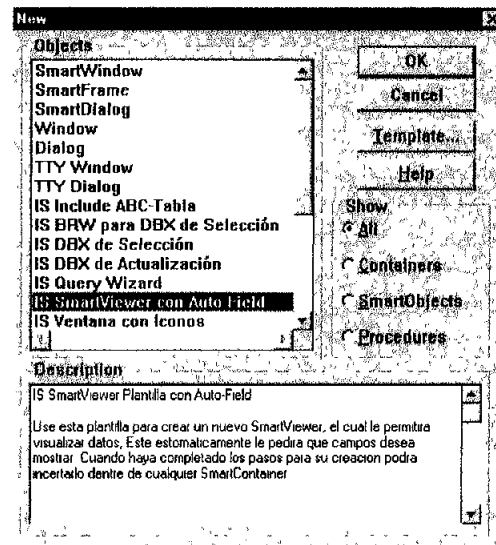
7. Por último se guardó el archivo como Q-AVPS.W.

5.3.4.3 Creación del SmartViewer de Avance de Programa de Salud (V-AVPS.W)

Para construir el SmartViewer de Avance de Programa de Salud se llevaron a cabo los siguientes pasos:

1. Se hizo click en el botón **New** de la barra de herramientas del UIB. Al hacer esto se el cuadro de diálogo **New** que se observa en la Figura 5.52 y de éste se eligió la plantilla IS SmartViewer con Auto-Field.

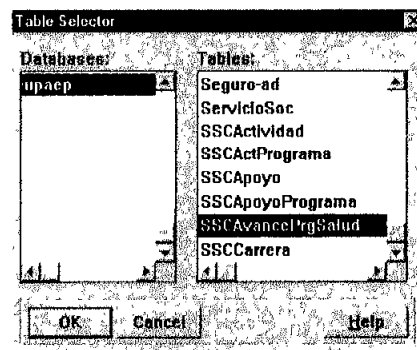
Figura 5.52. Selección de la Plantilla IS SmartViewer con Auto-Field.



Fuente: Progress v8.

2. A continuación se despliega el cuadro de diálogo **Table Selector**, donde se selecciona la tabla de la que se tomaron los campos. Este cuadro diálogo se puede observar en el Figura 5.53. En este caso se seleccionó la tabla de `SSCAvancePrgSalud` y se oprimió el botón **OK** para terminar con el cuadro de diálogo.

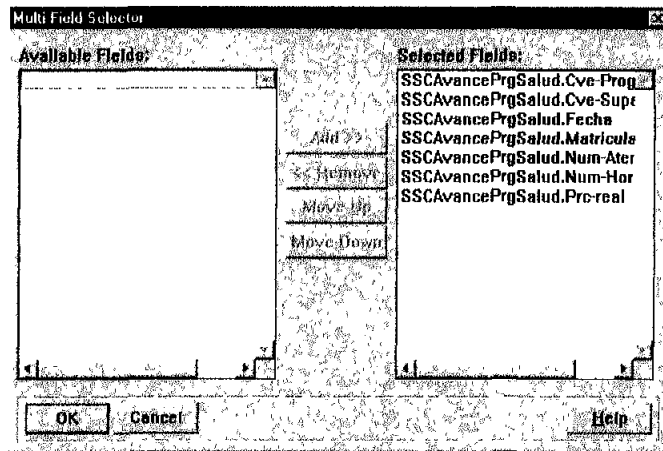
Figura 5.53. Table Selector, Tabla SSCAvancePrgSalud.



Fuente: Progress v8.

3. A continuación se despliega el cuadro de diálogo **Multi-Field Selector** que se muestra en la Figura 5.54. En este se eligieron los siguientes campos: Matrícula, Fecha, Cve-Programa, Cve-Supervisor, Num-Horas, Num-Atenciones, Prc-Real.

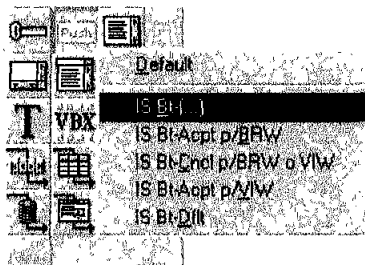
Figura 5.54. Multi-Field Selector de la Tabla SSCAvancePrgSalud.



Fuente: Progress v8.

4. Para cerrar el cuadro de diálogo se hizo click en el botón **OK**.
5. Se agregó un botón de 3 puntos por cada campo clave además de un editor . Para agregarlo, se hizo click con el botón derecho en el icono de botón en la Paleta de Objetos. El botón tiene todos los estándares definidos para los widgets que se observa en la Figura 5.55; de esa manera, los cambios que se requirieron son mínimos.

Figura 5.55. Paleta de Objetos - Botón de 3 puntos.



Fuente: Ingeniería de Software.

- 6 El botón de 3 puntos contiene el código para llamar al Cuadro de Diálogo de Selección en el evento **CHOOSE** como se muestra en la Figura 5.56.

Figura 5.56. Código del Botón de 3 Puntos.

```

/* --> &RunABC = OPCION
*
*     donde OPCION:
*         1 es para el Fill-In.
*         2 es para el Botón de tres puntos
*
*     --> Los cambios en el código para el evento local-display-fiels son:
*
*         El Archivo de Inclusión se llama: IS_Btn3b.I
*         &RunABC = 1
* */
/*  RUN IS_RUNDialogo(INPUT "BModPROG",INPUT-OUTPUT ClaveRowid).
C:\REVPROC\GNR\I\IS_BTN3A.I &RunABC = 2
&Tabla = NomTabla
&ClaveCampo = CampoClave
&DescripcionCampo = CampoDescripcion
&FifClave = BaseDatos.NomTabla.CampoClave
&FifDescripcion = EDT-CampoDescripcion
&BtnTresPuntos = BTN-CampoClave
*/

```

Fuente: Ingeniería de Software.

7. En este código se sustituyeron las palabras que aparecen en negritas por la información relacionada con el módulo, tal como se muestra en la Figura 5.57.

Figura 5.57. Código Adaptado del Botón de 3 Puntos.

```

/* --> &RunABC = OPCION
*
*      donde OPCION:
*      1 es para el Fill-In.
*      2 es para el Botón de tres puntos
*
*      --> Los cambios en el código para el evento local-display-fields son:
*
*      El Archivo de Inclusión se llama: IS_Btn3b.I
*      &RunABC = 1
* */
RUN IS_RUNDialogo(INPUT "BSSCPROG",INPUT-OUTPUT ClaveRowid).
\REVPROC\GNR\I\IS_BTN3A.I &RunABC = 2
&Tabla = SSCPrograma
&ClaveCampo = Cve-Programa
&DescripcionCampo = Dscon-Programa
&FifClave = UPAEP.SSCAvancePrgSalud.Cve-Programa
&FifDescripcion = EDT-Dscon-Programa
&BtnTresPuntos = BTN-Programa
}

```

Fuente: Módulo de Servicio Social.

8. El código se copió al evento **LEAVE** de cada campo clave. Sin embargo, para que funcionará adecuadamente se quitó la primera línea de código (texto en negrita) y se agregó **OR ENTER OF SELF**, en el encabezado del evento, para modificar el comportamiento del objeto. El código quedó como se muestra en la Figura 5.58.

Figura 5.58. Código del Botón de 3 Puntos para el Campo Clave.

```

OR ENTER OF SELF DO:
/*RUN IS_RUNDialogo(INPUT "BSSCPROG",INPUT-OUTPUT ClaveRowid).*/
C:\REVPROC\GNR\I\IS_BTN3A.I &RunABC = 1
&Tabla = SSCPrograma
&ClaveCampo = Cve-Programa
&DescripcionCampo = Dscon-Programa
&FifClave = UPAEP.SSCAvancePrgSalud.Cve-Programa
&FifDescripcion = EDT-Dscon-Programa
&BtnTresPuntos = BTN-Programa
}
END.

```

Fuente: Módulo de Servicio Social.

9. Se copió el código del Fill-in al procedimiento **local-display-fields** donde se cambió el nombre del Archivo de Inclusión a **IS_BTN3B.I**. El código quedó como se muestra en la Figura 5.59.

Figura 5.59. Código del Botón de 3 Puntos del Procedimiento local-display-fields.

```

/* Code placed here will execute AFTER standard behavior.      */
C:\REVPROC\GNR\I\IS_BTN3B.I &RunABC = 1
&Tabla = SSCPrograma
&ClaveCampo = Cve-Programa
&DescripcionCampo = Dscon-Programa
&FifClave = UPAEP.SSCAvancePrgSalud.Cve-Programa
&FifDescripcion = EDT-Dscon-Programa
&BtnTresPuntos = BTN-Programa
}

```

Fuente: Módulo de Servicio Social.

10. Por último se guardó con el nombre de **V-AVPS.W**.

5.3.4.4 Creación del Cuadro de Diálogo de Selección de Programas (BSCCPROG.W)

El Cuadro de Diálogo de Selección tiene las siguientes características:

- Permite seleccionar un registro y regresarlo a la ventana de ABC de Movimientos.
- Permite actualizar la tabla llamando a una ventana de actualización.
- Permite la búsqueda de datos en la tabla ya sea por clave o por descripción.

El Cuadro de Diálogo de Selección se compone de dos elementos:

- Un archivo de inclusión que contiene las definiciones y el código necesario para el funcionamiento.
- Un SmartBrowse que permite visualizar los registros en forma de lista.

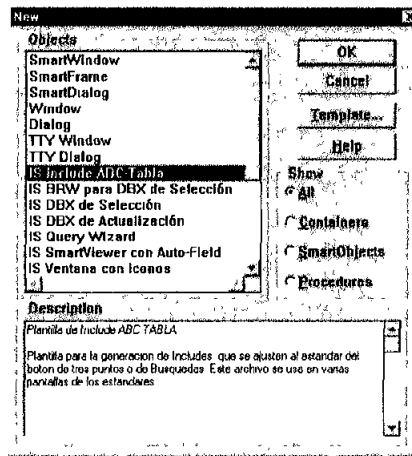
En las siguientes secciones se explica el procedimiento para crear cada elemento:

5.3.4.4.1 Creación del Archivo de Inclusión de Programas (ABC-PROG.I)

El archivo de inclusión contiene la información que necesitan las ventanas que utilizan la Ingeniería de Software desarrollada. Para crear este archivo se hicieron los siguientes pasos:

1. Se hizo click en el botón **New** de la barra de botones del UIB, lo cual despliega el cuadro de diálogo **New** que se observa en la Figura 5.60 y de éste se eligió la plantilla IS Include ABC-Tabla.

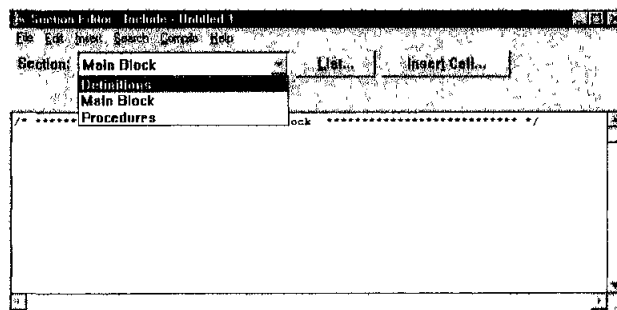
Figura 5.60. Selección de la Plantilla IS Include ABC-Tabla.



Fuente: Progress v8.

2. Inmediatamente después aparece el cuadro de diálogo de **Section Editor** que se observa en la Figura 5.61. Ahí se selecciona la sección de Definitions.

Figura 5.61. Ventana del Editor de Sección.



Fuente: Progress v8.

3. La sección de definiciones contiene las sentencias de código que se muestran en la Figura 5.62.

Figura 5.62. Código de la Sección de Definiciones.

```

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

DEFINE BUFFER Tabla FOR NomTabla.
DEFINE VARIABLE VIWArchivo AS CHAR INITIAL "VModTabla".

&GLOBAL-DEFINE ClaveCampo CampoClave
&GLOBAL-DEFINE DsconCampo CampoDescripcion
&GLOBAL-DEFINE VIWHandle h_V-Tabla
&GLOBAL-DEFINE BRWHandle h_Bn-Tabla
&GLOBAL-DEFINE QRYHandle h_Q-Tabla

DEFINE VARIABLE BRWArchivo AS CHAR INITIAL "BModTabla".
&GLOBAL-DEFINE FifClave FIFCampoClave
&GLOBAL-DEFINE FifDscon FIFCampoDescripcion

```

Fuente: Ingeniería de Software.

4. Las palabras que aparecen en negritas se sustituyeron por la información relacionada con el módulo, tal como se muestra en la Figura 5.63.

Figura 5.63. Código Adaptado del Include ABC-ACT.I.

```

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

DEFINE BUFFER Tabla FOR SSCPrograma.
DEFINE VARIABLE VIWArchivo AS CHAR INITIAL "VSSCprog".

&GLOBAL-DEFINE ClaveCampo Cve-Programa
&GLOBAL-DEFINE DsconCampo Dscon-Programa
&GLOBAL-DEFINE VIWHandle h_V1-PROG
&GLOBAL-DEFINE BRWHandle h_B1-Prog
&GLOBAL-DEFINE QRYHandle h_Q-PROG

DEFINE VARIABLE BRWArchivo AS CHAR INITIAL "BSSCPROG".
&GLOBAL-DEFINE FifClave UPAEP.SSCPrograma.Cve-Programa
&GLOBAL-DEFINE FifDscon FIF-Dscon-Programa

```

Fuente: Ingeniería de Software.

