



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO

INSTITUTO DE CIENCIAS BÁSICAS E INGENIERÍA

*Sistema para la Administración de la Información
en la Dirección Técnica del Gobierno del Estado de
Hidalgo.*

TESIS

QUE PARA OBTENER EL GRADO DE LICENCIADO EN
SISTEMAS COMPUTACIONALES

PRESENTA

FILIBERTO PÉREZ BARRERA

ASESOR
LIC. NORMA LAURA SALAZAR VIVEROS

Pachuca de Soto Hidalgo, México Noviembre de 2005.



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO
INSTITUTO DE CIENCIAS BÁSICAS E INGENIERÍA
SISTEMAS COMPUTACIONALES

P.D.L.S.C. PÉREZ BARRERA FILIBERTO
PRESENTE

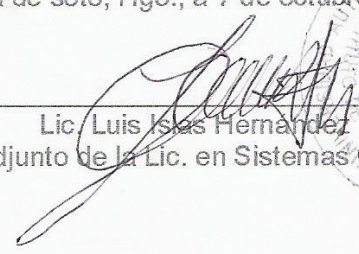
Por este conducto les comunico que el Jurado que les fue asignado a su trabajo de titulación denominado "SISTEMA PARA LA ADMINISTRACIÓN DE LA INFORMACIÓN EN LA DIRECCIÓN TÉCNICA DEL GOBIERNO DEL ESTADO DE HIDALGO" y que después de revisarlo en reunión de sinodales han decidido autorizar la impresión del mismo, hechas las correcciones que fueron acordadas.

A continuación se anotan las firmas de conformidad de los integrantes del Jurado:

PRESIDENTE:	ING. ALBERTO A. GALINDO GALINDO
PRIMER VOCAL:	L.C. NORMA LAURA SALAZAR VIVEROS
SEGUNDO VOCAL:	LIC. LUIS RAMÓN MACIAS PULIDO
TERCER VOCAL:	M EN C. LUIS H. GARCIA ISLAS
SECRETARIO:	M EN C. GONZALO A. TORRES SAMPERIO
PRIMER SUPLENTE:	LIC. LUIS ISLAS HERNÁNDEZ
SEGUNDO SUPLENTE:	LIC. MACEDONIO MARQUEZ MARTINEZ

Sin otro particular, reitero a usted la seguridad de mi atenta consideración.

ATENTAMENTE
"AMOR, ÓRDEN Y PROGRESO"
Pachuca de soto, Hgo., a 7 de octubre de 2005


Lic. Luis Islas Hernández
Coordinador adjunto de la Lic. en Sistemas Computacionales



Dedicatorias

A mis padres

Por brindarme la oportunidad de superarme personal y profesionalmente, por su apoyo incondicional, por los sacrificios, por su cariño, por ser el ejemplo a seguir y por todo Gracias...

A mis hermanos

Por su apoyo y tolerancia...

A mi hija

Por sus sacrificios, por su cariño...

Agradecimientos

A esta universidad

Universidad Autónoma del Estado de Hidalgo, por brindarme las herramientas necesarias para crecer profesionalmente...

A mi asesora

Lic. Norma Laura Salazar Viveros, por guiarme en este proceso, por su apoyo y consejos...

A mis compañeros y amigos

Por su apoyo, comprensión y consejos...

A la Subdirección Informática de la SEDESO del Gobierno del Estado de Hidalgo

Por el apoyo recibido para la realización de este trabajo.



**SECRETARÍA DE PLANEACIÓN
Y DESARROLLO REGIONAL**



Pachuca de Soto, Hidalgo a 12 de septiembre de 2005.

A quien corresponda:

Por medio del presente envío un cordial saludo y al mismo tiempo hago de su conocimiento que el **P.L.S.C. Filiberto Pérez Barrera** fue parte del equipo de desarrollo en el proyecto “Sistema para la administración de la información de la Dirección Técnica del Gobierno del estado de Hidalgo”.

Dicho proyecto fue llevado a cabo en el periodo Enero – Junio de 2004 cumpliendo satisfactoriamente con las necesidades de dicha Dirección.

Tomando como base de desarrollo el lenguaje *Visual Basic 6.0*, como visualizador de reportes *Crystal Reports 8.0* y para la administración de la base de datos *Access 97*.


Atentamente

Ing. Juan Enrique Martínez Barrera.
Encargado de Subdirección Informática.

Plaza Juárez s/n, 4º Piso, Col. Centro, C.P. 42000, Pachuca de Soto, Hgo. Tel.: (771) 763 55 y 760 00 Ext. 6513

Índice general

Índice de tablas	IX
Índice de figuras	XI
Objetivo General	1
Objetivos Específicos	1
Descripción de la problemática	2
Justificación	3
Alcances	4
Limitaciones	5
1. MARCO TEÓRICO	7
1.1. Introducción a los Sistemas de Información	8
1.1.1. Concepto	8
1.1.2. Elementos	8
1.1.3. Actividades	9
1.1.4. Tipos de Sistemas de Información	9
Sistemas de Procesamiento de Transacciones	10
Sistemas de Apoyo para la Decisión	11
Sistemas de Información Gerencial	11
Sistemas Expertos	11
1.2. Desarrollo de Sistemas de Información	12
1.2.1. Modelo lineal secuencial	13
Investigación de Sistemas	14
Análisis de los requisitos	15
Diseño del Sistema	16
Generación de Código	17

Implantación y Pruebas	18
Mantenimiento	18
1.2.2. Modelo de construcción de prototipos	19
1.2.3. Modelo de desarrollo rápido de aplicaciones	21
Modelado de Gestión	21
Modelado de datos	22
Modelado del proceso	23
Generación de Aplicaciones	23
Pruebas y entrega	23
1.2.4. Modelo incremental	23
1.2.5. Modelo en espiral	25
1.2.6. Modelo de ensamblaje de componentes	27
1.2.7. Proceso Unificado de Racional (<i>Rational Unified Process, RUP</i>)	28
1.2.8. Programación Extrema (<i>eXtreme Programming, XP</i>)	32
Introducción	32
Definición	34
Historia	35
Roles	36
Las cuatro variables	37
Los cuatro valores	39
Comunicación	40
Sencillez	40
Retroalimentación	40
Coraje	40
Proceso de desarrollo	41
Fase I: Exploración	42
Fase II: Planificación de la Entrega	43
Fase III: Iteraciones	44
Fase IV: Producción	44
Fase V: Mantenimiento	45
Fase VI: Muerte del Proyecto	45
Prácticas XP	45
Planificación	46
Entregas pequeñas (<i>Small releases</i>)	46
Metáfora	46
Diseño simple	46
Pruebas (<i>Testing</i>)	47
Refactorización (<i>Refactoring</i>)	47
Programación en parejas (<i>Pair programming</i>)	47
Propiedad colectiva del código	47
Integración continua	48

40 horas por semana	48
Cliente in-situ	48
Estándares de programación	48
1.2.9. Desarrollo Guiado por la Funcionalidad (<i>Feature Driven Development, FDD</i>)	49
2. IMPLEMENTACIÓN DE LA METODOLOGÍA	53
2.1. Introducción	54
2.2. Definición del material	55
2.3. Proceso de desarrollo	56
2.3.1. Roles	56
2.4. Fase I Exploración	57
2.4.1. Interacción con el cliente	57
2.4.2. Historias de usuario	58
2.5. Fase II Planificación de entrega	63
2.5.1. Plan de entrega	64
2.6. Fase III Iteraciones	66
2.7. Primera Iteración	66
2.7.1. Tarea 1 Diseño de la Base de Datos	67
2.7.2. Tarea 3 Diseño de la interfaz de principal del sistema	72
2.7.3. Tarea 3 Diseño de la interfaz de captura de propuestas	74
2.7.4. Tarea 4 Ingreso y lectura de datos	77
2.8. Segunda Iteración	80
2.8.1. Tarea 5 Agregar propuesta de obra	80
2.8.2. Tarea 6 Editar propuesta de obra	83
2.8.3. Tarea 7 Eliminar propuesta de obra	85
2.8.4. Tarea 8 Actualizar propuestas de obra	87
2.9. Tercera Iteración	88
2.9.1. Tarea 9 Ordenamiento de propuestas de obra	89
2.9.2. Tarea 10 Filtrado de propuestas de obra	91
2.9.3. Tarea 11 Búsqueda de propuesta de obra	93
2.10. Cuarta Iteración	95
2.10.1. Tarea 12 Diseño de catálogo de Localidades	95
2.10.2. Tarea 13 Diseño de catálogo de Fondo	96
2.10.3. Tarea 14 Diseño de catálogo de Programas	97
2.11. Quinta Iteración	98
2.11.1. Tarea 15 Creación de archivo de usuarios	99
2.11.2. Tarea 16 Diseño de ventana para identificación de usuarios	100
2.11.3. Tarea 17 Cambio de contraseña de usuario	101
2.11.4. Tarea 18 Abrir Base de Datos	103
2.11.5. Tarea 19 Crear nueva Base de Datos	106

2.11.6. Tarea 20 Crear respaldo de Base de Datos	107
2.11.7. Tarea 21 Diseño de formato de reporte general	109
2.11.8. Tarea 22 Integración del formato al sistema	110
2.12. Sexta iteración	111
2.12.1. Tarea 23 Diseño de formato de reporte de propuesta	112
2.12.2. Tarea 24. Integración del formato al sistema	113
2.12.3. Tarea 25. Diseño de archivo del manual de operación	115
2.12.4. Tarea 26. Integración del archivo al sistema	116
2.12.5. Tarea 27. Bloqueo del sistema	119
3. RESULTADOS	121
3.1. Fase IV Producción	122
3.1.1. Paquete de instalación	122
3.1.2. Instalación del sistema	129
3.1.3. Pruebas funcionales	133
3.2. Fase V Mantenimiento	133
3.3. Fase VI Muerte del proyecto	134
4. CONCLUSIONES	135
5. TRABAJOS FUTUROS	139
A. Planificación de desarrollo	141
B. Código	151
C. Pruebas funcionales	155
D. Glosario	201
Bibliografía	205

Índice de tablas

2.1. Historia de Usuario 1. Ingreso de nueva propuesta de obra	59
2.2. Historia de Usuario 2. Gestión de propuesta de obra de obra	59
2.3. Historia de Usuario 3. Operaciones secundarias sobre propuestas	59
2.4. Historia de Usuario 4. Consulta de catálogos	60
2.5. Historia de Usuario 5. Control de acceso a usuarios	60
2.6. Historia de Usuario 6. Gestión de datos de usuarios	60
2.7. Historia de Usuario 7. Administración de la Base de Datos	61
2.8. Historia de Usuario 8. Reporte general de propuestas	61
2.9. Historia de Usuario 9. Reporte de propuesta	61
2.10. Historia de Usuario 10. Presentación de manual de operación	62
2.11. Historia de Usuario 11. Bloqueo de sistema	62
2.12. Iteraciones del proyecto	63
2.13. Planificación de entrega	64
2.14. Tarea 1. Diseño de la Base de Datos	67
2.15. Estructura de la tabla <i>FONDO</i>	69
2.16. Estructura de la tabla <i>LOCALIDADES</i>	69
2.17. Estructura de la tabla <i>MUNICIPIOS</i>	70
2.18. Estructura de la tabla <i>PROGRAMA</i>	70
2.19. Estructura de la tabla <i>PROPUESTAS</i>	71
2.20. Estructura de la tabla <i>REGIONES</i>	71
2.21. Tarea 2. Diseño de la interfaz de principal del sistema	72
2.22. Tarea 3. Diseño de la interfaz de captura de propuestas	74
2.23. Tarea 4. Ingreso y lectura de datos	77
2.24. Tarea 5. Agregar propuesta de obra	80
2.25. Tarea 6. Editar propuesta de obra	83
2.26. Tarea 7. Eliminar propuesta de obra	86
2.27. Tarea 8. Actualizar propuestas de obra	87
2.28. Tarea 9. Ordenamiento de propuestas de obra	89
2.29. Tarea 10. Filtrado de propuestas de obra	91
2.30. Tarea 11. Búsqueda de propuestas de obra	93
2.31. Tarea 12. Diseño de catálogo de <i>Localidades</i>	95

2.32. Tarea 13. Diseño de catálogo de <i>Fondo</i>	97
2.33. Tarea 14. Diseño de catálogo de <i>Programas</i>	98
2.34. Tarea 15. Creación de archivo de usuarios	99
2.35. Tarea 16. Diseño de ventana para identificación de usuarios	100
2.36. Tarea 17. Cambio de contraseña de usuario	102
2.37. Tarea 18. Abrir Base de Datos	103
2.38. Tarea 19. Crear nueva Base de Datos	106
2.39. Tarea 20. Crear respaldo de Base de Datos	107
2.40. Tarea 21. Diseño de formato de reporte general	109
2.41. Tarea 22. Integración del formato al sistema	110
2.42. Tarea 23. Diseño de formato de reporte de propuesta	112
2.43. Tarea 24. Integración del formato al sistema	114
2.44. Tarea 25. Diseño de archivo del manual de operación	115
2.45. Tarea 26. Integración del archivo al sistema	117
2.46. Tarea 27. Bloqueo del sistema	119
A.1. Estimación de salarios	142

Índice de figuras

1.1. Modelo lineal secuencial	14
1.2. Modelo de construcción de prototipos	20
1.3. Modelo de desarrollo rápido de aplicaciones	22
1.4. Modelo incremental	24
1.5. Modelo en Espiral	26
1.6. Modelo de ensamblaje de componentes	28
1.7. Modelo RUP	30
1.8. Modelo de Programación Extrema	35
1.9. Modelo de Desarrollo Guiado por la Funcionalidad	50
2.1. Estructura de las tablas de la Base de Datos.	68
2.2. Interfaz principal del sistema.	72
2.3. Interfaz de captura de <i>Propuestas</i>	75
2.4. Ingreso y lectura de datos.	78
2.5. Movimiento entre registros de la tabla <i>PROPUESTAS</i>	79
2.6. Inhabilitación de botón <i>Guardar</i> y <i>Cancelar</i>	81
2.7. Habilitación de botón <i>Guardar</i> y <i>Cancelar</i>	82
2.8. Botón <i>Editar</i>	84
2.9. Editar propuesta de obra.	85
2.10. Eliminar propuesta de obra.	86
2.11. Actualizar propuestas de obra.	88
2.12. Ordenamiento de propuestas de obra.	90
2.13. Filtrado de propuestas de obra.	92
2.14. Búsqueda de propuesta de obra.	94
2.15. Ventana de catálogo de <i>Localidades</i>	96
2.16. Ventana de catálogo de <i>Fondo</i>	97
2.17. Ventana de catálogo de <i>Programas</i>	98
2.18. Ventana para identificación de usuarios.	101
2.19. Ventana para el cambio de contraseña de usuario.	103
2.20. Menú <i>Archivo</i> , <i>Abrir BD</i>	104
2.21. Ventana <i>Abrir BD</i>	105

2.22.	Menú <i>Archivo, Crear nueva BD.</i>	106
2.23.	Menú <i>Archivo, Crear respaldo de BD.</i>	108
2.24.	Diseño de formato de reporte general.	110
2.25.	Menú <i>Consulta General, Reporte.</i>	111
2.26.	Diseño de formato de reporte de propuesta.	113
2.27.	Menú <i>Archivo, Imprimir Previo.</i>	114
2.28.	Menú <i>Ayuda, Manual de operación.</i>	117
2.29.	Manual de Operación.	118
2.30.	Menú <i>Archivo, Bloquear.</i>	120
2.31.	Ventana de identificación de usuarios.	120
3.1.	Ventana de desarrollo de <i>Visual Basic 6.0.</i>	123
3.2.	Asistente para empaquetado y distribución.	124
3.3.	Asistente para empaquetado y distribución, Secuencia de comandos.	125
3.4.	Asistente para empaquetado y distribución, Tipo de empaquetado.	125
3.5.	Asistente para empaquetado y distribución, Carpeta de paquete.	126
3.6.	Asistente para empaquetado y distribución, Controladores DAO.	127
3.7.	Asistente para empaquetado y distribución, Opciones de .cab.	127
3.8.	Asistente para empaquetado y distribución, Título de instalación.	128
3.9.	Asistente para empaquetado y distribución, Elementos del menú inicio.	128
3.10.	Asistente para empaquetado y distribución, Finalizado.	129
3.11.	Informe de empaquetado.	130
3.12.	Inicio de instalación.	130
3.13.	Instalación de Sistema de Captura de Propuestas v1.3.	131
3.14.	Instalación, carpeta de instalación.	131
3.15.	Instalación, nombre de grupo.	132
3.16.	Instalación, porcentaje de avance.	132
3.17.	Finalización de instalación.	133
A.1.	Planificación de desarrollo del sistema, Hoja 1 de 7.	143
A.2.	Planificación de desarrollo del sistema, Hoja 2 de 7.	144
A.3.	Planificación de desarrollo del sistema, Hoja 3 de 7.	145
A.4.	Planificación de desarrollo del sistema, Hoja 4 de 7.	146
A.5.	Planificación de desarrollo del sistema, Hoja 5 de 7.	147
A.6.	Planificación de desarrollo del sistema, Hoja 6 de 7.	148
A.7.	Planificación de desarrollo del sistema, Hoja 7 de 7.	149

Introducción

Objetivo General

Desarrollar un sistema capaz de administrar de forma rápida y correcta la información de las propuestas de obra, que los ochenta y cuatro municipios del estado de Hidalgo envían a la Dirección Técnica del Gobierno del Estado de Hidalgo, para su evaluación y posterior aprobación.

Objetivos Específicos

- Diseñar una base de datos, para dar soporte a los datos necesarios para el almacenamiento de la información.
- Diseñar una aplicación con base en la metodología de desarrollo de sistemas denominada *Programación Extrema*, para permitir la conexión y administración de la base de datos.
- Integrar a la aplicación, los formatos necesarios para los reportes pertinentes de la base de datos.
- Diseñar e integrar a la aplicación un archivo HTML, que sirva como manual de

operación del sistema.

Descripción de la problemática

El objetivo principal la Dirección Técnica de la Secretaría de Desarrollo Social del Gobierno del Estado es el de analizar la viabilidad y rentabilidad de las obras propuestas por los municipios del Estado de Hidalgo con la finalidad de realizar su aprobación y financiamiento, dicha información se recibe por escrito en forma de expedientes técnicos.

Un expediente técnico es un documento, que consta de un oficio con la solicitud de análisis de las obras propuestas y de una lista de todas ellas con sus datos correspondientes. Estos datos no se encuentran estandarizados por lo que cada municipio envía las características que cree convenientes.

Estos documentos llegan a ser de gran volumen ya que en cada uno de ellos se describen todas las obras propuestas por el municipio, la cuales se planean realizar en todo un año de gestión administrativa, lo cual llega a ser un gran problema a la hora de almacenar toda esa información.

Después de ser recibida esta información, es necesario registrarla de manera electrónica, esto se hace por medio de una hoja de cálculo en *Microsoft Excel*, en la cual se especifican cada una de las características que tendrá la obra propuesta. Por ejemplo cada obra propuesta por cualquier municipio debe llevar por lo menos los siguientes datos: *Nombre de la región, Nombre del municipio, Nombre de la localidad, Descripción de la obra, Fondo de aplicación, Programa, Número de beneficiarios, Inversión Fiscal, Inversión de los*

Beneficiarios e Inversión Total.

En promedio cada municipio propone realizar treinta obras y además si esto se multiplica por los ochenta y cuatro municipios, se obtiene un total de dos mil quinientas veinte obras propuestas, esto en realidad es muy engorroso de capturar por el volumen, además de que en algunos casos la información se repite, por ejemplo el nombre de la región, municipio y localidad.

Justificación

En la actualidad, el manejo de la información en las organizaciones, sea del tipo que sea, es vital para el funcionamiento y desarrollo de la misma. Sus actividades giran entorno a ella por lo cual es importante un manejo rápido y eficiente para alcanzar el objetivo principal de la organización.

Con el avance tecnológico en el campo de la informática, las organizaciones se han dado cuenta de la importancia del uso de *sistemas de información*, los cuales aportan grandes beneficios en el procesamiento de la información, de una manera fácil, rápida y confiable.

Además, estos sistemas de información han servido como pieza fundamental en la toma de decisiones, que han de realizar los directivos para la solución de problemas concernientes al giro de la organización.

Es por ello que el desarrollo de un sistema que permita el manejo y administración de la información, en la Dirección Técnica de la Secretaría de Desarrollo Social del Gobierno

del Estado, es de suma importancia para desempeñar sus actividades de manera rápida y eficiente y así alcanzar el objetivo principal de la Dirección.

Además de la reducción de tiempo de captura, este sistema permitirá el acceso y visualización de manera eficiente de los datos, ya que no será necesario trasladarse al archivo y buscar en primer lugar el año, municipio, localidad y descripción de cada obra propuesta por el mismo municipio. Esto se logrará por medio de una opción dentro del sistema que visualizará la información requerida por medio de un formato de reporte capaz de agrupar la información de acuerdo a ciertos parámetros de selección.

El acceso a la información se verá favorecido con la implantación de este sistema, ya que solo permitirá el acceso a la base de datos a usuarios permitidos, por medio de nombres de usuario y claves únicas.

Alcances

El presente trabajo tiene como objeto mostrar al lector una introducción breve acerca de lo que son los sistemas de información, sus elementos, actividades, así como los tipos que existen.

Además pretende mostrar de manera general los métodos de desarrollo de sistemas. Empezando por los métodos tradicionales como lo son el *Modelo Lineal Secuencial* hasta llegar al *Modelo de Ensamblaje de Componentes*.

También se presentan algunos de los métodos ágiles y pesados del desarrollo de sistemas.

En este caso se realiza una revisión general de *RUP* que es considerado como pesado, y de *FDD* que puede ser considerado como ágil o pesado según el criterio del desarrollador.

Se hace un análisis mas profundo del método ágil *XP*, para mostrar al lector las bases del presente trabajo. Posteriormente se detalla paso por paso la implementación de la metodología de desarrollo.

Con el análisis anterior se pretende mostrar las bases y alcances de *XP*, para que puedan ser evaluados y así *XP* pueda ser implementada en cualquier otro proyecto de desarrollo de sistemas.

Limitaciones

Existe una gran cantidad de métodos ágiles y pesados, en este trabajo solo se analizan tres. Un ágil y un pesado y otro que puede ser clasificado por el desarrollador, ya que si se ha trabajado con *RUP* le parecerá ligero, sin embargo, si tiene experiencia en *XP* seguramente lo clasificará como pesado.

MARCO TEÓRICO

Resumen.

En este capítulo se presenta al lector la información necesaria para entender que es un sistema de información, así como su clasificación general. También se presentan los modelos de desarrollo de sistemas de información más comunes, desde los modelos tradicionales, hasta los modelos ágiles y pesados.

Objetivo del Capítulo.

- Presentar la definición y características de un sistema de información.
- Presentar la descripción general de cada uno de los tipos de sistemas de información.
- Proporcionar la descripción de los principales modelos tradicionales de desarrollo de sistemas de información.
- Proporcionar la descripción de los principales modelos ágiles y pesados de desarrollo de sistemas de información.

1.1. Introducción a los Sistemas de Información

1.1.1. Concepto

Un sistema de información es un conjunto de elementos organizados que permiten el procesamiento de la información, cumpliendo con los requerimientos de la organización para la cual fue creado, para soportar la toma de decisiones, la coordinación y el control de una organización. [1] [3]

1.1.2. Elementos

Para que un sistema de información pueda ser considerado como tal, debe tener los siguientes elementos para su correcto funcionamiento: [3]

- **Físicos**, constituido por las computadoras y sus periféricos, que permiten la ejecución de los elementos lógicos.
- **Lógicos**, todos los programas que son necesarios para la manipulación de la información.
- **Humanos**, aquellos individuos que alimentan al sistema con información y que a su vez lo mantienen en funcionamiento.
- **Básicos**, constituidos por los datos que son procesados para generar la información rápida y confiable.

Con estos elementos aseguramos que el sistema de información implantado en la organización sea considerado como un canal de flujo de información, con el propósito de tomar decisiones y ejecutar acciones.

1.1.3. Actividades

Un sistema de información puede desarrollar cuatro actividades básicas, que permiten el procesamiento eficaz de la información de manera rápida y confiable. [3]

La primera actividad es denominada *Entrada de información*, por que es en ella donde se lleva a cabo el proceso en el cual se toman los datos requeridos para procesar la información. Este proceso se puede hacer de dos maneras, manual o automáticamente. En el primer caso el usuario introduce la información directamente y en el segundo, los datos provienen de otros sistemas de información.

El *Almacenamiento de la información* es la actividad posterior, es un proceso en el que se guarda la información en bases de datos que pueden ser recuperadas y manipuladas en cualquier momento.

La actividad denominada *Procesamiento de la información* permite la transformación de los datos en información organizada y completamente útil para la toma de decisiones del usuario.

La *Salida de información* es la actividad final, en ella el sistema presenta la información procesada y organizada, para ser tomada por el usuario o por otro sistema de información.

1.1.4. Tipos de Sistemas de Información

El propósito de un sistema de información, puede ser muy amplio, todo depende de las necesidades de la organización. Existen distintos tipos de sistemas de información, entre

1.1. Introducción a los Sistemas de Información

los que destacan los siguientes:

Sistemas de Procesamiento de Transacciones

Se define como *transacción* un suceso que implica o afecta a una organización, y que está compuesta por datos referentes a ellas y que son de importancia para la organización.

[1]

Estos sistemas se encargan del procesamiento de los datos referentes a las transacciones, además de permitir la automatización de tareas y procesos operativos. La información que se obtiene como salida es utilizada posteriormente por los funcionarios de nivel operativo de la organización en la toma de decisiones. [3]

Las razones para el procesamiento de las transacciones son:

- **Clasificación**, implica agrupar todos los datos de acuerdo con características comunes.
- **Operaciones de cálculo**, consiste en realizar alguna operación para obtener resultados útiles.
- **Ordenamiento**, consiste en disponerlos de alguna forma o secuencia, facilita el procesamiento y la búsqueda.
- **Síntesis**, reduce los datos en información breve y concisa.
- **Almacenamiento**, permite el registro de todas y cada una de los suceso que afectan a la organización.

Sistemas de Apoyo para la Decisión

Los Sistemas de Apoyo para la Decisión son sistemas de información que tienen como propósito auxiliar al usuario con las decisiones únicas que no se repiten y que no tienen una estructura definida. Además de estar hechos a la medida de la persona o grupo que los usa en comparación con los Sistemas de Información Gerencial. [1]

El propósito de estos sistemas es el de responder correctamente a condiciones inesperadas y propias de la información. Esto permite que sean empleados en niveles altos de la organización.

Sistemas de Información Gerencial

Los Sistemas de Información Gerencial, también llamados Sistemas de Reportes de Gerencia, se dedican al apoyo de decisiones siempre que los requerimientos de información sean identificados, esto es que la información que necesita para la toma de decisiones haya sido analizada anteriormente, y que esta misma decisión pueda tomarse nuevamente. [1]

Estos sistemas pueden extraer la información necesaria de cualquier parte de la organización, por lo que la información necesaria ya se tiene almacenada al ser procesada por un sistema de transacciones.

Sistemas Expertos

Un caso especial de Sistemas de Información son los *Sistemas Expertos*, Estos pretenden resolver problemas, por medio de enfoques de razonamiento de la *Inteligencia Artificial*.

1.2. Desarrollo de Sistemas de Información

También llamados *Sistemas Basados en Conocimiento* puede resolver problemas de una organización por medio del uso del conocimiento capturado de un experto. Permitiendo obtener como salida la mejor solución a un problema. [1]

Los componentes de un Sistema Experto son *la base de conocimiento, la máquina de inferencia y la interfaz de usuario*. [3]

La base de conocimiento almacena toda la información, datos, reglas, casos y relaciones importantes y puede ser comparada con todos los conocimientos y experiencias de los expertos humanos que se obtienen a través de años de trabajo en un área específica.

La máquina de inferencias o motor de inferencias se encarga de realizar las búsquedas de la información y relaciones de la base de conocimiento, para proporcionar respuestas, pronósticos y sugerencias como lo haría un experto.

La interfaz de usuario permite a los usuarios el uso del sistema experto, para esto se utiliza software especializado que además permite diseñar, crear y actualizar el mismo sistema experto.

1.2. Desarrollo de Sistemas de Información

Cuando un integrante de la organización, con la capacidad de iniciar cambios dentro de ella, localiza un posible beneficio de un sistema nuevo o modificado, se inician las actividades del desarrollo de sistemas. [3]

Con frecuencia los administradores o gerentes son los integrantes de la organización quienes empiezan con los proyectos de desarrollo de sistemas. Además, dado que la propuesta se genera en un nivel alto, es muy probable que el proyecto tenga éxito. Pero esto no quiere decir que las iniciativas de desarrollo de sistemas no se planeen en cualquier nivel de la organización.

El proceso de desarrollo de sistemas también se denomina Ciclo de Vida del Desarrollo de Sistemas (SDLC, *Systems Development Life Cycle*), dado que las actividades relacionadas con dicho proceso son continuas. El SDLC es un enfoque formado por fases de análisis y diseño que sostiene que los sistemas son desarrollados de mejor manera mediante el uso de un ciclo específico de actividades. [3]

En general, cada sistema de información tiene un calendario y fechas límite, hasta que se aprueba y pone en marcha. La vida del sistema continúa con el mantenimiento y revisión ante fallas. Pero si el sistema requiere de cambios significativos, que sobrepasan el mantenimiento, es necesario poner en marcha un nuevo proyecto.

A continuación se presentan los ciclos de vida de desarrollo de sistemas más comunes.

1.2.1. Modelo lineal secuencial

El *Modelo lineal secuencial* también es llamado *Ciclo de vida clásico*, *Ciclo de vida tradicional* o *Modelo en cascada*. Sugiere un desarrollo secuencial del sistema dividida en fases distribuidas en el tiempo de desarrollo. Cuando se finaliza una fase se comienza la siguiente tomando como datos de entrada los datos obtenidos en la fase anterior. Por lo que permite que sea aplicado en cualquier tipo de proyecto. [2]

1.2. Desarrollo de Sistemas de Información

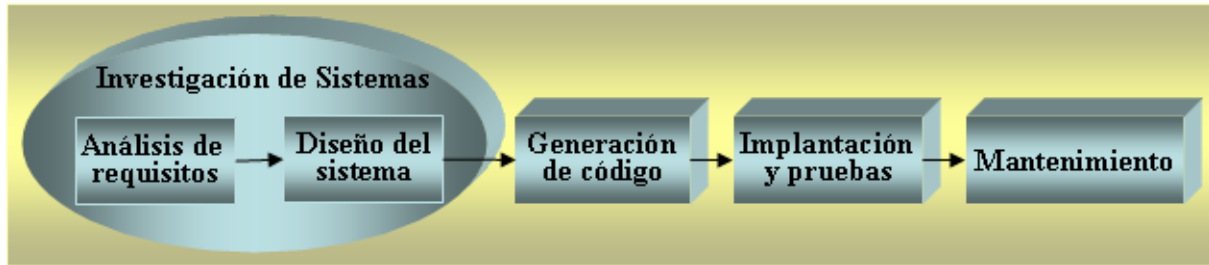


Figura 1.1: Modelo lineal secuencial

Las fases de este modelo son presentadas a continuación. Fig. 1.1

Investigación de Sistemas

En la primera fase se identifican los posibles problemas y oportunidades considerando los objetivos de la organización, esta etapa es fundamental para el éxito del proyecto. En ella se debe resolver a las preguntas “¿Cuál es el problema?” y “¿Vale la pena resolverlo?”. [3]

Para identificar el problema es necesario observar lo que sucede en la organización, de manera que pueda encontrar los problemas que afectan el desempeño de la organización.

Las oportunidades se presentan cuando se identifican los problemas y se considera que pueden ser resueltos por medio de sistemas de información. Esto permite un avance competitivo o la generación de estándares en la organización.

Se debe tener claro que es lo que esta tratando de hacer la organización, para ver si con la aplicación de sistemas de información, atacando problemas específicos, los objetivos se pueden alcanzar mas fácilmente.

Como salida de esta fase se obtiene un estudio de factibilidad que contiene una definición del problema y un monto específico de recursos, para que sea evaluado y ver si continúa con el proyecto propuesto.

Análisis de los requisitos

El Análisis de los requisitos debe responder a la pregunta “¿Qué debe hacer el sistema de información para resolver el problema?” Esta fase es dividida en dos subfases. [3]

La primer subfase consiste en recolectar las necesidades del usuario y a partir de ellas crear funciones que deberá realizar el sistema. Esta fase implica la creación de un documento de requisitos de usuario, que constituye la base de desarrollo para que al finalice acepte el sistema de acuerdo a este documento.[4]

Para determinar las necesidades del usuario se emplean herramientas de investigación, como cuestionarios, entrevistas, muestreos e investigación de los datos relevantes, etc.

La segunda subfase tiene por objeto realizar un análisis de los requerimientos del sistema, es decir, por medio de herramientas y técnicas especiales crear un modelo lógico del sistema de información, describiendo las funciones necesarias para la operación del sistema, para satisfacer los requerimientos del usuario.

Una herramienta para el modelo lógico, se puede aplicar los diagramas de flujo de datos para representar la entrada, proceso y salida de las funciones del sistema.

1.2. Desarrollo de Sistemas de Información

Como salida de esta fase se obtiene una lista de requisitos y prioridades así como un diagrama del sistema.

Diseño del Sistema

El diseño del sistema busca responder a la pregunta “¿De qué manera el sistema de información hará lo necesario para resolver el problema?”. [3]

En esta fase se pretende determinar una solución a los requisitos del sistema. Para esto se parte del diseño lógico del sistema de información, realizado en la fase anterior, para transformarlo en un modelo donde se define una estructura general de la solución, identificando grandes módulos (funciones que tendrá el sistema) y como es que se comunicaran entre si. A este proceso se le llama diseño de alto nivel. [4]

Parte del diseño lógico del sistema de información es diseñar la interfaz, con la cual el usuario tendrá contacto con el sistema, por lo cual es muy importante el uso de técnicas para el buen uso de pantallas y formas.

Posteriormente, realizado el diseño de alto nivel, es necesario establecer los algoritmos y como es que estará organizado el código para proceder con la implementación. Este proceso es llamado diseño detallado.

En el diseño del sistema se deben tomar en cuenta cosas muy importantes como el diseño los archivos y bases de datos que guardarán la mayor parte de la información que el usuario manipulará por medio del sistema. Una base de datos bien organizada es la base para cualquier sistema de información. [1]

Otro punto a considerar es el diseño de procedimientos para el control y respaldo del sistema y de la información.

Al igual que los requisitos, el diseño debe documentarse, para que si es el caso detectar errores, sea posible corregirlos fácilmente.

Generación de Código

En la fase de generación de código, se lleva a cabo la tarea de traducir el diseño realizado a una forma legible por la computadora, por medio de un lenguaje de programación. Dado que en la fase de diseño se sugiere realizar el diseño detallado este proceso se realiza mecánicamente. [2]

Es importante recalcar que las fases anteriores pueden ser desarrolladas de manera independiente del lenguaje de programación seleccionado. En esta fase, como se dijo anteriormente, se elige un lenguaje de programación. Por lo que es importante que esta etapa sea llevada a cabo por uno o un grupo de programadores expertos, para que estos mismos sean capaces de codificar correctamente el modelo lógico realizado anteriormente. [4]

Además de la generación de código, es necesario desarrollar documentación efectiva acerca del uso del sistema, esta documentación ayuda al usuario a manipular el sistema de manera correcta y también le dice qué hacer si existen problemas con el sistema.

1.2. Desarrollo de Sistemas de Información

Implantación y Pruebas

Antes de que el sistema sea usado por primera vez por el usuario final, este debe ser probado, las primeras pruebas son realizadas por los programadores solos, posteriormente se realizan las pruebas entre programadores y analistas de sistemas. En general se buscan problemas con los tipos de datos introduciendo datos erróneos y eventualmente con datos reales. [1]

Posteriormente, terminadas las pruebas necesarias, el siguiente paso es la implantación del sistema de información, esto incluye el adiestramiento de los usuarios para que puedan manejar el sistema.

Mantenimiento

Después de que ha sido instalado el sistema se le debe dar mantenimiento, para modificarlo o actualizarlo. Esto se debe a que después de un cierto tiempo los sistemas requieren cambios en su estructura, por errores, por cambios en su entorno externo (cambio de *sistema operativo*), o porque el usuario cambia de requerimientos. [2]

En general el mantenimiento se realiza por dos motivos, el primero es para corregir errores, dado que en ocasiones existen errores dentro del sistema que no son descubiertos en las fases anteriores, si no hasta que después de un tiempo de uso el sistema presenta anomalías en el funcionamiento.

La otra razón para dar mantenimiento, es para mejorar las capacidades del sistema, dado que la organización tiende a cambiar necesidades o evoluciona para brindar un mejor ser-

vicio a sus clientes. La evolución de la organización repercute de manera considerable en nuevas tecnologías de manejo de información sin importar en que caso se de, en software o hardware, un sistema que ha sido desarrollado con tecnología antigua debe ser modificado para poder aprovechar las capacidades de una tecnología nueva.

