



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO ⁶³

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES ²⁷

“ARAGON”

SISTEMA DE ADQUISICIONES DE LA
BIBLIOTECA DE LA ENEP ARAGON
EN SQLWINDOWS 5.0

T E S I S

Que para obtener el Título de:

INGENIERO EN COMPUTACION

P r e s e n t a :

MARIO SOTO MEDRANO

San Juan de Aragón, Edo. de Méx.

1996

TESIS CON
FALLA DE ORIGEN

TESIS CON
FALLA DE ORIGEN



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

TESIS

COMPLETA



ESTADOS UNIDOS MEXICANOS
SECRETARÍA DE EDUCACIÓN PÚBLICA

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
ARAGON
DIRECCION

MARIO SOTO MEDRANO
P R E S E N T E .

En referencia a su atento escrito de fecha 25 DE abril del año en curso, por el que solicita autorización para **cambio de título** de su trabajo de Tesis, mismo que propone se denomine **"SISTEMA DE ADQUISICIONES DE LA BIBLIOTECA DE LA ENEP ARAGON EN SQLWINDOWS 5.0"**, dirigido por , Ing. **ERNESTO PEÑALOZA ROMERO**, con fundamento en el punto 6 y siguientes del Reglamento para Exámenes Profesionales en esta Escuela, y toda vez que la documentación presentada por usted reúne los requisitos que establece el precitado Reglamento; me permito comunicarle que ha sido aprobada su solicitud.

Aprovecho la ocasión para reiterarle mi distinguida consideración.

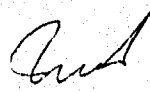
ATENTAMENTE
"POR MI RAZA HABLARA EL ESPIRITU"
San Juan de Aragón, Edo. de Méx., mayo 5 de 1995
EL DIRECTOR



M en I CLAUDIO C. MERRIFIELD CASTRO

c c p Unidad Académica.
c c p Jefe de Carrera de Ingeniería en Computación.
c c p Asesor de Tesis.

CCMC/AIR/vr



A mis padres: Teresa y Jela,
por enseñarme el camino.

A mis hermanos: Jela, Heracles, Javier y Estrella Svell,
por apoyarme.

To my dear Olga Leticia, for not leave fallme.
Because your support is an encouragement for me.

A todos mis amigos, no menciono sus nombres porque
faltaría espacio. Saben que los estimo mucho.

Gracias.

Mis uterlos:

**Es imposible distinguir entre
una tecnología lo suficientemente avanzada
y la magia.**

Arthur Clarke

**No sabemos quiénes somos
hasta que vemos
lo que somos capaces de hacer.**

Martha Grimes

**Sorprenderse,
extrañarse,
es comenzar a entender.**

José Ortega y Gasset

**Todo les parece imposible
a quienes no intentan nada.**

Jean-Louis Étienne

**Quien se sienta en el fondo de un pozo
a contemplar el cielo,
lo encontrará pequeño.**

Han Yu

**Quizá la imaginación no sea
sino inteligencia que se divierte.**

George Scialabba

**Lo mejor de todo es aprender.
El dinero puede perderse o ser robado,
la salud y la fuerza llegan a fallar,
pero lo que confiamos a la mente
es nuestro para siempre.**

Louis L'Amour

**La adversidad alimenta el valor.
Nadie puede ser valiente
si en la vida sólo le han sucedido
cosas maravillosas.**

Mary Tyler Moore

**Al tomar parte en una competición,
es preciso esforzarse como si hasta el último minuto
hubiera posibilidades de perder.
Así debe uno enfrentarse a un combate;
a la política;
a todo.**