5. Por último se guardó el archivo como ABC-PROG. I.

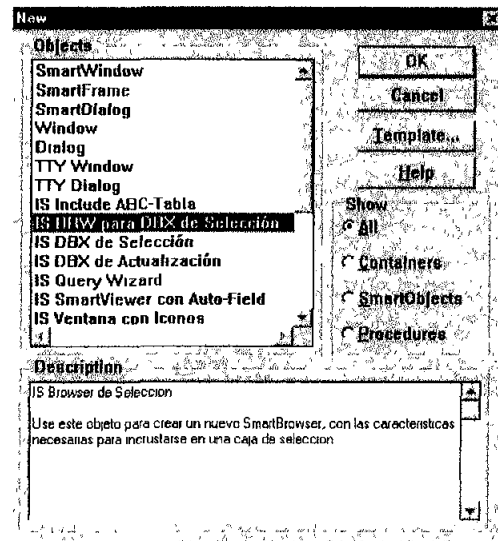
5.3.4.4.2 Creación del Archivo del SmartBrowse de Programas (B1-PROG.W)

Un SmartBrowse es un SmartObject que permite visualizar en forma de lista el resultado del Query. En ocasiones el SmartBrowse se puede configurar para actualizar los datos directamente en él. En este caso no se utilizó esa característica.

A continuación se detallan los pasos que se siguieron para la construcción del SmartBrowse.

1. Se hizo click en botón **New** de la barra de herramientas del UIB, lo cual despliega el cuadro de diálogo **New** que se observa en la Figura 5.64 y de este se eligió la plantilla IS BRW para DBX de Selección.

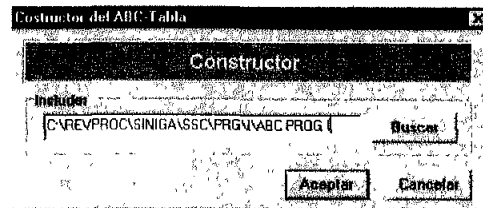
Figura 5.64. Selección de la Plantilla IS BRW para DBX de Selección.



Fuente: Progress v8.

2. A continuación se despliega el cuadro de dialogo **Constructor Del ABC-Tabla** que se observa en la Figura 5.65, donde se ingresó el nombre y la ruta del Archivo de Inclusión: C:\REVPROC\SINIGA\SSC\PRG\I\ABC-PROG.I.

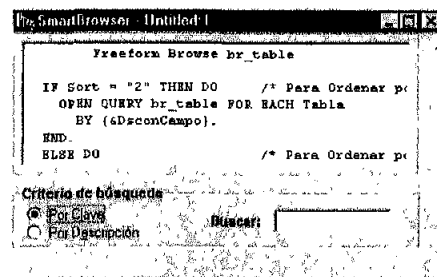
Figura 5.65 Constructor del ABC-Tabla.



Fuente: Ingeniería de Software.

3. En este caso, el Wizard no aparece, ya que el SmartBrowse está en modo 'Freeform Query'. Este modo permite modificar manualmente el código del Query que se genera. Los diferentes eventos y procedimientos a los que generalmente no se tiene acceso utilizando el SmartQuery Wizard, aparecen en este modo, en el Section Editor. Además, este modo es necesario para implementar la búsqueda por clave y descripción. Lo anterior se puede observar en la Figura 5.66.

Figura 5.66. SmartBrowse en Modo Freeform Query.



Fuente: Ingeniería de Software.

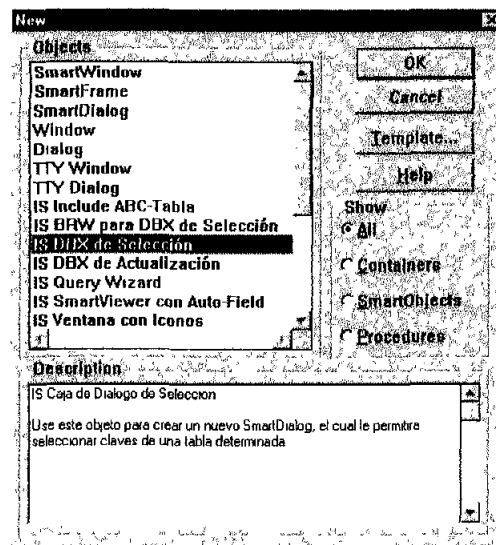
4. Por último se guardó el archivo como B1-PROG.W.

5.3.4.4.3 Integración del Cuadro de Diálogo de Selección de Programas (BSSCPROG.W)

Una vez creado el SmartBrowse y el archivo de inclusión de Programas (B1-PROG.W Y ABC-PROG.I) se procedió a integrarlos en un cuadro de diálogo como se explica a continuación:

1. Se hizo click en botón **New** de la barra de herramientas del UIB, lo cual despliega el cuadro de diálogo **New** que se observa en la Figura 5.67 y de este se eligió la plantilla IS DBX de Selección.

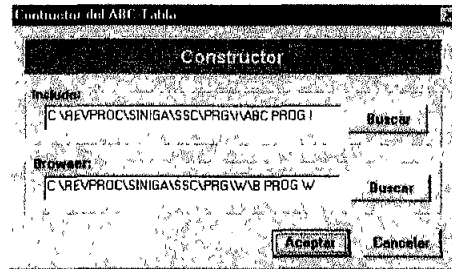
Figura 5.67. Selección de la Plantilla IS DBX de Selección.



Fuente: Progress v8.

2. A continuación se despliega el cuadro de dialogo **Constructor Del ABC-Tabla** que se observa en la Figura 5.68, donde se ingresó el nombre y la ruta del archivo de inclusión creado anteriormente: C:\REVPROC\SINAF\SSC\PRG\I\ABC-PROG.I y el nombre y la ruta del archivo del SmartBrowse: C:\REVPROC\SINAF\SSC\PRG\W\B1-PROG.W.

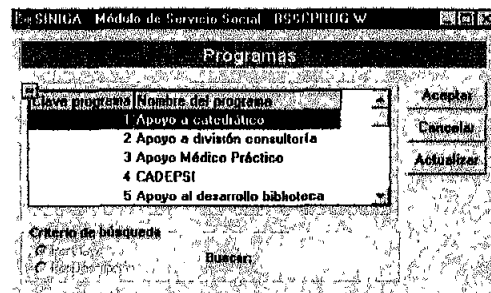
Figura 5.68. Constructor del ABC-Tabla.



Fuente: Ingeniería de Software.

3. Después de hacer click en el botón **Aceptar**, se cambió el título de la ventana a SINIGA - Módulo de Servicio Social y el título del encabezado a Programas. El cuadro de diálogo quedó como se muestra en la Figura 5.69.

Figura 5.69. Cuadro de Diálogo de Selección de Programas.



Fuente: Módulo de Servicio Social.

4. Por último se guardó el archivo como BSSCPROG.W.

Cabe mencionar que el cuadro de diálogo tiene un botón con la etiqueta **'Actualizar'**. Este botón permite actualizar la tabla que se está examinando, mandando a llamar un Cuadro de Diálogo de **Actualización De Programas**. En la siguiente sección se explica la construcción del Cuadro de Diálogo de **Actualización De Programas**.

5.3.4.5. Creación del Cuadro de Diálogo de Actualización de Programas (VSSCPROG.I)

El Cuadro de Diálogo de Actualización permite dar altas, bajas y cambios a una tabla, en este caso a `SSCProgramas`. En este punto puede causar confusión, ya que anteriormente se creó una ventana de actualización de la tabla `SSCProgramas` (`PSSC1110.W`).

Las razones por las que se creó un cuadro de diálogo en lugar de utilizar la ventana `PSSC1110`, son las siguientes:

1. Un cuadro de diálogo no puede llamar a una ventana, sólo a otro cuadro de diálogo. La ventana de actualización `PSSC1110` es una ventana.
2. Paso de parámetros: cuando se creó la ventana, ésta no definía ningún parámetro de entrada, ya que se manejaría de forma autónoma. Cuando surgió la necesidad de llamar a una ventana de actualización, se creó un cuadro de diálogo que definía un parámetro de entrada. De esa forma, el cuadro de diálogo puede regresar el registro recién actualizado.
3. Si la ventana tenía la definición del parámetro no se podía ejecutar en forma autónoma, ya que el compilador indicaba un error de paso de parámetros. En base a esto se decidió crear un cuadro de diálogo que sería llamado por otro cuadro de diálogo. Sin embargo, después se descubrió cómo crear la ventana y que ésta se ejecutara en forma autónoma o pudiera ser llamada desde otra ventana.

Al igual que la ventana de actualización `PSSC1110`, los elementos que integran el Cuadro de Diálogo de Selección son los siguientes:

Un `SmartViewer` para visualizar y cambiar los datos de los registros.

Un `SmartQuery` que provea los registros al `SmartViewer`.

Un `SmartPanel` de Actualización para realizar las altas, bajas y cambios.

Un `SmartPanel` de Navegación para recorrer los registros.

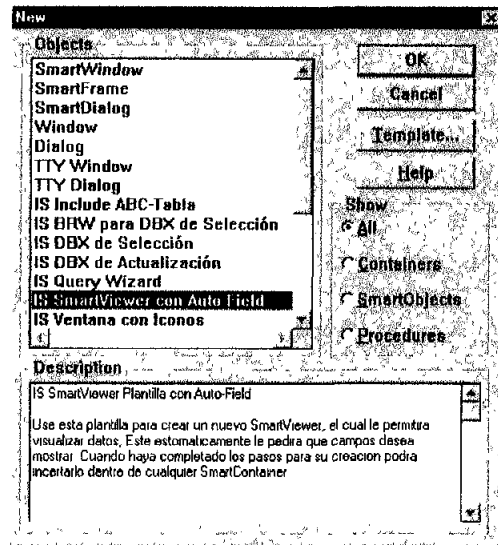
5.3.4.5.1. Creación del SmartViewer de Programas (V-PROG.W)

Los `SmartViewers` son `SmartObjects` que despliegan los campos de las tablas y permiten modificar los registros a través de la interacción con los `SmartPanels` de Actualización.

Para construir el `SmartViewer` de Programas se llevaron a cabo los siguientes pasos:

- 1 Se hizo click en el botón **New** de la barra de herramientas del UIB. Al continuación se despliega el cuadro de diálogo **New** que se observa en la Figura 5.70. En este cuadro de diálogo se eligió la plantilla **IS SmartViewer con Auto-Field**.

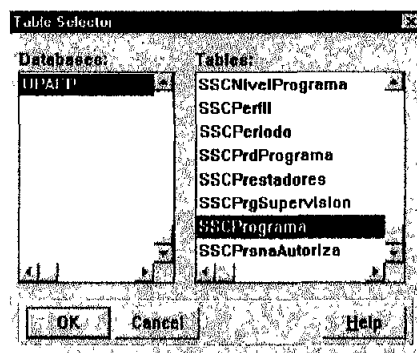
Figura 5.70. Selección de la Plantilla IS SmartViewer con AutoField.



Fuente: Progress v8.

2. Se despliega el cuadro de diálogo **Table Selector** donde se seleccionó la tabla **SSCPrograma** que se observa en la Figura 5.71. A continuación se oprimió el botón **OK**.

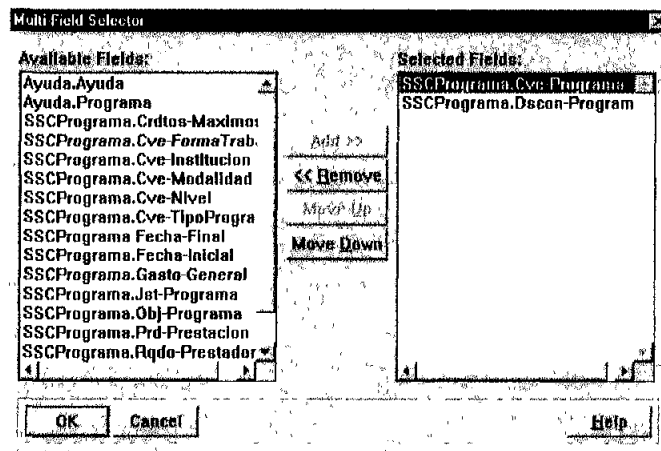
Figura 5.71. Table Selector, Tabla SSCProgramas.



Fuente: Progress v8.

3. Inmediatamente después aparece el cuadro de diálogo Multi-Field Selector donde se seleccionaron los campos que tendría el SmartViewer: Cve-Programa y Dskon-Programa. Esto se observa en la Figura 5.72. Para terminar se hizo click en el botón OK.

Figura 5.72. Multi-Field Selector de la Tabla SSCActividad.



Fuente: Progress v8.

4. Una vez que los campos quedaron colocados en el SmartViewer se procedió a aplicar los estándares definidos en el Anexo B 'Estándares de Ventana VGA'.
5. Por último se agregó el Archivo de Inclusión para el incremento de la secuencia: IS_SEC.I. Este cambio se realizó en el evento ENTRY del fill-in UPAEP.SSCPrograma.Dskon-Programa. En la Figura 5.73 se observa el llamado al Archivo de Inclusión con los parámetros necesarios

Figura 5.73. Llamado al Archivo de Inclusión.

```
{c:\revproc\gnr\i\is_sec.1 &Tabla = SSCPrograma
&Secuencia =SSC-Cve-Programa
&Campo = Cve-Programa
}
```

Fuente: Módulo de Servicio Social.