El mantenimiento se puede dividir en tres diferentes, de acuerdo al tiempo de aplicación. El preventivo pretende descubrir posibles fallas antes de que afecten la funcionalidad del sistema, el correctivo elimina problemas descubiertos durante la operación del sistema y que no fueron detectados anteriormente, y el evolutivo pretender modificar la funcionalidad del sistema para que pueda ser adaptado a las necesidades del usuario o con las necesidades del software o hardware. [4]

1.2.2. Modelo de construcción de prototipos

Podemos definir como *prototipo* un modelo parcial del sistema que puede ejecutarse, puede ser parcial ya que no es necesario que el sistema cubra todos los requerimientos del usuario, solo aquellas partes que deban ser analizadas. Y debe ser ejecutable para que la validación del sistema pueda hacerse a partir de la experimentación del prototipo. [4]

El *Modelo de construcción de prototipos* (Fig. 1.2) consiste en la ejecución de un método iterativo en el proceso de desarrollo de sistemas. En cada una de las iteraciones se analizan las posibles soluciones del problema, se diseña la solución y posteriormente se pone en operación una parte del sistema para que pueda ser evaluada por los usuarios y se genere retroalimentación. [3]

El modelo de construcción de prototipos comienza con el análisis del sistema, se deben identificar los objetivos generales, así como los requisitos con los que debe cumplir el sis-

1.2. Desarrollo de Sistemas de Información

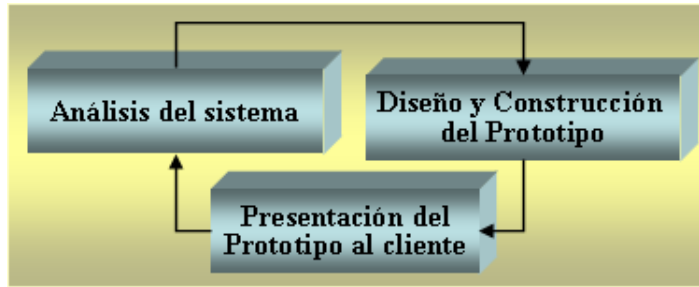


Figura 1.2: Modelo de construcción de prototipos

tema e identificar las necesidades del usuario. Terminada esta etapa se genera un modelo preliminar del sistema, o una parte a escala del sistema completo, este modelo se diseña de forma rápida para generar un prototipo. [2]

El prototipo presentará aspectos del sistema que serán visibles para el usuario, por ejemplo, pantallas de entrada y salida de datos, formatos de informes, etc. Posteriormente el prototipo es presentado al usuario para ser evaluado y refinar los requisitos del sistema a desarrollar. La iteración ocurre cuando el prototipo satisface las necesidades del usuario y permite que se comprenda de mejor manera lo que se debe hacer.

El prototipo presentado puede servir como base para el desarrollo de una segunda y tercera generación de sistemas, y así hasta que los requisitos son cumplidos completamente y se genera la versión final del sistema. De otra forma puede ser desechado y empezar de nuevo con los resultados obtenidos de la iteración con el usuario. [3]

Los prototipos pueden ser clasificados en operativos y no operativos, un prototipo operativo es aquel que permite una ejecución correcta de las funciones del sistema, es decir, permite la manipulación de la información contenida en las bases de datos o archivos. Por

el contrario un prototipo no operativo es aquel que no permite la manipulación de la información, solo presenta los formatos y especificaciones de los datos que serán introducidos al sistema. La ventaja de usar prototipos no operativos es que pueden ser desarrollados de manera más rápida que los operativos.

1.2.3. Modelo de desarrollo rápido de aplicaciones

El *Modelo de Desarrollo Rápido de Aplicaciones* (Rapid Application Development, RAD) es un modelo de desarrollo de sistemas lineal secuencial que se caracteriza por un ciclo de vida muy corto. El modelo RAD es una adaptación del modelo lineal secuencial que se ejecuta de manera rápida utilizando un enfoque de desarrollo basado en componentes. (Fig. 1.3) [2]

En el RAD se utilizan herramientas, *técnicas* y *métodos* diseñados para acelerar el proceso de desarrollo de sistemas, disminuye la creación de documentación, automatiza la generación de código y permite la participación del usuario en el proceso de desarrollo. Por lo que el RAD permite a un grupo de desarrollo crear sistemas completamente funcionales en períodos de tiempo cortos. [3]

El modelo RAD comprende las siguientes fases. [2]

Modelado de Gestión

Se realiza un *modelo* del flujo de la información entre las funciones de gestión, de tal manera que se pueda identificar las siguientes preguntas, ¿Qué información se genera?, ¿Quién la genera?, ¿A dónde va la información?, ¿Quién la procesa?

1.2. Desarrollo de Sistemas de Información

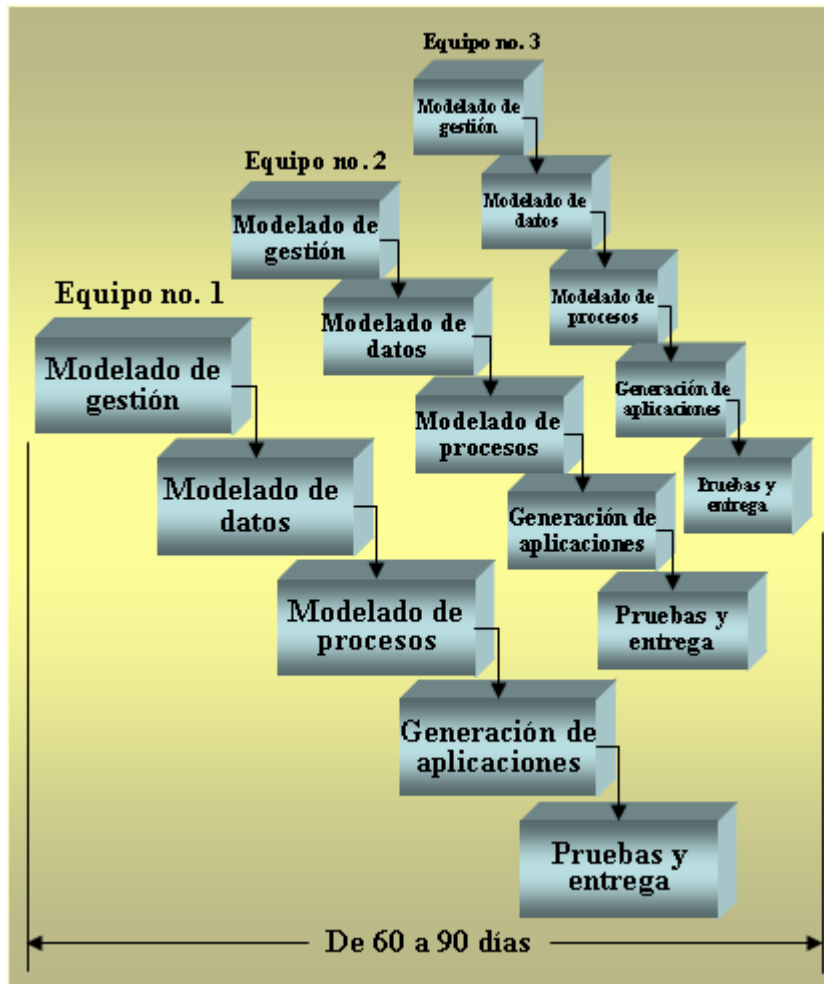


Figura 1.3: Modelo de desarrollo rápido de aplicaciones

Modelado de datos

Este modelado pretende responder a una serie de preguntas necesarias para cualquier sistema de procesamiento de la información. ¿Cuáles son los objetos de datos que va a procesar el sistema?, ¿Cuál es la composición de cada objeto de datos?, ¿Cuáles es la relación entre los objetos de datos y los procesos que lo transforman?

Modelado del proceso

En este modelo se describe la forma en que los objetos de datos se crean, modifican, suprimen y recuperan dentro del sistema. Para esto es necesario especificar de manera correcta el modelado de datos. En otras palabras, describe como es que son transformados los objetos de datos para lograr el flujo de la información, necesario para implementar el modelo de gestión.

Generación de Aplicaciones

El RAD permite el uso de herramientas de software que generan automáticamente el código fuente basándose en las especificaciones técnicas, a estas herramientas se les denomina *Técnicas de Cuarta Generación (T4G)*. Esto permite utilizar componentes de sistemas ya existentes o crear componentes reutilizables. En cualquier caso se utilizan herramientas automáticas para facilitar la construcción del sistema.

Pruebas y entrega

Dado que el RAD promueve la reutilización, muchos componentes del sistema ya han sido probados con anticipación, reduciendo tiempo de pruebas. Pero los componentes nuevos deben ser probados a fondo.

1.2.4. Modelo incremental

El *Modelo incremental* es una combinación de elementos del modelo lineal secuencial con el procedimiento iterativa de la construcción de prototipos. El modelo incremental (Fig. 1.4) aplica secuencias lineales en las cuales cada una produce un incremento del sistema. [2]

1.2. Desarrollo de Sistemas de Información

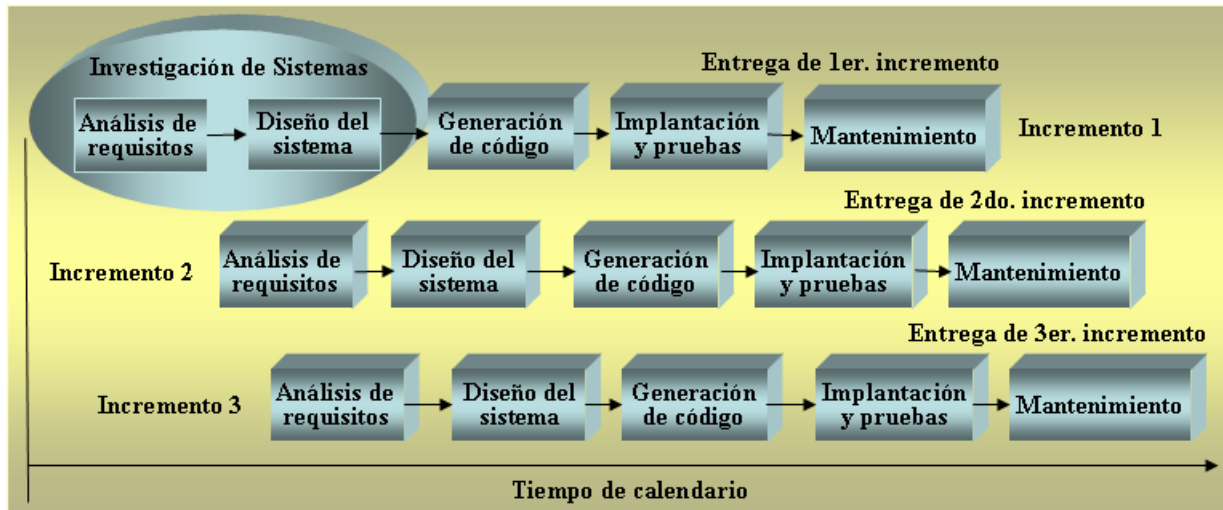


Figura 1.4: Modelo incremental

Al utilizar el modelo incremental, el primer incremento constituye el núcleo del sistema. Esto quiere decir que el sistema considerará los requisitos básicos, pero muchos de las funcionalidades del sistema no aparecerán. El usuario utiliza y evalúa el núcleo del sistema, posteriormente se desarrolla un plan para el incremento siguiente.

El plan del incremento siguiente especifica los cambios que deben hacerse al núcleo del sistema, con el fin de cumplir de una manera más eficaz las necesidades del usuario y de agregar las funciones y características adicionales. Este proceso se repite hasta que se obtiene un sistema completo que satisface completamente las necesidades del *usuario*.

El modelo incremental, como el modelo de construcción de prototipos, es iterativo, pero la diferencia entre estos modelos, es que el primero entrega un producto operacional en cada incremento. Este incremento permite que el usuario haga uso del sistema y evalúe de manera concreta los aspectos del sistema.

Una característica importante del modelo incremental es que es muy útil cuando el personal de desarrollo no es muy numeroso, los primeros incrementos pueden ser desarrollados por un número pequeño de desarrolladores. Posteriormente si el núcleo del sistema es aceptado es posible incrementar el número de desarrolladores para implementar el siguiente incremento.

1.2.5. Modelo en espiral

El *Modelo en espiral* fue propuesto hace algunos años por Boehm, es un modelo de desarrollo de sistemas evolutivo, por lo que combina la naturaleza iterativa de la construcción de prototipos con aspectos del modelo lineal secuencial, proporciona la característica de permitir el desarrollo de forma rápida de versiones incrementales. [2]

En las primeras iteraciones la versión incremental puede ser representada con un modelo en papel o un prototipo. Así las últimas iteraciones se producen versiones cada vez más completas.

Se divide en *actividades estructurales* o *regiones de tareas*. La Fig. 1.5 presenta el modelo en espiral que se divide en las *regiones de tareas* siguientes.

- **Comunicación con el cliente**, aquí se especifican las tareas a desarrollar para establecer comunicación entre el desarrollador y el usuario.
- **Planificación**, tareas en donde se definen los recursos, el tiempo y otros aspectos importantes para el proyecto.
- **Análisis de riesgos**, las tareas requeridas para evaluar riesgos técnicos y de gestión.

1.2. Desarrollo de Sistemas de Información

- **Ingeniería**, las tareas para construir una o más representaciones del sistema.
- **Construcción y adaptación**, las tareas requeridas para construir, probar, instalar y proporcionar soporte al usuario.
- **Evaluación del cliente**, las tareas requeridas para obtener la reacción del cliente según la evaluación de las representaciones del sistema.

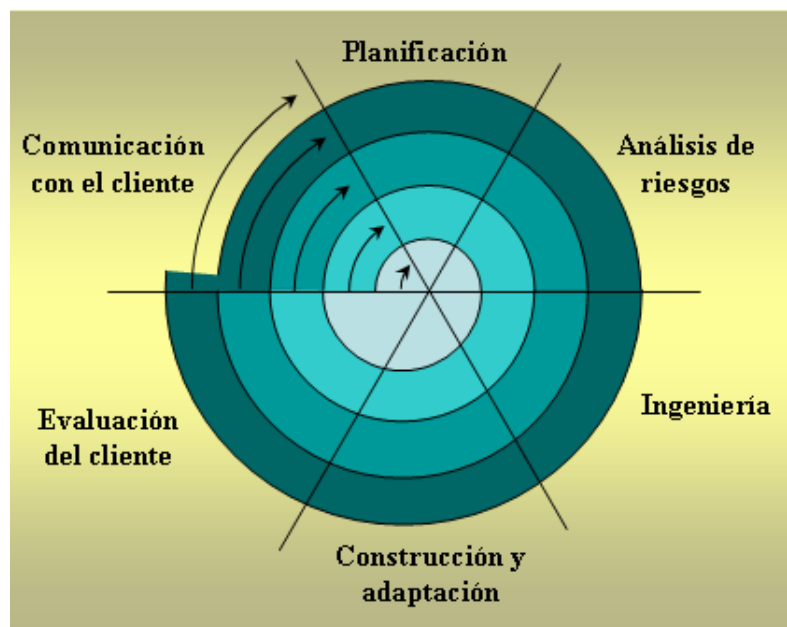


Figura 1.5: Modelo en Espiral

Cada región de tareas tiene una serie de tareas que permiten adaptarse a las características del sistema que se desarrolla. Cuando se desarrolla un sistema pequeño, el número de tareas y su formalidad es baja, sin embargo, en sistemas grandes se definen tareas para lograr un mayor nivel de formalidad.

Al empezar este modelo el desarrollador debe empezar en la dirección de las manecillas del reloj, comenzando en el centro de la espiral. El primer circuito genera una especificación

del sistema, los siguientes circuitos desarrollan un prototipo del sistema y posteriormente versiones mas sofisticadas del sistema.

1.2.6. Modelo de ensamblaje de componentes

En el *Modelo de ensamblaje de componentes*, se utilizan las tecnologías orientadas a objetos como proceso de desarrollo de sistemas, tomando como base los componentes. Estas tecnologías orientadas a objetos enfatizan en la creación de *clases* que encapsulan los datos y los algoritmos que se utilizan para manejarlos. Estas clases pueden ser implementadas o reutilizadas por diferentes aplicaciones y arquitectura de sistemas de información. [2]

El modelo de ensamblaje de componentes, Fig. 1.6, incorpora muchas de las características del modelo en espiral, es evolutivo y requiere de un enfoque iterativo para la creación de sistemas, con la diferencia de que este modelo desarrolla las aplicaciones a partir de componentes de software (clases) ya creados.

El modelo comienza con la actividad de identificación de clases candidatas, para ello se examinan los datos que serán utilizados por la aplicación y el algoritmo que se va a aplicar para manipular los datos. Y finalmente se encapsulan en una clase y son almacenadas en una biblioteca de clases o depósito, para posteriores proyectos de desarrollo.

Identificadas las clases, se examina la biblioteca de clases para determinar si las clases ya existen, en caso de que ya existan se extraen de la biblioteca y se vuelven a utilizar. De lo contrario se aplican los métodos orientados a objetos y se crea la clase.

1.2. Desarrollo de Sistemas de Información

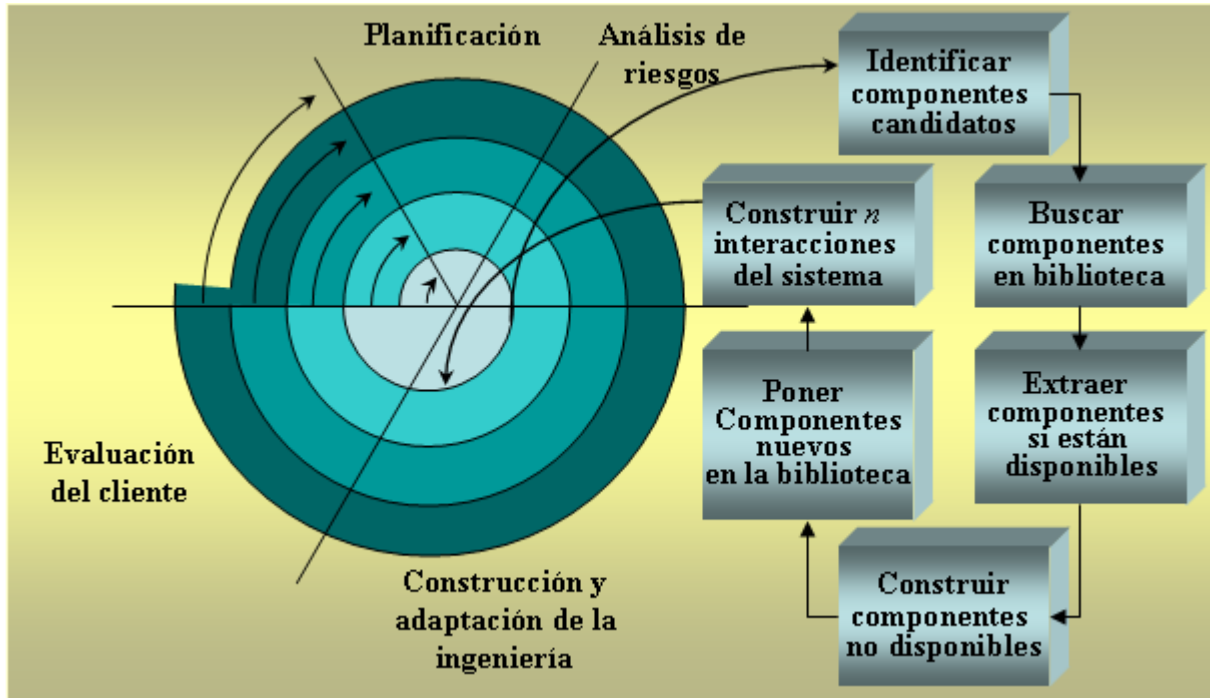


Figura 1.6: Modelo de ensamblaje de componentes

Posteriormente se compone la primera interacción de la aplicación a construirse, por medio de las clases extraídas de la biblioteca de clases y de las nuevas, para cubrir las necesidades del sistema.

Dada la reutilización de clases, el tiempo de desarrollo se reduce drásticamente así como también el costo de proyecto se reduce de manera sustancial.

1.2.7. Proceso Unificado de Racional (*Rational Unified Process, RUP*)

En la actualidad la cantidad y variedad de los procesos de desarrollo de sistemas han aumentado de forma muy rápida, dejando atrás el *paradigma* de modelo de SDLC lineal

secuencial.

Dando lugar a dos corrientes, los métodos de desarrollo de sistemas ligeros y los pesados. La diferencia entre estos, radica en que los métodos pesados intentan conseguir el objetivo común a través del seguimiento de sus fases y de la documentación, y los métodos ligeros tratan de mejorar la calidad de los sistemas a través de la comunicación directa con los clientes. [14]

El *Proceso Unificado de Rational* fue desarrollado por Philippe Kruchten, Ivar Jacobson y otros de la Rational como el proceso complementario al UML. [14]

El Proceso Unificado de Rational (Rational Unified Process), Fig. 1.7, es considerado como un proceso de desarrollo de sistemas pesado y es uno de los procesos de desarrollo más generales, por lo que permite adaptarse a cualquier proyecto, no solo a desarrollo de sistemas.

Para realizar cualquier proyecto se debe tener en cuenta cuatro fases del RUP:

1. Intercepción (puesta en marcha)
2. Elaboración (definición, análisis, diseño)
3. Construcción (implementación)
4. Transacción (fin del proyecto y puesta en producción)

En cada una de las fases del RUP se llevaran a cabo una o varias iteraciones y dentro de cada una se seguirá un modelo lineal secuencial o también llamado cascada (waterfal),

1.2. Desarrollo de Sistemas de Información

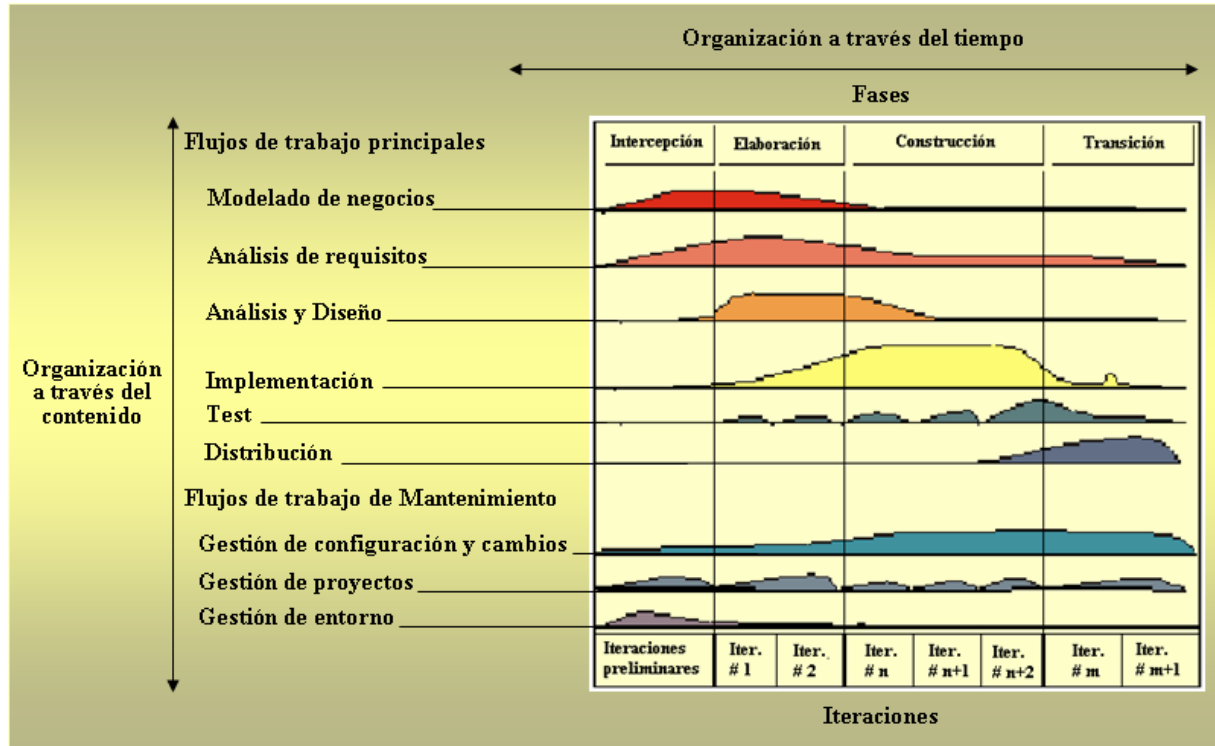


Figura 1.7: Modelo RUP

para los flujos de trabajo que requieren las actividades antes mencionadas. [14]

También se definen nueve actividades a realizar en cada fase del proyecto: [14]

1. Modelado de negocio
2. Análisis de requisitos
3. Análisis y diseño
4. Implementación
5. Test
6. Distribución

7. Gestión de configuración y cambios
8. Gestión de proyecto
9. Gestión de entorno

Entre las actividades antes mencionadas existe un flujo de trabajo, en base a los diagramas de actividad.

Además RUP permite asignar roles al personal involucrado en el proceso, estos roles definen las tareas y los resultados que se espera de ellos.

RUP se basa en UML como herramienta principal, para describir lo que se espera del sistema se utilizan los *casos de uso*, por lo que se crea la documentación necesaria para posteriores modificaciones o actualizaciones del proyecto. [14]

RUP esta pensado para proyectos y equipos grandes, en cuanto a tamaño y tiempo.

Para la obtención de requisitos se basa en UML, por lo que utiliza los casos de uso.

El proceso de desarrollo es controlado por diferentes elementos de planificación (plan de desarrollo, plan de iteración, plan de calidad, etc.)

Por medio de un esquema definido de escalabilidad y gestión de riesgos, se reconocen previamente problemas y fallos para prevenirlos o corregirlos.

RUP define en cada momento del ciclo de vida, que artefactos, con que detalle y por

1.2. Desarrollo de Sistemas de Información

que rol, deben ser creados. [14]

Con RUP al final de cada una de sus fases se presentarán al cliente los artefactos obtenidos y se valorarán las precondiciones para la siguiente fase, y solo se continuará con la siguiente fase hasta que el cliente apruebe los artefactos generados.

RUP genera *releases* basados en los artefactos después de cada fase, vienen acompañados de todos lo que traería el sistema final, (notas de la versión, instrucciones de instalación, ayuda, etc.) [14]

1.2.8. Programación Extrema (*eXtreme Programming, XP*)

Introducción

A principios del año 2001, en una reunión de diversos expertos del desarrollo de software surge el termino *ágil*, aplicado al desarrollo de sistemas con el fin de ofrecer una alternativa a los procesos de desarrollo de sistemas ya conocidos, que como ya se sabe se caracterizan por ser estrictos en la documentación de cada una de las actividades desarrolladas. [6]

Para esto se creó una organización que tuviera como objetivo promover los conceptos relacionados con el desarrollo ágil de software. Estos conceptos están plasmados en lo que es llamado *Manifiesto Ágil*, el cual resume la filosofía ágil. [6]

- **Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas.** El principal factor para el éxito del proyecto es la gente, por lo que es mejor construir un buen equipo de desarrollo que construir el entorno en

el cual se desenvuelvan.

- **Desarrollar software que funciona más que conseguir una buena documentación.** No se debe crear documentación alguna, a menos que sea muy necesaria. En tal caso deben ser cortos y específicos.
- **La colaboración con el cliente más que la negociación de un contrato.** Es necesario que el cliente tenga constante interacción con el equipo de desarrollo para asegurar el éxito del proyecto.
- **Responder a los cambios más que seguir estrictamente un plan.** La planificación debe ser flexible y abierta para poder responder a los cambios que puedan surgir en el proceso de desarrollo.

A continuación se presentan los principios plasmados también en el manifiesto. Estos doce principios están inspirados en los valores antes mencionados, por lo que son características que diferencian un proceso ágil de uno tradicional. [6]

- i. La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.*
- ii. Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.*
- iii. Entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.*
- iv. La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.*

1.2. Desarrollo de Sistemas de Información

- v. *Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.*
- vi. *El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.*
- vii. *El software que funciona es la medida principal de progreso.*
- viii. *Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.*
- ix. *La atención continua a la calidad técnica y al buen diseño mejora la agilidad.*
- x. *La simplicidad es esencial.*
- xi. *Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.*
- xiii. *En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.*

Definición

Programación Extrema (eXtreme Programming, XP) es una metodología ágil de desarrollo por lo que es implementada para proyectos cortos y equipos pequeños. XP propone potenciar las relaciones interpersonales como punto clave para el éxito en el desarrollo del sistema, promueve el trabajo en equipo y se preocupa por que los integrantes del equipo de trabajo aprendan y tengan un buen clima de trabajo. Se basa también en una constante retroalimentación entre el *cliente* y el *equipo de desarrollo*. [6]

XP intenta minimizar el riesgo de fallo del proceso por medio de la disposición permanente de un representante del cliente, este representante deberá ser capaz de contestar rápida y correctamente a cualquier cuestionamiento del equipo de desarrollo.

Los principios y prácticas son comunes pero llevados al extremo, de ahí el nombre, establecido por Kent Beck, el padre de *XP*. Fig. 1.8. [17]

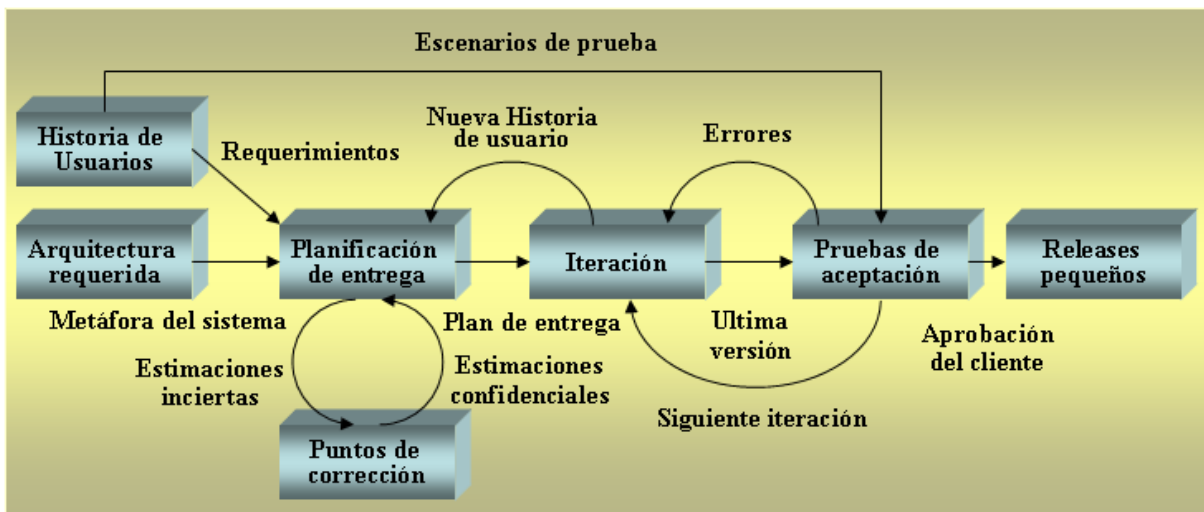


Figura 1.8: Modelo de Programación Extrema

Historia

XP nace como proceso de desarrollo de software de la mano de Kent Beck, cuando Beck forma parte de *Chrysler Corporation* en el verano de 1996, es entonces cuando se le encomienda la tarea de desarrollar un sistema de nominas. Anterior a la contratación de Beck, dicho sistema había sido desarrollado por empleados de la compañía pero con poco éxito debido a la carencia de una metodología eficaz para el desarrollo. [17]

Beck reconoció esta carencia y concluyó que era necesario cambiar la estructura en la

1.2. Desarrollo de Sistemas de Información

metodología y la manera en que los programadores debían trabajar. [11]

En 1999 en una entrevista en la revista *C++ Magazine*, comunicó sus ideas principales sobre una mejor metodología que enfatizaba en la comunicación dentro del equipo de desarrollo, la implementación y programación sencillas, y que el usuario debía estar involucrado estrechamente con el proceso y que las decisiones se debían tomar rápida y eficazmente.

Roles

El equipo de desarrollo debe estar compuesto por personas que asumen un papel muy importante para el desarrollo del sistema, este papel es denominado *rol* el cual especifica las tareas de cada integrante del grupo debe realizar. A continuación se describen los roles que son implementados en *XP*. [7]

Programador (*Programmer*)

Escribe las pruebas unitarias y el código del sistema. Debe haber una buena comunicación y coordinación entre los programadores y otros miembros del equipo.

Cliente (*Client*)

El cliente escribe las historias de usuario y las pruebas funcionales. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración. Solo debe haber un cliente dentro del proyecto pero puede representar a varias personas que se verán afectadas por el sistema.

Encargado de pruebas (*Tester*)

El encargado de pruebas ayuda al cliente a escribir las pruebas funcionales. En ocasiones

ejecuta las pruebas, da a conocer los resultados y avances en el equipo y es responsable de las herramientas para pruebas.

Encargado de seguimiento (*Tracker*)

Proporciona retroalimentación al equipo, tiene por responsabilidad verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, comunicando los resultados para mejorar futuras estimaciones. También evalúa si los objetivos son alcanzables con las restricciones de tiempo y recursos presentes. Determina cuándo es necesario realizar algún cambio para lograr los objetivos de cada iteración.

Entrenador (*Coach*)

Es el responsable del proceso global. Debe ser experto en el proceso *XP* para guiar a los miembros del equipo de forma que se apliquen correctamente las prácticas *XP*.

Consultor

Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto.

Gestor (*Big boss*)

Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

Las cuatro variables

XP define cuatro variables para cualquier proyecto software: *coste*, *tiempo*, *calidad* y *alcance*. De estas cuatro variables, sólo tres de ellas pueden establecerse por el cliente y

1.2. Desarrollo de Sistemas de Información

jefes de proyecto, mientras que el valor de la cuarta variable será establecido por el equipo de desarrollo en función de los valores de las otras tres. [8]

Normalmente los clientes y jefes de proyecto se creen capaces de fijar el valor de todas las variables, por lo que XP propone que se juegue con todas las partes implicadas en el proyecto hasta que el valor que alcancen las cuatro variables sea el correcto para todas las partes.

Ocurre además que las cuatro variables no guardan entre sí una relación tan obvia como a menudo se quiere ver.

El incremento del número de programadores no repercutirá de manera lineal en el *tiempo* de desarrollo del proyecto, siendo de todos conocido el dicho: "*nueve mujeres no pueden tener un hijo en un mes*". XP hace especial énfasis en equipos de desarrollo pequeños, que se podrán ir incrementando a medida que sea necesario, pero no antes, o los resultados serán generalmente contrarios a lo esperado. [13]

En cambio es bueno, incrementar el *coste* del proyecto en aspectos como equipo de cómputo más rápido, más especialistas técnicos en determinadas áreas o mejores oficinas para el equipo de desarrollo.

Con la *calidad* suele suceder un fenómeno extraño: frecuentemente un proyecto al cual se trate de aumentar la calidad conduce a que el proyecto pueda realizarse en menos tiempo.

Es verdad que cuando un equipo de desarrollo se acostumbra a realizar pruebas intensivas,

se siguen estándares de codificación, poco a poco se comenzará a avanzar más rápido de lo que lo hacía antes, mientras la calidad del proyecto se mantiene asegurada, lo que conlleva mayor confianza en el código y, por tanto, mayor facilidad para adaptarse al cambio, sin estrés, lo que hace que se programe más rápido. [13]

En cuanto al *alcance* del proyecto, es una buena idea dejar que sea esta la variable libre, de manera que, una vez fijadas las otras tres, el equipo de desarrollo determinaría el alcance mediante:

- La estimación de las tareas a realizar para satisfacer los requisitos del cliente.
- La implementación de los requisitos más importantes primero, de manera que el proyecto tenga en cada instante tanta funcionalidad como sea posible.

Los cuatro valores

El grupo de desarrollo debe tener muy claro que en el desarrollo de un proyecto software los cambios van a aparecer, cambiarán los requisitos, las reglas de negocio, el personal, la tecnología, todo va a cambiar. Por tanto el problema no es el cambio en si, ya que este va a suceder sino la incapacidad de enfrentarnos a estos cambios.

Como en otra cualquier actividad humana necesitamos valores para desarrollar nuestro trabajo y conseguir los planteamientos iniciales. Estos cuatro valores son: [12] [11]

- Comunicación
- Sencillez
- Retroalimentación

1.2. Desarrollo de Sistemas de Información

- Valentía

Comunicación

Algunos problemas en los proyectos surgen cuando un integrante del equipo no dice algo en su momento. La mala comunicación no surge por casualidad y hay circunstancias que conducen a la ruptura de la comunicación. XP ayuda mediante sus prácticas a fomentar la comunicación.

Sencillez

Mantener todo tan simple como sea posible. En relación al proceso y codificación, XP propone el principio de hacer las cosas más simples y que funcionen correctamente. Es mejor haber algo simple hoy que algo complicado sin probabilidades de uso mañana.

Retroalimentación

Por medio de pruebas funcionales del sistema, este mantendrá informado del grado de fiabilidad que presenta, esta información realmente no tiene precio. Los clientes y las personas que escriben pruebas tienen una retroalimentación real de su sistema.

Coraje

Este valor existe en el contexto de los otros tres valores, cada uno de ellos se apoyan en los demás:

- Se requiere valor para comunicarse con los demás integrantes del equipo.
- Se requiere valor para mantener el sistema simple.

- Se requiere coraje para confiar en que la retroalimentación es mejor que tratar de adivinar todo con anticipación.
- Sin un sistema simple, una comunicación constante y una retroalimentación, es difícil mantenerse con coraje.

Proceso de desarrollo

De manera general el proceso de desarrollo consiste en: [6]

- El cliente define el valor de negocio a implementar. El cliente determinará la importancia de cada una de las funcionalidades que tendrá el sistema.
- El programador estima el esfuerzo necesario para su implementación. El *grupo de desarrollo* determinara el tiempo necesario para su desarrollo.
- El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo. El cliente deberá establecer el orden en el cual se llevaran a cabo las funcionalidades del sistema.
- El programador construye ese valor de negocio.
- Vuelve al paso 1.

En cada una de las iteraciones del proyecto el *grupo de desarrollo* y el cliente aprenden, además para garantizar la calidad del software nunca se debe presionar al grupo de programadores para realizar mas trabajo de lo establecido.