Dwight Eisenhower

**Si el saber
es un placer,
no te lo niegues.**

Mario Soto

**Nada sucede
hasta que sucedió.**

Mario Soto

**Si las estrellas aparecieran
en el firmamento sólo una noche
cada milenio,
¿cómo podría el hombre
creer y adorar?**

Ralph Waldo Emerson

**No tener ilusiones es respetable;
y seguro,
y benéfico...
y absurdo.**

Joseph Conrad

**... he de cumplir mis promesas,
y caminar muchos kilómetros,
y caminar muchos kilómetros
antes de dormir ...**

Robert Frost

```

* Programa: RECUERDA.prg ...lo que algunas veces olvidamos...
*
* NOTA : Leer y ejecutar todas las mañanas al despertar.
*
* Fecha : HOY

USE Tu_mismo      && Despierta
SET TALK OFF      && No hablar por hablar
SET ESCAPE OFF    && No escapar a lo que se debe enfrentar

CLEAR             && Contemplar el panorama claramente

Seas_consciente = .Yes.

DO WHILE Seas_consciente

    STORE "amor"      TO _corazón
    STORE "salud"     TO _cuerpo
    STORE "paz"       TO _mente
    STORE "compasión" TO _otros
    STORE "estima"    TO _a_tí_mismo
    STORE "confianza" TO _humanidad
    STORE "fe"        TO _DIOS                && Constante

    REPLACE Negativo WITH Positivo, ;
             Juzgar   WITH Aceptar, ;
             Rencor   WITH Perdón

    REPLACE Desesperación WITH Opciones, ;
             Confusión  WITH Claridad, ;
             Dilación   WITH Participación

    REPLACE Separación WITH Unión, ;
             Escasez   WITH Abundancia, ;
             Penas     WITH Alegrías

    @ todo, el_tiempo SAY tu_verdad

    IF es_tiempo
        EXIT
    ENDIF

ENDDO

SAVE TO Siempre      && No lo olvides

CLEAR ALL

RETURN

```


**Sistema de Adquisiciones de la Biblioteca
de la
ENEP Aragón
en
SQLWindows 5.0**

Índice

Introducción	1
Capítulo 1 Antecedentes	3
1.1 La Biblioteca de la ENEP Aragón	5
1.2 WINDOWS 3.1	7
1.2.1 Ventajas	8
1.2.2 Desventajas	9
1.2.3 Windows ante dos perspectivas	9
1.3 SQLWindows 5.0	12
1.3.1 ¿Qué es SQL.Windows?	12
1.3.2 Lo que se necesita saber para utilizar SQL.Windows	13
1.3.3 Diseño visual y la paleta de herramientas	14
1.3.4 Personalizador	15
1.3.5 Ventana de código fuente	16
1.3.6 Barra de opciones	17
1.3.7 Lenguaje de aplicación de SQL.Windows	18
1.3.8 Facilidad de orientación a los objetos	18
1.3.9 Depurador	19
1.3.10 Compilador	19
1.3.11 Creación de un archivo ejecutable (.EXE)	20
1.3.12 Requerimientos mínimos	20

Capítulo 2 El proceso manual	23
2.1 Situación actual	24
2.1.1 Solicitudes	24
2.1.2 Pedidos	25
2.1.3 Formas F-2	27
2.2 Propuesta del Sistema de Control de Adquisiciones	30
2.2.1 Solicitudes	31
2.2.2 Pedidos	32
2.2.3 Formas F-2	34
2.2.4 Consultas	35
2.2.5 Catálogos	39
Capítulo 3 SQLWindows 5.0	41
3.1 Objetos	43
3.1.1 ¿Qué son los objetos?	43
3.1.2 ¿Cómo interactúan los objetos?	45
3.1.3 Clases e instancias	46
3.1.4 Polimorfismo	47
3.1.5 Herencia	48
3.2 Objetos de SQLWindows	51
3.2.1 Objetos de alto nivel	53
3.2.2 Objetos hijos	63
3.2.3 Diálogos estándar de Windows	73
3.3 SAL - Lenguaje de programación	74
3.3.1 Tipos de datos	75
3.3.2 Variables	76
3.3.3 Constantes	77
3.3.4 Arreglos	78
3.3.5 Operadores	79
3.3.6 Expresiones	79
3.3.7 Sentencias SAL	80
3.3.8 Comentarios	86
3.3.9 Funciones	87
3.3.10 Argumentos de la línea de comandos	89
3.4 Mensajes	89
3.4.1 Tipos de Mensajes	89
3.5 SQL - Manejo de bases de datos	92
3.5.1 Interfaz multi-paso	95
3.5.2 Interfaz de un-solo-paso	103
3.5.3 List Boxes y Combo Boxes	104

Capítulo 4	Sistema de Control de Adquisiciones	105
4.1	Descripción de Módulos	106
4.1.1	Módulo de Solicitudes	107
4.1.2	Módulo de Pedidos	109
4.1.3	Módulo de Formas F-2	109
4.1.4	Módulo de Catálogos	110
4.1.5	Módulo de Consultas	112
4.1.6	Módulo de Ayuda	113
4.2	Entidades y relaciones para la base de datos	114
4.3	Desarrollo	120
4.3.1	Qué se necesita hacer	121
4.3.2	Cómo hay que hacerlo	123
Conclusión		155
Bibliografía		157

Introducción

El desarrollo de un sistema administrativo por computadora aumenta la eficiencia, velocidad y exactitud en el procesamiento de grandes cantidades de datos. Esto, gracias a que alivia las actividades que ocurren con frecuencia y en la misma forma, lo cual podría provocar errores en cualquiera de las etapas del proceso.