6. Por último se guardó el archivo como V-PROG.W.

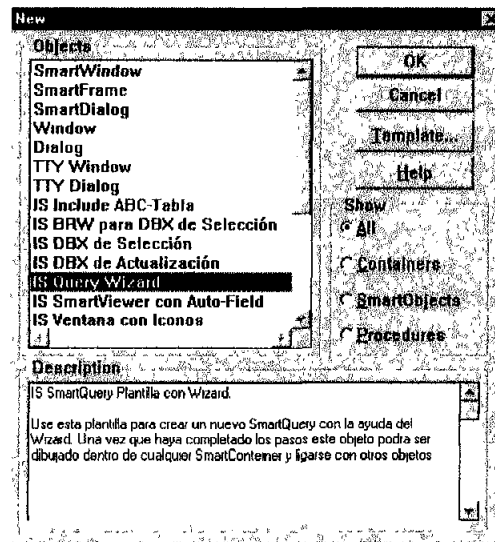
5.3.4.5.2. Creación del SmartQuery de Programas (Q-PROG.I)

Un SmartQuery es un SmartObject que obtiene los registros que cumplen el criterio de selección. El SmartQuery interactúa con el SmartViewer para proveer a este los registros y con el SmartPanel de Navegación para recorrerlos. A continuación se explica la construcción del SmartQuery Q-PROG.W.

Los pasos que se llevaron a cabo son los siguientes:

1. Se hizo click en el botón **New** de la barra de herramientas del UIB. Al hacer esto se despliega el cuadro de diálogo **New** que se observa en la Figura 5.74. En este cuadro diálogo se eligió la plantilla "IS Query Wizard".

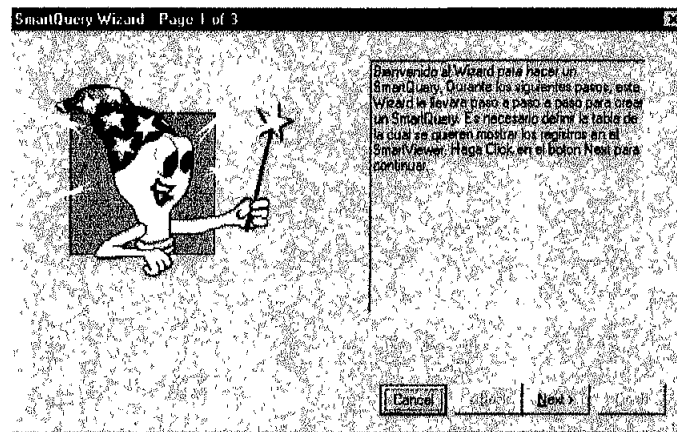
Figura 5.74. Selección de la Plantilla IS Query Wizard.



Fuente: Progress v8.

2. A continuación aparece el primer cuadro de diálogo del SmartQuery Wizard, el cual se observa en la Figura 5.75.

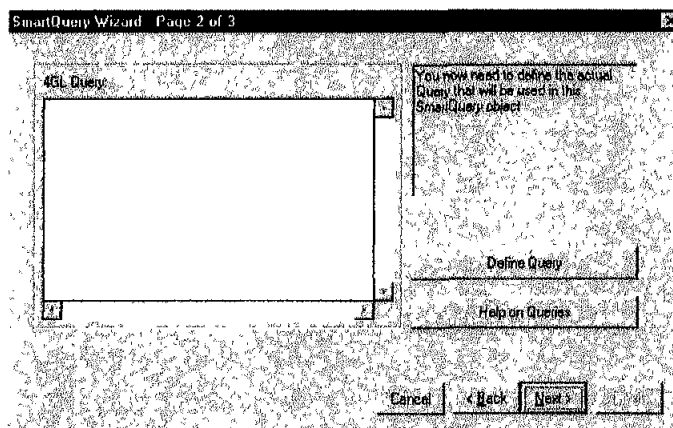
Figura 5.75. SmartQuery Wizard de Programas. Página 1 de 3.



Fuente: Ingeniería de Software.

3. Se continua con el siguiente cuadro de diálogo del Wizard que se observa en la Figura 5.76, oprimiendo el botón **Next**. En este cuadro de diálogo se pasa a definir el Query haciendo click en el botón Define Query.

Figura 5.76. SmartQuery Wizard de Programas. Página 2 de 3.

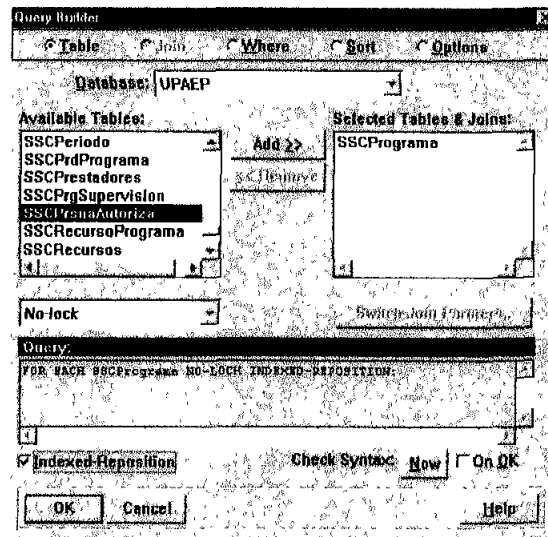


Fuente: Ingeniería de Software.

4. Lo anterior despliega el cuadro de diálogo **Query Builder** (Figura 5.77) donde se especificó la tabla **SSCProgramas**, a la cual se le daría mantenimiento. Para mejorar la velocidad de búsqueda de los registros se seleccionó la casilla **Indexed-Reposition**.

- Por último se oprimió el botón de **OK**. Esta última acción regresa al usuario al cuadro de diálogo de la Figura 5.76..

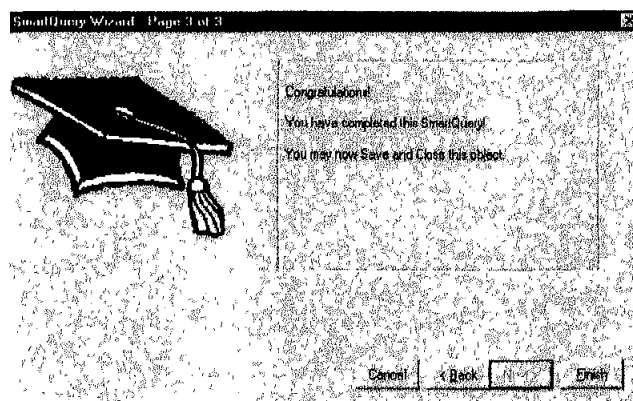
Figura 5.77. Cuadro de Diálogo Query Builder de Programas.



Fuente: Progress v8.

- Se continuó con el siguiente cuadro de diálogo del Wizard (Figura 5.78), oprimiendo el botón **Next**. Este cuadro de diálogo notifica la terminación de la construcción del SmartQuery.

Figura 5.78. SmartQuery Wizard de Programas. Página 3 de 3.



Fuente: Progress v8.

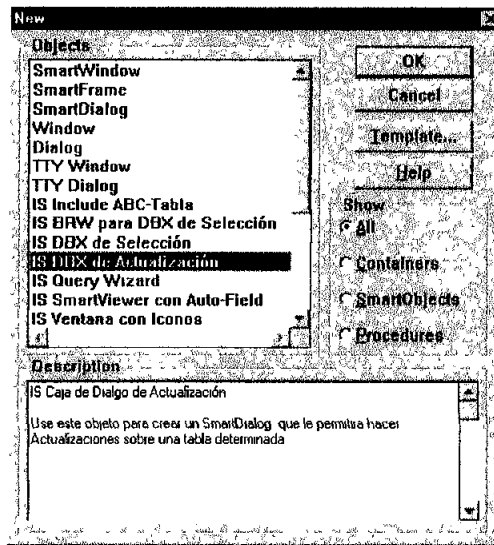
- Por último se guardó el archivo como **Q-PROG.W**.

5.3.4.5.3. Integración de los Objetos en el Cuadro de Diálogo de Actualización (VSSCPROG.I)

Una vez contruidos los archivos anteriores ya se puede elaborar el Cuadro de Diálogo de Actualización de Programas. Esta ventana se generó con una de las plantillas especialmente desarrolladas para tal propósito. A continuación se detallan los pasos que se siguieron para su creación:

1. Se hizo click en botón **New** de la barra de herramientas del UIB, lo cual despliega el cuadro de diálogo **New** que se muestra en la Figura 5.79 y de ésta se escogió la plantilla "IS DBX de Actualización"

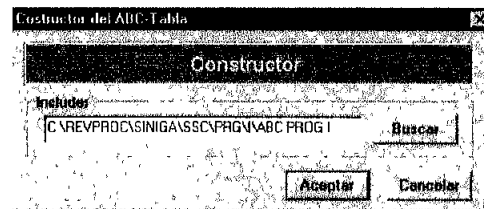
Figura 5.79 Selección de la Plantilla IS DBX de Actualización.



Fuente: Progress v8.

2. A continuación se despliega el cuadro de dialogo del constructor que se observa en la Figura 5.80, donde se ingresó el nombre y la ruta del Archivo de Inclusión:
C:\REVPROC\SINIGA\SSC\PRG\I\ABC-PROG.I.

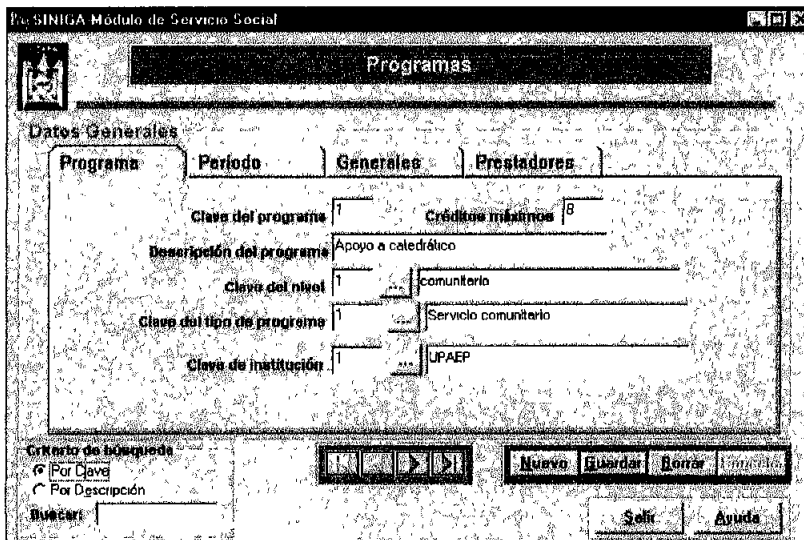
Figura 5.80. Constructor del Include ABC-PROG.I



Fuente: Ingeniería de Software.

3. Como los SmartObjects que contenía la plantilla no eran los que se necesitaban. Se cambiaron sus instancias por el archivo correcto:
 - Se cambió la instancia del SmartViewer por V-PROG.W.
 - Se cambió la instancia del SmartQuery por Q-PROG.W.
4. Se cambió el título de la ventana a SINIGA-Módulo de Servicio Social.
5. Se cambió el título del encabezado a Programas, el cuadro de dialogo ya terminado se muestra en la Figura 81.

Figura 5.81. Pantalla de Consulta de Programa.



Fuente: Módulo de Servicio Social.

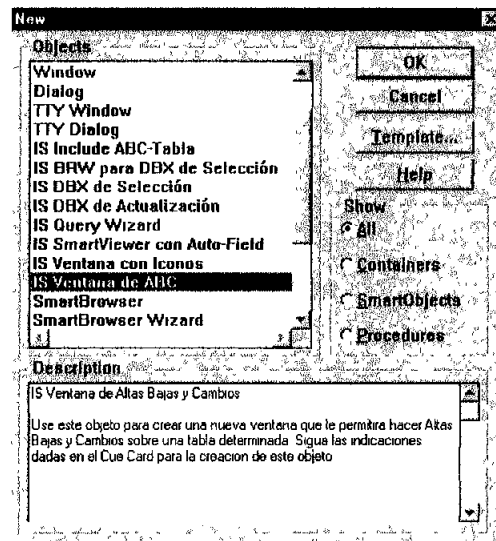
6. Por último se guardó el archivo con el nombre de PSSC1110.W.

5.3.4.6 Integración de los objetos en la Ventana de ABC (PSSC1210.W)

Una vez contruidos los archivos anteriores, ya se puede elaborar la ventana de ABC de Avance de Programa de Salud. Esta ventana utiliza también la Ingeniería de Software desarrollada, por lo que es importante destacar que se necesita crear con las plantillas especialmente desarrolladas. A continuación se detallan los pasos que se siguieron:

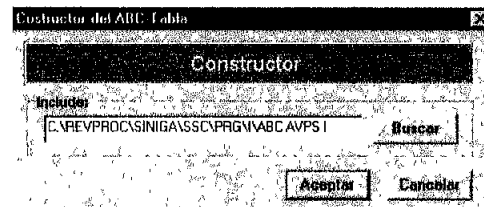
1. Se hizo click en botón **New** de la barra de herramientas del UIB, lo cual despliega el cuadro de diálogo **New** que se muestra en la Figura 5.82 y de esta se escogió la plantilla "IS Ventana de ABC".

Figura 5.82. Selección de la Plantilla IS Ventana de ABC.



Fuente: Progress v8.

2. A continuación se despliega el cuadro de dialogo del constructor que se observa en la Figura 5.83, donde se ingresó el nombre y la ruta del include: "C:\REVPROC\SINIGA\SSC\PRG\I\ABC-AVPS.I".