El ciclo de vida ideal de *XP* consta de seis fases listadas a continuación: [7]

1. Exploración

1.2. Desarrollo de Sistemas de Información

2. Planificación de la Entrega (*Release*)
3. Iteraciones
4. Producción
5. Mantenimiento
6. Muerte del Proyecto.

Fase I: Exploración

En la primera fase del desarrollo, el cliente debe definir las *historias de usuario*, estas *historias de usuario* son la manera con la cual se especifican los requisitos del software. Una *historia de usuario* es una tarjeta de papel en la que el cliente describe de manera breve las características que el sistema debe poseer, para que el *grupo de desarrollo* pueda implementarla en cierto tiempo.

La información contenida en la tarjeta puede variar según los requerimientos del sistema y del *grupo de desarrollo*, siendo los más esenciales los siguientes: *Número*, *Nombre*, *Puntos estimado*, *Iteración asignada*, *Riesgo*, *Prioridad*, *Programador responsable*, *Descripción*, *Observaciones*.

Estas *historias de usuario* deben ser desarrolladas en un plazo no mayor a tres semanas, para no superar el tiempo límite de una iteración. Para esto se pueden dividir o descomponerse en *tareas de programación*, las cuales son asignadas a cada uno de los programadores para ser implementadas durante la iteración.

Al mismo tiempo el *grupo de desarrollo* debe familiarizarse con las herramientas, tecnologías y prácticas que serán empleadas en el proyecto. Para esto el *grupo de desarrollo* debe crear un prototipo de la estructura que puede tener el sistema, para que sea evaluado posteriormente por el *cliente*, probando así las posibilidades de la tecnología y herramientas a utilizar.

Esta fase de Exploración tiene una duración mínima, tomando pocas semanas según la familiaridad del grupo de trabajo con la tecnología y herramientas.

Fase II: Planificación de la Entrega

Una vez escritas las *historias de usuario*, el cliente deberá establecer la prioridad de cada una de ellas, con esto el *grupo de desarrollo* hará la estimación del esfuerzo necesario para su implementación, dando como resultado un cronograma de trabajo donde se especifican las fechas de la primera entrega y de las posteriores. Este cronograma es analizado por ambas partes y llegando a un acuerdo es validado para ser seguido por el *grupo de desarrollo*.

La primera entrega debe entregarse en menos de tres meses.

Para realizar las estimaciones de esfuerzo asociado a la implementación de las historias, se establece como unidad de medida el punto, el cual equivale a una semana de programación, esto es cuarenta horas de trabajo, ocho por día.

Esta fase dura pocos días según el número de *historias de usuario* a estimar.

1.2. Desarrollo de Sistemas de Información

Fase III: Iteraciones

El proceso de desarrollo incluye varias iteraciones antes de ser entregado, por lo que debe realizarse un *Plan de entrega* en el cual se especifican el número de iteraciones necesarias para el proceso desarrollo. Estas iteraciones clasifican un número determinado de *historias de usuario* que tendrán que ser implementadas dentro de cierto tiempo. Una iteración no debe pasar de tres semanas.

Por lo general en la primera iteración se debe intentar diseñar la estructura del sistema, para que pueda servir como base y posteriormente ser pulida y perfeccionada durante el resto del proyecto. Para esto es recomendable desarrollar las *historias de usuario* que impliquen diseñar funcionalidades dentro de la arquitectura del sistema.

El trabajo de cada iteración es medido en tareas de programación, las cuales son asignadas a un programador responsable, pero desarrolladas por parejas de desarrolladores.

Fase IV: Producción

En esta fase se realizan pruebas adicionales y revisiones antes de que el sistema sea instalado al *cliente*, así mismo se debe determinar si es necesario incluir nuevas características al sistema actual, esto depende de los resultados de las pruebas y revisiones anteriores.

Si es necesario esto, las nuevas ideas o sugerencias deben ser documentadas para su implementación posterior.

Fase V: Mantenimiento

Una vez que la primera versión se encuentra en producción, se debe mantener el sistema en funcionamiento al mismo tiempo que se desarrollan nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura.

Fase VI: Muerte del Proyecto

Esta fase empieza cuando el *cliente* no tiene más historias para ser incluidas en el sistema. Con esto se asegura que el sistema satisface las necesidades del *cliente* en aspectos como rendimiento y confiabilidad del sistema.

Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura.

Esta fase también ocurre cuando el sistema no genera los beneficios esperados por el *cliente* o cuando no hay presupuesto para seguir con su implementación o para mantenerlo.

Prácticas XP

Como se ha demostrado, los largos ciclos de desarrollo de los métodos tradicionales son incapaces de adaptarse al cambio, tal vez lo que haya que hacer sea ciclos de desarrollo más cortos. Esta es una de las ideas centrales de XP. [9]

Gracias a las tecnologías disponibles que ayudan al desarrollo de software y a la aplicación de las prácticas de *XP*, es posible disminuir la curva exponencial del costo del cambio

1.2. Desarrollo de Sistemas de Información

que se presenta en los procesos de desarrollo de software tradicionales. A continuación se describen dichas prácticas. [5] [10]

Planificación

Se define a la planificación como un constante dialogo entre el cliente y el *grupo de desarrollo*, en la cual el *grupo de desarrollo* realiza la estimación necesaria para llevar a cabo las *historias de usuario*, mientras que el *cliente* decidirá el alcance, la prioridad y el tiempo de entrega. Posteriormente se ordenan las *historias de usuario* según la prioridad y el esfuerzo requerido y se establece el contenido de las iteraciones.

Entregas pequeñas (*Small releases*)

Lo ideal es desarrollar de forma rápida versiones del sistema que sean operativas, esto es que tengan algunas de las funcionalidades requeridas por el usuario, pero que funcionen correctamente. Una entrega no debe tardar más de tres meses.

Metáfora

El sistema es definido por una o varias metáforas definidas por el *cliente* y el *grupo de desarrollo*. Una metáfora se define como una historia que describe como debería funcionar el sistema, dicho de otra manera, formar un conjunto de nombres que sirvan como vocabulario para hablar sobre el dominio del problema, y así ayudar a la nomenclatura de clases.

Diseño simple

Se debe conseguir la funcionalidad deseada de la forma más sencilla posible, debe evitarse la complejidad y el código extra. Se deben mantener las cosas lo mas sencillas

posibles con el fin de que se implemente y funcione correctamente en cualquier momento del proyecto.

Pruebas (*Testing*)

Las pruebas unitarias dirigen la producción del código, estas son establecidas por el *cliente* antes de escribir el código y son ejecutadas constantemente cuando se realiza alguna modificación al sistema.

Refactorización (*Refactoring*)

Se refiere a la actividad constante de reestructuración del código con el objetivo de eliminar duplicación de código, mejorar la legibilidad, simplificarlo y hacerlo mas flexible a posibles cambios. Se debe mejorar la estructura interna del código sin afectar el comportamiento externo.

Programación en parejas (*Pair programming*)

La producción de todo el código debe realizarse en parejas de programadores, dos personas sentadas frente a un monitor y un teclado, un programador escribirá el código y el otro aportara ideas sobre el diseño. En cualquier momento esto puede cambiar siendo uno de los valores de la programación en parejas. Con esto se reducen y corrigen errores disminuyendo el tiempo de desarrollo.

Propiedad colectiva del código

Cualquier programador puede cambiar cualquier parte del código en cualquier momento. Esta práctica permite que todos los programadores contribuyan con nuevas ideas

1.2. Desarrollo de Sistemas de Información

en todos los segmentos del sistema. Con esto se evita que algún programador sea imprescindible para realizar cambios en alguna porción de código.

Integración continua

Al cabo de un día de programación el sistema será integrado completamente, para esto se debe tener una computadora en la cual una pareja de programadores integrará una clase que haya sido probada unitariamente. Se realizarán pruebas para que el nuevo código sea incorporado definitivamente. El sistema podrá ser integrado más de una vez al día.

40 horas por semana

Se debe trabajar un máximo de 40 horas por semana. No se trabajan horas extras en dos semanas seguidas. Si esto sucede, probablemente se está presentando un problema que debe ser corregido. En lugar de esto se debe realizar un cambio en la planificación para modificar el ámbito del proyecto o la fecha de entrega.

Cliente in-situ

El *cliente* tiene que estar presente y disponible todo el tiempo para el *grupo de desarrollo*. Éste es uno de los principales factores de éxito del proyecto *XP*. El *cliente* dirige constantemente el trabajo hacia lo que aportará mayor valor de negocio y los programadores pueden resolver de manera inmediata cualquier duda asociada. La comunicación oral es más efectiva que la escrita.

Estándares de programación

XP enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación (del equipo, de la organi-

zación u otros estándares reconocidos para los lenguajes de programación utilizados). Los estándares de programación mantienen el código legible para los miembros del equipo, facilitando los cambios.

El beneficio que aportan las prácticas se consigue con su aplicación conjunta y equilibrada puesto que se apoyan unas en otras.

La mayoría de las prácticas propuestas por *XP* no son novedosas, sino que en alguna forma ya habían sido propuestas en ingeniería del software e incluso demostrado su valor en la práctica, el mérito de *XP* es integrarlas de una forma efectiva y complementarlas con otras ideas desde la perspectiva del negocio, los valores humanos y el trabajo en equipo. [9]

1.2.9. Desarrollo Guiado por la Funcionalidad (*Feature Driven Development, FDD*)

El *Desarrollo Guiado por la Funcionalidad* (Feature Driven Development FDD) es un proceso cuya catalogación depende del gusto del desarrollador, pues lo consideran ligero la gente que esta mas apegada a los procesos pesados y pesado para los que gustan de los procesos ligeros. Fig. 1.9. [14]

Fue diseñado por Meter Coad, Erich Lefebvre y Jeff de Luca, y como se dijo anteriormente puede ser considerado como el intermedio entre RUP y XP.

FDD esta pensado para proyecto con poco tiempo de desarrollo, aproximadamente menos

1.2. Desarrollo de Sistemas de Información

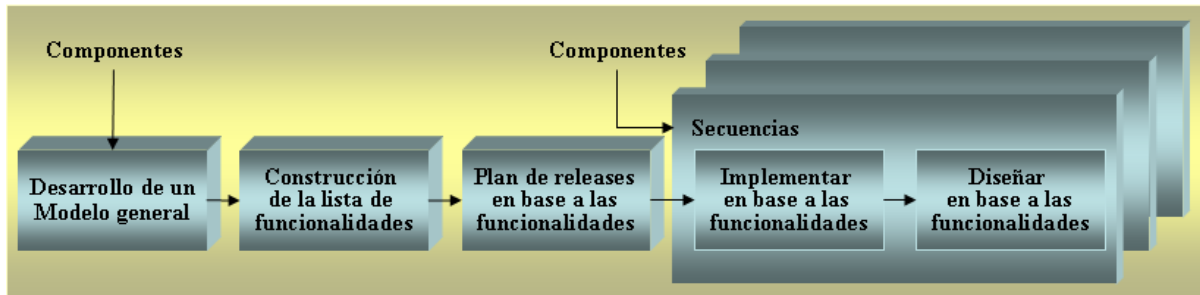


Figura 1.9: Modelo de Desarrollo Guiado por la Funcionalidad

de un año. Se basa en un proceso con iteraciones cortas de más o menos dos semanas que producen un sistema que el cliente y la dirección de la organización pueden ver y controlar.

[14]

Las iteraciones se basan en pequeñas partes del sistema llamadas *features* o funcionalidades, que tienen significado para el cliente.

Este proceso esta constituido por cinco fases: [15]

1. Desarrollo de un modelo general
2. Construcción de la lista de funcionalidades
3. Plan de *releases* en base a las funcionalidades a implementar
4. Diseñar en base a las funcionalidades
5. Implementar en base a las funcionalidades

Las primeras tres fases requieren de una gran parte del tiempo en las primeras iteraciones y posteriormente las dos ultimas requerirán mas tiempo según lo avanzado del proyecto.

Todo el trabajo se realiza en grupo, pero siempre existirá un jefe o responsable, esta persona deberá ser el que tenga mayor experiencia, el cual que tendrá la última decisión en caso de desacuerdos. [15]

Las *features* a implementar en cada *release* serán dividida entre cada uno de los sub-grupos del grupo de desarrollo, se escriben clases y cada una de ellas tiene uno o varios propietarios, por lo que solo ellos pueden cambiarlas o modificarlas según los requerimientos de cada *features*. Con todas las clases requeridas se procede a la implementación de las clases. [15]

También se debe considerar la preparación y ejecución de pruebas, las revisiones de código y la integración de las partes del sistema.

FDD define métricas para seguir de cerca el proceso de desarrollo del sistema, estas son útiles para el cliente porque le ayudan a conocer el estado actual del proceso, además de proporcionar una mejor estimación para proyectos futuros. [14]

IMPLEMENTACIÓN DE LA METODOLOGÍA

Resumen.

En este capítulo se define la metodología con la que se llevó a cabo el desarrollo del sistema, así como también se presenta a detalle las actividades desempeñadas por el grupo de trabajo.

Objetivo del Capítulo.

- Establecer la metodología de desarrollo.
- Llevar a cabo cada una de las fases de la metodología *Programación Extrema*.
- Detallar las actividades del grupo de desarrollo.
- De manera implícita proporcionar una explicación del funcionamiento del sistema.

2.1. Introducción

Para poder realizar un proceso eficaz de desarrollo de sistemas se requiere de una planificación total de las estrategias que el grupo de desarrolladores debe seguir, estas estrategias pueden estar definidas como procesos, métodos y actividades, que en conjunto describen la manera en que se desarrolla gradualmente el sistema de información. Estas estrategias pueden ser consideradas como modelos de proceso de desarrollo de sistemas, lo que en el capítulo anterior se denomina Ciclo de Vida del Desarrollo de Sistemas (SDLC, *Systems Development Life Cycle*).

Como ya se ha mencionado anteriormente, el presente trabajo pretende dar solución a la problemática del manejo y administración de la información, en la Dirección Técnica de la Secretaría de Desarrollo Social del Gobierno del Estado de Hidalgo, por medio de un sistema capaz de administrar de forma rápida y correcta la información de las propuestas de obra, que los ochenta y cuatro municipios del Estado de Hidalgo envían a la Secretaría de Planeación del Gobierno del Estado de Hidalgo, para su evaluación y posterior aprobación.

La Secretaría de Desarrollo Social del Gobierno del Estado de Hidalgo, cuenta con un área de trabajo encargada del desarrollo de sistemas para el manejo de la información. Esta área tiene el nombre de Subdirección de Informática, en este lugar es donde se realizará el sistema bajo la dirección del Ing. Juan Enrique Martínez Barrera y con la colaboración del P.L.S.C. Filiberto Pérez Barrera, así como también del C. Miguel de la Rosa Martínez.

2.2. Definición del material

Para la elaboración de este sistema, se ha decidido usar un lenguaje de programación simple y que brinde un soporte sólido a la base de datos necesaria para el almacenamiento de la información, es por esto que *Visual Basic* será la base de desarrollo de este sistema.

Así mismo el diseño de la base de datos debe ser fácil de entender y sobre todo fácil de manipular por lo que se ha elegido a *Microsoft Access* para su diseño.

Estas dos herramientas cumplen con las características necesarias para proveer al sistema de un diseño y comunicación sólidas, además de ser fáciles de entender y usar, por lo que no se requerirá demasiado tiempo en cursos de entrenamiento y programación.

En cuanto a recursos tecnológicos, la Subdirección de Informática cuenta con 3 computadoras para el desarrollo de este sistema. De las cuales 2 estarán disponibles totalmente para este fin, estas computadoras cuentan con características de procesamiento que ofrece un Pentium III de 700Mhz, 128MB en memoria RAM y 4GB en Disco Duro, con sistema operativo *Microsoft Windows 98*, *Microsoft Visual Basic 6.0* y *Microsoft Access 97*, aunque modestas pero suficientes para el desarrollo del sistema.

La tercera computadora esta dedicada para el uso exclusivo del Ing. Juan Enrique Martínez Barrera, quien dirigirá y evaluará los resultados obtenidos por el grupo de trabajo. Como es de esperarse esta computadora presenta mejores características de desempeño, con un procesador Pentium 4 a 1.7Ghz, 512MB de memoria RAM, con sistema operativo *Microsoft Windows XP*, *Microsoft Visual Basic 6.0* y *Microsoft Access 97*.

2.3. Proceso de desarrollo

Por lo tanto, en base al análisis, realizado en el capítulo anterior, de los modelos de proceso de desarrollo de sistemas, se llega a la conclusión de que el proceso de desarrollo de Programación Extrema (*eXtreme Programming*), *XP* cumple con todas las características necesarias para el desarrollo de dicho sistema.

Como se podrá observar en las siguientes secciones se describen las características más importantes de la implementación de *XP*, tomando como base [16] para la presentación del proceso de desarrollo.

2.3.1. Roles

Con la definición anterior podemos establecer los roles de cada uno de los participantes del equipo de desarrollo, tomando en cuenta que el sistema es relativamente pequeño no será necesario un equipo muy grande, por lo que tres personas desarrollando diversas tareas son suficientes.

- **Manager:** Juan Enrique Martínez Barrera

- **Tracker y Tester:** Filiberto Pérez Barrera y Miguel de la Rosa Martínez

- **Programmer:** Juan Enrique Martínez Barrera

- **Programmer:** Filiberto Pérez Barrera

2.4. Fase I Exploración

2.4.1. Interacción con el cliente

El primer paso para el desarrollo del sistema es la interacción con el *cliente*, por lo que según las características de la metodología *XP* son necesarias las *historias de usuario*, las *historias de usuario* son la base para el desarrollo de la primera iteración, estas dan un aspecto general de las operaciones que deberá realizar el sistema y que poco a poco, al transcurrir las iteraciones, serán pulidas y perfeccionadas para lograr un sistema robusto que sea capaz de satisfacer las necesidades del *cliente*.

En la primera entrevista con el *cliente*, en este caso el encargado de la recepción y captura de expedientes técnicos, el C. Miguel de la Rosa Martínez, se determinó que el sistema sería capaz de cumplir con los siguientes puntos:

- Crear y respaldar la Base de datos.
- Conectarse a la base de datos, para la manipulación de la información.
- Realizar operaciones como: agregar, eliminar y editar los registros de dicha base de datos.
- Presentar informes sobre la información almacenada en la base de datos.
- Mostrar gráficamente el manual de operación del sistema.
- Minimizar el riesgo de manipulación de la información por personal no autorizado.

2.4. Fase I Exploración

2.4.2. Historias de usuario

De los puntos tratados anteriormente, se desprenden las primeras *historias de usuario* que son enlistadas a continuación:

- Ingreso de nueva propuesta de obra
- Gestión de propuesta de obra
- Operaciones secundarias sobre propuestas
- Consulta de catálogos
- Control de acceso a usuarios
- Gestión de datos de usuario
- Administración de Base de Datos
- Reporte general de propuestas
- Reporte de propuesta
- Presentación de manual de operación
- Bloqueo de sistema

Cabe mencionar que los campos de estas *historias de usuario* pueden variar según la información deseada por el grupo de desarrollo, dependiendo del control que se quiera tener sobre el proceso.

2. IMPLEMENTACIÓN DE LA METODOLOGÍA

Historia de Usuario	
Número: 1	Usuario: Capturista
Nombre historia: Ingreso de nueva propuesta de obra	
Puntos estimados: 4	Iteración asignada: 1
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: El expediente técnico llega en formato impreso, donde se localiza la lista de propuestas. Se accede a la base de datos se almacena la información de acuerdo a sus campos en la tabla correspondiente.	
Observaciones:	

Tabla 2.1: Historia de Usuario 1. Ingreso de nueva propuesta de obra

Historia de Usuario	
Número: 2	Usuario: Capturista
Nombre historia: Gestión de propuesta de obra	
Puntos estimados: 3	Iteración asignada: 2
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: Un segundo expediente técnico llega en formato impreso, donde se localiza la corrección de propuestas. Se accede a la base de datos se localiza la propuestas y se edita, elimina o actualiza la información de la propuesta en la tabla correspondiente.	
Observaciones: Existe la posibilidad de que el primer expediente técnico contenga errores, por lo que se envía nuevamente un expediente para que se hagan las modificaciones correspondientes.	

Tabla 2.2: Historia de Usuario 2. Gestión de propuesta de obra de obra

Historia de Usuario	
Número: 3	Usuario: Encargado de departamento
Nombre historia: Operaciones secundarias sobre propuestas	
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: Se realizarán operaciones secundarias sobre las propuestas, esta operaciones pueden ser, búsqueda, ordenamiento y filtrado según criterio.	
Observaciones:	

Tabla 2.3: Historia de Usuario 3. Operaciones secundarias sobre propuestas

2.4. Fase I Exploración

Historia de Usuario	
Número: 4	Usuario: Capturista/Encargado de departamento
Nombre historia: Consulta de catálogos	
Puntos estimados: 3	Iteración asignada: 4
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: En ocasiones es necesario consultar las tablas secundarias de la base de datos para realizar modificaciones o actualizaciones.	
Observaciones:	

Tabla 2.4: Historia de Usuario 4. Consulta de catálogos

Historia de Usuario	
Número: 5	Usuario: Capturista/Encargado de departamento
Nombre historia: Control de acceso a usuarios	
Puntos estimados: 0.5	Iteración asignada: 5
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: Antes de iniciar el sistema, los usuarios deberán introducir una contraseña de ingreso para poder manipular la información y así evitar el uso no autorizado del sistema.	
Observaciones:	

Tabla 2.5: Historia de Usuario 5. Control de acceso a usuarios

Historia de Usuario	
Número: 6	Usuario: Capturista/Encargado de departamento
Nombre historia: Gestión de datos de usuarios	
Puntos estimados: 0.5	Iteración asignada: 5
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: Los usuarios del sistema deberán cambiar su contraseña de ingreso de forma periódica para evitar uso no autorizado del sistema.	
Observaciones:	

Tabla 2.6: Historia de Usuario 6. Gestión de datos de usuarios

2. IMPLEMENTACIÓN DE LA METODOLOGÍA

Historia de Usuario	
Número: 7	Usuario: Capturista/Encargado de departamento
Nombre historia: Administración de la Base de Datos	
Puntos estimados: 1	Iteración asignada: 5
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: Es necesario respaldar la base de datos para evitar la pérdida de información, también es necesario la creación de nuevas bases de datos para una mejor administración de la información.	
Observaciones:	

Tabla 2.7: Historia de Usuario 7. Administración de la Base de Datos

Historia de Usuario	
Número: 8	Usuario: Encargado de departamento
Nombre historia: Reporte general de propuestas	
Puntos estimados: 1.5	Iteración asignada: 5
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: Se presentarán en pantalla todas las propuestas de obra, clasificadas y listas para su impresión.	
Observaciones:	

Tabla 2.8: Historia de Usuario 8. Reporte general de propuestas

Historia de Usuario	
Número: 9	Usuario: Encargado de departamento
Nombre historia: Reporte de propuesta	
Puntos estimados: 1.5	Iteración asignada: 6
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: Se presentarán en pantalla la información correspondiente a una propuesta de obra, para su posterior impresión.	
Observaciones:	

Tabla 2.9: Historia de Usuario 9. Reporte de propuesta

2.4. Fase I Exploración

Historia de Usuario	
Número: 10	Usuario: : Capturista/Encargado de departamento
Nombre historia: Presentación de Manual de operación	
Puntos estimados: 1.5	Iteración asignada: 6
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: Se presenta un archivo en formato HTML con los puntos más importantes sobre la manipulación del sistema.	
Observaciones:	

Tabla 2.10: Historia de Usuario 10. Presentación de manual de operación

Historia de Usuario	
Número: 11	Usuario: : Capturista/Encargado de departamento
Nombre historia: Bloqueo de sistema	
Puntos estimados: 0.5	Iteración asignada: 6
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: Cuando sea necesario el sistema deberá ser bloqueado y solo con la contraseña correcta podrá ser puesto en marcha nuevamente.	
Observaciones:	

Tabla 2.11: Historia de Usuario 11. Bloqueo de sistema

2.5. Fase II Planificación de entrega

Realizada la valoración por parte del cliente de las *Historias de usuario* y junto con el análisis requerido para la estimación del esfuerzo, se presenta a continuación la tabla donde se especifican el número de iteraciones que se acordaron para el desarrollo del sistema.

No.	Nombre	Prioridad	Iteración	Esfuerzo
1	Ingreso de nueva propuesta de obra	Alta	1	4.0
2	Gestión de propuesta de obra	Alta	2	3.0
3	Operaciones secundarias sobre propuestas	Alta	3	2.0
4	Consulta de catálogos	Alta	4	3.0
5	Control de acceso usuarios	Alta	5	0.5
6	Gestión de datos de usuarios	Media	5	0.5
7	Administración de Base de Datos	Media	5	1.0
8	Reporte general de propuestas	Media	5	1.5
9	Reporte de propuesta	Media	6	1.5
10	Presentación de Manual de operación	Baja	6	1.5
11	Bloqueo de sistema	Media	6	0.5
			Total:	19.0

Tabla 2.12: Iteraciones del proyecto

Como se mencionó anteriormente, la prioridad de las historias de usuario fue establecida por el *cliente*, por lo que realizando una estimación de los requisitos necesarios para el cumplimiento y desarrollo de cada una de ellas, el grupo de trabajo establece el número de puntos necesarios para llevarlo a cabo.

Cabe aclarar que un punto es la unidad de medida para el esfuerzo necesario para la implementación de una historia de usuario, un punto es equivalente a una semana de ocho horas diarias, o sea cuarenta horas a la semana, otro de los puntos importantes en esta metodología.

2.5. Fase II Planificación de entrega

La estimación de esfuerzo y los costos necesarios para llevar a cabo el desarrollo del sistema se detallan en el Diagrama de planificación de desarrollo, ver apéndice A.

2.5.1. Plan de entrega

Definidos los puntos anteriores se han acordado junto con el cliente las fechas posibles de entrega de cada una de las iteraciones, a continuación se presenta la tabla donde se especifican las características del *Plan de entrega*.

No. de Iteración	No. de <i>historias de usuario</i>	Puntos	Fecha de entrega
1	1	4	27/02/04
2	1	3	19/03/04
3	1	2	02/04/04
4	1	3	23/04/04
5	4	3.5	19/05/04
6	3	3.5	14/06/04

Tabla 2.13: Planificación de entrega

En la primera iteración se desarrollarán las funcionalidades descritas en la primera historia de usuario. Se llevará a cabo el diseño de la base de datos, la interfaz principal del sistema, la ventana de captura de propuestas, funciones y opciones para poder introducir un expediente técnico a la base de datos.

La segunda iteración tendrá por objeto desarrollar todos los métodos y funciones para permitir al usuario realizar las operaciones primarias del sistema, como es editar, eliminar y actualizar los campos de las propuestas de obra.

2. IMPLEMENTACIÓN DE LA METODOLOGÍA

En la tercera iteración se pretende permitir al usuario realizar operaciones secundarias sobre la información de propuestas de obra, esto es, ordenar, filtrar y buscar.

En la cuarta iteración se llevarán a cabo todas las funcionalidades relacionadas con los catálogos, esto es, agregar, editar y eliminar registros de cada uno de ellos. Para ello se diseñaran ventanas de captura independientes entre si.

En la quinta iteración se desarrollarán los métodos para asegurar la manipulación autorizada del sistema por medio de nombres de usuario y contraseñas, que serán guardados en un archivo oculto para el cliente.

También se desarrollarán las opciones para permitir la administración de la base de datos, abrir una nueva base de datos, crear un a nueva base de datos y crear un respaldo de la base de datos actual.

Y por último en esta iteración se llevará a cabo el diseño e integración del formato de reporte general de propuestas, que permitirá al cliente visualizar la información en forma impresa.

En la iteración final se llevarán a cabo las funcionalidades para generar reportes individuales de cada una de las propuestas y la opción de bloqueo del sistema para evitar el uso no autorizado del sistema. También se desarrollará el manual de operación del sistema.

Y por último, terminadas las iteraciones y evaluadas por el cliente se ha llegado al acuerdo de que el día veintiocho de junio de dos mil cuatro, se entregará el Sistema para la admi-

2.6. Fase III Iteraciones

nistración de la información en la Dirección Técnica del Gobierno del Estado de Hidalgo, con un máximo de cinco días hábiles para su posible retraso.

2.6. Fase III Iteraciones

2.7. Primera Iteración

En la primera iteración se procede al desarrollo de la primera *historia de usuario*, para ello procede a dividirla en *tareas* mas simples.

Una *tarea* también puede ser representada como una tarjeta donde se especifican las acciones que se deben realizar para lograr desarrollar la *historia de usuario*.

La primera *historia de usuario* llamada *Ingreso de nueva propuesta de obra* se divide en las siguientes tareas.

- Diseño de la Base de Datos, para dar soporte a la información
- Diseño de la interfaz principal del sistema
- Diseño de la interfaz de Captura de Propuestas, necesaria para la manipulación de la Base de Datos
- Ingreso y lectura de datos, desde la interfaz de Captura de Propuestas

2. IMPLEMENTACIÓN DE LA METODOLOGÍA

Tarea	
Número de Tarea: 1	No. Historia: 1
Nombre Tarea: Diseño de la Base de Datos	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Fecha Inicio: 02/02/2004	Fecha Fin: 06/02/2004
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: Se realizará un análisis para determinar la estructura de las tablas de la base de datos, tomado en cuenta otras bases de datos existentes. Y se diseñara conforme los resultados obtenidos del análisis la nueva base de datos.	

Tabla 2.14: Tarea 1. Diseño de la Base de Datos

2.7.1. Tarea 1 Diseño de la Base de Datos

Realizado el análisis requerido para establecer la estructura de las tablas de la base de datos, en la Fig. 2.1 se muestra la estructura gráfica general de ellas.

Como se mencionó anteriormente, se tomaron en cuenta bases de datos existentes en la SubDirección para verificar la estructura que debería tener la nueva base de datos, con esto aseguramos que si en un futuro se desea integrar todos los sistemas existentes, puedan manipular una sola base de datos general sin necesidad de modificar nada.

Cada uno de los campos de cada tabla ha sido analizado de forma minuciosa para lograr una estructura bien definida y que nos permita el almacenamiento y la manipulación correcta de la información. A continuación se muestra a detalle cada uno de los campos de las tablas de la base de datos.

La primera tabla en orden alfabético es la denominada *FONDO*, en ella se encuentra almacenada la información concerniente a los diferentes fondos programáticos, estos al-

2.7. Primera Iteración

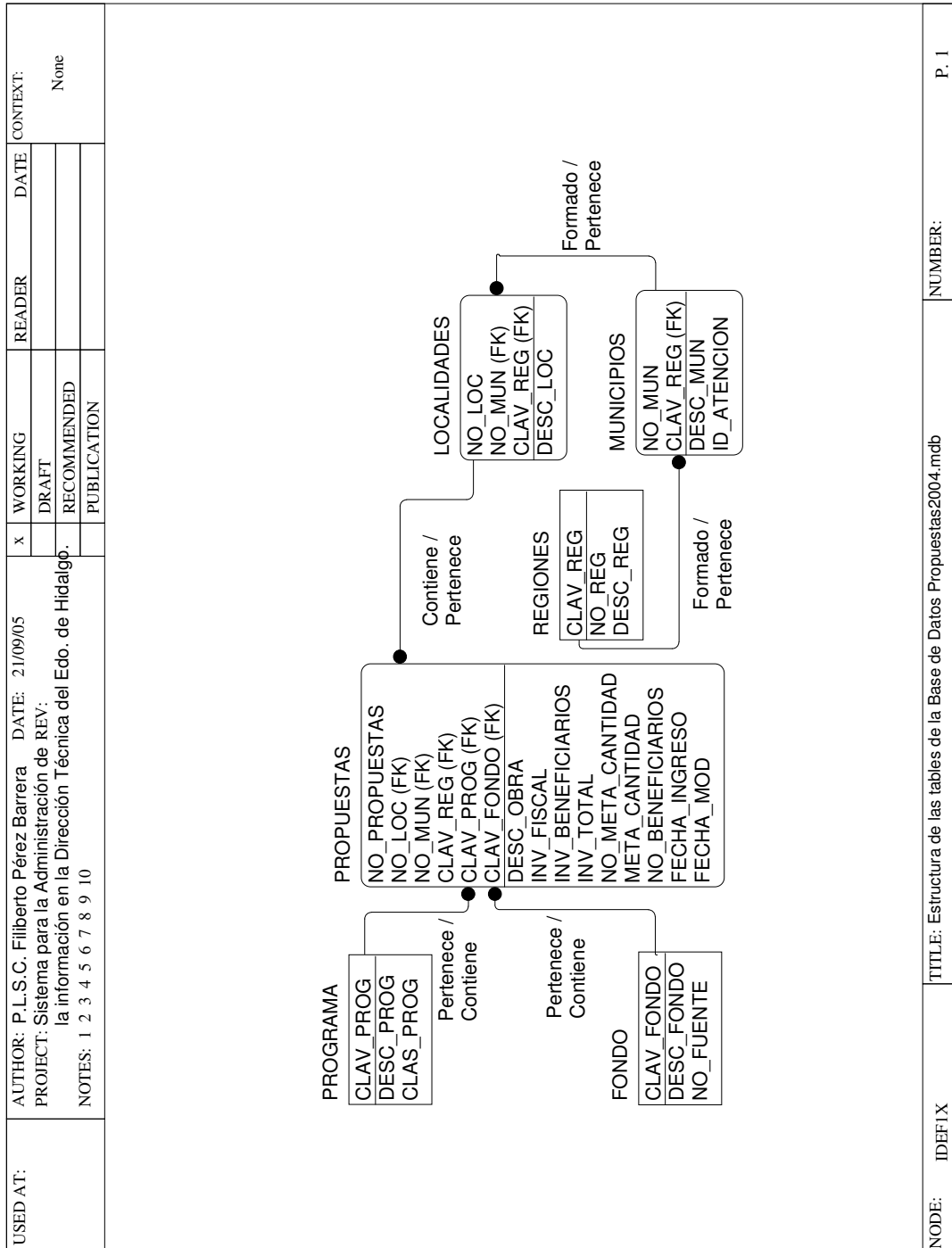


Figura 2.1: Estructura de las tablas de la Base de Datos.

2. IMPLEMENTACIÓN DE LA METODOLOGÍA

macenan los diferentes programas en los cuales se dividen los tipos de obras. Existen una gran cantidad de fondos programáticos, pero para este sistema solo se usaran tres que son: *Fondo para el Fortalecimiento Municipal, FAFM; Fondo de Infraestructura Social Municipal, FAISM y el Fondo Único De Participaciones, FUP.*

Tabla	FONDO		
Nombre	Tipo de campo	Tipo de datos	Tamaño
CLAV_FONDO	Llave Primaria	Texto	10
DESC_FONDO	Normal	Texto	70
NO_FUENTE	Normal	Texto	2

Tabla 2.15: Estructura de la tabla *FONDO*

La siguiente tabla se denomina *LOCALIDADES*, en ella se encuentran todas y cada una de las localidades de todo el estado de Hidalgo, es necesaria para determinar la ubicación de la propuesta que se ha de recibir.

Tabla	LOCALIDADES		
Nombre	Tipo de campo	Tipo de datos	Tamaño
NO_LOC	Llave Primaria	Texto	5
NO_MUN	Llave Foránea	Texto	5
DESC_LOC	Normal	Texto	45

Tabla 2.16: Estructura de la tabla *LOCALIDADES*

La tabla *MUNICIPIOS*, es la encargada de almacenar todos y cada uno de los 84 municipios que componen al estado de Hidalgo, el campo *ID_ATENCION* de esta tabla fue tomado de otras bases de datos con el fin de asegurar la compatibilidad de sistemas a futuro.

2.7. Primera Iteración

Tabla	MUNICIPIOS		
Nombre	Tipo de campo	Tipo de datos	Tamaño
NO_MUN	Llave Primaria	Texto	5
CLAV_REG	Llave Foránea	Texto	2
DESC_MUN	Normal	Texto	28
ID_ATENCION	Normal	Número	Entero Largo

Tabla 2.17: Estructura de la tabla *MUNICIPIOS*

La tabla siguiente se denomina *PROGRAMA*, como se mencionó anteriormente, un programa es la clasificación que se le da a un propuesta de obra, existen ciento treinta programas, que se clasifican en veintidós apartados generales dentro de los tres fondos programáticos.

Tabla	PROGRAMA		
Nombre	Tipo de campo	Tipo de datos	Tamaño
CLAV_PROG	Llave Primaria	Texto	3
DESC_PROG	Normal	Texto	50
CLAS_PROG	Normal	Texto	50

Tabla 2.18: Estructura de la tabla *PROGRAMA*

La tabla siguiente es la principal dentro del sistema *PROPUESTAS*, porque en ella se almacenara toda la información que es enviada por los 84 municipios a la Secretaría para que sea evaluada y posteriormente aprobada para su ejecución municipal. En general todas los expedientes técnicos presentan una lista de propuestas de obras con los campos que se presentan a continuación, en caso de que no se cumpla con todos los campos es necesario introducir algún o algunos símbolos que represente la ausencia de estos datos.

La última tabla de la base de datos se denomina *REGIONES*, el estado de Hidalgo está dividido en diecisiete regiones que agrupan a todos los municipios, esto permite un

2. IMPLEMENTACIÓN DE LA METODOLOGÍA

Tabla	PROPUESTAS		
Nombre	Tipo de campo	Tipo de datos	Tamaño
NO_PROPUESTA	Llave Primaria	Autonumérico	Entero largo
NO_MUN	Llave foránea	Texto	5
CLAV_REG	Llave foránea	Texto	2
CLAV_FONDO	Llave foránea	Texto	10
CLAV_PROG	Llave foránea	Texto	3
NO_LOC	Llave foránea	Texto	5
DESC_OBRA	Normal	Texto	50
INV_FISCAL	Normal	Número	Doble 2 decimales
INV_BENEFICIARIOS	Normal	Número	Doble 2 decimales
INV_TOTAL	Normal	Número	Doble 2 decimales
NO_META_UNIDAD	Normal	Texto	3
META_CANTIDAD	Normal	Texto	5
NO_BENEFICIARIOS	Normal	Número	Doble
FECHA_INGRESO	Normal	Texto	10
FECHA_MOD	Normal	Texto	10

Tabla 2.19: Estructura de la tabla *PROPUESTAS*

mejor control tanto político como administrativo de los municipios del estado.