El Departamento de Adquisiciones de la biblioteca de la ENEP Aragón no es la excepción al manejo de actividades rutinarias propensas al error. Este sistema resuelve en gran medida tal efecto al lograr que los datos se almacenen en la computadora para su consulta, impresión, actualización, etc.

El objetivo de esta tesis no es el describir las etapas del desarrollo de software. La intención es dar a conocer los conceptos de la Programación Orientada a Objetos y entender cómo es que utilizados con SQL Windows 5.0, una de las herramientas más modernas para la creación de sistemas, se logra agilizar el diseño del Sistema de Control de Adquisiciones.

SQL Windows hace más dinámico el diseño de la interfaz del usuario ya que se elabora visualmente. Los elementos que intervienen en el diseño ya cuentan con código predefinido, lo que acelera la codificación de las acciones a ejecutar.

SQL Windows permite que se utilice el enfoque de Programación Orientada a Objetos, de tal manera que ofrece una mejora en la productividad al lograr un desarrollo más flexible con la reutilización de código y con su facilidad de extensibilidad.

El humano al interactuar con su medio ambiente tiende a clasificar y abstraer cosas para su clasificación y comprensión. La Programación Orientada a Objetos imita la manera de este proceder al conceptualizar un programa (o un sistema) como un conjunto de objetos que se comunican para cumplir con su objetivo.

Por otra parte, al ser Windows un entorno gráfico, simplifica el manejo de las computadoras. Windows estandariza las aplicaciones al hacer que funcionalmente y visualmente sean parecidas. Este sistema al ser desarrollado sobre Windows goza de las ventajas que ofrece este entorno y así se logra que el sistema sea verdaderamente una herramienta acorde a los tiempos modernos de la computación.

En el capítulo 1 se dan a conocer de manera global la situación y funcionamiento actual de la biblioteca de la ENEP Aragón y se precisa en el Departamento de Adquisiciones. También se habla sobre las generalidades de Windows y SQLWindows.

En el capítulo 2 se hace un análisis del proceso manual del Departamento de Adquisiciones para conocer la situación actual, también se presenta la normatividad a la que se apegará el Sistema de Control de Adquisiciones propuesto por esta tesis.

El capítulo 3 hace una presentación de la teoría de los conceptos de orientación a objetos. También se dan a conocer las características de SQL Windows 5.0.

Finalmente, en el capítulo 4 se hablará sobre la aplicación de la teoría de los objetos sobre la programación del sistema.

Capítulo 1 Antecedentes

Objetivo General

- Conocer de manera global los elementos que intervendrán en el presente trabajo de tesis

Objetivos Particulares

- Conocer la estructura de la biblioteca y ubicar estratégicamente la importancia del Departamento de Adquisiciones
- Conocer las características y elementos principales del entorno de Windows 3.1, plataforma que se utilizará para desarrollar el Sistema de Control de Adquisiciones
- Conocer las características y elementos principales de SQL Windows 5.0, una de las herramientas más avanzadas para el diseño visual en Windows

La biblioteca de la ENEP Aragón, "Jesús Reyes Heróles", es uno de los recursos más importantes de la universidad, ya que su función es la de proporcionar apoyo bibliográfico, esencial para la formación académica, la investigación y la cultura en general.

Para lograr sus objetivos, la biblioteca cuenta con diversos departamentos, los cuales son:

- Departamento de Adquisiciones
- Departamento de Procesos Técnicos
- Departamento de Servicios Externos
- Departamento de Servicios Internos

De éstos, el Departamento de Adquisiciones es el área medular de la biblioteca. Es aquí donde se hace la selección y compra del nuevo material bibliográfico. La adquisición de nueva bibliografía es un gasto que de primera instancia afecta al presupuesto de la biblioteca y de manera global al presupuesto de la ENEP Aragón.

El control del presupuesto, aunado al seguimiento del proceso de selección y compra es una tarea ardua donde es necesario el apoyo informático que coadyuva las labores del Departamento de Adquisiciones.

En este capítulo, en la sección 1.1, se abordará de manera general la estructura de la biblioteca de la ENEP Aragón, siendo hasta el capítulo 2 donde se observe con detalle las funciones del Departamento de Adquisiciones.