Figura 5.83. Constructor del Include ABC-AVPS.I

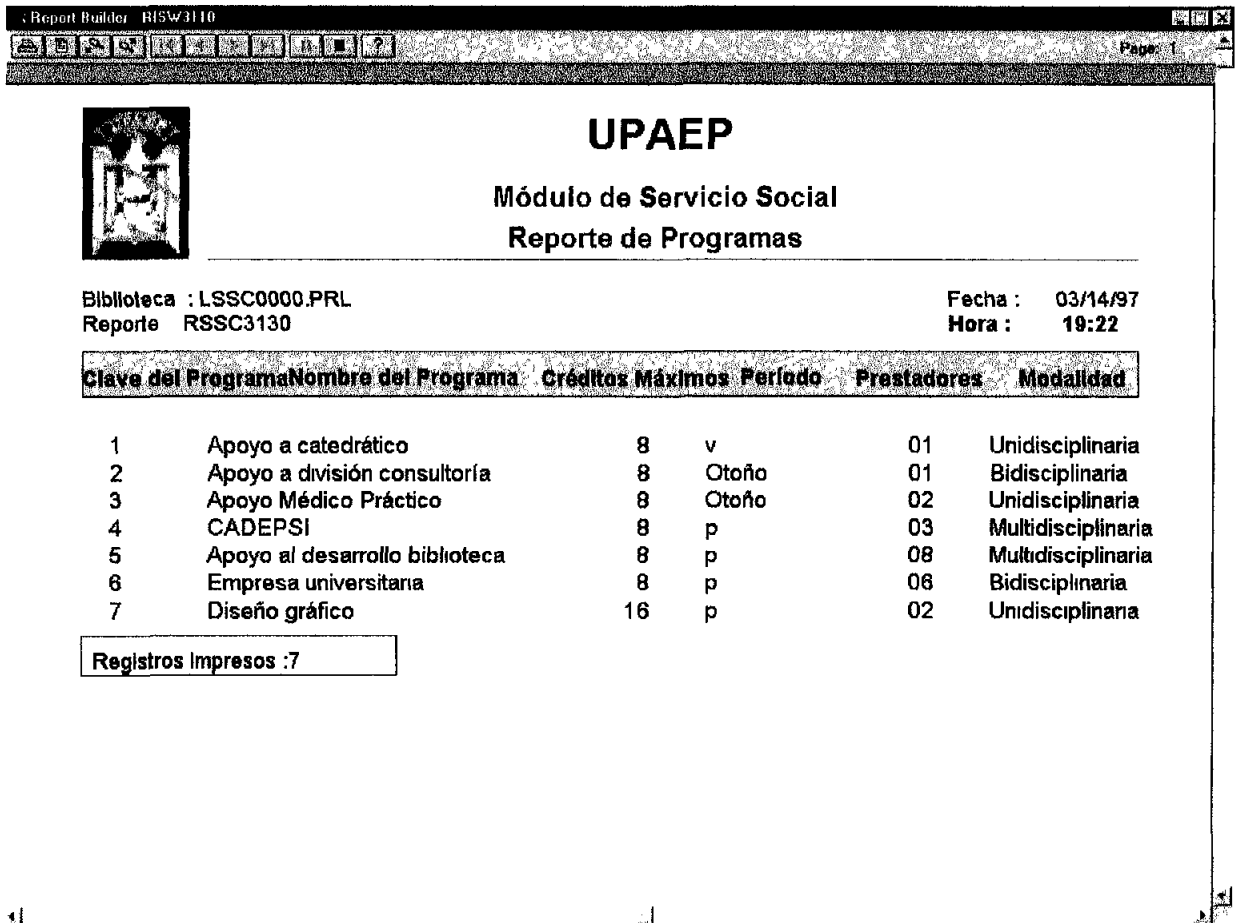
Fuente: Módulo de Servicio Social.

3. Como los SmartObjects que contenía la plantilla no eran los que se necesitaban, se cambiaron sus instancias por el archivo correcto:
 - Se cambió la instancia del SmartViewer por V-AVPS.W.
 - Se cambió la instancia del SmartQuery por Q-AVPS.W.
4. En el Section Editor se acceso el evento LEAVE del fill-in **FIF-Buscar** y se cambió el nombre del Archivo de Inclusión a IS_BUSC5.I. Este Archivo de Inclusión hace búsquedas de campos clave de tipo entero. Si se desea hacer búsquedas de campos clave de tipo carácter.
5. Se cambió el título de la ventana a SINIGA-Módulo de Servicio Social.
6. Se cambió el título del encabezado a Avance de Programa de Salud.
7. Por último se guardó el archivo con el nombre de PSSC1210.W.

5.3.5 Construcción de un Reportes de Maestros (RSSC3110)

El reporte terminado del Maestro de Programas se puede observar en la Figura 5.84.

Figura 5.84. Reporte de Programas.

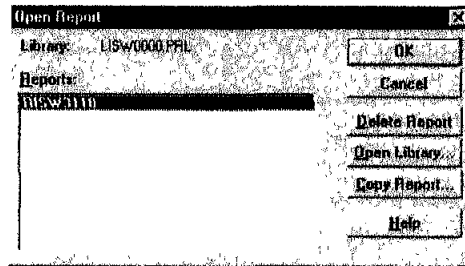


Fuente: Módulo de Servicio Social.

Para construir este reporte es necesario antes instalar la Ingeniería de Software para los reportes. Una vez hecho esto se realizan los siguientes pasos:

1. Se ejecuta el Report Builder.
2. Se abre la librería LISW0001.PRL y se selecciona el reporte RISW3110 como se observa en la Figura 5.85

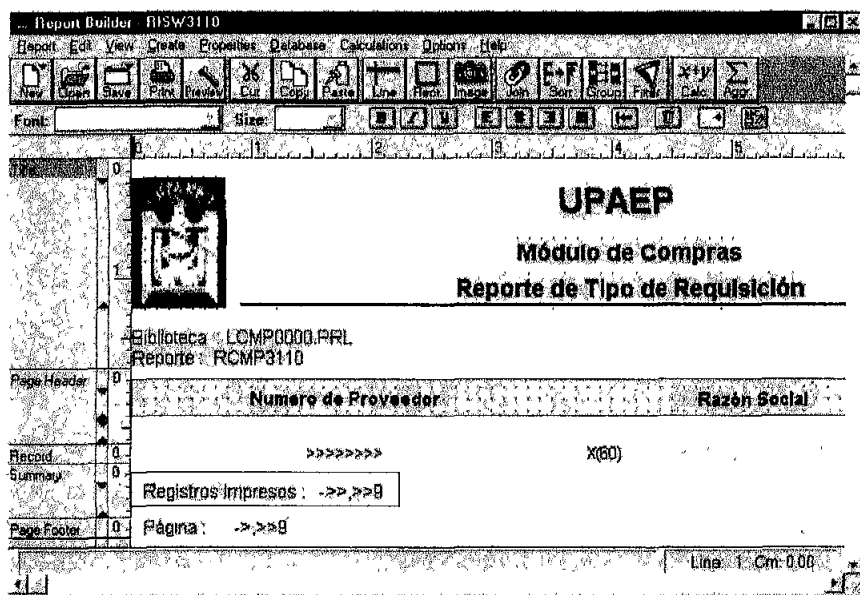
Figura 5.85. Cuadro de Diálogo Open Report.



Fuente: Progress v8.

3. A continuación se despliega una plantilla que se muestra en la Figura 5.86.

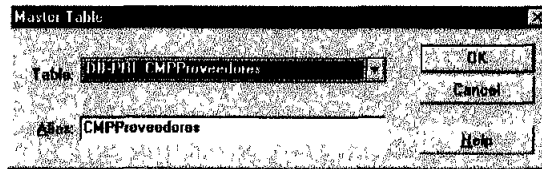
Figura 5.86. Plantilla de Reporte.



Fuente: Ingeniería de Software.

4. Se borraron los registros que contiene este reporte, después se conectó la base de datos UPAEP.
5. Se hizo click en el menú Database y se seleccionó la opción de Master table, dicha acción despliega un cuadro de diálogo que se observa en la Figura 5.87.

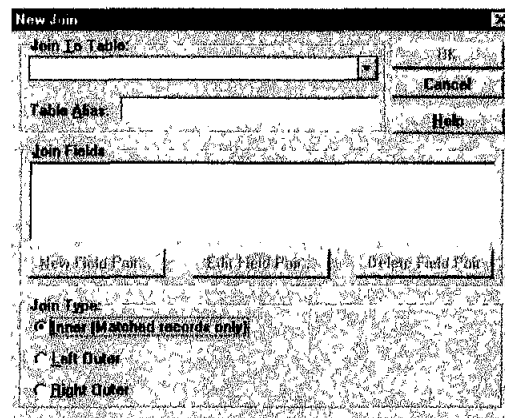
Figura 5.87. Cuadro de Diálogo Master Table.



Fuente: Progress v8.

5. Se seleccionó la tabla de SSCPrograma y se hizo click en el botón de **OK**.
6. A continuación se desconectó la base de datos DB-demo.
7. Se hizo click en el botón de **Join** de la barra de botones del Report Builder, lo que nos despliega un cuadro de diálogo como el que se observa en la Figura 5.88.

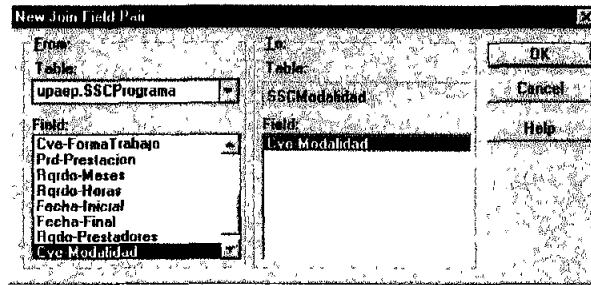
Figura 5.88. Cuadro de Diálogo New Join.



Fuente: Progress v8.

8. Se selecciono la tabla SSCModalidad, y se hizo click en el botón **New Field Pair** lo cual despliega el cuadro de diálogo que se observa en la Figura 5.89. Se seleccionaron los campos Cve-Modalidad y se hizo click en el botón **OK**.

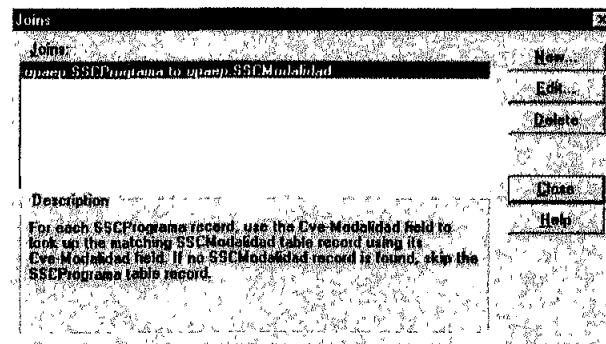
Figura 5.89. Cuadro de Diálogo New Join Field Pair.



Fuente: Progress v8.

9. Esta acción regresa al usuario al cuadro de diálogo que se muestra en la Figura 5.90. A continuación se hizo click en el botón de **OK** para regresar al cuadro de diálogo que se muestra en la Figura 5.90, donde se hizo click en el botón de **Close** para terminar.

Figura 5.90. Cuadro de Diálogo Joins.



Fuente: Progress v8.

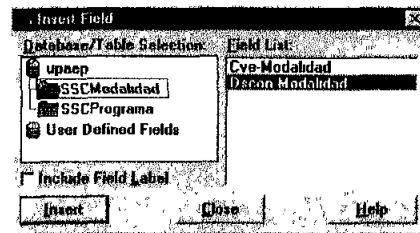
10. Se cambiaron los siguientes datos:

- Módulo de Compras **por** Modulo de Servicio Social.
- Reporte de Tipo de Requisición **por** Reporte de Programas.
- Biblioteca : LCMP0000.PRL **por** Biblioteca : LSSC0000.PRL.
- Reporte: RCMP3110 **por** Reporte: RSSC3130.
- Número de Proveedor **por** Clave del Programa.
- Razón Social **por** Nombre del Programa.

11. Se agregaron las siguientes etiquetas: Créditos Máximos, Período, Prestadores, Modalidad; y se cambio la fuente a Arial Negrita y el tamaño a 11 puntos.

12. Después se hizo click en el menú **Edition** y se seleccionó la opción **Insert Field**, dicha acción despliega un cuadro de diálogo como el que se muestra en la Figura 5.91.

Figura 5.91. Cuadro de Diálogo Insert Ffield.



Fuente: Progress v8.

Posteriormente se insertan los registros de:

- Cve-Programa
- Dscon-Programa
- Crdtos-Maximos
- Prd-Prestacion
- Rqdo-Prestadores
- Dscon-Modalidad

13. Los campos y las etiquetas se distribuyeron a lo ancho del reporte.

14. El reporte se guardó con el nombre PSSC3110. Este reporte se usó como plantilla para los demás reportes, y lo único que se cambió fue:

- El nombre del reporte
- El número del reporte
- Las etiquetas de los campos
- Los campos que se deseen mostrar

15. Se creó la librería para el módulo, guardándolo con el de LSSC000.PRL.

Capítulo 6. Resultados del Trabajo de Ingeniería de Software

Dado el objetivo principal de este trabajo, que es el desarrollo de Ingeniería de Software para el Departamento de Revisión de Procesos, se presentan en este capítulo los resultados del trabajo de Ingeniería de Software a lo largo del proceso de desarrollo.

La tesis de Ingeniería de Software tuvo entre sus objetivos cubrir algunos aspectos de desarrollo de los sistemas. Los aspectos que se cubrieron fueron los siguientes :

- Ingeniería de Software
- Capacitación
- Establecimiento de Estándares
- Coordinación de Grupo
- Soporte Técnico

Estos aspectos se fueron cubriendo en paralelo; algunos de ellos duraron hasta el final de la tesis. Por ejemplo: la Capacitación se dio hasta el momento en que los tesisistas manejaban las herramientas lo suficiente como para desarrollar sin muchos contratiempos; pero la Ingeniería se siguió desarrollando incluso después del cierre de la tesis.