Tabla	REGIONES		
Nombre	Tipo de campo	Tipo de datos	Tamaño
CLAV_REG	Llave Primaria	Texto	2
NO_REG	Normal	Texto	7
DESC_REG	Normal	Texto	30

Tabla 2.20: Estructura de la tabla *REGIONES*

El nombre de la nueva base de datos esta compuesto por dos partes, la primera es la palabra que identifica el contenido de la base de datos, esta palabra es Propuestas y la segunda parte es el año en el cual han sido enviadas las propuestas a la Secretaría, por lo que la primera base de datos se denomina *Propuestas2004*.

2.7. Primera Iteración

2.7.2. Tarea 3 Diseño de la interfaz de principal del sistema

Tarea	
Número de Tarea: 2	No. Historia: 1
Nombre Tarea: Diseño de la Interfaz de principal del sistema	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Fecha Inicio: 09/02/2004	Fecha Fin: 13/02/2004
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: Se diseñara una ventana en donde se enlisten todas las propuestas almacenadas en la base de datos, con ayuda de los diferentes controles disponibles en el lenguaje de programación. También se incluirán botones para la gestión de las propuestas .	
Obsevaciones: Los botones mencionados no tendrán funcionalidad hasta la siguiente historia de usuario.	

Tabla 2.21: Tarea 2. Diseño de la interfaz de principal del sistema

Para que el cliente tenga una buena interacción con el sistema es necesario proporcionarle una interfaz agradable, fácil de usar y muy práctica. Para esto se ha decidido tomar como base sistemas desarrollados en la Subsecretaría para asegurarnos de que el cliente conozca la forma de operar del sistema.

El formato de la interfaz principal se presenta en la Fig. 2.2

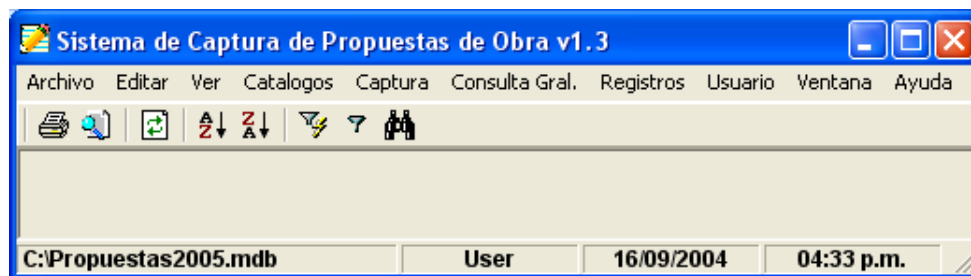


Figura 2.2: Interfaz principal del sistema.

2. IMPLEMENTACIÓN DE LA METODOLOGÍA

En la Fig. 2.2 se puede observar la interfaz principal del sistema. Esta compuesta por un menú principal, una barra de herramientas, el área de trabajo y por una barra de estado.

El menú *Archivo* permitirá al usuario realizar operaciones relacionadas con la Base de Datos, así como opciones de impresión de reportes, y por último opciones para bloquear y salir del sistema.

El menú *Editar* permitirá al usuario realizar la operación de búsqueda dentro de las ventanas de captura.

En caso de que el usuario prefiera no visualizar las barras de herramientas y estado, el menú *Ver* le permitirá ocultarlas.

En el menú *Catálogos* el usuario podrá visualizar el catálogo necesario para su análisis.

El menú *Captura* es el más importante, porque es ahí donde se encontrará la opción de captura de propuestas, que es el objetivo del sistema.

El menú *Consulta Gral.* permitirá al usuario ingresar a la opción para generar reporte sobre la información almacenada en la base de datos.

El menú *Registros* contendrá las opciones para filtrar y ordenar registros según la ventana de captura activa.

2.7. Primera Iteración

El menú *Usuario* permitirá la operación de cambio de contraseña de usuario.

Para una mejor administración de las ventanas el menú *Ventana* permitirá al usuario visualizar las ventanas de diferentes formas.

El menú *Ayuda* presentará las opciones para la visualización de los datos generales del sistema así como del manual de operación.

La barra de herramientas representa los accesos rápidos a las opciones de los menús, cabe aclarar que se han colocados los de mayor uso.

La barra de estado presentará al usuario información sobre la ubicación de la base de datos actual, el nombre usuario actual y por último la hora y fecha actual.

2.7.3. Tarea 3 Diseño de la interfaz de captura de propuestas

Tarea	
Número de Tarea: 3	No. Historia: 1
Nombre Tarea: Diseño de la Interfaz de Captura de Propuestas	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Fecha Inicio: 16/02/2004	Fecha Fin: 20/02/2004
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: Se diseñara una ventana en donde se enlisten todas las propuestas almacenadas en la base de datos, con ayuda de los diferentes controles disponibles en el lenguaje de programación. También se incluirán botones para la gestión de las propuestas.	
Obsevaciones: Los botones mencionados no tendrán funcionalidad hasta la siguiente historia de usuario.	

Tabla 2.22: Tarea 3. Diseño de la interfaz de captura de propuestas

2. IMPLEMENTACIÓN DE LA METODOLOGÍA

Se ha realizado el diseño de la interfaz de captura de propuestas junto el *cliente* y se ha obtenido la primera ventana que permitirá la manipulación de la información. Esta ventana se muestra en la Fig. 2.3.

The screenshot shows a software window titled "Propuestas de Obra". The window has a blue title bar with standard Windows window controls (minimize, maximize, close). The main content area is light beige and contains the following elements:

- A title bar with the text "Propuestas de Obra".
- A text input field labeled "No. de Propuesta".
- Two dropdown menus labeled "Región" and "Municipio".
- Two more dropdown menus labeled "Localidad" and "Número de Beneficiarios".
- A text input field labeled "Descripción".
- A dropdown menu labeled "Fondo".
- A dropdown menu labeled "Programa".
- Three columns of input fields:
 - Inversión:** "Fiscal", "Beneficiarios", "Total".
 - Metas:** "Cantidad", "Unidades".
 - Fecha:** "Ingreso", "Modificación".
- A navigation bar at the bottom with left and right arrows.
- A toolbar at the bottom with icons for document, edit, trash, checkmark, printer, and a red hand icon.

Figura 2.3: Interfaz de captura de *Propuestas*.

Como se puede observar en la figura, los diferentes controles nos permiten visualizar los campos de la tabla llamada *Propuestas*, todos ellos tienen por nombre los datos que deben ser proporcionados para ser almacenados en la base de datos.

El primer campo denominado *No. de Propuesta* permite indicar al usuario el número de propuesta que tiene asignado ese conjunto de datos, este no puede ser manipulado por

2.7. Primera Iteración

el usuario ya que es actualizado automáticamente por el sistema.

El campo de *Región* nos permitirá indicar la región en la cual se llevará a cabo la propuesta de obra, la lista desplegable permitirá hacer la selección de la opción necesaria. Un punto importante a mencionar es que debido a que después de haber seleccionado una región, no será necesario cargar todos los municipios del estado, ya que solo es posible la selección de un municipio que se encuentra en la región seleccionada.

Es por eso que el campo *Municipio* solo cargará los municipios de la región seleccionada anteriormente, así solo bastará que el usuario elija el municipio deseado y se prosiga con la selección de la localidad.

Así el campo de *Localidad* tiene un funcionamiento similar a las listas desplegables anteriores.

El campo *No. de Beneficiarios* y *Descripción* permiten la captura de datos con las restricciones propias de los campos en la base de datos, por ejemplo el primer campo solo acepta números enteros y el segundo caracteres alfanuméricos.

Las listas desplegables de *Fondo* y *Programa* permiten elegir las opciones correctas según la propuesta de obra.

El conjunto de datos de *Inversión* solo aceptan caracteres numéricos por lo que al llenar los dos primeros automáticamente se actualiza el tercero con la suma de los dos anteriores. Mientras que los datos de *Metas* permiten caracteres alfanuméricos.

2. IMPLEMENTACIÓN DE LA METODOLOGÍA

El conjunto de datos de *Fecha* permiten un máximo de diez caracteres donde debe ser colocada la fecha según sea el caso, el formato es *dd/mm/aaaa*.

El control de desplazamiento sobre los campos de la base de datos permite la visualización de las propuestas de obra que se encuentran en la tabla de *Propuestas*.

Y por último los botones permiten la manipulación de los datos de cada propuesta, ya sea para agregar, editar, eliminar, actualizar, cancelar o guardar algún registro.

2.7.4. Tarea 4 Ingreso y lectura de datos

Tarea	
Número de Tarea: 4	No. Historia: 1
Nombre Tarea: Ingreso y lectura de datos	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Fecha Inicio: 23/02/2004	Fecha Fin: 27/02/2004
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: Se realizará el proceso de ingreso de datos por medio de los controles, además se realizará la lectura de la base de datos, poniendo en ejecución la ventana. Los datos deberán ser visualizados por los controles correspondientes y se podrá realizar un recorrido por todos los registros de la tabla de Propuestas.	

Tabla 2.23: Tarea 4. Ingreso y lectura de datos

Como es obvio todos los campos de la ventana de Captura de Propuestas están ligados a los campos de la tabla de Propuestas de la base de datos, por lo que al ejecutar la ventana se pueden visualizar las propuestas que se encuentran en la tabla.

En la Fig. 2.4 se puede observar que para introducir una nueva propuesta de obra será nece-

2.7. Primera Iteración

sario presionar *Nuevo*, el primer botón de izquierda a derecha, esto ocasionará que, en caso de que exista información en la tabla, se borre todos los datos de los controles de la ventana y se permita el ingreso de datos nuevos para ser guardados posteriormente.

Para guardar esta información será necesario presionar *Guardar* el quinto botón. Esto permite que al presionar este botón los datos queden almacenados en la tabla correspondiente.

The screenshot shows a software window titled "Propuestas de Obra" with a blue title bar. The window contains a form with the following fields and values:

- No. de Propuesta:** 1
- Región:** VIIIA - ZACUALTIPAN
- Municipio:** 13068 - TIANGUISTENGO
- Localidad:** 68055 - ACATENCO
- Número de Beneficiarios:** 22222
- Descripción:** ELECTRIFICACION DE CALLES
- Fondo:** FAFM - FONDO PARA EL FORTALECIMIENTO MUNICIPAL
- Programa:** S11 - MEJORAMIENTO URBANO

Below these fields are three sections:

- Inversión:** Fiscal: 22222, Beneficiarios: 22222, Total: 44444
- Metas:** Cantidad: 22222, Unidades: M2
- Fecha:** Ingreso: 01/01/2004, Modificación: 01/01/2004

At the bottom of the window, there is a navigation bar with a list box showing "1 de 3 registros..." and several icons: a document with a pencil (Agregar), a trash can, a document with a checkmark, a printer, and a stop sign. Below the icons are two buttons labeled "Agregar" and "Guardar".

Figura 2.4: Ingreso y lectura de datos.

Para visualizar los datos almacenados en la tabla es necesario presionar las flechas del control de desplazamiento. La primera flecha de izquierda a derecha nos permite posi-

2. IMPLEMENTACIÓN DE LA METODOLOGÍA

cionarnos en el primer registro de la tabla, la segunda flecha permite desplazarnos de un registro a otro hacia atrás, la parte intermedia desplegara la posición actual en la que nos encontramos así como el total de registros en la tabla, y por último las flechas siguientes nos permitirán desplazarnos hacia delante, de registro a registro y hasta el último respectivamente. Fig. 2.5

The screenshot shows a software window titled "Propuestas de Obra" with a blue title bar. The main content area is a form with the following fields and values:

- No. de Propuesta:** 2
- Región:** IA - ATOTONILCO EL GRANDE
- Municipio:** 13001 - ACATLAN
- Localidad:** 01001 - ACATLAN
- Número de Beneficiarios:** 11111
- Descripción:** PAVIMENTACION DE CALLES
- Fondo:** FAISM - FONDO DE INFRAESTRUCTURA SOCIAL MUNICIPAL
- Programa:** BN - PLANEACION DE LA POLITICA DE TRANSPORTE

Below these fields are three summary boxes:

- Inversión:** Fiscal: 11111, Beneficiarios: 11111, Total: 22222
- Metas:** Cantidad: 11111, Unidades: ML
- Fecha:** Ingreso: 01/01/2004, Modificación: 01/01/2004

At the bottom, there is a navigation bar with a list box showing "2 de 3 registros..." and several icons for navigation and actions.

Figura 2.5: Movimiento entre registros de la tabla *PROPUESTAS*.

2.8. Segunda Iteración

En la segunda iteración se procede al desarrollo de la segunda historia de usuario, llamada *Gestión de propuesta de obra*, se divide en las siguientes tareas. El código de esta iteración puede consultarse en el Apéndice B.

- Agregar propuesta de obra
- Editar propuesta de obra
- Eliminar propuesta de obra
- Actualizar propuesta de obras

2.8.1. Tarea 5 Agregar propuesta de obra

Tarea	
Número de Tarea: 5	No. Historia: 2
Nombre Tarea: Agregar propuesta de obra	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Fecha Inicio: 01/03/2004	Fecha Fin: 05/03/2004
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: Se realizarán los métodos para permitir el proceso de ingreso de datos por medio de los controles, tomando en cuenta aspectos estéticos y funcionales de la ventana de ingreso.	

Tabla 2.24: Tarea 5. Agregar propuesta de obra

Como se mencionó anteriormente para el ingreso de nuevos datos en la tabla de *Propuestas*, la ventana de ingreso debe comportarse de manera diferente en comparación a los resultados obtenidos en la primera *historia de usuario*.

2. IMPLEMENTACIÓN DE LA METODOLOGÍA

El primer punto a considerar es que al principio del proceso de ingreso, y ya seleccionada la base de datos correcta, la ventana de ingreso debe permitir la visualización de los registros en la tabla, pero no debe permitir su edición hasta que sea presionado el botón de *Agregar* nueva propuesta de obra.

Tomando en cuenta que solo se puede guardar un registro si y solo si este, es nuevo o modificado, y que para cancelar la operación de guardado, anteriormente se debió haber sido presionado el botón de *Agregar*, los botones *Guardar* y *Cancelar* deberán estar deshabilitados hasta que se presione el botón de *Agregar*, esto se puede observar en la Fig. 2.6.



The screenshot shows a Windows-style application window titled "Propuestas de Obra". The window contains a form with the following fields and controls:

- No. de Propuesta:** Text input field containing the value "1".
- Región:** Dropdown menu with "VIII A - ZACUALTIPAN" selected.
- Municipio:** Dropdown menu with "13068 - TIANGUISTENGO" selected.
- Localidad:** Dropdown menu with "68055 - ACATENCO" selected.
- Número de Beneficiarios:** Text input field containing "22222".
- Descripción:** Text input field containing "ELECTRIFICACION DE CALLES".
- Fondo:** Dropdown menu with "FAFM - FONDO PARA EL FORTALECIMIENTO MUNICIPAL" selected.
- Programa:** Dropdown menu with "S11 - MEJORAMIENTO URBANO" selected.
- Inversión:** A sub-section containing:
 - Fiscal:** Text input field with "22222".
 - Beneficiarios:** Text input field with "22222".
 - Total:** Text input field with "44444".
- Metas:** A sub-section containing:
 - Cantidad:** Text input field with "22222".
 - Unidades:** Text input field with "M2".
- Fecha:** A sub-section containing:
 - Ingreso:** Text input field with "01/01/2004".
 - Modificación:** Text input field with "01/01/2004".
- Navigation:** A row of icons including a table, a pencil, a trash can, a checkmark, a printer, and a hand cursor. To the left of these icons is a status bar showing "1 de 3 registros..." with navigation arrows.

Figura 2.6: Inhabilitación de botón *Guardar* y *Cancelar*.

2.8. Segunda Iteración

Al presionar el botón *Agregar* los controles deben limpiarse para permitir la captura de los nuevos datos de la propuesta. Además los botones *Guardar* y *Cancelar* se deberán habilitar para permitir la operación correspondiente, por lógica los demás botones no tendrán funcionalidad puesto que no son necesarios. Fig. 2.7.

The screenshot shows a Windows-style application window titled "Propuestas de Obra". The window contains a form with the following fields and sections:

- No. de Propuesta:** A text input field.
- Región:** A dropdown menu with "I - PACHUCA" selected.
- Municipio:** A dropdown menu with "13051 - MINERAL DE LA REFORMA" selected.
- Localidad:** A dropdown menu with "51047 - COL. CEUNI" selected.
- Número de Beneficiarios:** A text input field containing "10545".
- Descripción:** A text input field containing "PAVIMENTACION DE CALLES".
- Fondo:** A dropdown menu with "FAISM - FONDO DE INFRAESTRUCTURA SOCIAL MUNICIPAL" selected.
- Programa:** A dropdown menu with "S12 - PAVIMENTACION EN COLONIAS POPULARES" selected.
- Inversión:** A section with three text input fields: "Fiscal:" (2568910), "Beneficiarios:" (12500), and "Total:" (2,581,410.00).
- Metas:** A section with two text input fields: "Cantidad:" (10) and "Unidades:" (KM).
- Fecha:** A section with two text input fields: "Ingreso:" (01/01/2004) and "Modificación:" (01/01/2004).
- Navigation:** A set of navigation buttons (back, forward, search, etc.) and a "1 de 3 registros..." indicator.
- Buttons:** A row of icons for various actions, including a "Guardar" button.

Figura 2.7: Habilitación de botón *Guardar* y *Cancelar*.

Mientras son capturados los datos de la propuesta, se verifican para que concuerden con los tipos establecidos por la estructura de la base de datos. Por ejemplo en el campo de *Número de Beneficiarios* solo acepta caracteres numéricos, por lo que las teclas alfabéticas y los signos de puntuación están deshabilitados en este campo.

2. IMPLEMENTACIÓN DE LA METODOLOGÍA

Capturados todos los datos de la propuesta, se deberá presionar el botón *Guardar* para almacenar la información. En este proceso se verificará que todos los campos de la ventana contengan la información necesaria para poder ser almacenada. Si todos los datos cumplen con la estructura establecida son guardados en el registro correspondiente y se actualiza la tabla.

En caso de presionar el botón *Cancelar* la ventana borrará los datos de los controles, actualizará la tabla y se posicionará en el último registro visualizado.

2.8.2. Tarea 6 Editar propuesta de obra

Tarea	
Número de Tarea: 6	No. Historia: 2
Nombre Tarea: Editar propuesta de obra	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Fecha Inicio: 08/03/2004	Fecha Fin: 12/03/2004
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: Se realizarán los métodos para permitir el proceso de edición de datos de la tabla de <i>Propuestas</i> , tomando en cuenta aspectos estéticos y funcionales de la ventana de ingreso.	

Tabla 2.25: Tarea 6. Editar propuesta de obra

En esta *tarea* también deberán ser considerados los mismos puntos de la *tarea* anterior.

En primera instancia una propuesta debe estar almacenada para poder editarla, por lo que hasta que no se presione el botón *Editar* los datos visualizados en los controles no podrán ser editados. Fig. 2.8

Una vez presionado el botón, el usuario deberá modificar los datos según sus necesi-

2.8. Segunda Iteración

Propuestas de Obra

No. de Propuesta: 3

Región: V - ZIMAPAN **Municipio:** 13084 - ZIMAPAN

Localidad: 84001 - ZIMAPAN **Número de Beneficiarios:** 33333

Descripción: ELECTRIFICACION DE CALLES

Fondo: FAISM - FONDO DE INFRAESTRUCTURA SOCIAL MUNICIPAL

Programa: S11 - MEJORAMIENTO URBANO

Inversión		Metas		Fecha	
Fiscal:	33333	Cantidad:	33333	Ingreso:	01/01/2004
Beneficiarios:	33333	Unidades:	M3	Modificación:	01/01/2004
Total:	66666				

3 de 4 registros...

Editar

Figura 2.8: Botón *Editar*

dades. Y todos los botones, excepto *Guardar* y *Cancelar*, estarán deshabilitados puesto que no son necesarios.

Para almacenar la información se deberá presionar el botón *Guardar*, el proceso de verificación de los datos es el mismo que en la tarea anterior. La diferencia de este método y el de *Agregar* radica en que la ventana tiene que determinar que tipo de proceso debe realizarse, agregar un nuevo registro o actualizar un registro existente, por lo que será necesaria una variable para determinar el proceso a seguir. Fig. 2.9

2. IMPLEMENTACIÓN DE LA METODOLOGÍA

El usuario podrá modificar cuantos campos del registro quiera, por lo que no es necesario repetir el proceso para editar campo por campo.

The screenshot shows a software window titled "Propuestas de Obra". The form contains the following data:

Field	Value
No. de Propuesta	3
Región	V - ZIMAPAN
Municipio	13084 - ZIMAPAN
Localidad	84001 - ZIMAPAN
Número de Beneficiarios	33333
Descripción	[Empty]
Fondo	FALTA DE SERVICIOS BÁSICOS MUNICIPALES
Programa	[Empty]
Inversión Fiscal	33333
Inversión Beneficiarios	33333
Inversión Total	66666
Metas Cantidad	33333
Metas Unidades	M3
Fecha Ingreso	01/01/2004
Fecha Modificación	01/01/2004

Figura 2.9: Editar propuesta de obra.

En caso de presionar el botón *Cancelar* la ventana borrará los datos de los controles, actualizará la tabla y se posicionará en el último registro visualizado.

2.8.3. Tarea 7 Eliminar propuesta de obra

Una vez seleccionada la propuesta a eliminar, se deberá presionar el botón *Eliminar* con lo cual el método correspondiente seleccionará el registro correspondiente y lo eliminará de

2.8. Segunda Iteración

Tarea	
Número de Tarea: 7	No. Historia: 2
Nombre Tarea: Eliminar propuesta de obra	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.6
Fecha Inicio: 15/03/2004	Fecha Fin: 17/03/2004
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: Se realizarán los métodos para permitir el proceso de eliminación de registros de la tabla de <i>Propuestas</i> , tomando en cuenta aspectos estéticos y funcionales de la ventana de ingreso.	

Tabla 2.26: Tarea 7. Eliminar propuesta de obra

la tabla de *Propuestas*. Fig. 2.10

The screenshot shows a software window titled "Propuestas de Obra". The window contains a form with the following fields and values:

- No. de Propuesta:** 3
- Región:** V - ZIMAPAN
- Municipio:** 13084 - ZIMAPAN
- Localidad:** 84001 - ZIMAPAN
- Número de Beneficiarios:** 33333
- Descripción:** ELECTRIFICACION DE CALLES
- Fondo:** FAISM - FONDO DE INFRAESTRUCTURA SOCIAL MUNICIPAL
- Programa:** S11 - MEJORAMIENTO URBANO

Below the main form, there are three sections:

- Inversión:** Fiscal: 33333, Beneficiarios: 33333, Total: 66666
- Metas:** Cantidad: 33333, Unidades: M3
- Fecha:** Ingreso: 01/01/2004, Modificación: 01/01/2004

At the bottom of the window, there is a navigation bar with a status indicator "3 de 4 registros..." and a set of icons. The "Eliminar" (Delete) icon, which shows a trash bin, is highlighted, and a button labeled "Eliminar" is visible below it.

Figura 2.10: Eliminar propuesta de obra.

2. IMPLEMENTACIÓN DE LA METODOLOGÍA

Al presionar *Eliminar* el método preguntará al usuario si realmente quiere eliminar el registro, para ello se desplegara una ventana donde el usuario debe elegir la opción para decidir la siguiente operación.

Si el usuario presiona el botón afirmativo el método deberá elegir el registro correspondiente y ejecutar la instrucción para la eliminación del registro de la tabla de *Propuestas*, posteriormente actualizará la tabla y se posicionará en el primer registro, el cual es presentado por los controles.

En caso de presionar el botón de cancelación el método deberá actualizar la tabla y se posicionará en el último registro visualizado, el cual es presentado por los controles.

2.8.4. Tarea 8 Actualizar propuestas de obra

Tarea	
Número de Tarea: 8	No. Historia: 2
Nombre Tarea: Actualizar propuesta de obra	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.4
Fecha Inicio: 18/03/2004	Fecha Fin: 19/03/2004
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: Se realizarán los métodos para permitir el proceso de actualización de registros de la tabla de Propuestas.	

Tabla 2.27: Tarea 8. Actualizar propuestas de obra

El botón llamado *Actualizar* permitirá ejecutar el método para actualizar la tabla de *Propuestas*, este método ejecutará las instrucciones necesarias para solucionar problemas que puedan surgir en cuanto al ingreso, modificación o eliminación de registros de la tabla de *Propuestas*. Fig. 2.11

2.9. Tercera Iteración

Al actualizar la tabla, los controles desplegarán los datos del primer registro de la tabla.

The screenshot shows a Windows-style application window titled "Propuestas de Obra". The window contains a form with the following fields and values:

- No. de Propuesta:** 3
- Región:** V - ZIMAPAN
- Municipio:** 13084 - ZIMAPAN
- Localidad:** 84001 - ZIMAPAN
- Número de Beneficiarios:** 33333
- Descripción:** ELECTRIFICACION DE CALLES
- Fondo:** FAISM - FONDO DE INFRAESTRUCTURA SOCIAL MUNICIPAL
- Programa:** S11 - MEJORAMIENTO URBANO

Below these fields are three summary boxes:

- Inversión:** Fiscal: 33333, Beneficiarios: 33333, Total: 66666
- Metas:** Cantidad: 33333, Unidades: M3
- Fecha:** Ingreso: 01/01/2004, Modificación: 01/01/2004

At the bottom of the window, there is a navigation bar showing "3 de 4 registros..." and a toolbar with icons for navigation, editing, deletion, and a button labeled "Actualizar".

Figura 2.11: Actualizar propuestas de obra.

2.9. Tercera Iteración

Para la tercera iteración del proyecto se desarrollará la historia de usuario llamada *Operaciones secundarias sobre propuestas*, la cual se divide en las siguientes tareas:

- Ordenamiento de propuestas de obra

2. IMPLEMENTACIÓN DE LA METODOLOGÍA

- Filtrado de propuestas de obra
- Búsqueda de propuestas de obra

2.9.1. Tarea 9 Ordenamiento de propuestas de obra

Tarea	
Número de Tarea: 9	No. Historia: 3
Nombre Tarea: Ordenamiento de propuestas de obra	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.6
Fecha Inicio: 22/03/2004	Fecha Fin: 24/03/2004
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: : Se realizarán los métodos necesarios para permitir el ordenamiento descendente y ascendente de los de registros de la tabla de <i>Propuestas</i> , para ser presentados en pantalla.	

Tabla 2.28: Tarea 9. Ordenamiento de propuestas de obra

Con estas funcionalidades el usuario tendrá la posibilidad de ordenar de forma ascendente y descendente los registros de la tabla de *Propuestas*, para ello deberá seleccionar el campo, bastará con posicionar el cursor y dar click sobre el campo, por el cual se deben ordenar los registros y seleccionar la opción de ordenar.

Para ello es necesario añadir a la ventana principal del sistema, diseñada en la primera iteración, una opción para el ordenamiento. En el menú *Registros*, se encuentra la opción *Ordenar*, es aquí donde se elegirá el método de ordenamiento requerido por el usuario.

A su vez en la barra de herramientas se proporciona esta misma opción, por medio de los botones de ordenamiento, en la Fig. 2.12 se presentan estas opciones.

2.9. Tercera Iteración

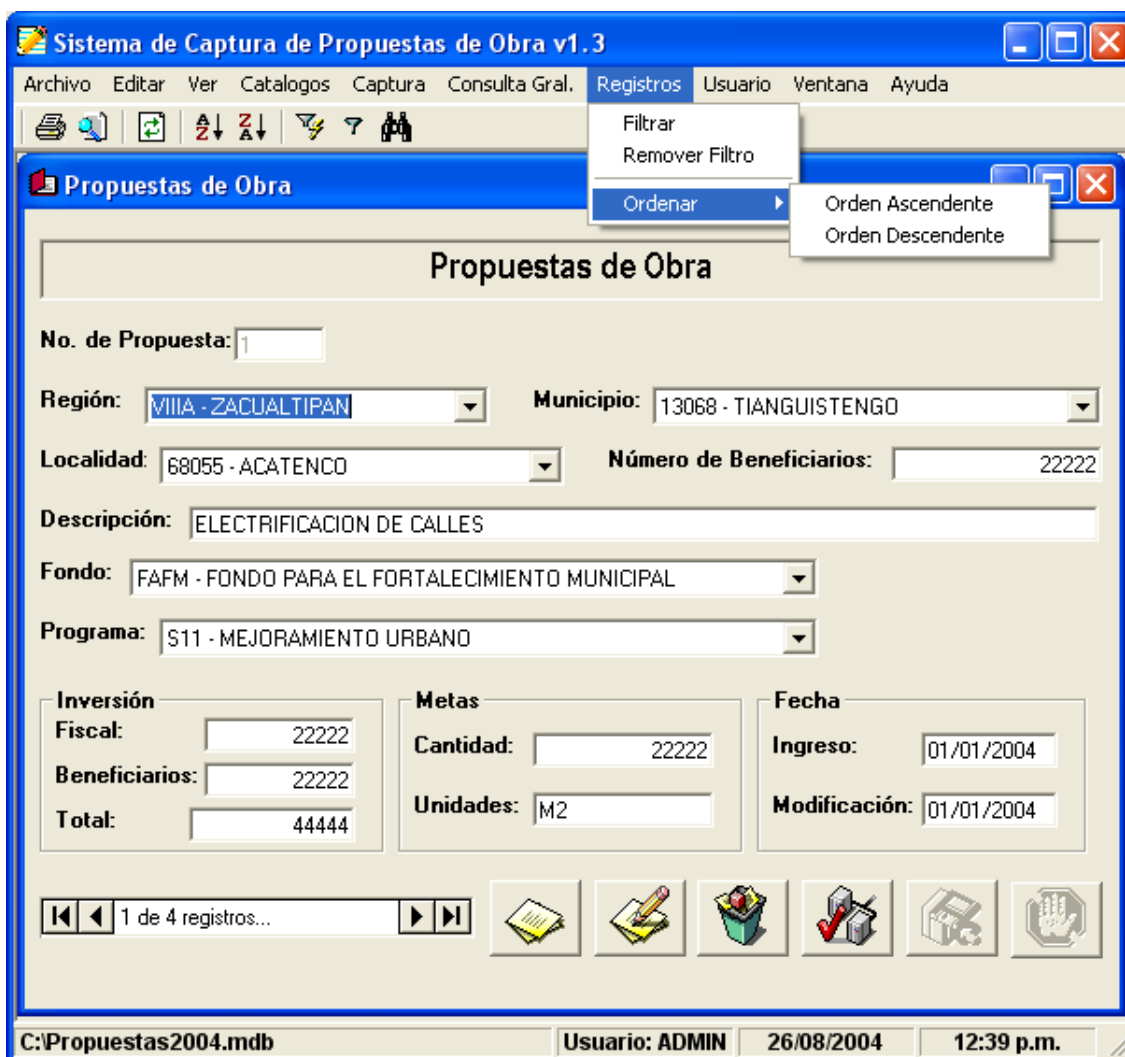


Figura 2.12: Ordenamiento de propuestas de obra.

2. IMPLEMENTACIÓN DE LA METODOLOGÍA

Al ejecutar esta función, la pantalla mostrará los registros ordenados y el usuario podrá visualizarlos por medio del control de desplazamiento.

Para regresar al estado normal de la ventana de ingreso, bastará con presionar el botón *Actualizar*.

2.9.2. Tarea 10 Filtrado de propuestas de obra

Tarea	
Número de Tarea: 10	No. Historia: 3
Nombre Tarea: Filtrado de propuestas de obra	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.8
Fecha Inicio: 25/03/2004	Fecha Fin: 30/03/2004
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: : Se realizarán los métodos necesarios para permitir el filtrado de los de registros de la tabla de <i>Propuestas</i> , por medio de un criterio. Los cuales serán presentados en pantalla.	

Tabla 2.29: Tarea 10. Filtrado de propuestas de obra

Este método permitirá al usuario realizar un filtro de los registros de la tabla de *Propuestas*, por medio de un criterio que permitirá dicha operación.

El usuario deberá elegir cualquier campo de la ventana de ingreso, bastará con posicionar el cursor y dar click sobre el campo, para que posteriormente se presione la opción *Filtrar*.

Esta opción al igual que la de ordenar, se encuentra en el menú *Registros* seleccionando la opción *Filtrar*. También estará disponible en la barra de herramientas. Fig. 2.13

Este proceso seleccionará todos los registros en los cuales se tenga como coincidencia

2.9. Tercera Iteración

Sistema de Captura de Propuestas de Obra v1.3

Archivo Editar Ver Catalogos Captura Consulta Gral. **Registros** Usuario Ventana Ayuda

Filtrar
Remover Filtro
Ordenar

Propuestas de Obra

No. de Propuesta:

Región: **Municipio:**

Localidad: **Número de Beneficiarios:**

Descripción:

Fondo:

Programa:

Inversión		Metas		Fecha	
Fiscal:	<input type="text" value="22222"/>	Cantidad:	<input type="text" value="22222"/>	Ingreso:	<input type="text" value="01/01/2004"/>
Beneficiarios:	<input type="text" value="22222"/>	Unidades:	<input type="text" value="M2"/>	Modificación:	<input type="text" value="01/01/2004"/>
Total:	<input type="text" value="44444"/>				

1 de 4 registros...

C:\Propuestas2004.mdb Usuario: ADMIN 26/08/2004 12:44 p.m.

Figura 2.13: Filtrado de propuestas de obra.

2. IMPLEMENTACIÓN DE LA METODOLOGÍA

el campo seleccionado por el usuario, y los presentará en la ventana de ingreso.

El método inverso, será necesario para realizar otras operaciones sobre la tabla de *Propuestas*, por lo que se encontrará en el mismo menú *Registros* y realizará una actualización de la tabla y eliminará la selección realizada anteriormente.

2.9.3. Tarea 11 Búsqueda de propuesta de obra

Tarea	
Número de Tarea: 11	No. Historia: 3
Nombre Tarea: Filtrado de propuestas de obra	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.6
Fecha Inicio: 31/03/2004	Fecha Fin: 20/04/2004
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: : Se realizarán los métodos necesarios para permitir la búsqueda de registros en la tabla de <i>Propuestas</i> , para lo que será necesario un criterio de búsqueda.	

Tabla 2.30: Tarea 11. Búsqueda de propuestas de obra

Estos métodos permitirán al usuario realizar búsquedas rápidas dentro de la tabla de *Propuestas*, para esto será necesario como en todos estos métodos elegir el campo a buscar en los registros.

Posteriormente será necesario elegir la opción *Buscar*, ya sea en el menú *Editar* o en la barra de herramientas.

Hecho lo anterior se mostrará una ventana en la cual se debe especificar la cadena a buscar, el campo donde se debe buscar y por último una de las opciones de la lista desplegable, la primera permite buscar la cadena en cualquier parte del campo, la segunda

2.9. Tercera Iteración

buscará exactamente la cadena en el campo y la ultima buscará la cadena al principio del campo. Fig. 2.14

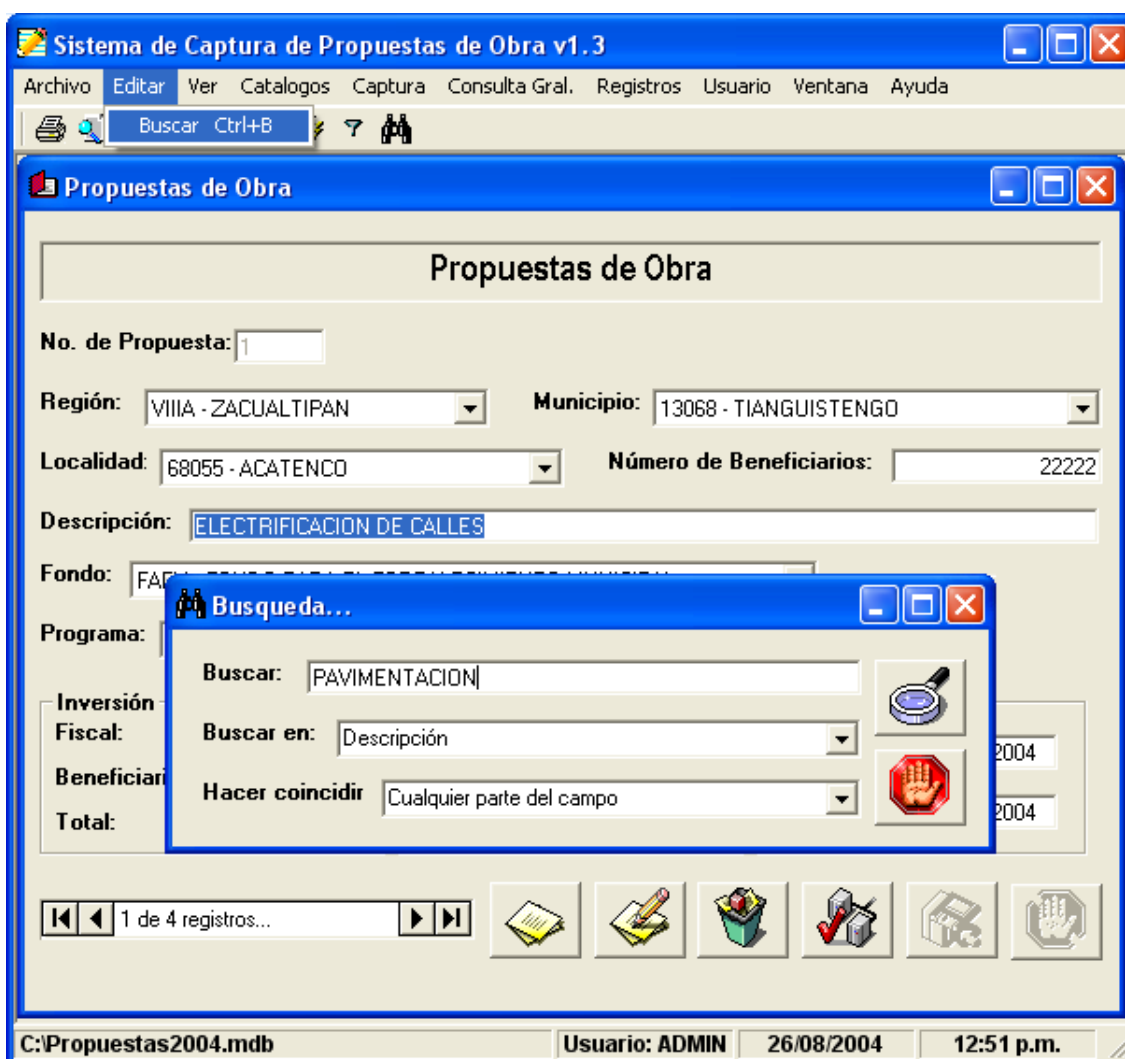


Figura 2.14: Búsqueda de propuesta de obra.