En la sección 1.2 se habla de Windows 3.1, una de las interfaces para usuario más extendida y que facilita en gran medida la utilización de las microcomputadoras. Windows es uno de los programas más importantes de la actualidad, y aunque no sea el mejor se ha logrado perfeccionar a tal grado que es la plataforma donde ya hay cientos de programas: desde entretenidos juegos hasta poderosos programas de diseño. Windows será la plataforma para el nuevo sistema por los beneficios que reporta al usuario final.

En la sección 1.3 se habla sobre SQL Windows 5.0, que es la herramienta software utilizada para el desarrollo de este proyecto. SQL Windows es uno de los programas más sofisticados para implementar sistemas en el entorno de Windows. Tanto por el diseño visual como por su esquema de codificación, es un ambiente muy propicio para que los desarrolladores inmediatamente sean productivos.

1.1 La Biblioteca de la ENEP Aragón

La biblioteca de la ENEP Aragón pone a disposición del estudiante un gran acervo que fortalece la formación académica y profesional de la comunidad universitaria. Aquí se encontrarán, por ejemplo, enciclopedias, diccionarios, atlas, mapas, audiovisuales, revistas, folletos, etc.

La biblioteca se divide básicamente en cuatro departamentos con diversas funciones cada uno. A continuación se analizará brevemente la función de cada departamento.

El **Departamento de Adquisiciones** es el área más importante de la biblioteca, considerando que su labor es la de dar a conocer a las jefaturas de carrera los catálogos de material bibliográfico nacional y del extranjero para su revisión y selección, a fin de que elaboren y remitan las solicitudes para la compra del nuevo material de tal manera que los alumnos cuenten con la más variada colección de títulos para llevar a buen término sus labores escolares.

Se levantan pedidos por proveedor y se hace un seguimiento del pedido desde su emisión hasta que es surtido. Esto para evitar anomalías y que se adquieran los ejemplares que realmente se solicitaron. Se informa a la Dirección General de Bibliotecas (DGB) y ésta a su vez hace llegar los datos de identidad de los libros, los cuales son: Clasificación, Número de Matriz y Número de Adquisición. A este proceso de recabar los datos de identidad se le conoce como proceso mayor.

El **Departamento de Procesos Técnicos** es el encargado del proceso menor, es decir, de poner los sellos de propiedad, datos de identidad y proteger magnéticamente los nuevos libros. Aquí también se repara material que, por haber estado en servicio, se ha deteriorado. Al terminarse el proceso en los libros, éstos pasan al acervo general.

El **Departamento de Servicios Externos** está compuesto de tres áreas que permiten, en general, la consulta de ejemplares fuera de la biblioteca, ya sea la propia biblioteca de la ENEP Aragón u otra biblioteca que tenga convenios con la UNAM. Las áreas son:

- **Préstamo a domicilio**

Este préstamo permite, de acuerdo con el reglamento de la biblioteca, retirar material que se necesita consultar fuera de las instalaciones por cinco días hábiles.

- Préstamo interbibliotecario

Es una alternativa para obtener material bibliográfico no existente en esta biblioteca al tener convenios de préstamo con otras instituciones. De esta manera se subsanan ciertas carencias de la biblioteca de la FNEP Aragón.

- Estantería abierta

Este servicio funciona también como préstamo interno, al permitir la consulta directa y poder leer los libros en la sala de estudio. Esto también permite seleccionar los libros deseados y solicitar la autorización para obtenerlo como préstamo a domicilio.

Por su parte, el **Departamento de Servicios Internos** está compuesto por diversas áreas cuya función es poner al alcance del alumno o cualquier usuario todos aquellos materiales o servicios con los cuales cuenta la biblioteca. Algunos de estos materiales por su importancia, ya sea por su valor o por su mínima existencia, no podrán salir de la biblioteca. Además de la estantería abierta, se tiene:

- Área de Consulta

Aquí se cuenta con material de información específica como: enciclopedias, diccionarios, leyes, códigos, anuarios, tesis, etc.

- Hemeroteca

Se encuentran diarios, revistas, agendas estadísticas y material periódico.

- Área de Reserva

Aquí se resguarda un ejemplar de cada título del acervo general.

- Mapoteca

Cuenta con mapas, cartas geográficas, fotos aéreas, planímetros, estereoscopios, etc.

- Videoteca

Contiene material audiovisual de interés para todas las carreras y de cultura en general.