El desarrollo de Ingeniería de Software, que es el objetivo de la tesis, permitió en gran medida resolver problemas comunes de programación a los que se enfrentaban los tesisistas. Este aspecto se detalla en la Sección 6.1.

El establecimiento de estándares surgió por la necesidad de llevar un desarrollo homogéneo y controlado. Estos estándares abarcan aspectos de programación, diseño y uso de las herramientas. El establecimiento de estándares se explica a detalle en la Sección 6.2.

El aspecto de la capacitación se fue cubriendo de acuerdo a las necesidades detectadas en los tesisistas. Los cursos no sólo abarcaron capacitación en las

herramientas que se tenían sino de la Ingeniería de Software que se desarrolló. Este aspecto se explica a detalle en la Sección 6.3.

La coordinación de grupo fue una parte muy importante en el desarrollo, ya que permitió la correcta integración de los sistemas. Debido a la cantidad de personas que debía coordinarse, surgieron algunos problemas de coordinación, que se resolvieron de forma satisfactoria reuniendo a los tesisistas en un solo lugar. En la Sección 6.4 se explica a detalle este aspecto.

Debido a la uso de herramientas nuevas en ese momento, surgieron problemas en su utilización, con lo cual se dio la necesidad de dar soporte técnico que ayudara a resolver esos problemas. Ingeniería de Software tuvo la obligación de aprender de antemano las herramientas que se utilizarían para poder dar soporte técnico. Este aspecto se explica a detalle en la Sección 6.5.

Una parte importante que debía cubrirse era la calidad. Por eso Ingeniería de Software trabajó en paralelo con la tesis de Calidad que auxilió en la correcta implantación de los Estándares además de verificar la calidad de los programas que se creaban. Este aspecto se explica a detalle en la Sección 6.6

6.1 Ingeniería de Software

Una de las mayores ventajas de la Ingeniería de Software es que permite crear herramientas, procedimientos y rutinas que auxilien en la elaboración de programas ayudando a reducir el tiempo de desarrollo.

Progress es una herramienta que apoya el desarrollo de Ingeniería, ya que tiene características como la orientación a objetos que permite la elaboración de plantillas que pueden ser utilizadas cuantas veces se desee.

6.1.1 Panorama General

Al inicio de desarrollo no se tenía conocimiento de Progress y sus herramientas. El tiempo que los tesisistas dedicaron al análisis y al diseño permitió a Ingeniería de Software aprender el lenguaje de Progress, el uso del User Interface Builder y lo más importante: la metodología de desarrollo de Progress basado en componentes.

Una vez que se tuvieron los conocimientos necesarios se empezaron a impartir cursos, en los cuales se enseñó lo más básico. Una vez hecho esto los tesisistas estuvieron en la posibilidad de empezar a desarrollar programas.

Los programas en realidad eran prototipos, y en las primeras entregas se detectaron las primeras necesidades que Ingeniería de Software debía cubrir. Se listan a continuación las observaciones que se llevaron a cabo.

- Los programas debían contar con un menú de iconos. Aunque se les especificó lo que debía llevar, se observó que cada módulo lo implementó a su criterio.
- Como los programas consistían en ventanas y cuadros de diálogo, estos se llamaban uno a otros, además de que en ocasiones era necesario enviar y/o recibir parámetros. Se tuvieron muchos problemas con esto porque se tenía que hacer el código del llamado cada vez que se necesitaba, haciendo difícil en ocasiones la depuración del programa.
- Cuando se necesitaron los primeros reportes no se tenía una forma clara de llamarlos y no se sabía la forma que el Report Builder manejara la base de datos. Esto originó, en ocasiones, daños en las bases de datos.

A continuación se explican las acciones que se llevaron a cabo para resolver estos problemas:

- Para los menús de iconos se inició con la elaboración de un manual de estándares donde se detalla las características que debía tener cada elemento. Esto se explica a detalle en la Sección 6.2 'Establecimiento de Estándares'.
- Con respecto a las ventanas y cuadro de diálogo se empezaron a crear archivos de inclusión (includes) que contenían el código para realizar los llamados y el manejo de las ventanas y cuadros de diálogo. Uno de los includes que se generó fue `is_lib.i`. Este archivo de inclusión contiene procedimientos para llamar los programas y los reportes. Gracias a esto el código para el llamado se redujo a una línea que incluía solamente el nombre del archivo, tal como se muestra en la Figura 6.1.

Figura 6.1. Llamado de una Ventana con `is_lib.i`.

```
RUN IS_RunVentana ("PCMP1000", 1, {&WINDOW-NAME}).
```

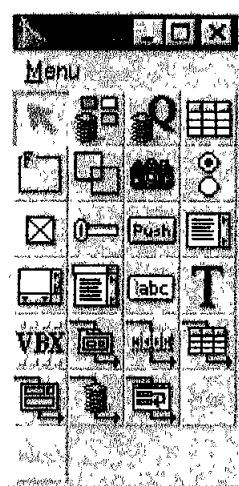
Fuente: Módulo de Compras.

Más adelante, cuando se pasó a la programación formal de los módulos, surgieron otras necesidades como las siguientes.

- La elaboración de los menús de iconos era una tarea repetitiva y llegó a ser tediosa ya que tenía que aplicarse los estándares, ya definidos para ese entonces: tipo y tamaño de letra, color, disposición de los objetos en la ventana, alto y ancho de los objetos, etc.

- Ingeniería de Software elaboró plantillas de menús de iconos. Estas plantillas ya tenían todos los elementos de un menú de iconos con todos los estándares requeridos. Además se modificó la Paleta de Objetos para que tuviera los botones, rectángulos, textos y demás objetos con todos los estándares, como se muestra en la Figura 6.2.

Figura 6.2. Paleta de Objetos Modificada.



Fuente: Progress v8.

- Una de las adiciones más importantes a la paleta fue la incorporación del estándar del botón de 3 puntos. Con este botón se manda a llamar a las ventanas de consulta en las pantallas de Altas, Bajas y Cambios. Con esto se resolvía el problema de los parámetros en el llamado a los programas. En la Figura 6.3 se puede observar el botón de 3 puntos.

Figura 6.3. Botón de 3 Puntos.



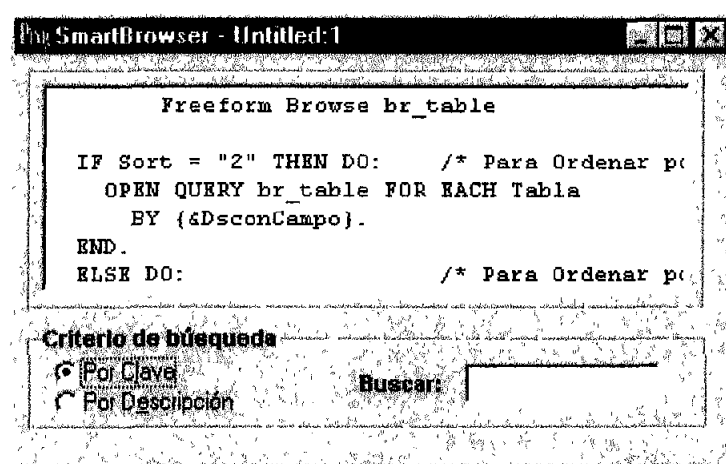
Fuente: Módulo de Compras.

- Las pantallas de movimientos se auxiliaban de pantallas especialmente diseñadas para consulta. Estas pantalla tenían un SmartBrowse del cual se podía seleccionar cualquier registro recorriendo la lista con la barra de desplazamiento. Sin embargo se

observó que en ocasiones la cantidad de datos eran muy grande y recorrerlos hasta encontrar el dato que se buscaba no era idea muy agradable.

Para solucionar esto se pensó dar la opción de búsqueda. Esta búsqueda podía ser por clave o descripción. Para eso se creó una plantilla que contenía tanto el browse como el código necesario para realizar la búsqueda (ver Figura 6.3). El código se puso en un include para que se tuviera la posibilidad de mejorar el código sin cambiar toda la plantilla.

Figura 6.4. Plantilla de Browse con Opción de Búsqueda.



Fuente: Ingeniería de Software.

Otra observación que se hizo fue en las pantallas de altas, bajas y cambios. Como las tablas tienen un campo clave para llevar el conteo de los registros dados de altas, estos campos debían aumentar su valor automáticamente cada vez que se daba de alta un nuevo registro. Esos campos claves toman su valor de las secuencias, que son campos que se definen en el Data Dictionary (Diccionario de Datos) y cuyo valor se incrementa o decrementa automáticamente. Aunque Progress permite escribir el código para manejar las secuencias directamente en el Diccionario de Datos, no era muy conveniente para el usuario, ya que el valor aparecía después de guardar el registro.

Tomando nota de esto, se creó un archivo de inclusión que realizaba el incremento y lo mostraba al momento de crear el nuevo registro. A este archivo se le llamó is_sec.i. Con esto sólo era necesario llamar al include con los parámetros necesarios en el evento adecuado. En la Figura 6.4 se puede observar un ejemplo del llamado a la secuencia.

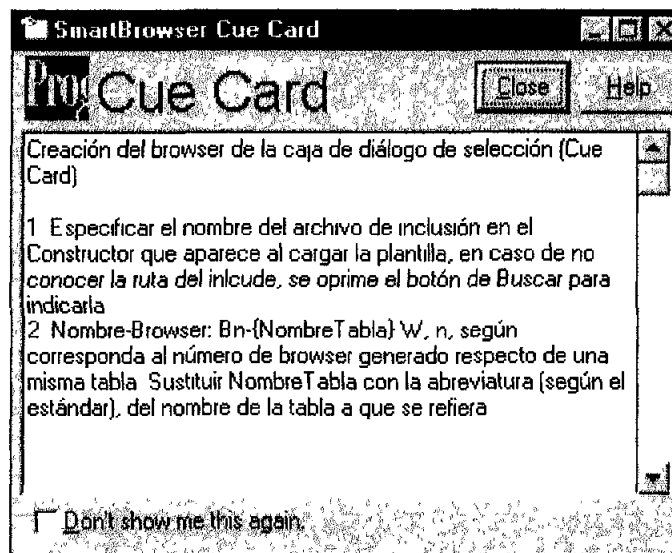
Figura 6.5. Llamado del Include is_sec.i

```
{c:\revproc\gnr\i\is_sec.i  &Tabla = CMPProvedores
                             &Secuencia = SCMP-Num-Proveedor
                             &Campo = Num-Proveedor
}
```

Fuente: Módulo de Compras.

Después de crear la Ingeniería para la creación de las pantallas de Altas, Bajas y Cambios se observó que se podía automatizar un poco la integración de los diferentes componentes de dichas pantallas.

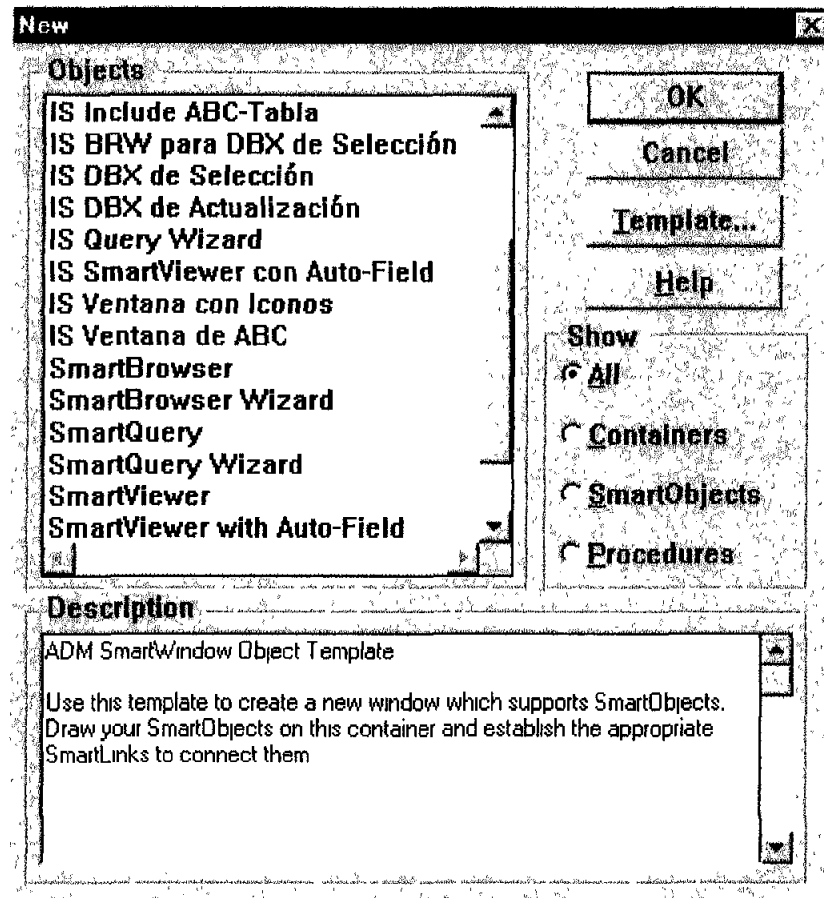
La forma de hacer eso, consistió en tomar como ejemplo las plantillas con Wizard que incluye el User Interface Builder. Estas pantallas guían al desarrollador en la creación de los SmartObjects. Cuando se selecciona una plantilla con Wizard (por ejemplo SmartQuery Wizard), se despliegan una serie de cuadros de diálogo que indican lo que se debe hacer para construir el SmartObject. Al mismo tiempo, al seleccionar una de esas plantillas se despliega un Cue Card, que no es más que una ventana pequeña que describe los pasos, de forma general, para construir el SmartObject. Un ejemplo de Cue Card se muestra en la Figura 6.6.