2.10. Cuarta Iteración

La siguiente historia de usuario de la cuarta iteración es la llamada *Consulta de Catálogos*, la cual se divide en las siguientes tareas:

1. Diseño de catálogo de *Localidades*
2. Diseño de catálogo de *Fondo*
3. Diseño de catálogo de *Programa*

2.10.1. Tarea 12 Diseño de catálogo de Localidades

Tarea	
Número de Tarea: 12	No. Historia: 4
Nombre Tarea: Diseño de catálogo de <i>Localidades</i>	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Fecha Inicio: 05/04/2004	Fecha Fin: 09/05/2004
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: : Se diseñará la ventana correspondiente a la tabla de <i>Localidades</i> con sus respectivos controles para la manipulación de los registros.	

Tabla 2.31: Tarea 12. Diseño de catálogo de *Localidades*

Los catálogos permitirán al usuario visualizar la información almacenada en las tablas secundarias de la base de datos. El primer catálogo es el denominado *Localidades*, en este se permitirá la manipulación de la información de la tabla correspondiente.

Dado que estos métodos necesarios para el desarrollo de esta tarea, ya se han llevado a cabo antes en las anteriores historias de usuario, bastará con adecuarlos a los propósitos de esta tarea y con esto se ahorra tiempo de desarrollo.

2.10. Cuarta Iteración

Por ejemplo para este catálogo, se permitirán las operaciones básicas de manipulación, como son agregar, editar, eliminar, actualizar, guardar y cancelar operación, sobre los registros de la tabla de *Localidades*. Así como también las operaciones secundarias, como búsqueda, ordenamiento y filtrado de registros.

En la Fig. 2.15 se presenta la pantalla que corresponde al catálogo de Localidades.



Figura 2.15: Ventana de catálogo de *Localidades*.

2.10.2. Tarea 13 Diseño de catálogo de Fondo

Como en la tarea anterior este catálogo tendrá la misma funcionalidad, todos los métodos necesarios han sido desarrollados anteriormente y se adecuarán a las necesidades de la tarea para llevarla a cabo en un tiempo mínimo. Así las operaciones permitidas son las mismas, solo cambiará la tabla en la cual se ejecutarán las operaciones, esta tabla se denomina *FONDO*.

2. IMPLEMENTACIÓN DE LA METODOLOGÍA

Tarea	
Número de Tarea: 13	No. Historia: 4
Nombre Tarea: Diseño de catálogo de <i>Fondo</i>	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Fecha Inicio: 12/04/2004	Fecha Fin: 16/04/2004
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: : Se diseñará la ventana correspondiente a la tabla de Fondo con sus respectivos controles para la manipulación de los registros.	

Tabla 2.32: Tarea 13. Diseño de catálogo de *Fondo*

En la Fig. 2.16 se presenta la pantalla que corresponde al catálogo de Fondo.



Figura 2.16: Ventana de catálogo de *Fondo*.

2.10.3. Tarea 14 Diseño de catálogo de Programas

Esta ventana permitirá al usuario manipular la información de la tabla *PROGRAMAS*, con todas las funcionalidades de las ventanas anteriores.

En la Fig. 2.17 se presenta la pantalla que corresponde al catálogo de Programas.

2.11. Quinta Iteración

Tarea	
Número de Tarea: 14	No. Historia: 4
Nombre Tarea: Diseño de catálogo de <i>Programas</i>	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Fecha Inicio: 19/04/2004	Fecha Fin: 23/04/2004
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: : Se diseñará la ventana correspondiente a la tabla de Fondo con sus respectivos controles para la manipulación de los registros.	

Tabla 2.33: Tarea 14. Diseño de catálogo de *Programas*



Figura 2.17: Ventana de catálogo de *Programas*.

2.11. Quinta Iteración

En la quinta iteración se procede al desarrollo de cuatro historias de usuario, la primera historia de usuario es la llamada *Control de acceso a usuarios* la cual se divide en las siguientes tareas:

- Creación de archivo de usuarios
- Diseño de ventana para identificación de usuarios

2.11.1. Tarea 15 Creación de archivo de usuarios

Tarea	
Número de Tarea: 15	No. Historia: 5
Nombre Tarea: Creación de archivo de usuarios	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.2
Fecha Inicio: 26/04/2004	Fecha Fin: 26/04/2004
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: : Será necesaria la creación de un archivo, en donde puedan ser almacenados los datos de los usuarios del sistema.	

Tabla 2.34: Tarea 15. Creación de archivo de usuarios

La creación de un archivo para almacenar los datos del usuario permite al sistema comprobar que los usuarios que intentan acceder al sistema son realmente quienes dicen ser.

Para dar solución a esto se utilizará un archivo de texto en el que por medio de etiquetas pueda ser leído por el sistema. A continuación se presenta el contenido de dicho archivo.

```
[DEFAULT]
USER=ADMIN
[ADMIN]
PASSWORD=NIMDA
[USER]
PASSWORD=RESU
```

Las etiquetas son las palabras que se encuentran entre corchetes “[]”, mientras que las variables son las palabras que se encuentran en la parte izquierda del símbolo de igual que

2.11. Quinta Iteración

“=” y por último el contenido de la variable son las palabras del lado derecho del símbolo de igual que “=”.

Este archivo será guardado con extensión punto ini, para que pueda ser leído por la API a emplearse por el sistema.

2.11.2. Tarea 16 Diseño de ventana para identificación de usuarios

Tarea	
Número de Tarea: 16	No. Historia: 5
Nombre Tarea: Diseño de ventana para identificación de usuarios	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.4
Fecha Inicio: 27/04/2004	Fecha Fin: 28/04/2004
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: : Se diseñará la ventana correspondiente para la identificación de usuarios, en la que serán proporcionados el nombre de usuario y password.	

Tabla 2.35: Tarea 16. Diseño de ventana para identificación de usuarios

Esta funcionalidad permitirá restringir el uso no autorizado del sistema, por lo que el usuario deberá introducir un identificador y una contraseña.

El método encargado de este proceso validará los datos proporcionados por el usuario, realizando una lectura del archivo, creado en la tarea anterior, y verificando que los datos sean correctos, con lo que el usuario tendrá acceso completo al sistema.

En la Fig. 2.18 se muestra la ventana de control de acceso a usuarios, el primer campo es el denominado *Usuario*, en el que se deberá registrar el nombre con el que pretende

2. IMPLEMENTACIÓN DE LA METODOLOGÍA

acceder al sistema, el segundo *Password*, permite comprobar que realmente es el usuario que dice ser.



Figura 2.18: Ventana para identificación de usuarios.

Este método leerá el archivo de manera secuencial, por lo que se debe proporcionar una etiqueta destino y la variable de la cual se desea saber el contenido. Una vez proporcionado el nombre de usuario, se valida que sea una etiqueta válida en el archivo y se procede con la lectura de la variable. Si el contenido de la variable y el valor proporcionado por el usuario en el campo *Password* es igual, se ejecuta el sistema para su manipulación.

La siguiente historia de usuario de esta quinta iteración se denomina *Gestión de datos de Usuario*, con la siguiente tarea:

- Cambio de contraseña de usuario.

2.11.3. Tarea 17 Cambio de contraseña de usuario

Dado que es de suma importancia mantener la seguridad de la información manipulada por el sistema, este método proveerá una herramienta útil para lograr dicho objetivo.

2.11. Quinta Iteración

Tarea	
Número de Tarea: 17	No. Historia: 6
Nombre Tarea: Diseño de Ventana para identificación de usuarios	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.4
Fecha Inicio: 29/04/2004	Fecha Fin: 30/04/2004
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: : Se diseñará la ventana correspondiente para el proceso de cambio de contraseña de usuario.	

Tabla 2.36: Tarea 17. Cambio de contraseña de usuario

Las contraseñas deberán ser cambiadas periódicamente para que no puedan ser encontradas por personal no autorizado.

Para esto el sistema proporcionara la opción para cambiar la contraseña del usuario que se encuentra activo en el sistema. Por ejemplo si el usuario ADMIN, ha accedido al sistema y desea cambiar su contraseña, bastará con elegir del menú Usuario, la opción Cambiar contraseña.

Con esto se mostrará una ventana donde se pide que ingrese el nombre de usuario, contraseña anterior y la contraseña nueva, con esto se ejecutará el método necesario para buscar el usuario en el archivo creado para dicho fin, y si este existe se procede a la escritura de la nueva contraseña en la variable correspondiente. Fig.2.19

La historia de usuario llamada *Administración de la BD*, es la siguiente de la quinta iteración, la cual se divide en las siguientes tareas:

- Abrir Base de Datos
- Crear nueva Base de Datos

2. IMPLEMENTACIÓN DE LA METODOLOGÍA

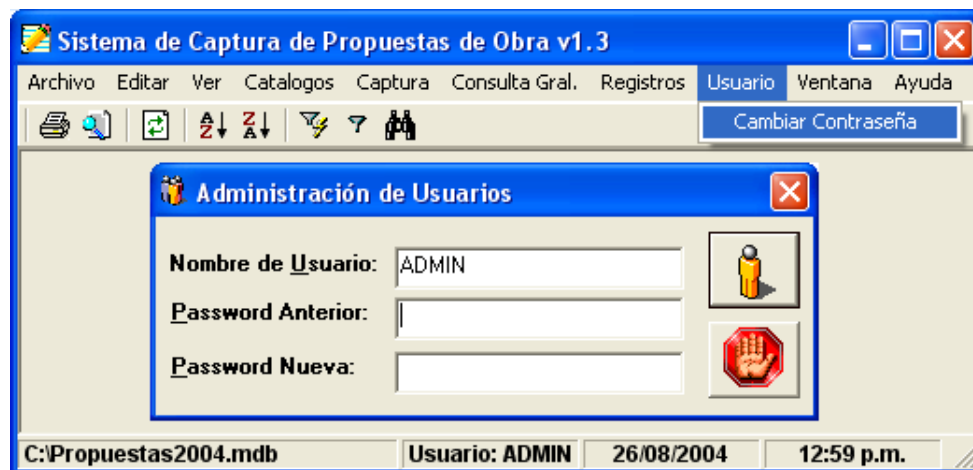


Figura 2.19: Ventana para el cambio de contraseña de usuario.

- Crear Respaldo de Base de Datos

2.11.4. Tarea 18 Abrir Base de Datos

		Tarea
Número de Tarea: 18	No. Historia: 7	
Nombre Tarea: Abrir Base de Datos		
Tipo de Tarea: Desarrollo	Puntos estimados: 0.4	
Fecha Inicio: 03/05/2004	Fecha Fin: 04/05/2004	
Programador responsable: Enrique Martínez, Filiberto Pérez		
Descripción: Se realizarán los métodos necesarios para permitir el acceso a la base de datos para la manipulación de la información.		

Tabla 2.37: Tarea 18. Abrir Base de Datos

Después de haberse identificado, el usuario deberá seleccionar la base de datos que desea manipular, para esto se desarrollarán los métodos necesarios para poder elegir, conectar y manipular la base de datos.

El primer paso es elegir la base de datos, para esto se debe agregar a la ventana prin-

2.11. Quinta Iteración

principal del sistema un menú donde se localice la opción de *Abrir BD*, con esto se permite al usuario manipular diferentes bases de datos, ya que si en cualquier momento se desea cambiar de base de datos solo es necesario elegir esta opción y los datos deberán actualizarse automáticamente.

Esta opción se encuentra en el menú *Archivo* como se muestra en la Fig. 2.20

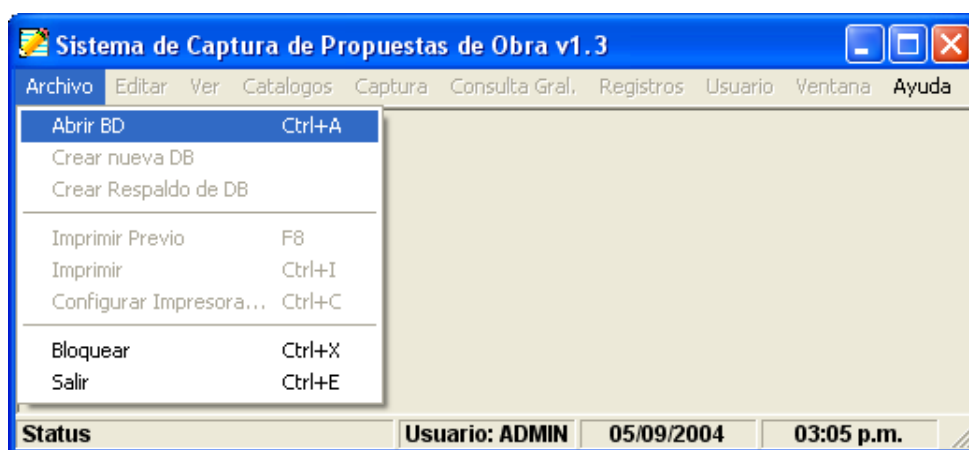


Figura 2.20: Menú *Archivo*, *Abrir BD*.

Posteriormente el sistema pedirá al usuario que elija la base de datos, por medio del cuadro de dialogo Base de Datos origen que se muestra en la Fig.2.21

Realizado el paso anterior el sistema deberá realizar la conexión a la base de datos por medio de una cadena de conexión, donde se especifica el tipo de controlador que se ha de usar, así como algunos otros parámetros necesarios. Esta cadena de conexión es creada por el sistema de acuerdo a la ubicación de la base de datos seleccionada, no importando si la base de datos se encuentra en la misma PC o si se encuentra en una PC dentro de la red de ña SEDESOS.

2. IMPLEMENTACIÓN DE LA METODOLOGÍA

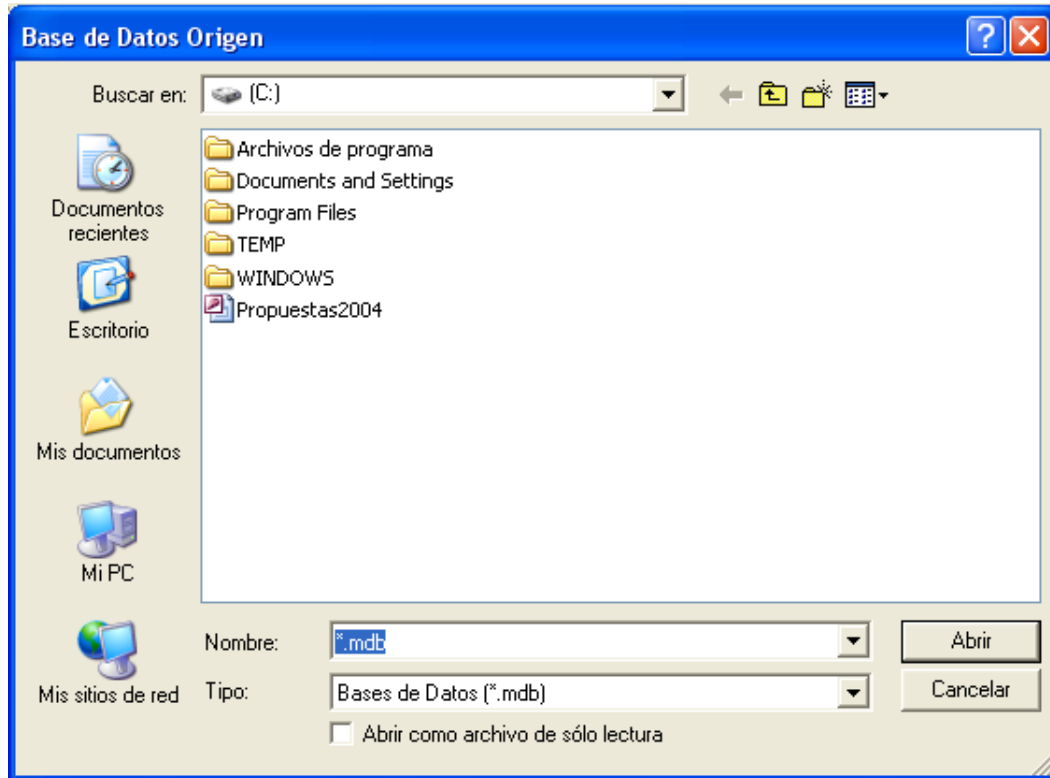


Figura 2.21: Ventana *Abrir BD*.

El siguiente paso es el de verificar si la BD cumple con las características necesarias para su manipulación, esto es, que contenga todas las tablas necesarias para poder manipularla, para esto el sistema debe evaluar internamente la BD por medio de los nombres de las tablas incluidas, si el sistema encuentra las tablas necesarias se procede con la lectura de cada una de ellas.

Una vez evaluada la BD el usuario podrá manipular la información de acuerdo a sus necesidades.

2.11. Quinta Iteración

2.11.5. Tarea 19 Crear nueva Base de Datos

Tarea	
Número de Tarea: 19	No. Historia: 7
Nombre Tarea: Crear nueva Base de Datos	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.4
Fecha Inicio: 05/05/2004	Fecha Fin: 06/05/2004
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: Se realizarán los métodos necesarios para crear una nueva base de datos con la misma estructura requerida por el sistema.	

Tabla 2.38: Tarea 19. Crear nueva Base de Datos

En ocasiones el usuario se vera en la necesidad de crear una nueva BD, ya sea por cambio de año, cambio de administración, etc. Para dar solución a este problema, el sistema permitirá al usuario crear nuevas BD que cumplan con la estructura requerida.

Así como en la tarea anterior esta opción se encontrará en el menú *Archivo* como se muestra en la Fig. 2.22

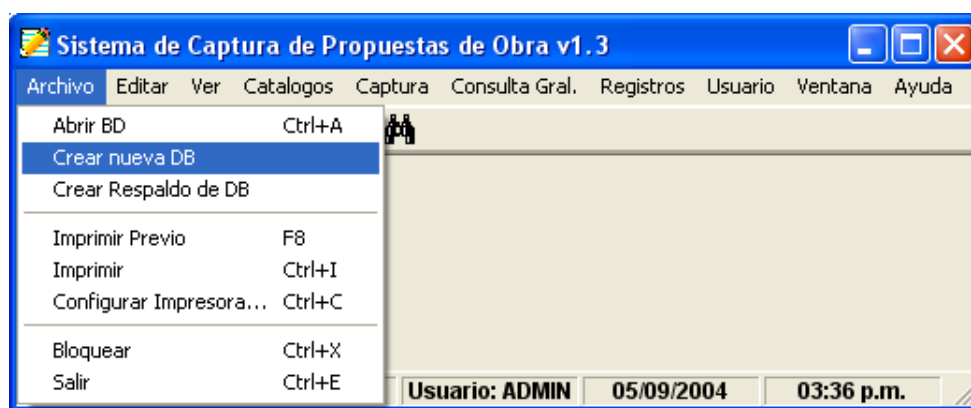


Figura 2.22: Menú *Archivo*, *Crear nueva BD*.

Una vez seleccionada la opción, el sistema creará un nombre para la base de datos,

2. IMPLEMENTACIÓN DE LA METODOLOGÍA

como se mencionó anteriormente el nombre de la base de datos consta de la palabra *Propuestas* y el año de creación. Por lo que el sistema creará una base de datos con nombre *Propuestas200x*, en el mismo directorio donde se encuentra la última base de datos a la cual se tuvo acceso.

El sistema creará una copia temporal de la base de datos actual para compactarla, esta base de datos temporal se encontrará en la raíz *c* y con el nombre de *bd.mdb*. Posteriormente se eliminarán todos los datos de la tabla *Propuestas*, y por último se volverá a compactar esta base de datos temporal para cambiarla de nombre y ubicación según las restricciones del nombre antes mencionadas.

La base de datos temporal es eliminada por el mismo método, liberando así espacio en disco y evitando la posible saturación de bases de datos innecesarias.

2.11.6. Tarea 20 Crear respaldo de Base de Datos

Tarea	
Número de Tarea: 20	No. Historia: 7
Nombre Tarea: Crear respaldo de Base de Datos	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.2
Fecha Inicio: 07/05/2004	Fecha Fin: 07/05/2004
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: Se realizarán los métodos necesarios para crear un respaldo de la base de datos actual, para mantener la seguridad de la información.	

Tabla 2.39: Tarea 20. Crear respaldo de Base de Datos

Este método permitirá al usuario crear respaldo de la base de datos actual con la finalidad de mantener la seguridad de la información, esta opción se encontrará en el

2.11. Quinta Iteración

menú *Archivo* como los anteriores. Fig.2.23

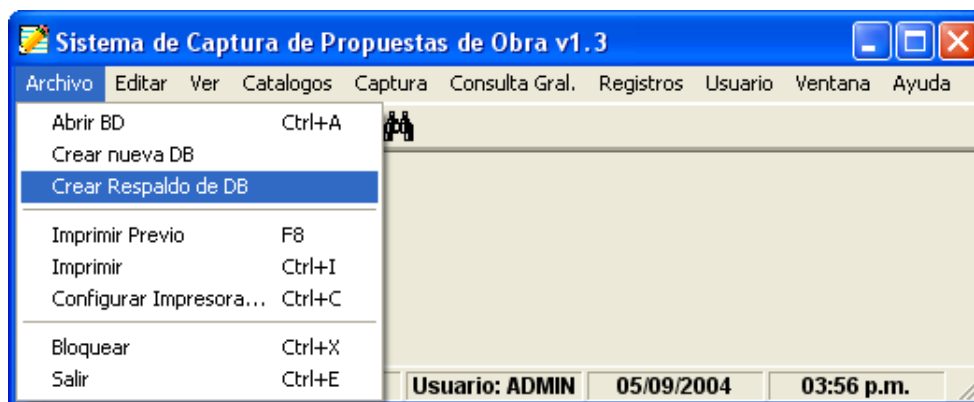


Figura 2.23: Menú *Archivo*, *Crear respaldo de BD*.

Al elegir esta opción el sistema realizará una copia de la base de datos actual y le asignará el mismo nombre de la base de datos, solo que al final del nombre se agregará la letra *A* con la cual se identificará un respaldo.

Por ejemplo se la base de datos actual tiene por nombre *Propuestas2004.mdb*, al realizar el respaldo el sistema nombrará a la copia *Propuestas2004A.mdb*. Y se ubicará en el mismo directorio que la original.

Por último la siguiente historia de usuario de la quinta iteración es la llamada *Reporte General de Propuestas* la cual se divide en las siguientes tareas:

- Diseño de formato de reporte general
- Integración del formato al sistema

2. IMPLEMENTACIÓN DE LA METODOLOGÍA

Tarea	
Número de Tarea: 21	No. Historia: 8
Nombre Tarea: Diseño de formato de reporte general	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.8
Fecha Inicio: 10/05/2004	Fecha Fin: 13/05/2004
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: Se realizarán los métodos necesarios para crear un respaldo de la base de datos actual, para mantener la seguridad de la información.	

Tabla 2.40: Tarea 21. Diseño de formato de reporte general

2.11.7. Tarea 21 Diseño de formato de reporte general

Para la construcción del formato de reporte se utilizará el programa de diseño *Seagate Crystal Reports*, el cual provee de herramientas especializadas para el manejo y presentación de información.

Este reporte deberá contar con todos los datos de la propuesta de obra, y tendrá que estar clasificado de acuerdo a criterios establecidos por otros reportes existentes en la Dirección.

El reporte deberá contar con los datos generales de la organización, en este caso se incluirá el nombre de la Secretaría de Desarrollo Social, el nombre de la Subsecretaría de Planeación y por último el nombre de la Dirección Técnica. También deberá incluir los datos generales del tipo de documento, así como también la fecha de solicitud de reporte.

Los datos deberán estar clasificados de acuerdo a dos criterios, en primera instancia se deberá ordenar los datos de acuerdo a la *Región* en que se encuentre la obra, posteriormente se deberán clasificar los datos de acuerdo a el Municipio. Después los demás datos

2.11. Quinta Iteración

se clasificarán en columnas donde puedan ser visualizados de forma clara.

En la Fig.2.24 se presentan el diseño del formato para el reporte general.

SECRETARÍA DE DESARROLLO SOCIAL SUBSECRETARÍA DE PLANEACIÓN Área Técnica Reporte general de Propuestas de Obra 2004, Ramo 33							
Localidad	Programa	Descripción de la obra	Fondo	I. Total	I. Fiscal	I. Beneficiarios	Metas
Región:	REG_DESC_REG						
Municipio:	DESC_MUN						
DESC_LOC	CLAV_PP	DESC_OBRA	CLAV_FG	INV_TOTAL	INV_FISCAL	V_BENEFICIARIOS	CANTIDAD

Figura 2.24: Diseño de formato de reporte general.

2.11.8. Tarea 22 Integración del formato al sistema

Tarea	
Número de Tarea: 22	No. Historia: 8
Nombre Tarea: Integración del formato al sistema	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.8
Fecha Inicio: 14/05/2004	Fecha Fin: 19/05/2004
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: Se integrará el formato de reporte de propuestas al sistema, por medio de los métodos necesarios para que pueda ser visualizado desde la ventana principal del sistema.	

Tabla 2.41: Tarea 22. Integración del formato al sistema

El sistema contará con los métodos necesarios para la manipulación del formato de reporte, para esto será necesario añadir un objeto de control. Este objeto de control per-

2. IMPLEMENTACIÓN DE LA METODOLOGÍA

mitirá proporcionar los parámetros necesarios para la correcta visualización de los datos.

El primer parámetro necesario para la visualización del formato es la ubicación del formato, para esto se debe especificar el nombre y ruta de destino en donde se encuentra en formato. El siguiente parámetro es la ubicación del origen de datos, esto es donde se encuentra la base de datos. Los siguientes datos pertenecen al estilo de visualización requerido por el sistema, en este caso se deberá presentar el informe en pantalla, de forma maximizada y con algunas barras de herramientas.

Una vez establecido lo anterior será necesario ubicar la opción para que el usuario pueda acceder al reporte desde la ventana principal del sistema. Para esto se agregará la opción *Reporte* al menú *Consulta General* de la barra de menú, como se muestra en la Fig. 2.25

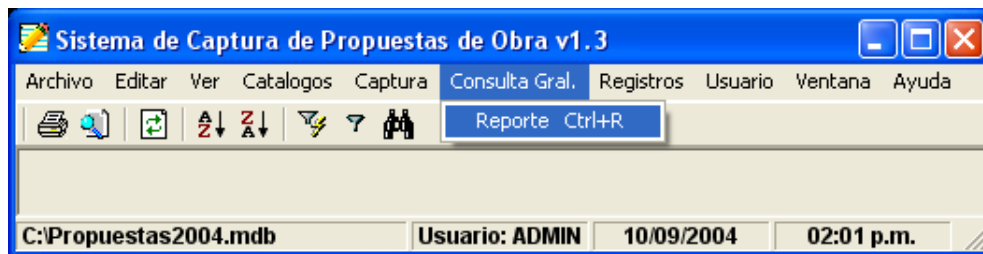


Figura 2.25: Menú *Consulta General*, *Reporte*.

2.12. Sexta iteración

La sexta iteración está compuesta por tres historias de usuario, la primera de ellas es la llamada *Reporte de Propuesta* la cual se divide en las siguientes tareas:

- Diseño de formato de reporte de propuesta

2.12. Sexta iteración

- Integración del formato al sistema

2.12.1. Tarea 23 Diseño de formato de reporte de propuesta

Tarea	
Número de Tarea: 23	No. Historia: 9
Nombre Tarea: Diseño de formato de reporte de propuesta	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.4
Fecha Inicio: 20/05/2004	Fecha Fin: 25/05/2004
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: Se desarrollará el formato para generar el reporte específico de una propuesta de obra, según la selección del usuario.	

Tabla 2.42: Tarea 23. Diseño de formato de reporte de propuesta

Como se mencionó anteriormente se utilizará *Seagate Crystal Reports* para el diseño del reporte.

En este caso el diseño del formato deberá presentarse de la misma manera que se presentan los datos en pantalla. Para esto será necesario que el usuario se encuentre en la ventana de captura de propuestas para que pueda visualizar el reporte de la propuesta actual.

Así como en el reporte anterior, este formato deberá contar con los datos generales de la organización, en este caso se incluirá el nombre de la Secretaría de Desarrollo Social, el nombre de la Subsecretaría de Planeación y por último el nombre de la Dirección Técnica. También deberá incluir los datos generales del tipo de documento, así como también la fecha de solicitud de reporte.

2. IMPLEMENTACIÓN DE LA METODOLOGÍA

La información de las propuestas de obra será presentada de la misma manera en que es presentada en la pantalla de captura de propuestas. *Los datos son Región, Municipio, Localidad, Número de beneficiarios, Descripción de obra, Fondo, Programa, Inversiones, Metas y Fechas.* Fig. 2.26

The image shows a screenshot of a report design interface. At the top, there is a header box containing the logo 'POR HIDALGO' and the text 'SECRETARÍA DE DESARROLLO SOCIAL', 'SUBSECRETARÍA DE PLANEACIÓN', 'Propuestas 2004 Ramo 33', and 'Área Técnica'. Below the header, there is a section titled 'Propuesta de obra.' with a 'Fecha de Impresión' field on the right. The main content area contains several input fields: 'Región: REG_DESC_REG', 'Municipio: O_MUN_DESC_MUN', 'Localidad: NO_LO_DESC_LOC', 'Número de Beneficiarios: NENEFICIARIOS', 'Descripción de Obra: DESC OBRA', and 'Fondo: FONDO_DESC_FONDO'. Below these fields are three summary tables: 'Inversión' with columns 'Beneficiarios', 'Fiscal', and 'Total'; 'Metas' with columns 'Cantidad' and 'Unidad'; and 'Fecha' with columns 'Ingreso' and 'Modificación'. The bottom of the page shows 'Página N de M'.

Figura 2.26: Diseño de formato de reporte de propuesta.

2.12.2. Tarea 24. Integración del formato al sistema

Así como en el reporte anterior, para poder incluirlo en el sistema, se deberá agregar un objeto de control para manipular el reporte.

Se deberán incluir los mismos parámetros de configuración, la ubicación del formato, origen de datos, y los parámetros de visualización.

Una vez establecido lo anterior será necesario ubicar la opción para que el usuario pueda

2.12. Sexta iteración

Tarea	
Número de Tarea: 24	No. Historia: 9
Nombre Tarea: Integración del formato al sistema	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.6
Fecha Inicio: 26/05/2004	Fecha Fin: 28/05/2004
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: Se integrará el formato de reporte de propuestas al sistema, por medio de los métodos necesarios para que pueda ser visualizado desde la ventana principal del sistema.	

Tabla 2.43: Tarea 24. Integración del formato al sistema

acceder al reporte desde la ventana principal del sistema. Para esto se agregará la opción *Imprimir Previo* al menú *Archivo* de la barra de menú y el icono correspondiente en la barra de herramientas, como se muestra en la Fig. 2.27

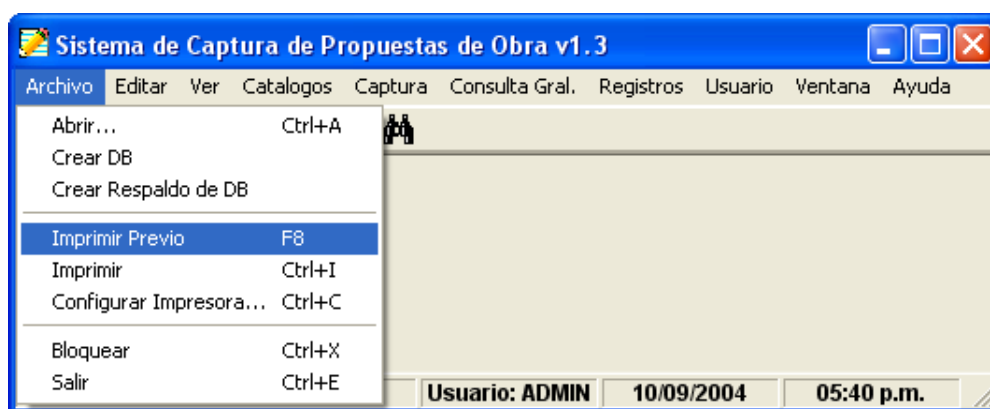


Figura 2.27: Menú *Archivo*, *Imprimir Previo*.

La segunda historia de usuario de la sexta iteración es la llamada *Manual de operación del sistema* la cual se divide en las siguientes tareas:

- Diseño de archivo del manual de operación
- Integración del archivo al sistema

2.12.3. Tarea 25. Diseño de archivo del manual de operación

Tarea	
Número de Tarea: 25	No. Historia: 10
Nombre Tarea: Diseño de archivo del manual de operación	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Fecha Inicio: 31/05/2004	Fecha Fin: 04/06/2004
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: Se diseñará los archivos, en formato HTML, para dar soporte al manual de operación del sistema, describiendo cada una de las opciones que se presentan en el sistema.	

Tabla 2.44: Tarea 25. Diseño de archivo del manual de operación

El manual de operación presentará al usuario una serie de datos donde se explica claramente como funciona el sistema, así como la forma en que se deben realizar los procesos para obtener mejores resultados.

Este manual estará compuesto por diferentes apartados, donde se especifican cada una de las funcionalidades del sistema.

El formato de estos archivos será HTML, por lo cual se utilizará *Dreamweaver* para su desarrollo. Una vez terminados estos archivos serán convertidos en un archivo de ayuda por medio de *HTML Help Workshop*.

El primer archivo contendrá la información general del sistema, por ejemplo, se mencionará el nombre de la organización, Secretaría de Desarrollo Social así como la Subsecretaría de Planeación y la Subdirección de Informática. Además se incluirá el nombre del sistema y una descripción breve del mismo. Posteriormente se incluirá un índice por medio

2.12. Sexta iteración

del cual es usuario elegirá la opción de su agrado.

Este índice tendrá la estructura siguiente:

- Iniciar la aplicación
- Descripción del Menú Principal
- Descripción de los Controles de Manipulación de Datos
- Descripción de Barra de Herramientas

En el primer punto se explicará al usuario la manera en la cual se debe iniciar el sistema, ejemplificando cada uno de los puntos con imágenes correspondientes.

En el segundo punto se explicarán la manera en que se debe utilizar el menú principal del sistema, está compuesta por varias opciones en cada una de las cuales se explica el funcionamiento a fondo de cada una. En la Descripción de los Controles de Manipulación de Datos se presentarán al usuario el manejo de los botones que se encuentran en cada una de las ventanas de captura. Y por último se explicará el funcionamiento de cada uno de los botones de la barra de herramientas.

2.12.4. Tarea 26. Integración del archivo al sistema

Una vez terminados y compilados los archivos, se procede a la incorporación de los mismos a la ventana principal del sistema. Para ello se agregará la opción *Manual de operación* al menú *Ayuda* de la barra de menú principal. Fig.2.28

2. IMPLEMENTACIÓN DE LA METODOLOGÍA

Tarea	
Número de Tarea: 26	No. Historia: 10
Nombre Tarea: Integración del archivo al sistema	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.6
Fecha Inicio: 07/06/2004	Fecha Fin: 09/06/2004
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: Se integrará el archivo del manual de operación al sistema, por medio de los métodos necesarios para que pueda ser visualizado desde la ventana principal del sistema.	

Tabla 2.45: Tarea 26. Integración del archivo al sistema

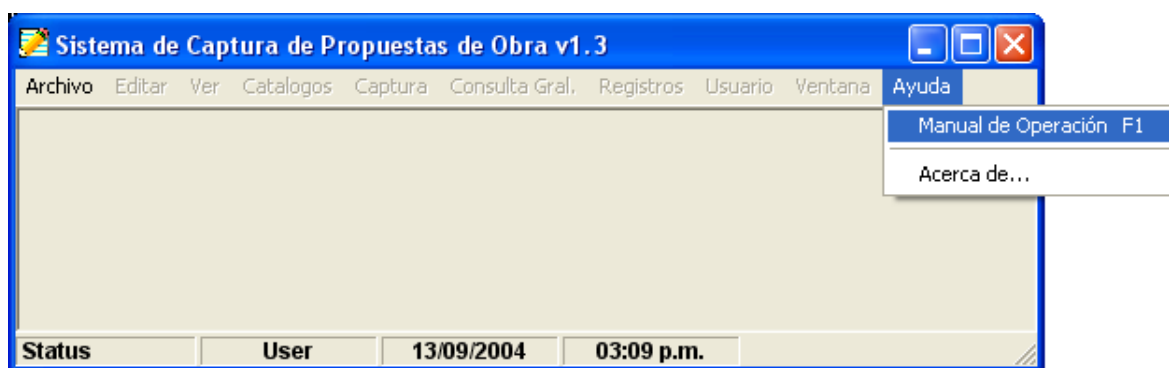


Figura 2.28: Menú *Ayuda*, *Manual de operación*.