- Servicio de préstamo de Máquinas de Escribir

Máquinas disponibles para elaborar trabajos mecanográficos. Se cuenta con una sala especial para trabajar con las máquinas sin interferir con las demás áreas de estudio donde se debe guardar silencio.

- Servicio de préstamo de Cubículos de Estudio

Son pequeños cuartos de estudio para realizar trabajos en equipo con un máximo de 6 personas y un mínimo de 3.

- Servicio de préstamo de Cubículos para Asesoría de tesis

Son cuartos de estudio donde las distintas jefaturas de carrera dan un servicio de orientación para alumnas que desean iniciar o para aquellos que ya se encuentran elaborando una tesis.

- Servicio de Fotocopiado

Es un servicio concesionado donde se ofrecen fotocopias con la comodidad para los alumnos de que se encuentran dentro de la misma biblioteca.

Hasta aquí se ha revisado de forma general el funcionamiento de la biblioteca. Dado que el nuevo Sistema de Control de Adquisiciones se desarrollará para el ambiente gráfico Windows, es en la siguiente sección donde se analizará el por qué Windows es una plataforma bastante aceptable para el usuario final.

1.2 WINDOWS 3.1

Windows es un software que hace más sencillo el uso diario de una computadora. Es una interfaz que se coloca sobre el conocido ambiente DOS (Disk Operating System - Sistema Operativo de Disco) logrando con esto una amplia utilización no sólo por expertos sino hasta por novatos de casi cualquier tipo de programa diseñado específicamente para Windows.

Windows 3.1 fue lanzado al mercado en abril de 1992, casi dos años después de Windows 3.0 que salió en mayo de 1990.

Entre Windows 3.0 y 3.1 sólo hubo una diferencia del 10% en capacidades y añadidos y no implicaba grandes cambios, razón por la cual el número de la versión varía en el decimal.

1.2.1 Ventajas

Hay algunas mejoras y también nuevos elementos que hacen de Windows 3.1 la plataforma más popular desde entonces:

TrueType

Windows incorpora tipos de letra a escala conocidos bajo el nombre de TrueType, éstos se pueden desplegar e imprimir en cualquier tamaño y casi en cualquier monitor e impresora que Windows permita.

OLE

La Vinculación e Incrustación de Objetos (OLE - Object Linking and Embedding) es una técnica que permite incluir de una manera consistente y estándar en un programa Windows información proveniente de un documento creado en otra aplicación. Así, es posible insertar información de una hoja de cálculo en un documento de un procesador de textos y después actualizar esa información directamente en la hoja de cálculo en que fue creada, evitando tener que hacer cambios también en el procesador de textos.

Ayuda

La aplicación que muestra el texto de ayuda en la mayoría de las aplicaciones fue mejorada en varios aspectos. La ventana de Ayuda siempre ha permitido a los usuarios hacer clic en las palabras claves que están en diferentes colores y cambiarse a las secciones relacionadas del texto de ayuda. Ahora se cuenta con botones o áreas activas dentro de la ventana para saltar a un subtema.

Multimedia y lápiz óptico

Ahora se incluyen controladores que permiten utilizar sonidos, animaciones y otros efectos de multimedia, así como controladores para un lápiz óptico.

Cerrar aplicaciones que no responden

Si hay una aplicación que no responde en Windows y se presiona Ctrl+Alt+Supr, previene que esta combinación de teclas no se debe utilizar para salir de las aplicaciones de una manera normal, las opciones son presionar Escape para no tomar ninguna acción, oprimir Retorno para cerrar sólo la aplicación actual o presionar de nuevo Ctrl+Alt+Supr para reinicializar el sistema.

DOS mejorado

Una aplicación del DOS se podrá ejecutar en una ventana de Windows con su propio menú e incluso con algún tipo de letra de los de Windows. Será posible utilizar el Ratón aunque la aplicación no haya sido diseñada para él y también se podrá copiar datos desde esa aplicación DOS a una aplicación Windows.

Redes

Windows 3.1 agregó ciertas funciones específicas para las redes, tales como las locales y aquellas donde se comparten periféricos como las impresoras.

Procedimiento de inicio

Windows 3.1 cuenta con un grupo especial: Inicio, donde se colocan los iconos de las aplicaciones que se quieren ejecutar al iniciar Windows. Funciona como el AUTOEXEC.BAT del DOS.

1.2.2 Desventajas

Con todo y sus mejoras, Windows 3.1 no es perfecto y aún tiene problemas. Algunos problemas se pueden encontrar en:

Programas residentes para compresión de discos

Ni Windows e incluso su programa de instalación son compatibles con todas las aplicaciones de compresión de discos, por lo que resulta muy problemático instalarlo en una computadora con software de este tipo.