Figura 6.6. Ejemplo de Cue Card.


Fuente: User Interface Builder.

Ingeniería de Software personalizó los Wizards y los Cue Cards, además de crear algunos más (ver Figura 6.7)

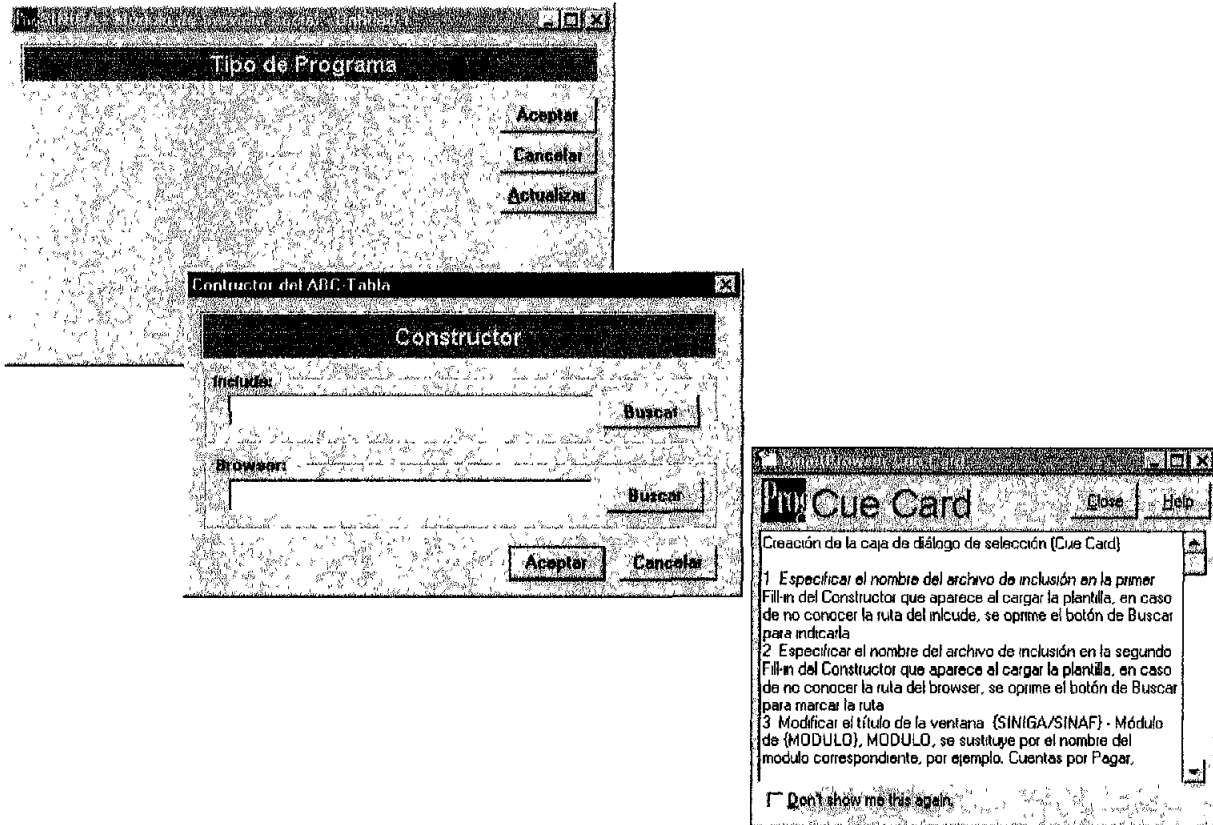
Figura 6.7. Plantillas de Ingeniería de Software.



Fuente: Ingeniería de Software.

Por ejemplo, cuando se abría la plantilla IS DBX de Selección se desplegaba el Cue Card de la Figura 6.8, además del cuadro de diálogo Constructor del ABC-Tabla.

Figura 6.8. Plantilla IS DBX de Selección y Cue Card.



Fuente: Ingeniería de Software.

Esto permitió que la integración de los componentes de una pantalla de Altas, Bajas y Cambios fuera más rápida, además de dar al desarrollador una idea clara de lo que se necesitaba para esa pantalla.

6.1.2 Resultados

Debido al poco conocimiento que se tenía de Progress la implementación de las soluciones descritas en la sección anterior resultó un tanto complicada en un principio. Sin embargo con la asistencia técnica que se proporcionó a los tesisistas, y el uso repetido de la Ingeniería, la implementación se volvió más fácil y la integración de los módulos se hizo sin mayor problema.

Esto se vio reflejado en las entregas que se hicieron de los prototipos de los módulos ante el Ing. Cabanas miembro del Patronato Fundador. En la primera entrega se notaron errores en la integración, los cuales se resolvieron para la segunda entrega. El Ing. Cabanas se mostró complacido con los prototipos e hizo algunos comentarios que se tomaron en cuenta para la próxima entrega.

Aún así surgieron problemas que dificultaron en cierto momento la implementación de la Ingeniería. Algunos de esos problemas fueron de comunicación entre los tesisistas e Ingeniería de Software y entre los mismos tesisistas, además de la renuencia en ciertos momentos de algunos de ellos.

Desde luego, toda la Ingeniería desarrollada quedó documentada para facilitar el mantenimiento de los programas.

6.2 Establecimiento de Estándares

El objetivo de la creación de estándares fue que los tesisistas desarrollaran, diseñaran y documentaran bajo los mismos lineamientos.

La estandarización ofrece muchas ventajas, algunas de ellas se listan a continuación :

- Se crean documentos con la misma estructura y el mismo formato dando un imagen común que ayuda a personas ajenas entender el sistema.
- Evita al desarrollador perder tiempo en pensar la manera en que debe hacer ciertas cosas, como nombrar archivos, nombrar objetos, guardar los programas en un sitio determinado, asignar colores, tipos de letras, etc.
- Respecto a la programación, ayuda a documentar los programas para que el código, los comentarios, etc., queden de tal forma que cualquier otro desarrollador entienda lo que se está haciendo.

Los estándares que se desarrollaron se dividen en tres partes principales :

1. Estándares de diseño de las aplicaciones.
2. Estándares orientados a la programación.
3. Estándares orientados al uso de las herramientas.

No se explicará a detalle cada uno de las partes, ya que de alguna forma u otra están interrelacionadas, tal como se explica en la siguiente sección.

6.2.1 Panorama General

El primer estándar que se creó estuvo destinado a establecer algunos puntos que se consideraron imprescindibles para iniciar el desarrollo. De esto surgió el Manual de Estándares de Progress 8. Estos estándares se establecieron después de la actividad de un día y una hora (para más información sobre estas actividades consulte el Capítulo 4 'Estudio de Desarrolladores'). En estas actividades se detectaron los puntos que debían cubrirse en el manual.

En este manual se establecieron algunos aspectos como:

- La estructura y asignación de los directorios de trabajo para cada módulo,
- El diseño de las aplicaciones: tipos de letra, colores, iconos, etiquetas para los objetos, archivos, mensajes, opciones de menú, etc.
- Especificaciones para la programación: manera de nombrar variables, ventanas, objetos, archivos, formato de comentarios, etc.
- Consideraciones en la creación de las tablas: manera de nombrar las tablas, índices, secuencias, etc.

Paralelamente al Manual de Estándares de Progress se creó el Manual de Estándares de EasyCASE. EasyCASE es la herramienta que se utilizó en la etapa de análisis para la elaboración de Diagramas de Flujo de Datos. En ese manual se establecieron algunos aspectos como:

- Nombre de los archivos
- Nombre y tamaño de los objetos
- Nombre de los flujos

En este manual también se estableció un algoritmo para la realización de la abreviatura de las palabras utilizadas en la descripción de los flujos, procesos, etc.

En el transcurso del desarrollo se observó que la mayoría de los usuarios trabajarían con una resolución de pantalla de VGA (640x480). Debido a esto se redefinieron los estándares que especificaban el tamaño de letra, la fuente, la geometría, etc., de los objetos, ya que anteriormente se habían definido para una resolución de SVGA (800x600). Los nuevos quedaron asentados en el Manual de Estándares de Pantalla VGA.

Otro manual que se creó fue el Manual de Estándares de Pantallas de Altas, Bajas y Cambios y Consulta para Progress 8. Uno de los objetivos de este manual fue el estandarizar las pantallas para que el usuario no se sintiera confundido al pasar de una pantalla a otra y así visualizara un producto integral.

Los estándares que se desarrollando fueron los siguientes:

- Para pantallas de captura :
 - ⇒ Estándar M-Alfa
 - ⇒ Estándar M-Beta

- Para pantallas de consulta :
 - ⇒ Estándar C-Alfa
 - ⇒ Estándar C-Beta

Además se incorporó el estándar del botón de 3 puntos y la ingeniería para realizar búsquedas. En la Figura 6.9 se observa un ejemplo de una pantalla de Altas, Bajas y Cambios con los estándares aplicados.

Figura 6.9. Pantalla de Altas, Bajas y Cambios con Estándares.

The screenshot shows a web application window titled "SINAI - Módulo de Compras" with a sub-header "Proveedores". The main content area is a form for "Datos Generales" (General Data) of a supplier. The form includes the following fields and values:

- No. proveedor: 1
- Nombre o razón social: LA TARJETA SA DE CV
- Nombre comercial: LA TARJETA SA DE CV
- RFC: PUE TAR-821213- Atención: SRA ALBA
- Proveedor sugerido: n Teléfono: 32 63-86 Fax: 32-26 52
- Domicilio: 12 ORIENTE #210 CP: [empty]
- Clave de colonia: 16 Las Palmas
- Clave de ciudad: 2 Puebla
- Clave de estado: 2 Puebla
- Clave de país: 1 México
- Clave giro comercial: 2 Papelería

At the bottom of the form, there is a "Criterio de búsqueda" (Search criteria) section with two radio buttons: "Por Clave" (selected) and "Por Descripción". Below this is a "Buscar:" input field. To the right of the search section are several buttons: "Nuevo", "Guardar", "Borrar", "Salir", and "Ayuda". There are also navigation arrows (back, forward, etc.) and a "Examinar" button near the top right of the form.

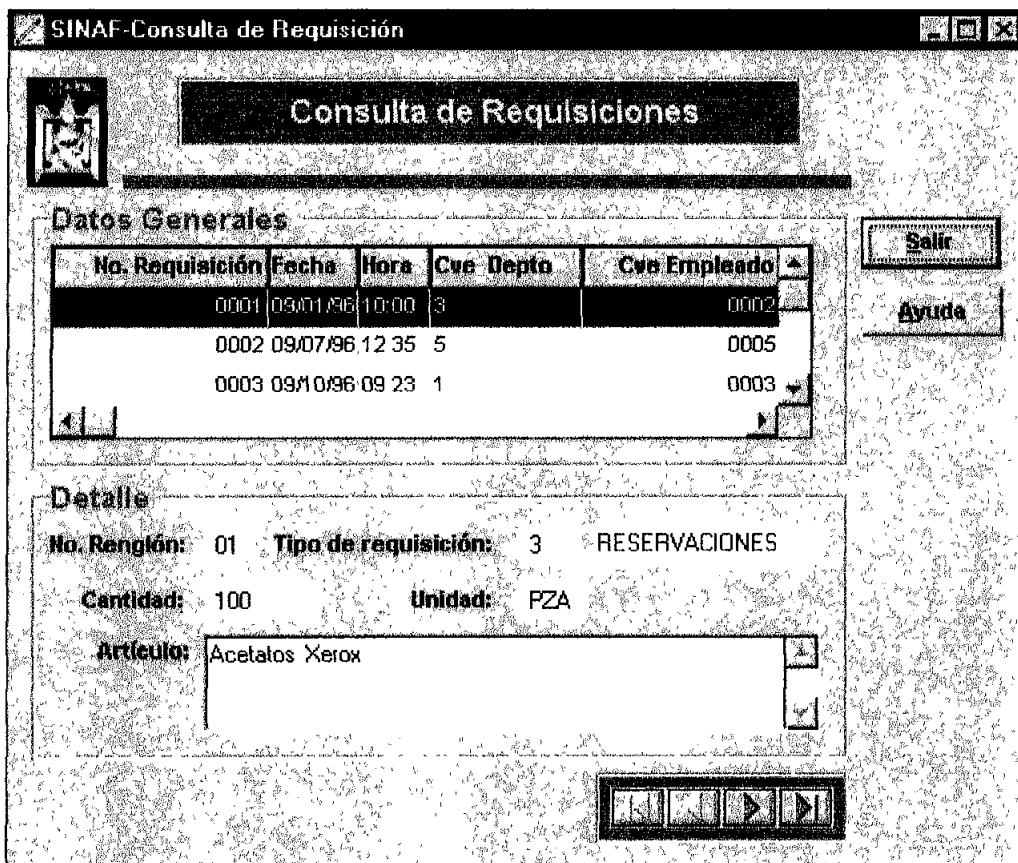
Fuente: Módulo de Compras.

El estándar de botón de 3 puntos se creó e incorporó a estas pantallas por la siguiente razón:

La normalización de las tablas exigía manejar campos clave en la captura de la información. Al inicio se tenía que teclear la clave sin visualizar el dato que representaba. Como los datos podían crecer en gran cantidad en ciertos casos, se pensó que el usuario pudiera consultar la lista existentes de claves y sus descripciones para hacer más cómoda y fácil la captura.

En las pantallas de consulta; se buscó que la forma de consultar fuera de la misma manera en todos los módulos, además de mostrar de forma clara las relaciones entre los datos. En la Figura 6.10 se puede observar un ejemplo de una pantalla de consulta estandarizada.