Una vez seleccionada la opción el sistema ejecutará la llamada a una API encargada de poner en ejecución el explorador. Una vez hecha esta llamada, se deben pasar los parámetros necesarios para visualizar el proyecto creado anteriormente. Algunos de estos datos son el nombre del archivo a abrir, el tipo de ventana y el tipo de presentación.

En la Fig. 2.29 se muestra la página inicial del *Manual de Operación*.

La última historia de usuario de esta iteración es la llamada *Bloqueo del sistema* la cual se divide en la siguiente tarea:

2.12. Sexta iteración

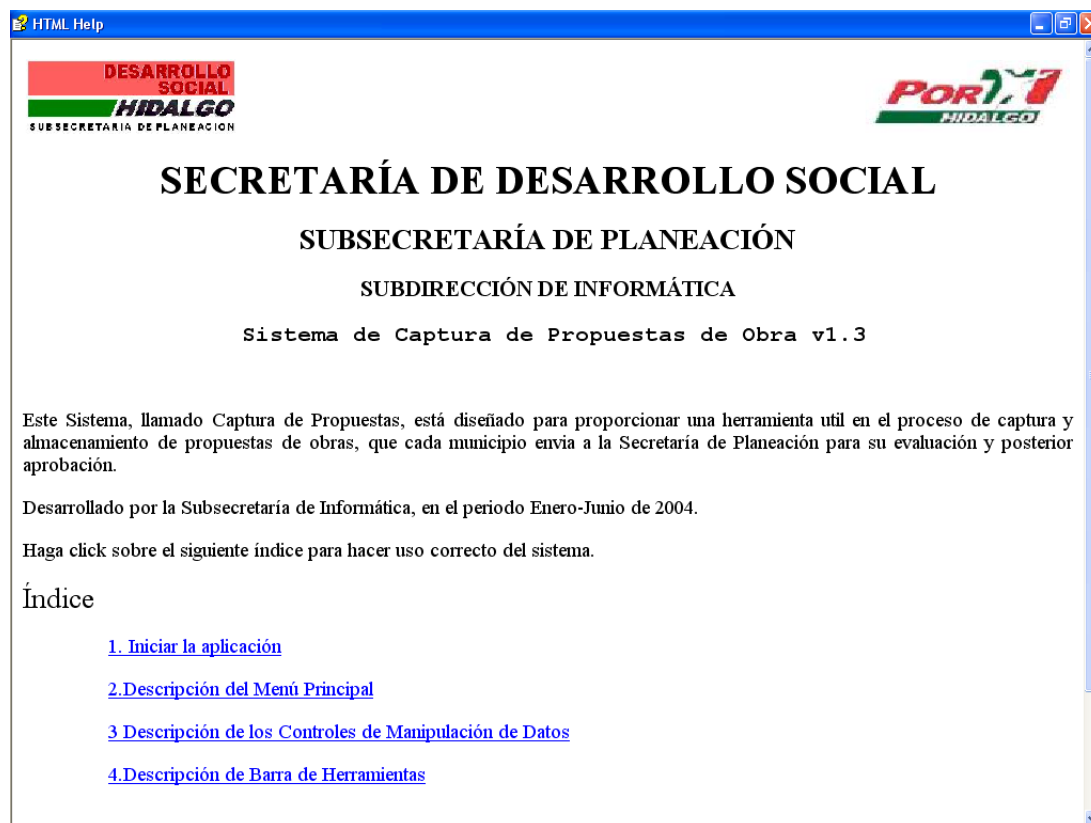


Figura 2.29: Manual de Operación.

- Bloqueo del sistema

2.12.5. Tarea 27. Bloqueo del sistema

Tarea	
Número de Tarea: 27	No. Historia: 11
Nombre Tarea: Bloqueo del sistema	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.4
Fecha Inicio: 10/06/2004	Fecha Fin: 11/06/2004
Programador responsable: Enrique Martínez, Filiberto Pérez	
Descripción: Se desarrollarán los métodos necesarios para permitir la opción de bloque del sistema completa, para evitar el uso no autorizado del mismo.	

Tabla 2.46: Tarea 27. Bloqueo del sistema

Dado que esta *historia de usuario* es muy importante para el *cliente*, se decidió tratarla por separado, y aunque en términos de desarrollo no implica gran trabajo, fue considerada de forma independiente para mostrarle al *cliente* que su implementación es de suma importancia para el manejo de la información.

Para el desarrollo de esta opción será necesario ubicarla en el menú principal para su fácil acceso, por lo anterior se agregara la opción *Bloquear* al menú *Archivo* como se muestra en la Fig. 2.30

Al seleccionar la opción *Bloquear* el sistema minimizará la ventana principal y mostrará la pantalla de identificación de usuario. Fig. 2.31

Para regresar a la ventana principal se deberá proporcionar el usuario y contraseña correcta, por lo cual en este caso esta ventana no podrá ser cerrada y lo único que se

2.12. Sexta iteración

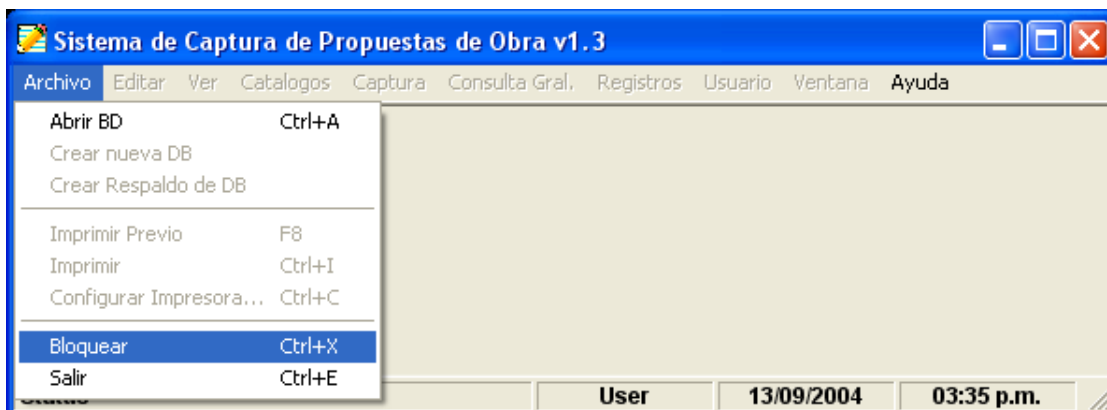


Figura 2.30: Menú *Archivo*, *Bloquear*.

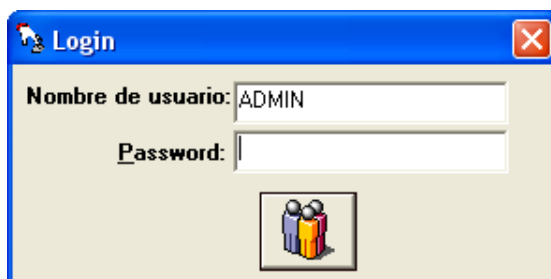


Figura 2.31: Ventana de identificación de usuarios.

podrá realizar en ella es la validación de usuario.

Por otro lado la ventana principal permanecerá minimizada y aunque el usuario pretenda maximizarla esto no será posible hasta introducir los datos correctos en la ventana de identificación de usuarios.

RESULTADOS

Resumen.

En este capítulo se presentan los resultados obtenidos al final del desarrollo de la metodología, se describen las pruebas funcionales, el proceso de empaquetado y la instalación del sistema.

Objetivo del Capítulo.

- Describir el empaquetamiento del proyecto del sistema
- Describir el proceso de instalación del sistema
- Realizar las pruebas funcionales

3.1. Fase IV Producción

Una vez concluidas las iteraciones del proyecto, en las cuales se especifica claramente la forma en que se llevaron a cabo, el siguiente paso es el de verificar que cada una de las *historias de usuario* cumplan con las necesidades del usuario.

Aunque cada una de ellas ha sido probada de forma unitaria, toca el turno de integrar el sistema para realizar las pruebas funcionales del sistema.

Para la integración del sistema el equipo de desarrollo deberá ensamblar todos los elementos del sistema, compilarlos y ponerlos en ejecución para su verificación.

3.1.1. Paquete de instalación

Como ya se ha mencionado antes, el sistema fue desarrollado bajo el lenguaje de programación *Visual Basic 6.0* por lo que por medio de éste, se procede a compilarlo. Para ello es necesario generar el archivo de la aplicación *.exe*, esto se logra por medio del menú *Archivo*, en la opción *Generar Sistema de Captura de Propuestas de Obra v1.3.exe...* del menú principal de la ventana de trabajo de *Visual Basic 6.0*. Fig. 3.1

Creado el archivo *.exe* se procede a generar el archivo de instalación del sistema, con esto aseguramos que el código se quede en el lugar de trabajo del grupo de desarrollo, sin necesidad de transportar todo el código y el ambiente de trabajo a la computadora de pruebas.

Para esto se hace uso de las *Herramientas de Microsoft Visual Studio*, en este caso se

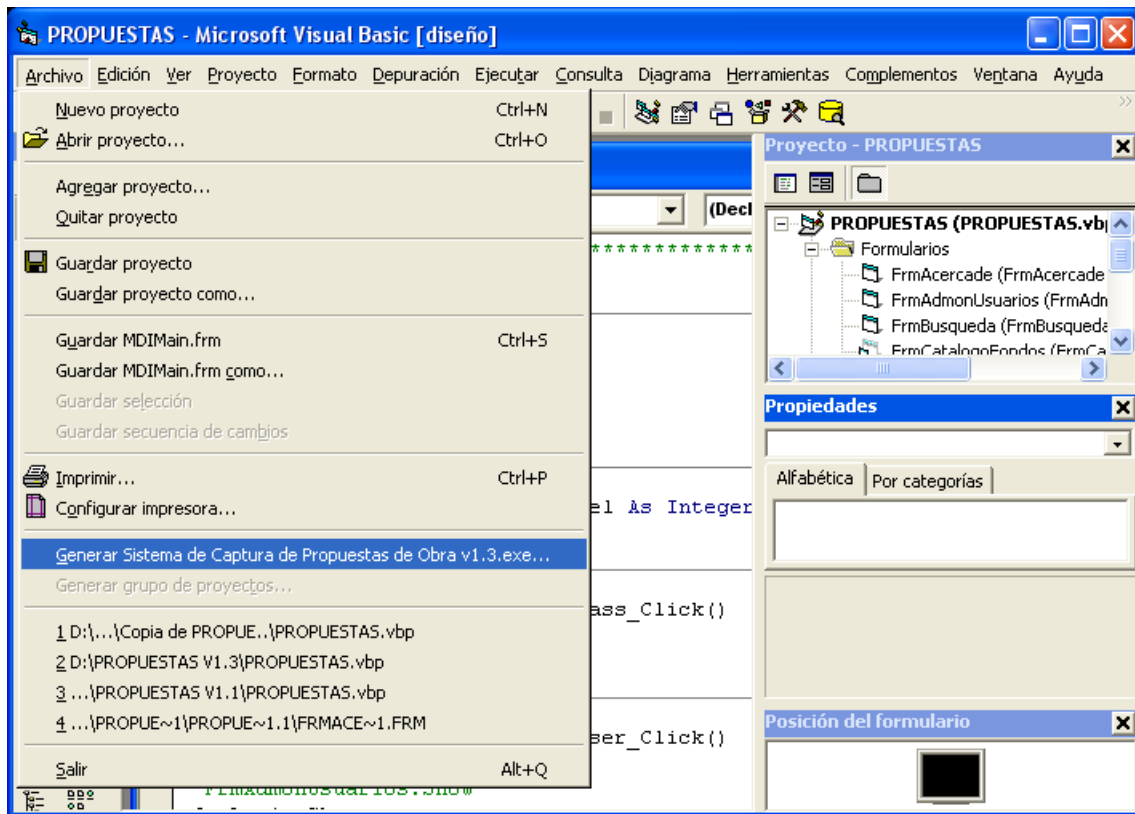


Figura 3.1: Ventana de desarrollo de *Visual Basic 6.0*.

3.1. Fase IV Producción

usará el *Asistente para empaquetado y distribución*, el cual se encuentra en el menú *Inicio / Todos los programas / Microsoft Visual Studio 6.0 / Herramientas de Microsoft Visual Studio / Asistente para empaquetado y distribución*.

En la Fig. 3.2 se muestra el *Asistente para empaquetado y distribución*.

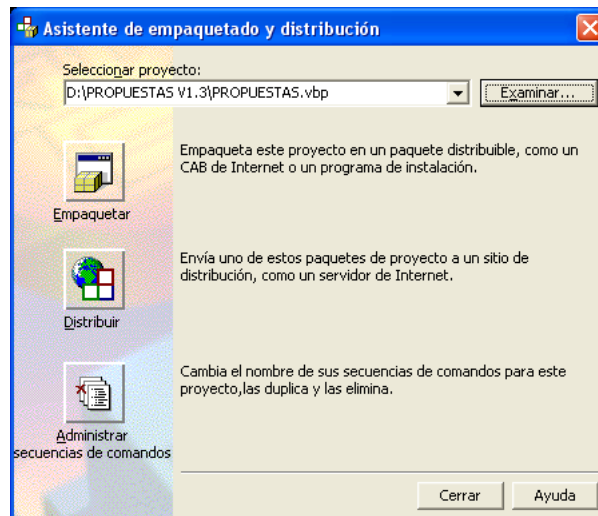


Figura 3.2: Asistente para empaquetado y distribución.

En esta ventana se deberá elegir la ruta donde se encuentra el proyecto, este proyecto fue creado para poder desarrollar el sistema, para esto se presiona el botón *Examinar*, una vez elegido el proyecto a empaquetar, se procede a elegir una de las tres opciones del asistente, se deberá elegir el botón *Empaquetar* para empezar el procedimiento.

Una vez presionado el botón *Empaquetar*, el asistente comenzará el proceso de integración de la información. Posteriormente preguntará sobre la secuencia de comandos que se desea utilizar para el empaquetado, para esto el asistente visualizará la siguiente ventana. Fig. 3.3 En la cual se deberá elegir de la lista desplegable la opción *Standard Setup Package 1*, y presionar el botón *Siguiente*.

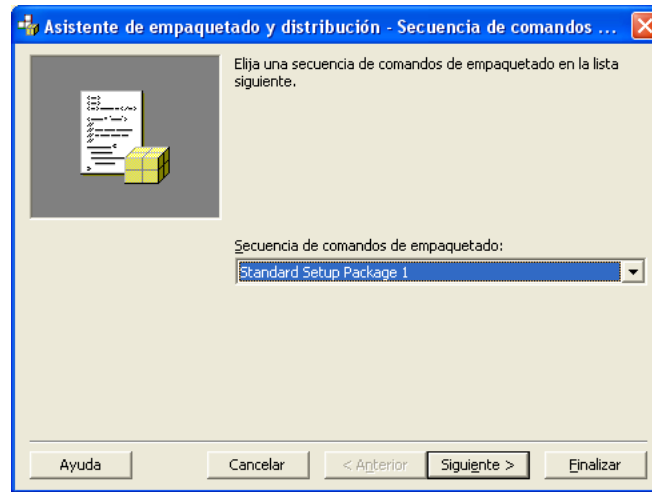


Figura 3.3: Asistente para empaquetado y distribución, Secuencia de comandos.

A continuación el asistente preguntará el tipo de paquete que se desea crear, se deberá elegir la opción *Paquete de Instalación Estándar*, el cual se usa para crear un paquete que se instalará con un programa *.exe*, por ejemplo *instalar.exe*. Fig. 3.4 Posteriormente se presiona el botón *Siguiente*.



Figura 3.4: Asistente para empaquetado y distribución, Tipo de empaquetado.

3.1. Fase IV Producción

Ahora el asistente preguntará la ubicación en la cual se creará el paquete de instalación, Fig. 3.5 Se deberá elegir la ubicación correcta, y proceder con el asistente, para seguir se deberá presionar el botón *Siguiente*. También se puede crear una carpeta nueva para evitar que los datos se sobrescriban o se pierdan.

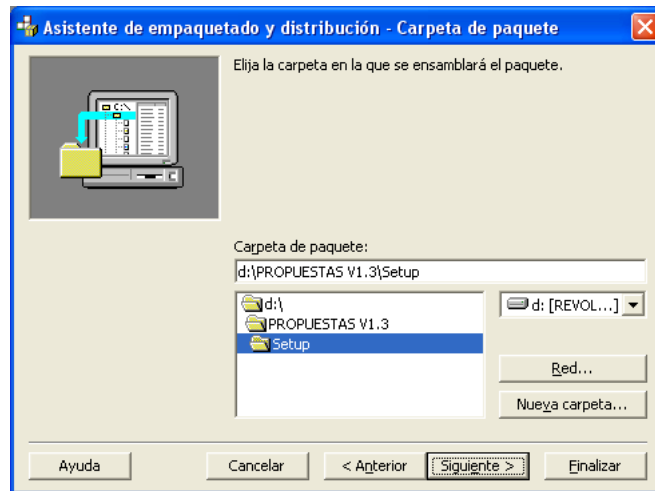


Figura 3.5: Asistente para empaquetado y distribución, Carpeta de paquete.

A continuación el asistente le pedirá que elija los controladores necesarios para la ejecución del sistema, estos controladores permitirán el acceso a los datos del sistema, una vez elegidos los controladores se procede con el proceso presionando el botón *Siguiente*. Fig. 3.6

El asistente agregará los archivos necesarios para el paquete de instalación, posteriormente el asistente preguntará si se desea realizar varios archivos *.cab*, para guardarlos en discos flexibles, o si se desea realizar uno solo, en este caso se deberá elegir la opción *Un único archivo .cab*. Fig. 3.7

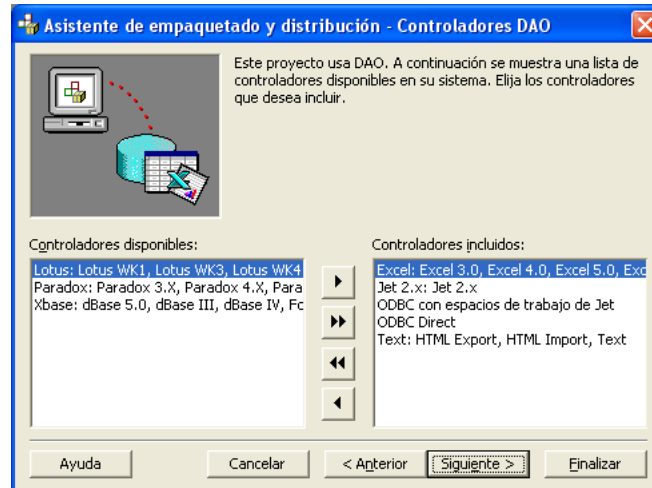


Figura 3.6: Asistente para empaquetado y distribución, Controladores DAO.

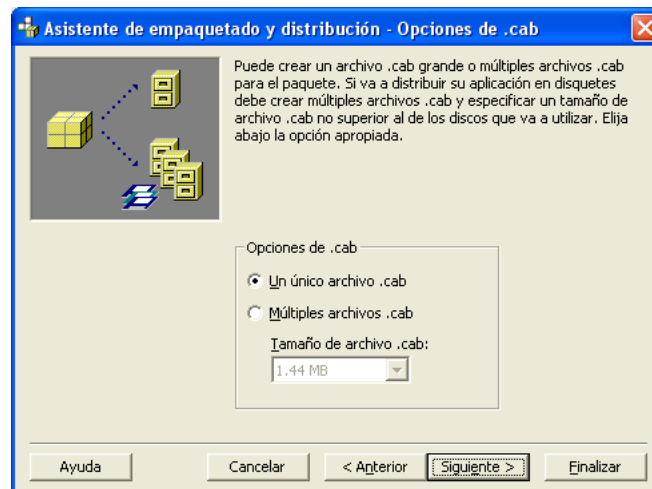


Figura 3.7: Asistente para empaquetado y distribución, Opciones de .cab.

3.1. Fase IV Producción

Como paso siguiente, se deberá establecer el nombre para el paquete a crear, en este caso será, *Sistema de Captura de Propuestas de Obra v1.3*. Fig. 3.8 Posteriormente se procede presionando el botón *Siguiente*.



Figura 3.8: Asistente para empaquetado y distribución, Título de instalación.

La siguiente ventana del asistente, permitirá especificar la ubicación el la cual se encontrará la carpeta de instalación dentro del menú *Inicio*, por lo que si es necesario puede ser modificada según los requerimientos del usuario. Fig. 3.9

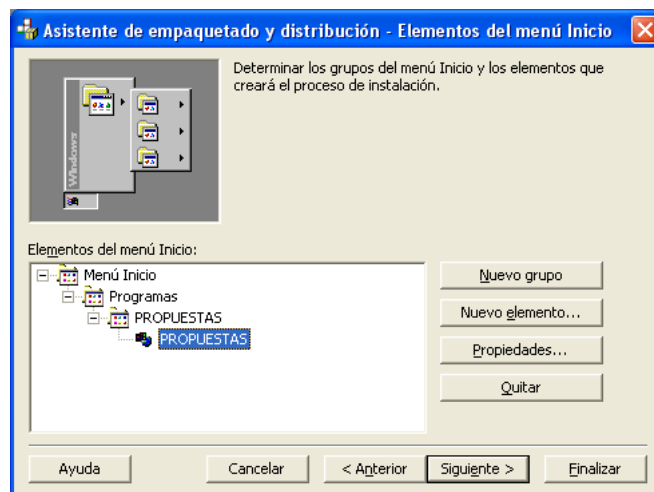


Figura 3.9: Asistente para empaquetado y distribución, Elementos del menú inicio.

En las ventanas siguientes no es necesario modificar ningún parámetro por lo que se procede presionando el botón *Siguiente*.

Como último paso para la creación del paquete de instalación el asistente notificará que ha concluido con la recopilación de información para el paquete y que esta listo para crearlo, ahora se presiona el botón *Finalizar*. Fig. 3.10



Figura 3.10: Asistente para empaquetado y distribución, Finalizado.

Así el asistente iniciará el proceso de creación del archivo *.cab*, al finalizar el proceso el asistente mostrará un informe de empaquetado, donde se mostrarán los detalles del proceso. Fig. 3.11

3.1.2. Instalación del sistema

Una vez creado el paquete de instalación, se procede a la instalación del sistema para realizar la pruebas funcionales, para esto es necesario ejecutar el archivo *setup.exe*, en la ubicación donde se creó el paquete.

3.1. Fase IV Producción

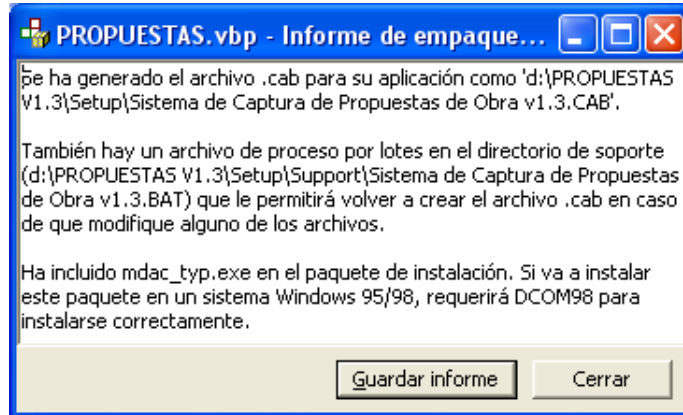


Figura 3.11: Informe de empaquetado.

Al ejecutar el archivo *setup.exe*, se visualizará la ventana de la Fig. 3.12 en la cual el asistente comenzará con la copia de archivos necesario para el sistema.

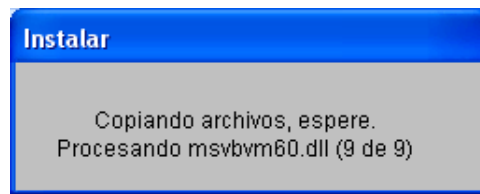


Figura 3.12: Inicio de instalación.

Al término de este proceso, el programa le da la bienvenida al programa de instalación y le pedirá que cierre todas las ventanas, para que pueda continuar de la instalación, Fig. 3.13 Se deberá presionar el botón *Aceptar*, para seguir con la instalación.

Ahora el asistente comenzará con la instalación el sistema, para ello es necesario presionar el botón de instalación, si es necesario se puede cambiar el directorio de instalación o cancelar la misma. Fig. 3.14

El programa de instalación agregará los elementos al grupo mostrado en la Fig. 3.15

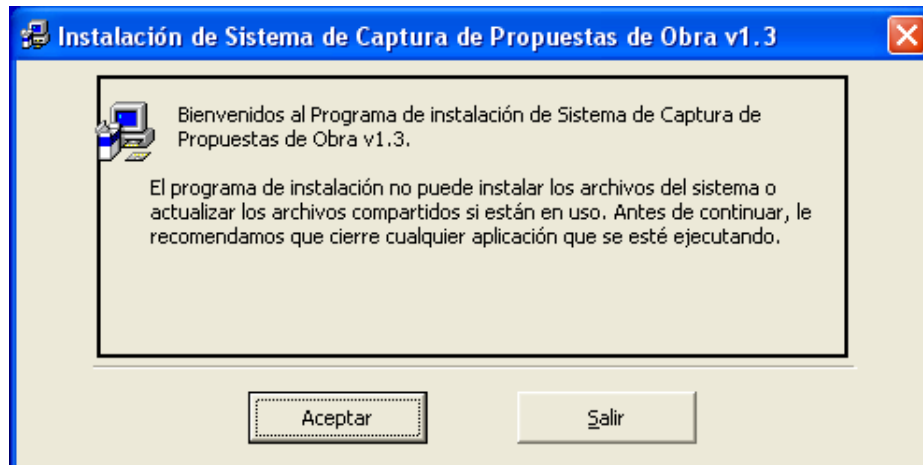


Figura 3.13: Instalación de Sistema de Captura de Propuestas v1.3.

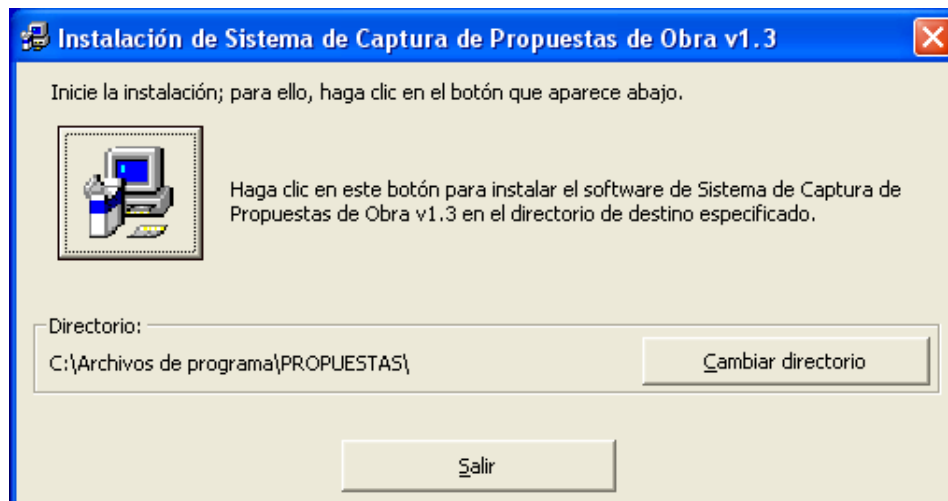


Figura 3.14: Instalación, carpeta de instalación.

3.1. Fase IV Producción

este nombre puede ser editado si es necesario.

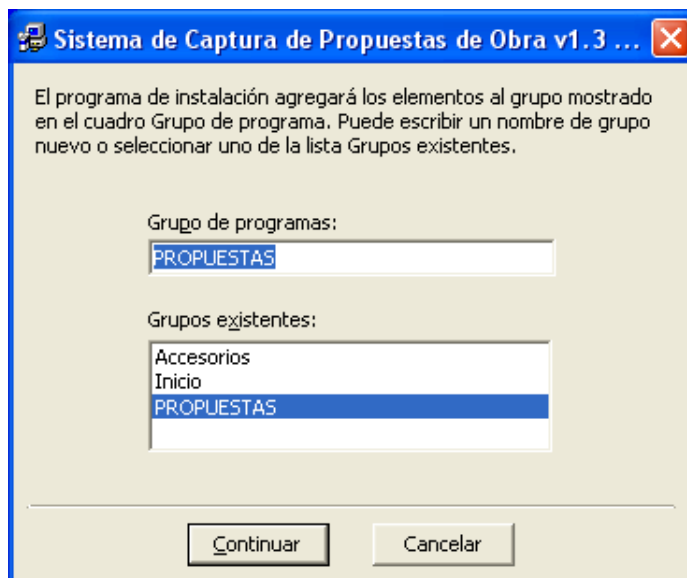


Figura 3.15: Instalación, nombre de grupo.

Al presionar el botón *Continuar*, el programa de instalación comenzará con la copia de los archivos necesarios para el sistema. Fig. 3.16

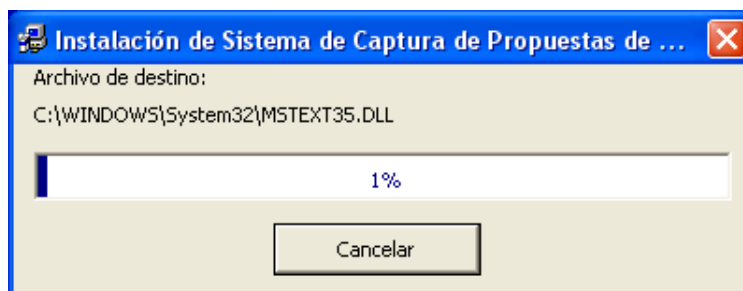


Figura 3.16: Instalación, porcentaje de avance.

Al finalizar la copia de archivos, el programa de instalación enviará un mensaje, en el cual se informa de la finalización de la instalación del sistema Fig. 3.17



Figura 3.17: Finalización de instalación.

3.1.3. Pruebas funcionales

Una vez instalado el sistema, se procede a realizar las pruebas funcionales de cada una de las *historias de usuario* desarrolladas.

Estas pruebas funcionales permiten verificar que cada una de las funcionalidades del sistema cumplen con los requerimientos del cliente, por lo que se realizan pruebas ingresando datos correctos e incorrectos para observar el comportamiento del sistema ante estas situaciones.

Las pruebas funcionales realizadas son vistas en el Apéndice C.

3.2. Fase V Mantenimiento

Una vez realizadas las pruebas funcionales, se procede a la entrega del sistema, para esto se realizó una cita con el *cliente*, con la fecha establecida para la capacitación del personal, ver Apéndice A.

Durante el periodo de la capacitación se presentó un *error de programación*, este se obtuvo al elegir la opción *Archivo / Configurar impresora...*, la cual no se ejecuto como debería,

3.3. Fase VI Muerte del proyecto

cabe mencionar que este tipo de opciones fueron integradas al sistemas como partes de código preestablecido, esto es, ya existían estas rutinas en sistemas desarrollados anteriormente, por lo que no fueron tomadas en cuenta por los requerimientos del *cliente* (*historias de usuario*).

Esta corrección al sistema fue clasificada dentro de la fase de mantenimiento porque el sistema ya había sido entregado al *cliente* y se habían hecho las pruebas funcionales, una vez corregido el error, se procede a la instalación del sistema funcionando al 100 %.

3.3. Fase VI Muerte del proyecto

Es así como el *Sistema de Captura de Propuestas de Obra*, llega a su fase final, dado que el cliente no tiene más *historias de usuario* que agregar al sistema.

Por otra parte, dado que se desarrollo un manual de operación, este mismo servirá como documentación, y en el código fuente fueron agregados pequeños comentarios sobre las instrucciones que se utilizaron para desarrollar cada una de las funcionalidades del sistema.

CONCLUSIONES

Como se mencionó anteriormente la tarea principal de la Dirección Técnica de la Secretaría de Desarrollo del Gobierno del Estado, es la de recibir, organizar y almacenar las obras propuestas que los municipios del estado envían con la finalidad de obtener su aprobación y financiamiento.

Al implementar el Sistema de Captura de Propuestas, el tiempo de captura de estas propuestas se reduce de manera considerable, ya que ahora solo es necesario entrar al sistema, seleccionar la ventana correspondiente y seleccionar con el ratón los datos que se repiten en cada expediente técnico, como son *Nombre de la región*, *Nombre del municipio*, *Nombre de la localidad* y *Fondo de aplicación*.

Otro punto a considerar es que al tener almacenada la información de manera organizada, los procesos de búsqueda y selección son mucho más eficaces que si se realizaran de manera manual. Es por esto que con el Sistema de Captura de Propuestas, estos procesos se realizan de manera más rápida y eficaz.

Es de suponerse que al realizar los procesos de búsqueda y filtrado, es necesario pre-

sentar la información para ser evaluada por el usuario del sistema, es por esto que el sistema permite la opción de generar reportes, siendo así posible presentar la información de manera impresa para su correspondiente evaluación y posible modificación.

También es importante resaltar que con la implantación del Sistema de Captura de Propuestas de Obras, la integridad de la información esta contemplada de manera significativa, ya que el sistema no puede ser usado por personal no autorizado, así solo el personal con contraseña puede acceder a el y manipular la información.

El incluir un *Manual de operación* permitirá a la Dirección Técnica reducir el tiempo de capacitación en el uso del sistema, ya que por lo general este sistema será operado por los prestadores de servicio social que se encuentran en el departamento.

Por otra parte, la metodología empleada para el desarrollo del sistema permitió al grupo de desarrollo trabajar de manera organizada, ya que no se contaba con una metodología para el desarrollo de sistemas dentro de la Subdirección de Informática.

Esta metodología fue en general fue fácil de llevar ya que la mayoría de sus actividades no son nuevas, es decir ya habían sido implementadas por otras metodologías, lo que permitió aplicarlas de manera rápida y sencilla.

Una de las actividades que representaban un reto para el grupo de desarrollo fue la denominada, *Programación en parejas*, ya que representaba mantener a dos programadores sentados frente a la misma computadora, lo cual no significaba que los resultados fueran los mejores, ya que el que se encuentren sentados provocaría algunas discusiones sobre

cuales serian las mejores rutinas para el desarrollo del sistema.

Aunque en un principio fue difícil el entendimiento, por la diferencia de nivel de programación de los desarrolladores, al final los resultados obtenidos cumplieron con los requerimientos del cliente, dejando atrás cualquier ánimo de discordia.

Otra práctica de la metodología que no pudo ser llevada a cabo como se establece fue la denominada *40 horas por semana*, ya que como se mencionó antes, el sistema fue desarrollado durante el periodo de préstamo de Servicio Social, lo cual impidió que se cumplieran las ocho hora diarias establecidas, cubriendo solo un total de cuatro horas diarias.

La flexibilidad de la metodología para aceptar los cambios constantes de los requerimientos del cliente, fue otra de las ventajas que permitió un gran avance en el desarrollo del sistema ya que no era necesario regresarse a determinada fase de la metodología, como lo harían las metodologías tradicionales, para hacer cambios en la estructura del sistema.

Con esto se puede concluir que la implementación del Sistema para la Administración de la información en la Dirección Técnica de la Secretaria de Desarrollo Social desempeñar una de sus actividades de una manera más eficaz y con un mínimo de tiempo.

Las funcionalidades que presenta el sistema permiten a la secretaria la recolección, manipulación y procesamiento de la información de manera rápida, evitando así el papeleo excesivo y la duplicidad de datos, cumpliendo con los objetivos generales y específicos de este trabajo.

TRABAJOS FUTUROS

Al término de la implementación del sistema, se encontraron varias características que podrían ser agregadas en un futuro.

Una de ellas es la posibilidad de que en cada una de las presidencias municipales del estado de Hidalgo se encuentre el sistema y que al término de la captura y evaluación de las propuestas, la base de datos se envíe a la Dirección Técnica para que sean analizadas y almacenadas por el departamento correspondiente con el fin de reducir al máximo la cantidad de papel que se genera de los expedientes técnicos.

Otra característica posible sería la conexión de cada uno de los servidores de las diferentes presidencias municipales con el servidor de la Secretaría de Desarrollo Social, para que directamente se capturen las propuestas desde cada municipio. Almacenando las bases de datos en el servidor de la Secretaría.

Esta se podría realizar por medio de una conexión remota, utilizando una línea telefónica realizando una conexión punto a punto. Otra conexión se podría realizar por medio de Internet, lo cual significaría adecuar el sistema para que sea manipulado desde una página

en Internet almacenada en el servidor de la Secretaría de Desarrollo Social

Planificación de desarrollo

En el diagrama de Gantt siguiente se describen las tareas que serán llevadas a cabo por el grupo de desarrollo. Como se puede observar, las actividades principales se refieren a las *historias de usuario* pactadas, y como dependientes de ellas, las tareas en las cuales se decidió dividir las.

Se muestra también el tiempo de ejecución de cada una de ellas, indicando la fecha de inicio y término así como el número de días necesarios para llevarla a cabo.

Así mismo, se presenta el costo calculado para cada una de las fases del proyecto, tomando como unidad de medida el término *hora / hombre*, que representa el número de horas por persona necesarias para desarrollar determinada actividad. Para conseguir esto se realizó una estimación del salario de cada uno de los integrantes del proyecto, esto se puede observar en la tabla A.1 .

Con lo anterior se calcula el costo total del proyecto que asciende a \$ 19,830.00, con un total de 852 horas / hombre.