Figura 6.10. Pantalla de Consultas con Estándares.



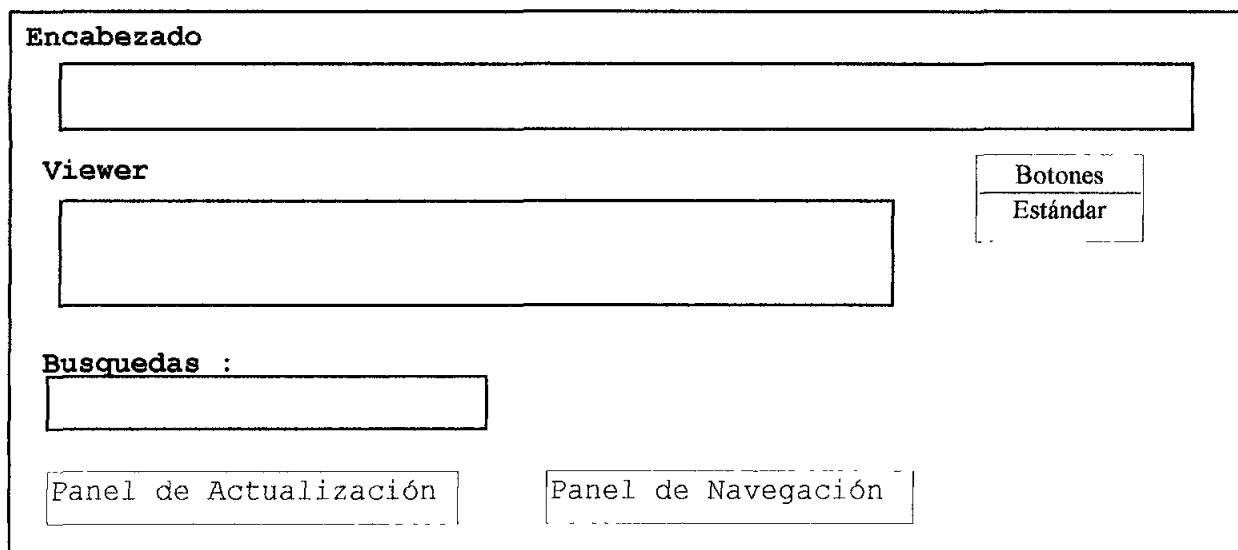
Fuente: Módulo de Compras.

Una vez que se creaban los programas, era necesario darles mantenimiento, ya sea por cambios en el diseño o por depuración de errores.

Debido a la metodología de desarrollo de aplicaciones de Progress basado en componentes surgió la necesidad de documentarlos de tal forma que permitiera saber a cualquiera como estaban contruidos los programas. La manera ideal era haciendo un catálogo de objetos de todos los programas, especificando las características de cada objeto. De ahí surgió el Manual para Documentar el Catálogo de Objetos.

Cada módulo incluyó en el manual técnico el catálogo de objetos correspondiente. En la Figura 6.11 se observa un ejemplo del Catálogo de Objetos de Servicio Social

Figura 6.11. Ejemplo del Catálogo de Objetos de Servicio Social.



Fuente: Manual Técnico de Servicio Social.

Por último, cuando se crearon las plantillas de Ingeniería de Software, fue un requisito indispensable elaborar una manual que enseñara el uso de dicha ingeniería: Manual de Uso de la Ingeniería.

En dicho manual se explica a detalle la construcción de cada uno de los componentes de las diferentes pantallas (de Altas, Bajas y Cambios, de Consulta, de Selección, etc.).

6.2.2 Resultados

El establecimiento de los estándares fue una tarea ardua y en ocasiones tediosa en su implementación, tanto para Ingeniería de Software como para los tesisistas.

Se presentaron algunos problemas:

- Las mejoras y cambios en ocasiones no eran bien recibidos por los tesisistas a causa del trabajo que implicaba.
- Los problemas de comunicación que se daban cuando se hacían cambios complicaba la correcta implementación de los estándares, ocasionando la molestia de algunos tesisistas ya que trabajaban doble ya que hacían pequeños modificaciones.
- A veces se entregaban archivos donde se aplicaban los cambios en los estándares, pero la documentación de tales cambios se entregaba después. Esto ocasionaba confusión y molestia en ciertos casos, ya que se establecían tiempos de entrega que no podían ser cumplidos debido a esta situación.
- Faltó definir líneas claras de comunicación para enterar a los tesisistas de los cambios realizados además de que no se hicieron reuniones para tal propósito. Este problema se refiere más que nada a la coordinación de grupo, el cual se detalla en la siguiente sección.

Cabe hacer la justa mención que algunos tesisistas colaboraron aportando ideas y cambios que mejoraron los estándares.

6.3 Capacitación

La primera necesidad que surgió, y a la que había que satisfacer antes que nada, fue la capacitación. La razón principal fue que se usarían herramientas hasta ese entonces desconocidas para la mayoría de las personas involucradas.

El equipo de Ingeniería de Software debía antes que nadie aprender, al menos lo básico, el uso de dichas herramientas para transmitir el conocimiento a los tesisistas. En la siguiente sección se detalla el curso de la capacitación de los tesisistas.

6.3.1 Panorama General

El primer curso que se dio fue de Progress versión 7.3B. La versión de Progress 8 no se obtuvo hasta el mes de mayo. Como la versión 7.3B no tenía la característica de SmartObjects, el curso consistió principalmente en enseñar el lenguaje de cuarta generación de Progress (4GL) y el uso del User Interface Builder .

Este curso se llevó a cabo en 8 horas, para lo cual se reunió a los tesistas en salones previamente seleccionados.

Para este curso y los siguientes se utilizaron televisiones y un PC/TV como medio de presentación. Como la cantidad de tesistas no era muy grande, no se tuvieron problemas con la visualización del curso.

Lo que se dio en este curso sirvió de base para llevar a cabo la Actividad de un Día. El detalle de la actividad y los resultados que se obtuvieron se encuentran en el Capítulo 4.

Después del curso se obtuvo Progress versión 8, lo cual hizo necesario dar nuevos cursos, en los cuales se enseñara las características de Progress, especialmente los SmartObjects. Debido a que la adquisición de Progress 8 no se esperaba tan pronto, estos cursos no se planearon adecuadamente por el poco tiempo y la necesidad de que los tesistas aprendieran todo lo posible para empezar a programar; por eso resultó un poco difícil darlos.

El primer curso trató de los SmartObjects. En éste se dieron los conceptos básicos de los SmartObjects :

- Ligas
- Propiedades
- Tipos de SmartObjects

Estos fueron los cursos propiamente dichos. Sin embargo algo que también cae en este aspecto son las explicaciones de los manuales de estándares. Aunque no se hizo una planeación necesario, si se tuvieron que llevar a cabo, debido a dificultades de los tesistas en aplicar los estándares, además de la poca experiencia del equipo de Ingeniería de Software en el desarrollo de los mismos. Un factor que ayudó mucho en adecuada elaboración de los manuales fue el equipo de Calidad, lo cual se trata en la Sección 6.6.

La liberación de los estándares implicaba una breve exposición de los mismos. Estas exposiciones servían además para resolver dudas sobre versiones anteriores de estándares. Las sesiones que se tuvieron, se llevaron cabo en el momento que era necesario. Se hicieron exposiciones de casi todos los manuales de estándares.

6.3.2 Resultados

Algunos de los resultados que se obtuvieron de los cursos fueron:

- No se aprovecharon totalmente los cursos, en especial el curso de Progress v7.3B. Esta aseveración se basa en los resultados del Capítulo 4 'Estudio de Desarrolladores'.
- El poco aprovechamiento no se debe solamente a los tesistas; otros factores que influyeron fueron:
 - ⇒ La falta del equipo adecuado para dar un curso de esa naturaleza (computadora para practicar, proyectores adecuados, etc.).
 - ⇒ La falta de experiencia en capacitación del equipo de Ingeniería de Software.
 - ⇒ No se tenía acceso a manuales y material de apoyo.
 - ⇒ Dificultades técnicas para manejar el equipo.
- Faltó una planeación más formal en lo que respecta al horario, sobre todo en los cursos rápidos.
- En ocasiones se dificultó el establecimiento del horario, porque algunos cursos se dieron durante los semestres.
- Los tesistas se dieron cuenta de la importancia de tratar de aprovechar al máximo los cursos que se les impartía, después de la actividad de un día.

6.4 Coordinación de Grupo

Como en todo proyecto de desarrollo que involucra a un grupo de personas llevando a cabo diferentes tareas, debe darse un control de todas las actividades que se lleven a cabo.

Este proyecto estuvo conformado por tres grupos :

- El equipo de tesistas
- El equipo de Ingeniería de Software
- El equipo de Calidad

El establecimiento de estándares así como el uso de la ingeniería no se podían dar sin una coordinación entre el grupo de desarrolladores e Ingeniería de Software.

Esta coordinación estuvo a cargo del equipo Ingeniería de Software, siendo una más de sus responsabilidades.

6.4.1 Panorama General

Al inicio del desarrollo no se planeó la elaboración de un horario de trabajo, ya que los tesisistas estaban en la etapa de análisis. Sin embargo se les citaba cuando se tenía listo un manual o se les tenía que dar información diversa.

Se inició la coordinación cuando empezaron a programar los módulos. Se tenía una fecha de entrega en la cual debía mostrarse el prototipo de los módulos, pero ya integrados. Se fijaron los días y las horas en la que debían entregar los módulos para que Ingeniería de Software integrara el producto. Si se encontraban errores se le comunicaba dándoles otra fecha para entregar las correcciones.

Cuando se presentó el primer prototipo, este tuvo muchos errores. Se notó la falta de consistencia en la integración de los módulos. Uno de los factores de la falta de consistencia fue que no se integró el producto en sesiones donde estuvieran los tesisistas y se les comunicara los errores inmediatamente para así tener los módulos funcionando correctamente e integrarlo de manera adecuada.

Teniendo en cuenta eso, se decidió empezar a trabajar en un solo lugar donde todos estarían reunidos: tesisistas, Ingeniería de Software y Calidad. Para ese entonces iniciaron las vacaciones, lo cual permitió fijar el siguiente horario:

Lun a Vie 9:00 AM a 2:00 PM y 4:00 PM a 8:00 PM

El lugar designado fue la casa de una de las personas del equipo de Calidad. El hecho de trabajar en un sólo lugar tuvo muchas ventajas :

- Coordinar a las personas se hizo más fácil ya que todos se enteraban al mismo tiempo de las decisiones que se tomaban.
- Mejoró la comunicación de los cambios en los estándares, de la ingeniería que se desarrollaba, etc.
- Mejoró la asistencia técnica. Los problemas que surgían se comunicaban en el momento, se entendía mejor el problema y la solución se daba en ese instante o se empezaba a trabajar en la solución cuando requería más tiempo.

Se trabajó así durante un mes hasta la segunda entrega del prototipo. En esta segunda entrega, el producto demostró tener más solidez en la integración, demostrando que el trabajo conjunto en un lugar arrojó muchos beneficios.

Las actividades más relevantes fueron las siguientes :

- Se trabajó en la revisión y corrección de los Diagramas de Entidad-Relación

- Se crearon las tablas generales, distribuyendo el desarrollo de las mismas a los tesisistas.
- Se creó el Manual para Documentar el Catálogo de Objetos.
- Se siguió desarrollando ingeniería, al detectar más necesidades.
- Se inició con la revisión de la calidad de los programas.

Después, se decidió cambiar de lugar de trabajo al edificio propiedad de un integrante del equipo de Ingeniería de Software. Esto se debió a que el lugar anterior quedaba muy lejos para la mayoría de las personas y causaba problemas de costos y horarios.

Las clases iniciaban cuando se llevó a cabo el cambio. En el edificio se trabajó durante 2 semanas. La actividad principal se centró en seguir desarrollando los módulos. No se dispuso mucho tiempo en este edificio debido a problemas con el dueño del mismo. Esto obligó a otro cambio.

El sitio elegido fue la casa de una persona perteneciente al grupo de tesisistas. Ahí se tuvo más estabilidad en cuestión de horario y ubicación. Se acercaba la fecha de una nueva entrega del prototipo del sistema y se trabajó más intensamente.

Las actividades más relevantes que se llevaron a cabo en ese lugar fueron :

- Se mejoró la integración de la base de datos, ya que había inconsistencia en las versiones anteriores.
- Se trabajó en el desarrollo de nuevos estándares e ingeniería, además de enseñar el uso de los mismos.
- Se inició con la definición de los datos de prueba.
- Los módulos se sometieron a revisiones más rigurosas de calidad.

En este lugar se trabajó durante mes y medio hasta la fecha de la última entrega.

6.4.2 Resultados

Se enfrentaron dificultades que se tienen siempre que se trata de coordinar a un grupo bastante heterogéneo de personas:

- Ponerse de acuerdo en un horario de trabajo en el que se vería el desarrollo de los módulos.
- Falta de experiencia del equipo de Ingeniería de Software.
- Falta de madurez para llevar a cabo las responsabilidades.
- Poca disciplina para hacer las cosas.

- En ocasiones no se tenía la flexibilidad necesaria para cumplir con los horarios de trabajo.
- No se definieron líneas de comunicación que permitieran hacer saber a los tesisistas los cambios realizados.

La coordinación fue una tarea bastante desgastante, tanto para el grupo de desarrolladores como del equipo de Ingeniería de Software y Calidad. Aun así, esta tarea fue bastante importante para apoyar la aplicación de los estándares y el uso de la ingeniería.