Puesto	Salario
Manager	\$30.00/hr
Programmer Sr	\$25.00/hr
Programmer Jr	\$20.00/hr
Tester	\$15.00/hr

Tabla A.1: Estimación de salarios

Id	Nombre de tarea	Duración	Comienzo	Fin	Costo	ene '04							01 feb '04							08 feb '0														
						L	M	M	J	V	S	S	D	L	M	M	J	V	S	S	D	L	M											
30	Abrir BD	2 días	lu 03/05/04	ma 04/05/04	\$360.00																													
31	Crear nueva BD	2 días	mi 05/05/04	ju 06/05/04	\$360.00																													
32	Crear respaldo de BD	1 día	vi 07/05/04	vi 07/05/04	\$180.00																													
33	Reporte general de Propuestas	8 días	lu 10/05/04	mi 19/05/04	\$1,440.00																													
34	Diseño del formato de reporte general	4 días	lu 10/05/04	ju 13/05/04	\$720.00																													
35	Integración del formato al sistema	4 días	vi 14/05/04	mi 19/05/04	\$720.00																													
36	Reporte de Propuesta	7 días	ju 20/05/04	vi 28/05/04	\$1,260.00																													
37	Diseño del formato de reporte	4 días	ju 20/05/04	ma 25/05/04	\$720.00																													
38	Integración del formato al sistema	3 días	mi 26/05/04	vi 28/05/04	\$540.00																													
39	Presentación de Manual de operación	8 días	lu 31/05/04	mi 09/06/04	\$1,440.00																													
40	Diseño del archivo del manual de operación	5 días	lu 31/05/04	vi 04/06/04	\$900.00																													
41	Integración del archivo al sistema	3 días	lu 07/06/04	mi 09/06/04	\$540.00																													
42	Bloqueo del Sistema	2 días	ju 10/06/04	vi 11/06/04	\$360.00																													
43	Bloqueo del Sistema	2 días	ju 10/06/04	vi 11/06/04	\$360.00																													
44	Fase IV Producción	2 días	lu 14/06/04	ma 15/06/04	\$600.00																													
45	Paquete de Instalación	1 día	lu 14/06/04	lu 14/06/04	\$300.00																													
46	Instalación del sistema	1 día	ma 15/06/04	ma 15/06/04	\$300.00																													
47	Pruebas generales	3 días	mi 16/06/04	vi 18/06/04	\$180.00																													
48	Capacitación del personal	5 días	lu 21/06/04	vi 25/06/04	\$600.00																													
49	Entrega final del sistema	1 día	lu 28/06/04	lu 28/06/04	\$120.00																													

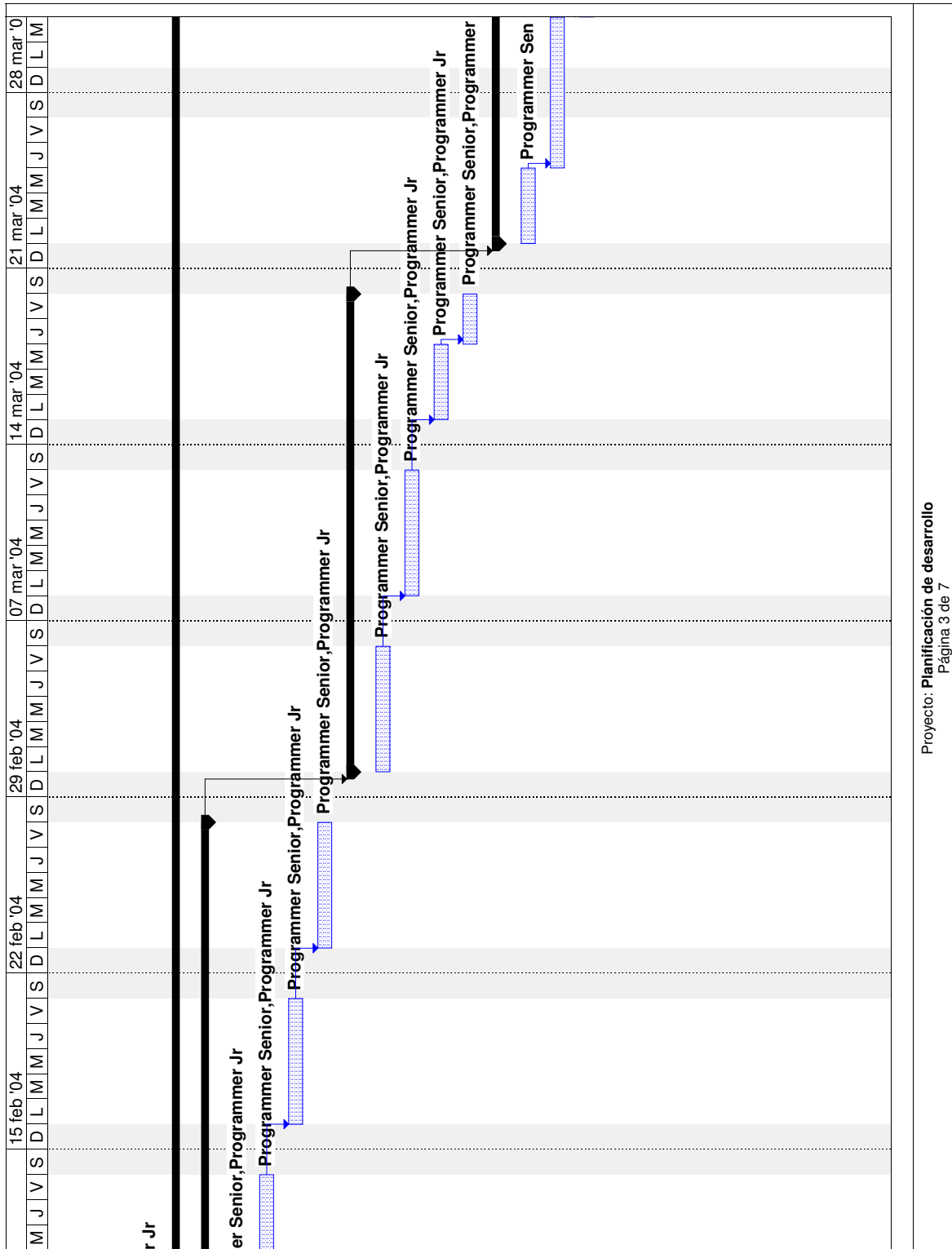
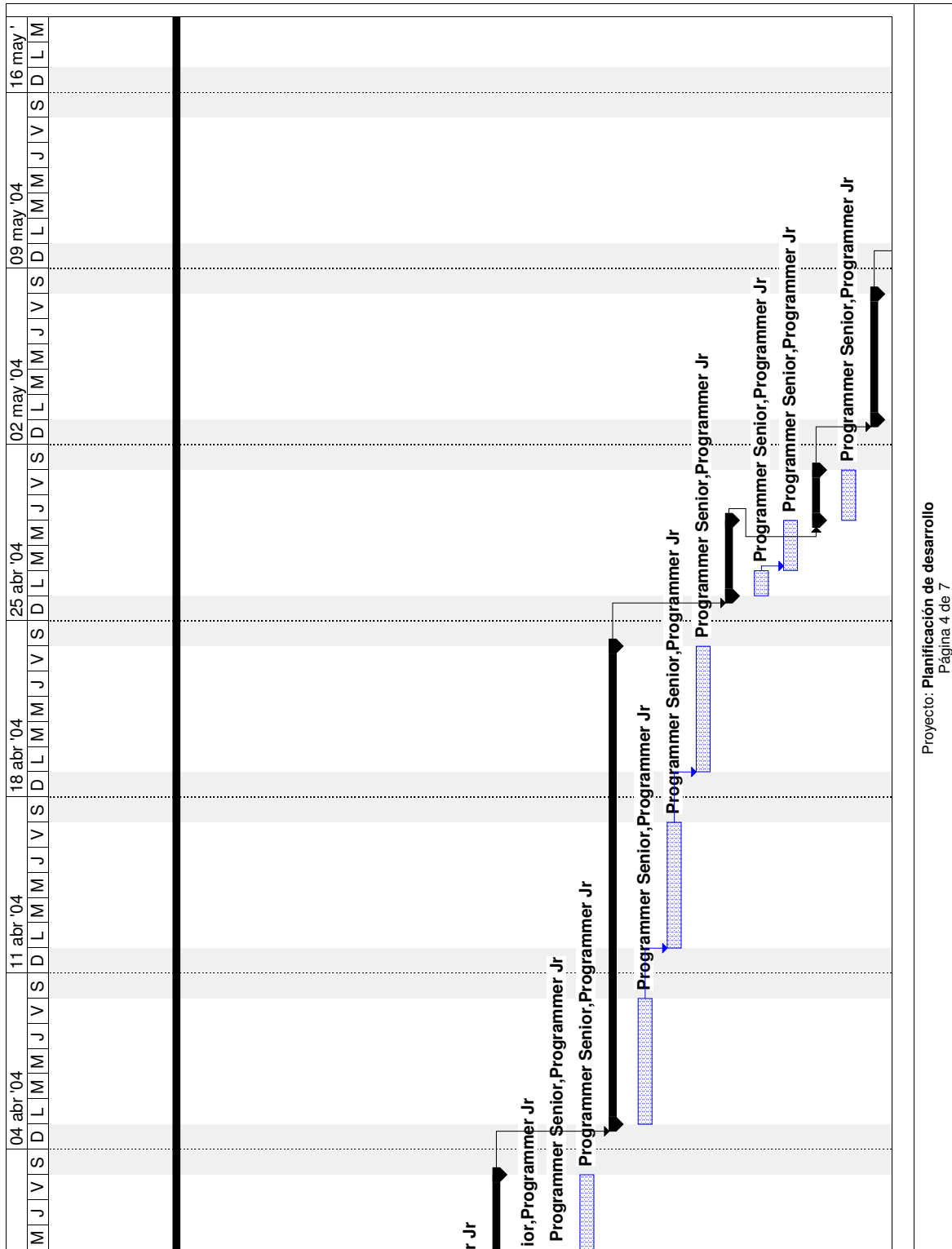


Figura A.3: Planificación de desarrollo del sistema, Hoja 3 de 7.



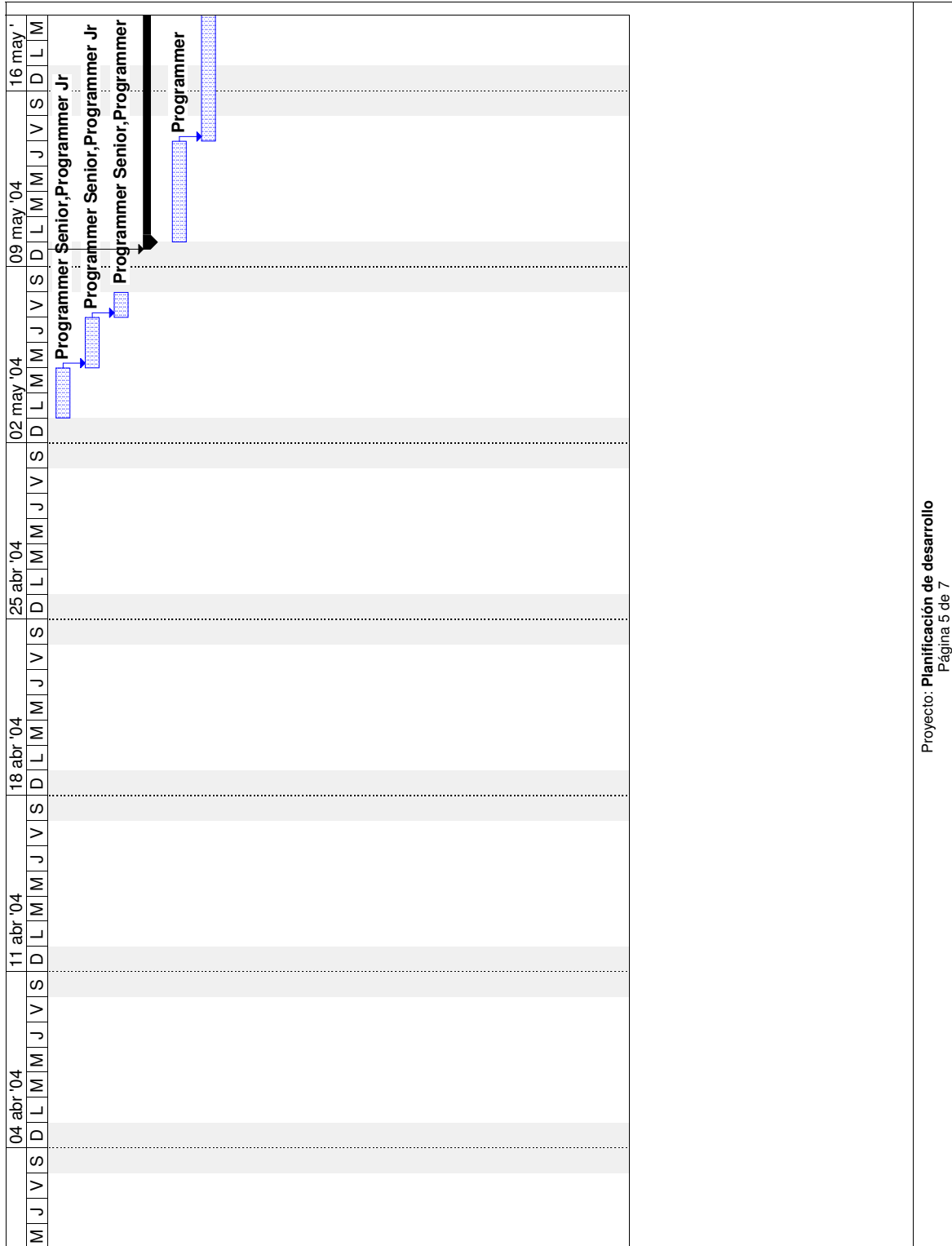
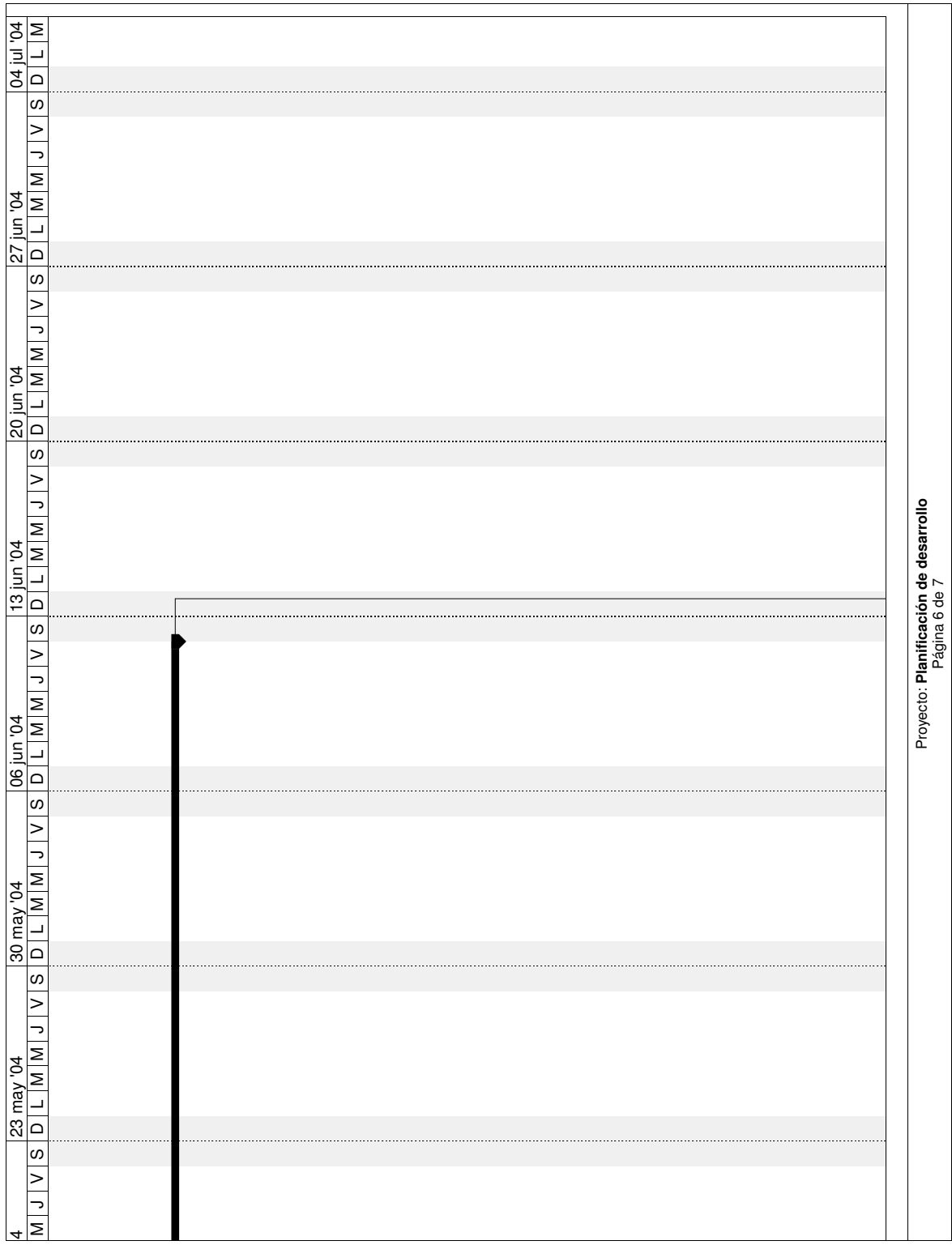


Figura A.5: Planificación de desarrollo del sistema, Hoja 5 de 7.



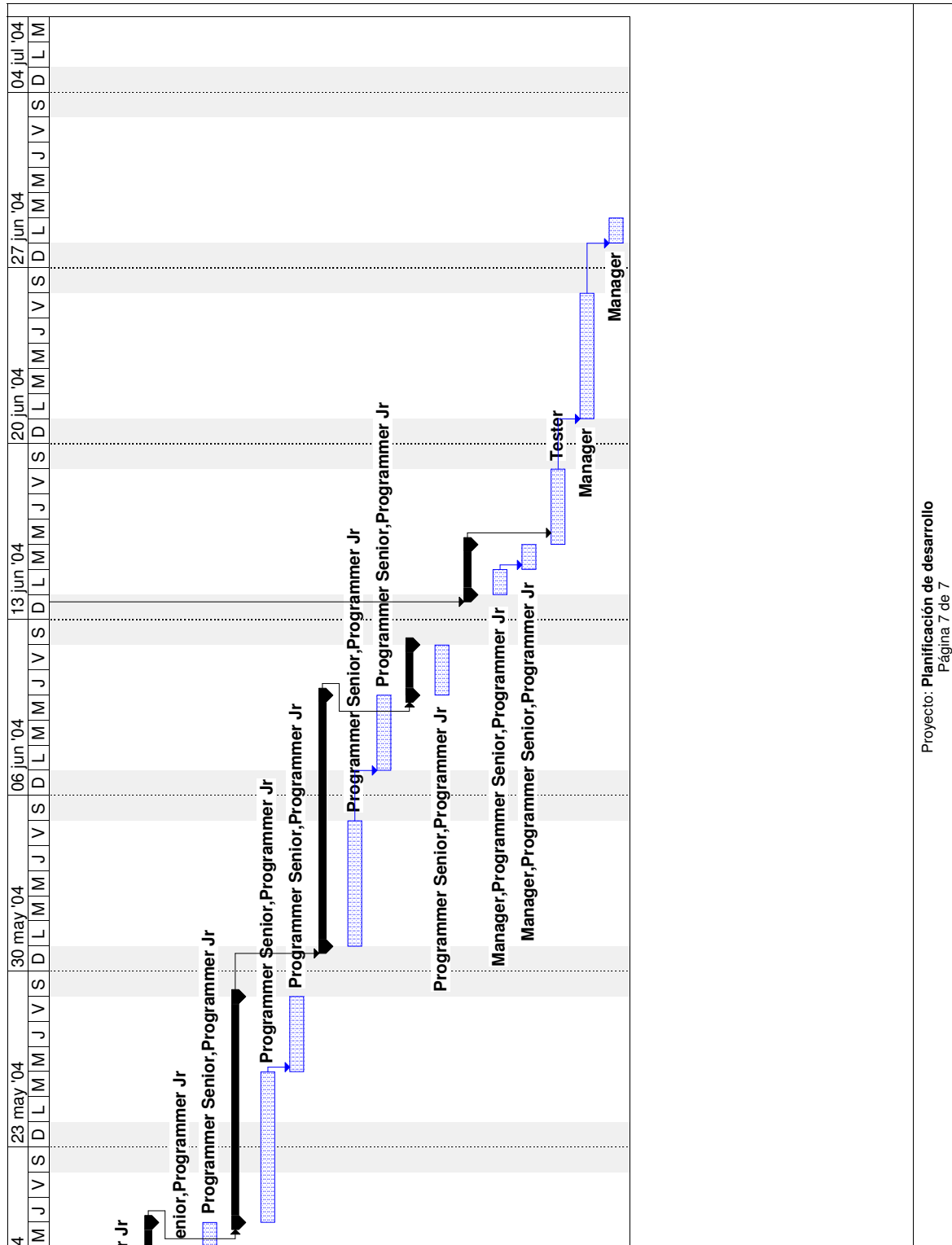


Figura A.7: Planificación de desarrollo del sistema, Hoja 7 de 7.

Apéndice B

Código

En la ventana de captura de propuestas de obra se pueden observar cada uno de los botones que permiten la manipulación de la información en la tabla principal de la base de datos. Estos botones forman un vector de objetos, con lo cual cada uno de ellos tiene un número de identificación dentro del vector.

Es por esto que la parte del código considerada de mayor relevancia es la función *CmndAcciones_Click(Index As Integer)* porque permite controlar todos los procesos necesarios para agregar, editar, actualizar, eliminar los datos de cada una de las tablas que se utilizan en este sistema. Como se puede observar, esta función recibe un parámetro el cual indica cual ha sido el botón que se ha presionado, con esto se selecciona el proceso a seguir.

```
Private Sub CmndAcciones_Click(Index As Integer)
    Dim Respuesta as Integer
    Dim IntContador As Integer
    Dim DBComando As Database
    Select Case Index
        Case 0
            On Error GoTo Salir
            BlnModificarDatos = True
            IntNuevooEditar = 0
            Call HabilitarBotones(False)
            Call BloquearInfo(False)
            AdodcPropuestas.Recordset.AddNew
            AdodcPropuestas.Enabled = False
            DbcboRegion.SetFocus
        Case 1
```

```

On Error GoTo Salir
BlnModificarDatos = False
IntNuevooEditar = 1
Call HabilitarBotones(False)
Call BloquearInfo(False)
AdodcPropuestas.Enabled = False
DbcboRegion.SetFocus
Case 2
On Error GoTo Salir
Respuesta = MsgBox(">Está seguro de querer eliminar el registro actual?", _
vbYesNo + vbCritical, "Eliminar registro")
Select Case Respuesta
Case 6
BlnModificarDatos = True
If AdodcPropuestas.Recordset.RecordCount > 0 Then
Set DBComando = OpenDatabase(StrRuta)
DBComando.Execute ("DELETE FROM PROPUESTAS WHERE NO_PROPUESTA=" & _
FrmPropuestas.TxtNoPropuesta.Text)
End If
DBComando.Close
AdodcPropuestas.Refresh
Case 7
Exit Sub
End Select
Case 3
On Error GoTo Salir
If AdodcPropuestas.Recordset.RecordCount > 0 Then
IntContReg = 1
Else
IntContReg = 0
End If
AdodcPropuestas.Refresh
Case 4
On Error GoTo Salir
If IntNuevooEditar = 0 Then
AdodcPropuestas.Recordset.Cancel
End If
AdodcPropuestas.Refresh
If AdodcPropuestas.Recordset.RecordCount > 0 Then
BlnModificarDatos = False
AdodcPropuestas.Refresh
Call BloquearInfo(True)
Call HabilitarBotones(True)
AdodcPropuestas.Enabled = True
Else
CmndAcciones(0).Enabled = True
For IntContador = 1 To 5
CmndAcciones(IntContador).Enabled = False
Next IntContador

```

```
        BlnModificarDatos = False
        AdodcPropuestas.Refresh
        Call BloquearInfo(True)
        AdodcPropuestas.Enabled = True
    End If
Case 5
    On Error GoTo Salir
    Select Case IntNuevoEditar
    Case 0
        For IntContador = 0 To 5
            If TxtCampos(IntContador).Text = "" Or DbcboMunicipios.BoundText = "" _
                Or DbcboRegion.BoundText = "" Or DbcboLocalidad.BoundText = "" Or _
                DbcboFondo.BoundText = "" Or DbcboPrograma.BoundText = "" Then
                MsgBox ("Registro no guardado, debe llenar todos los campos.", _
                    vbCritical, Error...)
                Exit Sub
            End If
        Next IntContador
        AdodcPropuestas.Recordset.Update
        MsgBox ("Registro guardado", vbInformation, "Nuevo Registro en BD")
    Case 1
        For IntContador = 0 To 5
            If TxtCampos(IntContador).Text = "" Or DbcboMunicipios.BoundText = "" _
                Or DbcboRegion.BoundText = "" Or DbcboLocalidad.BoundText = "" Or _
                DbcboFondo.BoundText = "" Or DbcboPrograma.BoundText = "" Then
                MsgBox ("Registro no actualizado, debe llenar todos los campos.", _
                    vbCritical, Error...)
                Exit Sub
            End If
        Next IntContador
        AdodcPropuestas.Recordset!CLAV_REG = DbcboRegion.BoundText
        AdodcPropuestas.Recordset!NO_MUN = DbcboMunicipios.BoundText
        AdodcPropuestas.Recordset!NO_LOC = DbcboLocalidad.BoundText
        AdodcPropuestas.Recordset!CLAV_FONDO = DbcboFondo.BoundText
        AdodcPropuestas.Recordset!CLAV_PROG = DbcboPrograma.BoundText
        AdodcPropuestas.Recordset.Update
        MsgBox ("Registro Actualizado", vbInformation, "Actualizando BD")
    End Select
    AdodcPropuestas.Refresh
    AdodcPropuestas.Recordset.MoveFirst
    AdodcPropuestas.Enabled = True
    Call HabilitarBotones(True)
    Call BloquearInfo(True)
    BlnModificarDatos = False
End Select
Exit Sub
Salir:
    MsgBox Err.Description
End Sub
```

Pruebas funcionales

Prueba funcional 1 Ingreso de nueva propuesta de obra

Descripción

Una vez que el expediente técnico llega en formato impreso, se revisa el contenido de la lista de propuestas para comprobar que cuenta con todos los datos necesarios para proceder a su captura. Un expediente técnico consta de un oficio de solicitud y una lista de propuestas de obra, en donde se especifican las obras que cada municipio planea realizar durante el año de gestión administrativa.

En caso de que los datos sean completos se accede a la base de datos se captura la información de acuerdo a sus campos en la tabla correspondiente y se procede a guardarla en la base de datos.

En caso de que la lista no contenga los datos necesarios para su captura, el expediente es rechazado y devuelto a la dependencia de origen para que sea editado y enviado nuevamente, en el caso de que se encuentren datos incompletos mínimos, se procede a la captura

de la lista y se avisa a la dependencia de origen para que sea integrada la información incompleta.

Ingreso correcto de nueva propuesta de obra

Descripción

El capturista, una vez ingresado al sistema y seleccionado la base de datos correcta, seleccionará la opción *Obras* del menú *Captura*. En la ventana el capturista deberá presionar el botón *Agregar* para llenar todos los campos disponibles con la información de la lista de propuestas de obra, posteriormente deberá presionar el botón *Guardar* y el sistema notificará con un mensaje el éxito de la operación.

Condición de ejecución

El capturista deberá estar dado de alta en el sistema, el nombre de usuario asignado al capturista es *USER*.

Datos de entrada

- El capturista ingresará con nombre de usuario y contraseña al sistema
- Del menú principal seleccionará *Archivo / Abrir BD*
- En la ventana de selección de *Base de datos Origen* deberá elegir la BD que desea manipular. Al seleccionar la BD el sistema notificará que la base de datos ha sido abierta correctamente
- Del menú principal seleccionará *Captura / Obras*

- En la ventana deberá presionar el botón *Agregar* e introducir todos los datos de la lista de propuestas en los campos de ingreso, a continuación deberá presionar el botón *Guardar*
- El sistema notificará el éxito de la operación con un mensaje con esto termina el proceso

Resultados

Al ingresar la nueva propuesta de obra, el sistema actualizará la tabla correspondiente en la BD y mostrará en la ventana de captura la nueva propuesta.

Evaluación de la prueba

Prueba satisfactoria.

Ingreso de nueva propuesta de obra con datos incorrectos

Descripción

El capturista, una vez ingresado al sistema y seleccionado la base de datos correcta, seleccionará la opción *Obras* del menú *Captura*. En la ventana el capturista deberá presionar el botón *Agregar* para llenar todos los campos disponibles con la información de la lista de propuestas de obra, posteriormente deberá presionar el botón *Guardar* y el sistema validará y encontrará un error en los tipos de datos que se han ingresado, por lo cual el sistema notificará con un mensaje el fracaso de la operación.

Condición de ejecución

El capturista deberá estar dado de alta en el sistema.

Datos de entrada

- El capturista ingresará con nombre de usuario y contraseña al sistema
- Del menú principal seleccionará *Archivo / Abrir BD*
- En la ventana de selección de *Base de datos Origen* deberá elegir la BD que desea manipular. Al seleccionar la BD el sistema notificará que la base de datos ha sido abierta correctamente.
- Del menú principal seleccionará *Captura / Obras*
- En la ventana deberá presionar el botón *Agregar* e introducir todos los datos de la lista de propuestas en los campos de ingreso, a continuación deberá presionar el botón *Guardar*
- El sistema validará los tipos de datos ingresados en los campos de la ventana de captura y notificará la incongruencia con un mensaje.
- El proceso se considera terminado hasta que el capturista corrija el error

Resultados

La información incorrecta proporcionada no puede ser almacenada en la BD por lo que los datos deberán ser corregidos para su almacenamiento.

Evaluación de la prueba

Prueba satisfactoria.

Prueba funcional 2 Gestión de propuesta de obra

Descripción

Existe la posibilidad de que el primer expediente técnico contenga errores o que no se haya incluido la información necesaria para su procesamiento, por lo que se envía nuevamente un expediente para que se hagan las modificaciones correspondientes.

Un segundo expediente técnico llega en formato impreso, donde se localiza la corrección de propuestas. Se accede a la base de datos se localiza la propuesta y se modifica o elimina la información de la propuesta en la tabla correspondiente.

Modificación correcta de propuesta de obra

Descripción

El capturista, una vez ingresado al sistema y seleccionado la base de datos correcta, seleccionará la opción *Obras* del menú *Captura*. En la ventana el capturista deberá desplazarse por las propuestas existentes en la tabla, hasta encontrar la que desea modificar, posteriormente deberá presionar el botón *Editar* para modificar los campos necesarios con la información de la lista de propuestas de obra, posteriormente deberá presionar el botón *Guardar* y el sistema notificará con un mensaje el éxito de la operación.

Condición de ejecución

La propuesta de obra a editar debe estar almacenada en la tabla correspondiente.

Datos de entrada

- El capturista ingresará con nombre de usuario y contraseña al sistema
- Del menú principal seleccionará *Archivo / Abrir BD*
- En la ventana de selección de *Base de datos Origen* deberá elegir la BD que desea manipular. Al seleccionar la BD el sistema notificará que la base de datos ha sido abierta correctamente.
- Del menú principal seleccionará *Captura / Obras*
- En la ventana deberá desplazarse por las propuestas existentes hasta encontrar la deseada, por medio del control de desplazamiento.
- Al encontrarla deberá presionar el botón *Editar* y modificar los datos necesarios de los campos de ingreso, a continuación deberá presionar el botón *Guardar*
- El sistema validará los tipos de datos ingresados en los campos de la ventana de captura y notificará el éxito de la operación con un mensaje.
- El proceso se considera terminado

Resultados

Al realizar la operación de modificación de propuesta de obra, el sistema actualizará la tabla correspondiente en la BD y mostrará en la ventana de captura la propuesta modificada.

Evaluación de la prueba

Prueba satisfactoria.

Modificación incorrecta de propuesta de obra

Descripción

El capturista, una vez ingresado al sistema y seleccionado la base de datos correcta, seleccionará la opción *Obras* del menú *Captura*. En la ventana el capturista deberá desplazarse por las propuestas existentes en la tabla, hasta encontrar la que desea modificar, posteriormente deberá presionar el botón *Editar* para modificar los campos necesarios con la información de la lista de propuestas de obra, posteriormente deberá presionar el botón *Guardar* y el sistema verificará que los tipos de datos coincidan y notificará con un mensaje el fracaso de la operación.

Condición de ejecución

La propuesta de obra a editar debe estar almacenada en la tabla correspondiente.

Datos de entrada

- El capturista ingresará con nombre de usuario y contraseña al sistema
- Del menú principal seleccionará *Archivo / Abrir BD*
- En la ventana de selección de *Base de datos Origen* deberá elegir la BD que desea manipular. Al seleccionar la BD el sistema notificará que la base de datos ha sido abierta correctamente.
- Del menú principal seleccionará *Captura / Obras*
- En la ventana deberá desplazarse por las propuestas existentes hasta encontrar la deseada, por medio del control de desplazamiento.

-
- Al encontrarla deberá presionar el botón *Editar* y modificar los datos necesarios de los campos de ingreso, a continuación deberá presionar el botón *Guardar*
 - El sistema validará los tipos de datos ingresados en los campos de la ventana de captura y notificará el fracaso de la operación con un mensaje.
 - El proceso se considera terminado, hasta que el capturista corrija el error de tipo de datos.

Resultados

La información incorrecta proporcionada no puede ser almacenada en la BD por lo que los datos deberán ser corregidos para su almacenamiento.

Evaluación de la prueba

Prueba satisfactoria.

Eliminación correcta de propuesta de obra

Descripción

El capturista, una vez ingresado al sistema y seleccionado la base de datos correcta, seleccionará la opción *Obras* del menú *Captura*. En la ventana el capturista deberá desplazarse por las propuestas existentes en la tabla, hasta encontrar la que desea eliminar, posteriormente deberá presionar el botón *Eliminar* para borrar la información de la lista de la tabla correspondiente, el sistema le preguntará si realmente desea eliminar la propuesta de obra, por lo que deberá presionar el botón afirmativo para hacerlo. El sistema notificará con un mensaje el éxito de la operación.

Condición de ejecución

La propuesta de obra a eliminar debe estar almacenada en la tabla correspondiente.

Datos de entrada

- El capturista ingresará con nombre de usuario y contraseña al sistema
- Del menú principal seleccionará *Archivo / Abrir BD*
- En la ventana de selección de *Base de datos Origen* deberá elegir la BD que desea manipular. Al seleccionar la BD el sistema notificará que la base de datos ha sido abierta correctamente.
- Del menú principal seleccionará *Captura / Obras*
- En la ventana deberá desplazarse por las propuestas existentes hasta encontrar la deseada, por medio del control de desplazamiento.
- Al encontrarla deberá presionar el botón *Eliminar*, el sistema le preguntará si realmente desea eliminarla, a continuación deberá presionar el botón afirmativo.
- El sistema realizará el proceso de eliminación y notificará el éxito de la operación con un mensaje.
- El proceso se considera terminado

Resultados

Al realizar la operación de eliminación de propuesta de obra, el sistema actualizará la tabla correspondiente en la BD y mostrará en la ventana de captura la primera propuesta en la tabla.

Evaluación de la prueba

Prueba satisfactoria.

Prueba funcional 3 Operaciones secundarias sobre propuestas

Descripción

Para la eliminación o modificación de una propuesta de obra, será necesario proporcionar una herramienta capaz de permitir la localización rápida de dicha propuesta. Para ello se realizarán operaciones secundarias sobre las propuestas, estas operaciones pueden ser, búsqueda, ordenamiento y filtrado según criterio.

La operación de búsqueda consiste en localizar una propuesta de obra que cumpla con dichos criterios de selección de manera única. En cuanto al ordenamiento se realizará la operación de acuerdo a dos criterios, realizar el proceso de manera ascendente o descendente y por último el filtrado deberá realizar una selección de una o varias propuestas que cumplan con criterios previamente establecidos.

Búsqueda correcta de propuesta de obra

Descripción

El encargado de departamento, una vez ingresado al sistema y seleccionado la base de datos correcta, seleccionará la opción *Obras* del menú *Captura*. En la ventana el encargado de departamento deberá seleccionar la opción *Buscar* del menú *Editar* o presionar el

botón *Buscar* de la barra de herramientas. Después en la ventana de búsqueda deberá especificar la cadena a buscar, en donde se debe realizar la búsqueda y el tipo de búsqueda. Deberá presionar el botón *Buscar* y el sistema le notificará el éxito de la operación.

Condición de ejecución

La tabla de propuesta debe tener por lo menos una propuesta para realizar la búsqueda correspondiente. El encargado de departamento debe estar dado de alta en el sistema, su nombre de usuario es ADMIN.

Datos de entrada

- El encargado de departamento ingresará con nombre de usuario y contraseña al sistema
- Del menú principal seleccionará *Archivo / Abrir BD*
- En la ventana de selección de *Base de datos Origen* deberá elegir la BD que desea manipular. Al seleccionar la BD el sistema notificará que la base de datos ha sido abierta correctamente.
- Del menú principal seleccionará *Captura / Obras*
- A continuación deberá seleccionar *Editar / Buscar* o presionar el botón *Buscar* de la barra de herramientas
- En la ventana deberá introducir la cadena a buscar y determinar el lugar y el tipo de búsqueda de las listas desplegables
- Al presionar el botón *Buscar* el sistema realizará el proceso de búsqueda, el sistema notificará al encargado de departamento que la cadena ha sido encontrada, con un

mensaje en el control de desplazamiento.

- El proceso se considera terminado

Resultados

Al realizar la operación de búsqueda, el sistema mostrará la propuesta de obra en donde se encontró la cadena de búsqueda.