En un principio, la coordinación no fue buena debido a la dispersión de los tesisistas. En ocasiones era difícil localizarlos cuando se tenía que comunicar algo y más difícil era reunirlos a tiempo en un lugar. Algo que complicaba la comunicación, era que algunos de los tesisistas no contaban con teléfono, tenían horarios poco flexibles o tenían que irse a sus casas porque no vivían en la ciudad de Puebla.

Sin embargo, la tarea de coordinación mejoró bastante cuando se tomó la decisión de reunir a las personas involucradas en el proyecto en un mismo lugar y con un horario bien definido. Se aprovechaba cuando estaban reunidos para comunicar cambios, mejoras, etc.

6.5 Soporte Técnico

Una de las responsabilidades que Ingeniería de Software tuvo durante el desarrollo fue el dar soporte técnico a los tesisistas en el momento que se requiriera.

El soporte técnico consistía en ofrecer ayuda en aspectos relacionados con el uso de las herramientas. En ningún momento el soporte debía estar destinado a elaborar los programas que se tenían que hacer. Este punto se enfatizó en juntas de carácter general.

6.5.1 Panorama General

La forma de dar soporte técnico cambió en el transcurso del desarrollo en base a la evolución de los tesisistas y el proyecto.

En un inicio se estableció proporcionar la asesoría en la computadora destinada para uso exclusivo de tesisistas de la Escuela de Sistemas Computacionales. Se fijó un horario en el que el equipo de Ingeniería de Software estaría resolviendo las dudas que tuvieran los tesisistas. Esto ocurrió durante el séptimo semestre. Cuando inició el período

de verano, este tipo de soporte dejó de darse y se optó por darlo vía telefónica y en el lugar de trabajo del equipo de Ingeniería de Software.

Se intentó hacer llevar un control de las llamadas y problemas, en una hoja de control. Este formato se les enseñó para que al momento de hacer la llamada dieran los datos necesarios. Sin embargo esto no funcionó, debido a lo siguiente :

Los tesisistas no proporcionaban los datos necesarios para el registro.

En ocasiones se les pedía que llamaran más tarde para darles la solución, ya que el problema no se podía resolver en ese momento. Sin embargo, la mayoría de las veces, no volvían a llamar.

Resultó difícil en la mayoría de las veces entender exactamente el problema que tenían, por lo tanto no se podía registrar adecuadamente.

A partir de ese momento, hasta el final del servicio que fue el 8 de diciembre, el soporte técnico se dio sin llevar un registro. Antes de reunirse en un sólo lugar para trabajar, los tesisistas hablaban o llegaban con Ingeniería de Software, el tiempo de respuesta variaba de acuerdo al tipo de problema.

Cuando se empezó a trabajar todos reunidos, el soporte mejoró, porque se daba en el momento, además de que los demás tesisistas contribuían a resolver algunos problemas.

6.5.2 Resultados

Se cumplió el objetivo del soporte técnico, por supuesto, en la medida de lo posible. El soporte estuvo muy ligado al proceso de estandarización, ya que no sólo consistía en resolver problemas relacionados con la programación, sino en resolver dudas de los manuales y la documentación que se liberaba.

6.6 Control de Calidad

Se debe hacer la aclaración que el título de esta sección se refiere a 'Aseguramiento de Calidad', es decir la revisión de los estándares de calidad del producto durante todo el proceso.

Como se sabe, el control de calidad se refiere a la revisión de los estándares de calidad del producto al final del proceso. Durante el desarrollo lo que se pretendió fue llevar un aseguramiento de calidad y no un control, con el fin de permitir a los

desarrolladores entregar aplicaciones que cubriesen en un gran porcentaje el objetivo de las mismas.

Para lograr esto se trabajó conjuntamente con el equipo de Calidad, el cual tenía como objetivo crear manuales de procedimientos para la implementación del estándar ISO9000-3. Aunque el equipo de Calidad consistía en dos personas, sólo una de ellas se encargó del control de calidad en el proyecto UPAEP.

6.6.1 Panorama General

El aseguramiento consistió en la revisión continua de los estándares que se establecieron y del funcionamiento correcto de los programas antes y durante la integración de los mismos.

Esto se dio cuando se llevó a cabo el trabajo conjunto en un solo lugar. Al inicio del proyecto no se pudo llevar a cabo el control de calidad debido a la dificultad de coordinación de los tesistas.

Las diferentes entregas de los prototipos permitió ver la calidad con la que se hacían los productos.

6.6.2 Resultados

Se sometieron a mejoras continuas tanto los estándares como las aplicaciones, con el fin de que las aplicaciones lucieran y funcionaran de manera similar y de manera integrada. Esto se alcanzó gracias a la aplicación del aseguramiento de calidad.

Se hace notar que en ocasiones no se lograba su aplicación al 100% por diferentes causas, algunas de ellas fueron:

- La falta de tiempo para aplicarlos. La presión para entregar los prototipos no permitía tomar el tiempo necesario para aplicar adecuadamente los estándares.
- La falta de comunicación entre tesistas y las personas que se encargaban del aseguramiento. Cuando se trabajaba por separado era muy difícil revisar los estándares.

Aun así, el trabajo de Calidad en conjunto con Ingeniería de Software fue satisfactorio en el sentido de que durante el proceso y al final de éste los productos son bastante sólidos y funcionan de manera integral.

Conclusión.

Este trabajo surgió para satisfacer diferentes necesidades relacionadas con el desarrollo de aplicaciones usando como herramienta un manejador de base de datos.

Se realizó un estudio comparativo donde se encontró que Progress V8 ofrecía un sin número de características muy importantes, las cuales necesitaban una estrategia para su explotación al máximo. Además se realizó una encuesta para determinar qué tanto sabían los tesisistas sobre cuestiones de Ingeniería de Software y se encontró que se tenían muchas deficiencias.

Se enfrentaron diversos problemas como el no contar con un buen material didáctico o de referencia sobre Progress, la apatía de las personas en seguir estándares e ingeniería, las dificultades intrínsecas para coordinar al grupo de tesisistas involucrados en el estudio, e inclusive la coordinación entre las áreas de Calidad e Ingeniería de Software.

Las diversas soluciones que se tomaron para resolver lo anterior son las siguientes :

- Impartición de cursos a los tesisistas, para incrementar el nivel de manejo del lenguaje de Progress.
- Elaboración de manuales (de estándares, de uso e implantación de ingeniería, y didácticos), los cuales sirvieron para normar los diversos aspectos que involucraron el desarrollo.
- Apoyo técnico, para resolver los problemas de programación y utilización tanto de Progress como de la documentación generada.
- Se optó porque todas las personas que estaban involucradas con el proyecto trabajaran en un solo lugar, con el fin de resolver los problemas de coordinación entre las áreas de calidad, Ingeniería de Software y el grupo de tesisistas.

Se finaliza diciendo que se alcanzaron los objetivos de capacitación, estandarización, coordinación y desarrollo de ingeniería, con lo que los tesisistas lograron desarrollar las aplicaciones en menos tiempo y con mayor facilidad. Así mismo se logró el objetivo de documentar adecuadamente los sistemas desarrollados.

Glosario

Aseguramiento de Calidad. Es la verificación y corrección de estándares de calidad en cada etapa de la producción o del desarrollo.

Back End. Aplicación que se encarga de la administración de los datos en un ambiente cliente-servidor.

Benchmarking. Es un proceso sistemático y continuo para evaluar los productos, servicios y procesos de trabajo de las organizaciones que son reconocidas como representantes de las mejores prácticas, con el propósito de realizar mejoras organizacionales.

CASE (Computer-Aided Software Engineering). Ingeniería de software asistida por computadora o ingeniería de sistemas asistida por computadora. Software que se utiliza en una o todas las fases del desarrollo de un sistema de información, incluyendo análisis, diseño y programación. Por ejemplo, los diccionarios de datos y herramientas de diagramación ayudan en las fases de análisis y diseño, mientras que los generadores de aplicaciones aceleran la fase de programación.

Ciclo de Vida del Software. Vida útil de un software. La duración del ciclo de vida depende de la naturaleza y volatilidad del negocio, así como de las herramientas de desarrollo de software utilizadas para generar las bases de datos y programas de aplicación.

Computación Paralela. Arquitectura de computadora hecha de múltiples procesadores.

Control de Calidad. Es la verificación de estándares de calidad al final de la producción o del desarrollo.

DD. véase Diccionario de Datos.

DE. véase Diagrama de Estructura.

DER. véase Diagrama de Entidad-Relación.

DFD. véase Diagrama de Flujo de Datos.

Diagrama de Estructura. Es la representación gráfica de los menús de la aplicación.

Diagrama Entidad-Relación. Es una representación gráfica que describe los atributos de entidades y las relaciones entre ellos en una base de datos.

Diagrama de Estructura de Datos. véase Diagrama de Flujo de Datos.

Diagrama de Flujo de Datos. Es la representación gráfica de las técnicas de análisis estructurado. Los diagramas muestran el flujo de datos entre los procesos, las entidades externas que proveen esos datos y las unidades de almacenamiento de los mismos.

Diccionario de Datos. Contiene el nombre, tipo, rango de valores, fuente y autorización para el acceso a cada elemento de datos en los archivos y bases de datos de la organización; indica así mismo qué programas de aplicación utilizan dichos datos.

Flujo de control. Es la gestión de la transmisión de datos. Asegura que la estación de recepción pueda procesar los datos antes de que se envíe el siguiente bloque.

Flujo de datos. En el diagrama de flujo de datos, es representado por una línea que indica la dirección de los datos entre los procesos.

Front End. Aplicación utilizada para el desarrollo de interfaces de usuario.

Indexación. Es la creación de índices basados en palabras o campos de datos clave. La indexación es el método más común para mantener el control de los datos en un dispositivo de almacenamiento.

Índices. Es un campo que contiene la ubicación de los registros de una tabla. Véase también Indexación.

Interfase. Es aquel dispositivo que establece un medio de comunicación entre el usuario y la computadora.

Internet. Red formada de 45,000 redes interconectadas en 70 países que utilizan el protocolo de comunicaciones en red TCP/IP. Es la red más grande de computadoras en el mundo.

Lower CASE. Herramientas CASE que generan código de programa.

Manejadores de Bases de Datos. Software que controla la organización, almacenamiento, recuperación, seguridad e integridad de los datos en una base de datos.

Metodologías estructuradas.

Métricas de Software.

Miniespecificaciones.

Modularidad. Es una técnica de programación que separa el diseño de un programa en componentes o módulos separados, cada uno de los cuales puede ser programado como una sola unidad. Permite que diferentes programadores desarrollen partes del programa, así cada parte puede probarse separadamente y luego unirse.

Multiusuario. Es un sistema de cómputo compartido por dos o más usuarios.

ODBC (Open DataBase Connectivity).

Paradigma. Un modelo, ejemplo o molde.

Productividad. Producir más con los mismos recursos y en el mismo tiempo.

Programación estructurada.

Protocolos de Red. Un conjunto de normas y regulaciones que gobiernan la transmisión y recepción de datos.

Prototipos. Se refiere a la creación de un sistema con carácter de prueba, para su verificación y aprobación.

Redes Neuronales Artificiales.

Reingeniería. Analizar un sistema ya terminado a fin de aislar sus bloques constructivos individuales.

Sistemas Distribuidos. Un sistema de computadoras conectadas entre sí por una red de comunicaciones. Cada sistema informático es elegido para manipular su carga local de trabajo y la red está diseñada para dar soporte al sistema como un todo.

Sistemas Expertos. Una aplicación de inteligencia artificial que usa una base de conocimiento de la experiencia humana para ayudar a la resolución de problemas.

SQL (Structured Query Language). lenguaje de consulta estructurado. Lenguaje utilizado para interrogar y procesar datos en una base de datos relacional.

Tecnología Orientada a Objetos. Paradigma que observa el mundo como objetos en lugar de procedimientos.

Transacciones. Actividad o solicitud. Ordenes de compra, ventas, cambios, altas y bajas son ejemplos de transacciones que se registran en un entorno de información para negocios.

Triggers. Es una sección de código que se ejecuta cuando ocurre un evento.

Upper CASE. Herramientas CASE que ayudan en el análisis y en el diseño de sistemas.



BIBLIOTECA CENTRAL
TESIS
USO ÚNICAMENTE EN SALA

Bibliografía

- Booch, Grady
Object-Oriented Analysis and Design with Applications
The Benjamin/Cummings Publishing Company, Inc., 1994.
- Jacobson, Ivar; Christerson, Magnus; Jonsson, Patrik; Övergaard, Gunnar
Object-Oriented Software Engineering; A use-case driven approach.
Addison-Wesley Publishing Company, 1993.
- Kassabgi George; Wood, William T.; Jannery, Michael; Feinstein, Steven J.; Seidl, Gerry.
Using Progress V8
QUE, 1995.
- Kendall, Kenneth E. y Kendall, Julie E
Análisis y Diseño de Sistemas
Prentice Hall, 1991.
- Martin, James
Information Engineering, Book I : Introduction
Prentice Hall, 1989.
- Martin, James
Information Engineering, Book II : Planning and Analysis
Prentice Hall, 1989.
- Martin, James
Information Engineering, Book III : Design and Construction
Prentice Hall, 1989.
- Pressman, Roger S.
Ingeniería del Software : Un enfoque práctico.
McGraw Hill, 1993.
- Scharch, Stephen R.
Software Engineering
IRWIN, USA, 1993.
- Senn, James A.



Análisis y Diseño de Sistemas de Información
MacGraw Hill, 1992.

Soluciones Avanzadas.
Julio, 1994.

Yourdon, Edward
Análisis Estructurado Moderno.
Prentice Hall, 1992.