Evaluación de la prueba

Prueba satisfactoria.

Búsqueda incorrecta de propuesta de obra

Descripción

El encargado de departamento, una vez ingresado al sistema y seleccionado la base de datos correcta, seleccionará la opción *Obras* del menú *Captura*. En la ventana el encargado de departamento deberá seleccionar la opción *Buscar* del menú *Editar* o presionar el botón *Buscar* de la barra de herramientas. Después en la ventana de búsqueda deberá especificar la cadena a buscar, en donde se debe realizar la búsqueda y el tipo de búsqueda. Deberá presionar el botón *Buscar* y el sistema le notificará el fracaso de la operación, debido a que la búsqueda no produjo ningún resultado.

Condición de ejecución

La tabla de propuesta debe tener por lo menos una propuesta para realizar la búsqueda correspondiente. El encargado de departamento debe estar dado de alta en el sistema.

Datos de entrada

- El encargado de departamento ingresará con nombre de usuario y contraseña al sistema
- Del menú principal seleccionará *Archivo / Abrir BD*
- En la ventana de selección de *Base de datos Origen* deberá elegir la BD que desea manipular. Al seleccionar la BD el sistema notificará que la base de datos ha sido abierta correctamente.
- Del menú principal seleccionará *Captura / Obras*
- A continuación deberá seleccionar *Editar / Buscar* o presionar el botón *Buscar* de la barra de herramientas
- En la ventana deberá introducir la cadena a buscar y determinar el lugar y el tipo de búsqueda de las listas desplegables
- Al presionar el botón *Buscar* el sistema realizará el proceso de búsqueda, el sistema notificará al encargado de departamento que la cadena no ha sido encontrada, con un mensaje en el control de desplazamiento.
- El proceso se considera terminado

Resultados

Al no realizar la operación de búsqueda, el sistema mostrará la última propuesta de obra que haya sido manipulada.

Evaluación de la prueba

Prueba satisfactoria.

Ordenamiento correcto de propuestas de obra

Descripción

El encargado de departamento, una vez ingresado al sistema y seleccionado la base de datos correcta, seleccionará la opción *Obras* del menú *Captura*. En la ventana el encargado de departamento deberá seleccionar, con un click en el campo de ingreso, la referencia por medio de la cual se deberá realizar el ordenamiento. Posteriormente deberá seleccionar el tipo de ordenamiento, seleccionando la opción *Registros / Ordenar / Ascendente* o *Descendente* o desde los iconos de ordenamiento de la barra de herramientas. El sistema realizará el proceso y le notificará el éxito de la operación con un mensaje.

Condición de ejecución

La tabla de propuesta debe tener por lo menos una propuesta para realizar el ordenamiento correspondiente. El encargado de departamento debe estar dado de alta en el sistema.

Datos de entrada

- El encargado de departamento ingresará con nombre de usuario y contraseña al sistema
- Del menú principal seleccionará *Archivo / Abrir BD*

- En la ventana de selección de *Base de datos Origen* deberá elegir la BD que desea manipular. Al seleccionar la BD el sistema notificará que la base de datos ha sido abierta correctamente.
- Del menú principal seleccionará *Captura / Obras*
- Deberá seleccionar el parámetro de ordenamiento, por medio de un click en el campo deseado
- A continuación deberá seleccionar *Registros / Ordenar / Ascendente* o *Descendente* o desde los iconos de ordenamiento de la barra de herramientas
- El sistema realizará el proceso de ordenamiento, y le notificará al encargado de departamento que el proceso se ha llevado a cabo satisfactoriamente, por medio de un mensaje
- El proceso se considera terminado

Resultados

Al realizar la operación de ordenamiento, el sistema mostrará las propuestas de obra de acuerdo al tipo de ordenamiento realizado, ya sea en forma ascendente o descendente.

Evaluación de la prueba

Prueba satisfactoria.

Ordenamiento incorrecto de propuestas de obra

Descripción

El encargado de departamento, una vez ingresado al sistema y seleccionado la base de datos correcta, seleccionará la opción *Obras* del menú *Captura*. Posteriormente deberá seleccionar el tipo de ordenamiento, seleccionando la opción *Registros / Ordenar / Ascendente o Descendente* o desde los iconos de ordenamiento de la barra de herramientas. El sistema realizará el proceso y le notificará el fracaso de la operación con un mensaje, debido a la falta de referencia para ordenar.

Condición de ejecución

La tabla de propuesta debe tener por lo menos una propuesta para realizar el ordenamiento correspondiente. El encargado de departamento debe estar dado de alta en el sistema.

Datos de entrada

- El encargado de departamento ingresará con nombre de usuario y contraseña al sistema
- Del menú principal seleccionará *Archivo / Abrir BD*
- En la ventana de selección de *Base de datos Origen* deberá elegir la BD que desea manipular. Al seleccionar la BD el sistema notificará que la base de datos ha sido abierta correctamente.
- Del menú principal seleccionará *Captura / Obras*

- Seleccionará *Registros / Ordenar / Ascendente* o *Descendente* o desde los iconos de ordenamiento de la barra de herramientas
- El sistema realizará el proceso de ordenamiento, y le notificará al encargado de departamento que el proceso ha fallado debido a la falta de referencia para ordenar, esto por medio de un mensaje
- El proceso se considera terminado

Resultados

Al no realizar la operación de ordenamiento, el sistema mostrará la última propuesta de obra que haya sido manipulada.

Evaluación de la prueba

Prueba satisfactoria.

Filtrado correcto de propuestas de obra

Descripción

El encargado de departamento, una vez ingresado al sistema y seleccionado la base de datos correcta, seleccionará la opción *Obras* del menú *Captura*. En la ventana el encargado de departamento deberá seleccionar, con un click en el campo de ingreso, la referencia por medio de la cual se deberá realizar el filtrado. Posteriormente deberá seleccionar la opción *Registros / Filtrar* o desde los iconos de filtrado de la barra de herramientas. El sistema realizará el proceso y le notificará el éxito de la operación con un mensaje en el control de desplazamiento.

Condición de ejecución

La tabla de propuesta debe tener por lo menos una propuesta para realizar el filtrado correspondiente. El encargado de departamento debe estar dado de alta en el sistema.

Datos de entrada

- El encargado de departamento ingresará con nombre de usuario y contraseña al sistema
- Del menú principal seleccionará *Archivo / Abrir BD*
- En la ventana de selección de *Base de datos Origen* deberá elegir la BD que desea manipular. Al seleccionar la BD el sistema notificará que la base de datos ha sido abierta correctamente.
- Del menú principal seleccionará *Captura / Obras*
- Deberá seleccionar la referencia para filtrar, por medio de un click en el campo deseado
- A continuación deberá seleccionar *Registros / Filtrar* o desde los iconos de filtrado de la barra de herramientas
- El sistema realizará el proceso de ordenamiento, y le notificará al encargado de departamento que el proceso se ha llevado a cabo satisfactoriamente, por medio de un mensaje en el control de desplazamiento
- El proceso se considera terminado

Resultados

Al realizar la operación de filtrado, el sistema mostrará las propuestas de obra que cumplan con la referencia con la cual se llevo a cabo el proceso.

Evaluación de la prueba

Prueba satisfactoria.

Filtrado incorrecto de propuestas de obra

Descripción

El encargado de departamento, una vez ingresado al sistema y seleccionado la base de datos correcta, seleccionará la opción *Obras* del menú *Captura*. Posteriormente seleccionará la opción *Registros / Filtrar* o desde los iconos de filtrado de la barra de herramientas. El sistema realizará el proceso y le notificará el fracaso de la operación con un mensaje, debido a la falta de referencia para filtrar.

Condición de ejecución

La tabla de propuesta debe tener por lo menos una propuesta para realizar el filtrado correspondiente. El encargado de departamento debe estar dado de alta en el sistema.

Datos de entrada

- El encargado de departamento ingresará con nombre de usuario y contraseña al sistema
- Del menú principal seleccionará *Archivo / Abrir BD*

-
- En la ventana de selección de *Base de datos Origen* deberá elegir la BD que desea manipular. Al seleccionar la BD el sistema notificará que la base de datos ha sido abierta correctamente.
 - Del menú principal seleccionará *Captura / Obras*
 - Seleccionará *Registros / Filtrar* o desde los iconos de filtrado de la barra de herramientas
 - El sistema realizará el proceso de ordenamiento, y le notificará al encargado de departamento que el proceso ha fallado, por medio de un mensaje
 - El proceso se considera terminado

Resultados

Al no realizar la operación de filtrado, el sistema mostrará la última propuesta de obra que haya sido manipulada.

Evaluación de la prueba

Prueba satisfactoria.

Prueba funcional 4 Consulta de catálogos

Descripción

En ocasiones será necesario consultar las tablas secundarias de la base de datos para realizar modificaciones o actualizaciones según sea el caso.

Los catálogos permiten mostrar la información almacenada en las tablas auxiliares, que sirven como apoyo a la tabla principal. Existen tres catálogos que podrán ser visualizados en el sistema, estos son *Localidades*, *Fondo* y *Programas*.

Estos catálogos están basados en la ventana de captura de propuestas, que es la ventana principal, por lo que tienen las mismas funcionalidades, esto es, podrán ser agregados, eliminados, actualizados, buscados, ordenados y filtrados los registros que se encuentran almacenados en cada uno de ellos.

Por tal motivo las pruebas siguientes se centran en la capacidad del sistema para mostrar cada uno de los catálogos y no sobre las opresiones que se pueden realizar sobre los mismos.

Consulta de catálogo de Localidades

Descripción

El encargado de departamento o capturista, una vez ingresado al sistema y seleccionado la base de datos correcta, seleccionará la opción *Localidades* del menú *Catálogos*.

Condición de ejecución

El encargado de departamento o capturista debe estar dado de alta en el sistema.

Datos de entrada

- El encargado de departamento o capturista ingresará con nombre de usuario y contraseña al sistema
- Del menú principal seleccionará *Catálogos / Localidades*
- El sistema visualizará una ventana en donde se muestran cada uno de los registros almacenados en la tabla *LOCALIDADES*.
- El proceso se considera terminado

Resultados

El sistema mostrará todos los registros guardados en la tabla de *LOCALIDADES*.

Evaluación de la prueba

Prueba satisfactoria.

Consulta de catálogo de Fondos

Descripción

El encargado de departamento o capturista, una vez ingresado al sistema y seleccionado la base de datos correcta, seleccionará la opción *Fondos* del menú *Catálogos*.

Condición de ejecución

El encargado de departamento o capturista debe estar dado de alta en el sistema.

Datos de entrada

- El encargado de departamento o capturista ingresará con nombre de usuario y contraseña al sistema
- Del menú principal seleccionará *Catálogos / Fondos*
- El sistema visualizará una ventana en donde se muestran cada uno de los registros almacenados en la tabla *FONDOS*
- El proceso se considera terminado

Resultados

El sistema mostrará todos los registros guardados en la tabla de *FONDOS*

Evaluación de la prueba

Prueba satisfactoria.

Consulta de catálogo de Programas

Descripción

El encargado de departamento o capturista, una vez ingresado al sistema y seleccionado la base de datos correcta, seleccionará la opción *Programas* del menú *Catálogos*.

Condición de ejecución

El encargado de departamento o capturista debe estar dado de alta en el sistema.

Datos de entrada

- El encargado de departamento o capturista ingresará con nombre de usuario y contraseña al sistema
- Del menú principal seleccionará *Catálogos / Programas*
- El sistema visualizará una ventana en donde se muestran cada uno de los registros almacenados en la tabla *PROGRAMAS*.
- El proceso se considera terminado

Resultados

El sistema mostrará todos los registros guardados en la tabla de *PROGRAMAS*.

Evaluación de la prueba

Prueba satisfactoria.

Prueba funcional 5 Control de acceso a usuarios

Descripción

Antes de iniciar el sistema, los usuarios deberán introducir una contraseña para poder manipular la información y así evitar el uso no autorizado del sistema.

El método encargado de este proceso validará los datos proporcionados por el usuario, realizando una lectura del archivo, y verificando que los datos sean correctos, con lo que el usuario tendrá acceso completo al sistema.

Verificación de nombre de usuario y contraseña incorrectos

Descripción

El usuario, al iniciar el sistema verá una ventana de identificación de usuario, en la que deberá introducir su nombre de usuario y contraseña. El sistema verificará los datos que se han ingresado y notificará que el usuario no está dado de alta en el sistema por medio de un mensaje.

Condición de ejecución

Ninguna

Datos de entrada

- El usuario inicia el sistema desde el acceso directo correspondiente
- Se visualizará una ventana donde ingresará el nombre de usuario y contraseña correspondientes
- El sistema verificará ambos campos con los del archivo correspondiente y comprueba que no concuerdan
- El sistema notifica el error con un mensaje y espera hasta que introduzcan datos correctos o cierren la ventana

Resultados

Solo los usuarios dados de alta en el sistema pueden manipular la información almacenada en la BD.

Evaluación de la prueba

Prueba satisfactoria.

Acceso correcto de USER

Descripción

El usuario, al iniciar el sistema verá una ventana de identificación de usuario, en la que deberá introducir su nombre de usuario y contraseña. El sistema verificará los datos que se han ingresado son correctos y mostrará la ventana principal del sistema.

Condición de ejecución

Ninguna

Datos de entrada

- El usuario inicia el sistema desde el acceso directo correspondiente
- Se visualizará una ventana donde ingresará el nombre de usuario y contraseña correspondientes y presiona el botón *Validar*
- El sistema verificará ambos campos con los del archivo correspondiente y comprueba que el usuario concuerda con *USER*
- El sistema presentará la ventana principal del sistema y parte del menú principal

Resultados

Al identificarse correctamente el nombre de usuario *USER* podrá ingresar al sistema y manipular la información almacenada en la BD.

Evaluación de la prueba

Prueba satisfactoria.

Acceso correcto de ADMIN

Descripción

El usuario, al iniciar el sistema verá una ventana de identificación de usuario, en la que deberá introducir su nombre de usuario y contraseña. El sistema verificará los datos que se han ingresado son correctos y mostrará la ventana principal del sistema.

Condición de ejecución

Ninguna

Datos de entrada

- El usuario inicia el sistema desde el acceso directo correspondiente
- Se visualizará una ventana donde ingresará el nombre de usuario y contraseña correspondientes y presiona el botón *Validar*
- El sistema verificará ambos campos con los del archivo correspondiente y comprueba que el usuario concuerda con *ADMIN*
- El sistema presentará la ventana principal del sistema y parte del menú principal

Resultados

Al identificarse correctamente el nombre de usuario *ADMIN* podrá ingresar al sistema y manipular la información almacenada en la BD.

Evaluación de la prueba

Prueba satisfactoria.

Prueba funcional 6 Gestión de datos de usuarios

Descripción

Los usuarios del sistema deberán cambiar su contraseña de ingreso de forma periódica para evitar uso no autorizado del sistema. Para eso deberán ingresar al sistema y una vez en dentro, deberán seleccionar la opción *Usuario / Cambiar Contraseña*.

En la ventana que se visualizará, los usuarios deberán introducir el nombre de usuario, la contraseña anterior y la que desea establecer. El método encargado de este proceso validará los datos proporcionados por el usuario, realizando una escritura del archivo, y verificando que los datos sean correctos, con lo que el usuario completará el cambio de contraseña.

Cambio de contraseña, usuario incorrecto

Descripción

El usuario, una vez ingresado al sistema, y elegido la BD correcta, seleccionará la opción *Usuario / Cambiar Contraseña*, con la que verá la ventana de *Administración de usuarios*, en la que deberá introducir su nombre de usuario, contraseña anterior y la que desea establecer. El sistema verificará los datos que se han ingresado y notificará el fracaso de la operación porque el usuario no está dado de alta en el sistema.

Condición de ejecución

El nombre de usuario debe estar dado de alta en el sistema

Datos de entrada

- El usuario ingresará con nombre de usuario y contraseña al sistema
- Del menú principal seleccionará *Usuario / Cambiar Contraseña*
- Se visualizará una ventana donde deberá introducir el nombre de usuario, contraseña anterior y la que desea establecer como nueva, posteriormente deberá presionar el botón *Actualizar*
- El sistema verificará los campos con los del archivo
- El sistema notificará con un mensaje que la contraseña no ha sido actualizada porque el usuario es incorrecto

Resultados

Solo los usuarios dados de alta en el sistema pueden manipular la información almacenada en la BD, y por lo tanto solo podrán modificar la contraseña aquellos que posean los datos correctos.

Evaluación de la prueba

Prueba satisfactoria.

Cambio de contraseña, contraseña no válida

Descripción

El usuario, una vez ingresado al sistema, y elegido la BD correcta, seleccionará la opción *Usuario / Cambiar Contraseña*, con la que verá la ventana de *Administración de usuarios*, en la que deberá introducir su nombre de usuario, contraseña anterior y la que desea establecer. El sistema verificará los datos que se han ingresado y notificará el fracaso de la operación porque la contraseña no es válida.

Condición de ejecución

El nombre de usuario debe estar dado de alta en el sistema

Datos de entrada

- El usuario ingresará con nombre de usuario y contraseña al sistema
- Del menú principal seleccionará *Usuario / Cambiar Contraseña*
- Se visualizará una ventana donde deberá introducir el nombre de usuario, contraseña anterior y la que desea establecer como nueva, posteriormente deberá presionar el botón *Actualizar*
- El sistema verificará los campos con los del archivo
- El sistema notificará con un mensaje que la contraseña no ha sido actualizada, porque la contraseña actual no es válida

Resultados

Solo los usuarios dados de alta en el sistema pueden manipular la información almacenada en la BD, y por lo tanto solo podrán modificar la contraseña aquellos que posean los datos correctos.

Evaluación de la prueba

Prueba satisfactoria.

Cambio de contraseña correcto

Descripción

El usuario, una vez ingresado al sistema, y elegido la BD correcta, seleccionará la opción *Usuario / Cambiar Contraseña*, con la que verá la ventana de *Administración de usuarios*, en la que deberá introducir su nombre de usuario, contraseña anterior y la que desea establecer. El sistema verificará los datos que se han ingresado y notificará el éxito de la operación con un mensaje.

Condición de ejecución

El nombre de usuario debe estar dado de alta en el sistema

Datos de entrada

- El usuario ingresará con nombre de usuario y contraseña al sistema
- Del menú principal seleccionará *Usuario / Cambiar Contraseña*

-
- Se visualizará una ventana donde deberá introducir el nombre de usuario, contraseña anterior y la que desea establecer como nueva, posteriormente deberá presionar el botón *Actualizar*
 - El sistema verificará los campos con los del archivo correspondiente y modificará el archivo con los nuevos datos
 - El sistema notificará con un mensaje que la contraseña ha sido actualizada

Resultados

Solo los usuarios dados de alta en el sistema pueden manipular la información almacenada en la BD, asegurando que al cambiar periódicamente la contraseña evitan la posibilidad de accesos no autorizados.

Evaluación de la prueba

Prueba satisfactoria.

Prueba funcional 7 Administración de Base de Datos

Descripción

En ocasiones será necesario respaldar la base de datos para evitar la pérdida de información, también es necesario la creación de nuevas bases de datos para una mejor administración de la información.

Para abrir una BD será necesario que cumpla con dos restricciones importantes, la primera

es que la BD deberá tener el formato requerido por el sistema y la segunda es que deberá contener todas las tablas necesarias para su manipulación.

Con esta prueba se pretende comprobar la funcionalidad de las opciones *Archivo / Abrir BD*, *Crear nueva BD* y *Crear respaldo de BD*.

Abrir BD correcta

Descripción

El usuario, una vez ingresado al sistema, seleccionará la opción *Archivo / Abrir BD*, con lo que verá la ventana *Base de datos origen*, en la que deberá buscar y elegir la BD que desea manipular, posteriormente deberá presionar el botón *Abrir*, el sistema verificará si la BD cumple con las condiciones para que pueda establecer una conexión. El sistema notificará el éxito de la operación con un mensaje.

Condición de ejecución

La BD debe cumplir con las condiciones necesarias para su manipulación

Datos de entrada

- El usuario ingresará con nombre de usuario y contraseña al sistema
- Del menú principal seleccionará *Archivo / Abrir BD*
- Se visualizará una ventana *Base de datos origen*, donde deberá elegir la BD que desea manipular, presionará *Abrir*

-
- El sistema verificará que la BD cumpla con las condiciones para que pueda realizar la conexión
 - El sistema notificará con un mensaje que la BD ha sido abierta con éxito

Resultados

Solo se podrán abrir BDs que cumplan con las condiciones necesarias, evitando así la manipulación de otra información ajena al sistema.

Evaluación de la prueba

Prueba satisfactoria.

Abrir BD incorrecta

Descripción

El usuario, una vez ingresado al sistema, seleccionará la opción *Archivo / Abrir BD*, con lo que verá la ventana *Base de datos origen*, en la que deberá buscar y elegir la BD que desea manipular, posteriormente deberá presionar el botón *Abrir*, el sistema verificará si la BD cumple con la condiciones para que pueda establecer una conexión. El sistema notificará el fracaso de la operación con un mensaje, dado que la BD no cumple con las condiciones necesarias.

Condición de ejecución

La BD debe cumplir con las condiciones necesarias para su manipulación

Datos de entrada

- El usuario ingresará con nombre de usuario y contraseña al sistema
- Del menú principal seleccionará *Archivo / Abrir BD*
- Se visualizará una ventana *Base de datos origen*, donde deberá elegir la BD que desea manipular, presionará *Abrir*
- El sistema verificará que la BD cumpla con las condiciones para que pueda realizar la conexión
- El sistema notificará con un mensaje que la BD no puede ser abierta porque no cumple las condiciones

Resultados

Al abrir la BD, el usuario podrá manipular la información contenida en ella.

Evaluación de la prueba

Prueba satisfactoria.

Crear nueva BD

Descripción

El usuario, una vez ingresado al sistema, seleccionará la opción *Archivo / Crear nueva BD*, con lo que el sistema creará una nueva BD, El sistema notificará el éxito de la operación con un mensaje donde se especificará el nombre de la BD.

Al crear una BD nueva, el sistema verificará si la BD correspondiente al año en curso ya fue creada, si no ha sido así el nombre de la BD será compuesto por la palabra *Propuesta* y el año en curso, p.e. *Propuestas 2005.mdb*.

Si ya fue creada con ese nombre, el sistema aumentará un año al nombre, p.e. *Propuestas 2006.mdb*.

Condición de ejecución

La BD debe cumplir con las condiciones necesarias para su manipulación

Datos de entrada

- El usuario ingresará con nombre de usuario y contraseña al sistema
- Del menú principal seleccionará *Archivo / Crear nueva BD*
- El sistema creará una nueva BD de acuerdo a los parámetros para nombrarla
- El sistema notificará con un mensaje que la BD ha sido creada con éxito, proporcionando el nombre de la misma

Resultados

Solo se podrán crear dos nuevas BDs posteriores al año en curso, debido a que el sistema solo podrá aumentar un año al nombre.

Evaluación de la prueba

Prueba satisfactoria.

Crear respaldo BD

Descripción

El usuario, una vez ingresado al sistema, seleccionará la opción *Archivo / Crear respaldo de BD*, con lo que el sistema creará un respaldo de la BD en uso, El sistema notificará el éxito de la operación con un mensaje donde se especificará el nombre de la BD.

Al crear el respaldo de la BD en uso, el sistema verificará si el respaldo de la BD correspondiente ya fue creada, si no ha sido así el nombre de la BD será compuesto por la palabra *Propuesta*, el año en curso y la letra *A*, p.e. *Propuestas 2005A.mdb*.

Si ya fue creada con ese nombre, el sistema eliminará la BD anterior y creará una nueva con el nombre anterior.

Condición de ejecución

La BD debe cumplir con las condiciones necesarias para su manipulación

Datos de entrada

- El usuario ingresará con nombre de usuario y contraseña al sistema
- Del menú principal seleccionará *Archivo / Crear respaldo de BD*
- El sistema creará el respaldo de la BD en uso, de acuerdo a los parámetros para nombrarla
- El sistema notificará con un mensaje que la BD ha sido creada con éxito, proporcionando el nombre de la misma

Resultados

Solo se podrá crear un respaldo de la BD en uso.

Evaluación de la prueba

Prueba satisfactoria.

Prueba funcional 8 Reporte general de propuestas

Descripción

Se presentarán en pantalla todas las propuestas de obra, clasificadas y listas para su impresión.

Este reporte deberá contar con todos los datos de la propuesta de obra, y tendrá que estar clasificado de acuerdo a criterios establecidos por otros reportes existentes en la Dirección.

El reporte deberá contar con los datos generales de la organización, en este caso se incluirá el nombre de la Secretaría de Desarrollo Social, el nombre de la Subsecretaría de Planeación y por último el nombre de la Dirección Técnica. También deberá incluir los datos generales del tipo de documento, así como también la fecha de solicitud de reporte.

La información contenida en los informes dependerá de las operaciones realizadas sobre las propuestas, por ejemplo si se realiza alguna operación como búsqueda, ordenamiento o filtrado la información presentada en los reportes provendrá de los resultados de la

operación realizada anteriormente.

Consulta de reporte general

Descripción

El encargado de departamento o capturista, una vez ingresado al sistema y seleccionado la base de datos correcta, seleccionará la opción *Reporte* del menú *Consulta Gral.*

Condición de ejecución

El encargado de departamento o capturista debe estar dado de alta en el sistema, y debe haber por los menos una propuesta en la BD.

Datos de entrada

- El encargado de departamento o capturista ingresará con nombre de usuario y contraseña al sistema
- Del menú principal seleccionará *Consulta Gral. / Reporte*
- El sistema visualizará una ventana en donde se mostrará una lista de todas y cada una de las propuestas almacenadas en la BD

Resultados

El sistema mostrará todos los registros guardados en la tabla de *PROPUESTAS*, proporcionando un formato para su ordenamiento y clasificación. Y tomando en cuenta la operación que haya podido ser ejecutada antes a esta ejecución de reporte.

Evaluación de la prueba

Prueba satisfactoria.

Prueba funcional 9 Reporte de propuesta

Descripción

En este caso el diseño del formato deberá presentarse de la misma manera que se presentan los datos en pantalla. Para esto será necesario que el usuario se encuentre en la ventana de captura de propuestas para que pueda visualizar el reporte de la propuesta actual.

Así como en el reporte anterior, este formato deberá contar con los datos generales de la organización, en este caso se incluirá el nombre de la Secretaría de Desarrollo Social, el nombre de la Subsecretaría de Planeación y por último el nombre de la Dirección Técnica. También deberá incluir los datos generales del tipo de documento, así como también la fecha de solicitud de reporte.

La información de las propuestas de obra será presentada de la misma manera en que es presentada en la pantalla de captura de propuestas. Los datos son Región, Municipio, Localidad, Número de beneficiarios, Descripción de obra, Fondo, Programa, Inversiones, Metas y Fechas.

Consulta de reporte específico

Descripción

El encargado de departamento o capturista, una vez ingresado al sistema y seleccionado la base de datos correcta, seleccionará la opción *Obras* del menú *Captura*, posteriormente seleccionará la opción *Archivo / Imprimir previo* o desde la barra de herramientas el icono del mismo nombre.

Condición de ejecución

El encargado de departamento o capturista debe estar dado de alta en el sistema, y debe haber por los menos una propuesta en la BD.

Datos de entrada

- El encargado de departamento o capturista ingresará con nombre de usuario y contraseña al sistema
- Del menú principal seleccionará *Archivo / Imprimir previo* o desde la barra de herramientas el icono del mismo nombre
- El sistema visualizará una ventana en donde se mostrará un formato en el que se presenta la propuesta actual, esto es, que se encuentra en pantalla

Resultados

El sistema mostrará los registros guardados en la tabla de *PROPUESTAS*, proporcionando un formato para su ordenamiento y clasificación. Y tomando en cuenta la operación que haya podido ser ejecutada antes a esta ejecución de reporte.

Evaluación de la prueba

Prueba satisfactoria.

Prueba funcional 10 Presentación de Manual de operación

Descripción

El manual de operación presentará al usuario una serie de datos donde se explica claramente como funciona el sistema, así como la forma en que se deben realizar los procesos para obtener mejores resultados.

Este manual estará compuesto por diferentes apartados, donde se especifican cada una de las funcionalidades del sistema.

Este índice tendrá la estructura siguiente:

- Iniciar la aplicación

- Descripción del Menú Principal

- Descripción de los Controles de Manipulación de Datos

- Descripción de Barra de Herramientas

Consulta de manual de operación

Descripción

El encargado de departamento o capturista, una vez ingresado al sistema, seleccionará la opción *Manual de operación* del menú *Ayuda*, posteriormente el sistema desplegará un archivo HTML con la información necesaria para manipular el sistema.

Condición de ejecución

El encargado de departamento o capturista debe estar dado de alta en el sistema.

Datos de entrada

- El encargado de departamento o capturista ingresará con nombre de usuario y contraseña al sistema
- Del menú principal seleccionará *Ayuda / Manual de operación*
- El sistema visualizará una ventana en donde se mostrará la información necesaria para manipular el sistema

Resultados

El sistema mostrará al usuario la manera en que debe manipular el sistema, por medio de un archivo HTML.

Evaluación de la prueba

Prueba satisfactoria.

Prueba funcional 11 Bloqueo del sistema

Descripción

Cuando sea necesario el sistema deberá ser bloqueado y solo con la contraseña correcta podrá ser puesto en marcha nuevamente.

Bloqueo del sistema

Descripción

El encargado de departamento o capturista, una vez ingresado al sistema, seleccionará la opción *Bloquear* del menú *Archivo*, posteriormente el sistema minimizará la ventana principal y desplegará la ventana de verificación de usuario.

Condición de ejecución

El encargado de departamento o capturista debe estar dado de alta en el sistema.

Datos de entrada

- El encargado de departamento o capturista ingresará con nombre de usuario y contraseña al sistema
- Del menú principal seleccionará *Archivo / Bloquear*
- El sistema minimizará la ventana principal y desplegará la ventana de verificación de usuario
- El sistema regresará a estado normal hasta que el encargado de departamento o capturista ingrese nuevamente con nombre de usuario y contraseña correctos

Resultados

El usuario evitará el uso del sistema mientras no se encuentra en uso.

Evaluación de la prueba

Prueba satisfactoria.

Apéndice D

Glosario

Algoritmo, conjunto finito de instrucciones o pasos que sirven para ejecutar una tarea o resolver un problema. La palabra algoritmo deriva del nombre del matemático árabe Al-Khwarizmi, (825 DC). Un programa de computadora puede considerarse como un algoritmo elaborado. Por ejemplo, el código para comprimir los datos de un archivo.

Base de datos, Conjunto de información guardada de forma correlativa y organizada, de forma tal que su contenido sea fácilmente accesado, administrado y actualizado.

Casos de uso, Es una técnica para capturar información de cómo un sistema o negocio trabaja, o de cómo se desea que trabaje. No pertenece estrictamente al enfoque orientado a objeto, es una técnica para captura de requisitos. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa.

Clases, Las clases son declaraciones o abstracciones de objetos, lo que significa, que una clase es la definición de un objeto. Cuando se programa un objeto y se definen sus características y funcionalidades, realmente se programa una clase.

Dato, Cantidad mínima de información no elaborada, sin sentido por sí misma, pero que convenientemente tratada se puede utilizar en la realización de cálculos o toma de decisiones.

HTML, Acrónimo inglés de Hyper Text Markup Language (Lenguaje de Marcación de Hipertexto), es un lenguaje de programación que permite estructurar textos y presentarlos como páginas web. Los documentos creados con este lenguaje contienen instrucciones que son interpretadas por un navegador, las cuales indican como debe desplegar el contenido del documento.

Información, Conjunto de datos organizados de tal manera que adquieren valor adicional más allá del que poseen por sí mismos.

Método, Proceso o camino sistemático establecido para realizar una tarea o trabajo con el fin de alcanzar un objetivo predeterminado.

Modelo, Abstracción o aproximación empleada para representar la realidad.

Paradigma, Aspecto o situación que se toma como patrón, ejemplo o modelo en forma esquemática.

Prototipo, Modelo o versión inicial de un sistema, previsto para probar y desarrollar el diseño, que no lleva a cabo la totalidad de las funciones necesarias del sistema final. Proporcionando una retroalimentación temprana por parte de los usuarios acerca del Sistema.

Releases, En español, revisión.º "versión". Es habitual que una aplicación software sufra modificaciones, mejoras o correcciones. El número de versión suele indicar el avance de los cambios. Suelen ser números correlativos, y frecuentemente son dos cifras separadas por un punto.

Rol, Papel desempeñado por las personas en algún sistema o en la sociedad

RUP, Rational Unified Process, método iterativo para el diseño de software creado por Rational Software Corporation (ahora división de IBM), Describe como generar software efectivo utilizando probadas técnicas comerciales. Comprende un gran número de diferentes actividades y fue diseñado para crear software a la medida.

Sistema Operativo, Conjunto de programas o aplicaciones base, que interactúan entre la máquina, los diferentes procesos y el usuario. Es el encargado administrar los diferentes recursos de la computadora (memoria, disco duro, llamadas al procesador para ejecutar instrucciones...). Imprescindible en cualquier computadora.

Técnicas de cuarta generación (T4G), El término (T4G) comprende un amplio rango de herramientas de software que facilitan la especificación inicial del software y la generación automática del código fuente basándose en la especificación. Este modelo está en auge debido a los grandes avances de la tecnología.

Tecnologías Orientadas a Objetos, Conjunto de disciplinas de ingeniería de software (análisis, diseño y programación) para el modelado y desarrollo que hacen más fácil construir sistemas complejos a partir de componentes individuales

UML, Acrónimo inglés de Unified Modelling Language (Lenguaje Unificado de Modelado) Es un lenguaje de propósito general para el modelado orientado a objetos. Es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir.

Usuario, Persona, organización u otra entidad que depende de los servicios de una computadora o sistema de información para obtener un resultado deseado.

Bibliografía

- [1] Senn, J. A. *Sistemas de Información para la Administración.*, Grupo Editorial Iberoamericana, México 1990. Tercera edición, pp 790.

- [2] Pressman, R. S. *Ingeniería de software.*, McGraw-Hill, España 1998. Cuarta edición, pp 824.

- [3] Stair, S. M. et al. *Principios de Sistemas de Información.*, International Thomson Editores, México 2000. Cuarta edición, pp 692.

- [4] León Serrano, G. *Ingeniería de sistemas de software.*, ISDEFE Ingeniería de Sistemas, Madrid España 1996. Primera edición, pp 221.

- [5] C. Wake, William *Extreme Programming Explored.*, Addison Wesley, 2002. Primera edición, pp 144.

- [6] *Métodologías ágiles en el desarrollo de software.*, Letelier Patricio, Penadés M^a Carmen, Laboratorio de Sistemas de Información. Departamento de Sistemas Informáticos y Computación. Facultad de Informática. Universidad Politécnica de Valencia. Taller realizado en el marco de las VIII Jornadas de Ingeniería del Software y Bases de Datos, JISBD 2003. Alicante, del 12 al 14 de Noviembre de 2003, 59 págs.

BIBLIOGRAFÍA

- [7] *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*, Letelier Patricio, Penadés M^a Carmen, Laboratorio de Sistemas de Información. Departamento de Sistemas Informáticos y Computación. Facultad de Informática. Universidad Politécnica de Valencia. 17 págs.
- [8] *Una explicación de la programación extrema (XP)*, Calero Solís Manuel., V Encuentro usuarios xBase 2003 Madrid, 11 págs. <http://www.apolosoftware.com/>
- [9] *eXtreme Programming (XP): Un nuevo método de desarrollo de software*, F. Acebal César, M. Cueva Juan Lovelle, Novática Edición digital, Marzo-Abril 2002, número 158, págs. 8-12. <http://www.ati.es/novatica/>
- [10] *Extreme Programming, Rapid Development for Web-Based Applications*, Maurer Frank, Martel Sebastien, IEEE Internet Computing, Enero-Febrero 2002, Vol. 6, No. 1, págs. 86-90. <http://www.computer.org/portal/site/internet//>
- [11] *Extreme Programming, Agil Project Management Advisory Service White Paper*, Highsmith Dim, Cutter Consortium, Febrero 2002. <http://www.cutter.com>
- [12] *Introducción a la Programación Extrema*, Aguilar S., Alejandro, Universidad Nacional Autónoma de México, Diciembre 2002. <http://www.willydev.net/descargas/Articulos/General/IntroXP.PDF>
- [13] *Introducción a la Programación Extrema*, Fernández E., Gerardo, Ingeniería de Software II, Universidad de Castilla - La Mancha, Departamento de Informática, Diciembre 2002. <http://www.info-ab.uclm.es/asignaturas/42551/trabajosAnteriores/Trabajo-XP.pdf>
- [14] *Procesos de desarrollo: RUP, XP y FDD*, Molpeceres A., Diciembre 2002. <http://www.javahispano.org/articles.article.action?id=76>

- [15] *Feature Driven Development Processes*, De Luca J., Diciembre 2002. <http://www.featuredrivendevelopment.com/files/fddprocessesA4.pdf>
- [16] *Ejemplo de desarrollo software utilizando la metodología XP*, Usero Ruiz Alejandro, Valero Vergara Francisco José, López Rodríguez César, Laboratorio de Sistemas de Información. Departamento de Sistemas Informáticos y Computación. Facultad de Informática. Universidad Politécnica de Valencia., 9 de Enero de 2004, <http://www.dsic.upv.es/asignaturas/facultad/lsi/ejemploxp/index.html>
- [17] *Extreme Programming: A gentle introduction.*, Dun Wells., 1999 - 2001, <http://www.extremeprogramming.org>
- [18] *An Agil Software Development Resource.*, Ronald E. Jefries., 1999 - 2005, <http://www.xprogramming.com>