Comandos del DOS

Existen varios comandos del DOS que son incompatibles con las operaciones de Windows y que si son ejecutados en una sesión de Windows son capaces de desestabilizarlo.

1.2.3 Windows ante dos perspectivas

Al encontrar ventajas y desventajas, es natural que existan opiniones encontradas respecto al acierto de Windows en el mundo de las computadoras.

Cabe señalar que además de los juicios que se tengan en contra o a favor de Windows, existen dos enfoques respecto al tipo de usuario de esta plataforma: el de usuario final y el de diseñador de software.

Punto de vista del usuario final

En los primeros tiempos, las computadoras estaban orientadas a producir la mayor cantidad de trabajo que fuera posible, ya que era muy costosa, quedando relegados a segundo término la comodidad y productividad del usuario.

En los tiempos modernos se puede adquirir una computadora personal relativamente barata, cuyo sistema operativo provee un entorno propicio para el desarrollo del usuario, lo cual implica el incremento de la productividad y creatividad del mismo.

El sistema operativo es reforzado con paquetes de software que colocan una interfaz que hace más accesible el manejo del propio sistema y de la computadora en general. En los últimos años la interfaz que predomina en las microcomputadoras es Windows, y al basarse en un entorno gráfico facilita la comunicación computadora-usuario.

Al hacer innecesario el aprendizaje de la sintaxis de muchos comandos, simplifica la utilización de las aplicaciones por lo que se convierte en una herramienta muy poderosa al permitir la selección de opciones en forma casi intuitiva.

Además, todas las aplicaciones Windows se apegan a ciertos estándares, lo que permite que todas sean similares en su interfaz. Al aprender a utilizar una aplicación, se aprenderá a utilizar otras en menos tiempo.

En una corporación se necesita que el usuario sea productivo y que no pierda demasiado tiempo en la capacitación de una nueva aplicación. El entorno de Windows ofrece eso y más.

Punto de vista del diseñador de software

Windows permite que las aplicaciones sean relativamente independientes de la computadora, abriendo el amplio mercado de software.

Windows permite la multitarea, por lo que puede haber en algún momento dos o más aplicaciones ejecutándose al mismo tiempo, es por esto que cada aplicación

debe contar con un espacio en la pantalla y se le asigna un rectángulo llamado **ventana** (window) donde aparecen menús y cuadros de diálogo, por ejemplo.

En el entorno DOS, la aplicación se adueña de la pantalla y también debe preocuparse por todos y cada uno de los detalles de lo que el usuario hace. Windows se ocupa de todos los detalles de forma automática: sabe qué tecla se presiona o libera, sabe dónde se encuentra el ratón, etc.

Windows puede manejar aplicaciones que crean y usan un gran número de ventanas para desplegar información, Windows controla la pantalla y evita que dos aplicaciones traten de utilizar la misma porción de pantalla al mismo tiempo.

Windows cuenta con controladores de dispositivo para lograr la independencia de los dispositivos. Esto significa que, por ejemplo, se puedan dibujar líneas o círculos en una impresora láser o de matriz de puntos, en un monitor VGA o uno EGA, sin importar la resolución.

Al permitir la ejecución de más de un programa al mismo tiempo, los recursos de la computadora se deben compartir (la memoria, dispositivos de Entrada/Salida, la pantalla y por supuesto, el propio CPU), es por esto que Windows controla cuidadosamente cada uno de los detalles para garantizar el control de los recursos.

En el entorno DOS, una aplicación se apropia del CPU y, por ejemplo, la memoria puede ser accesada en toda su extensión y por el método que sea. Windows no lo permite, para llevar así una correcta administración del sistema.

Windows tiene muchas características que no tiene el entorno del tradicional sistema operativo DOS. Por esta razón, el desarrollo de aplicaciones Windows puede ser más complejo que el desarrollo de aplicaciones para el DOS. Considerando lo que Windows puede ofrecer, en realidad se paga un justo precio.

En la siguiente sección, 1.3, se hablará de SQLWindows, la herramienta software motivo de esta tesis. Se revisarán de manera global las características innovativas que hacen de SQLWindows un sistema de desarrollo bastante completo y poderoso.

1.3 SQLWindows 5.0

1.3.1 ¿Qué es SQLWindows?

SQLWindows fue introducido al mercado en 1988, y es uno de los ambientes Cliente/Servidor más utilizados mundialmente. Actualmente lo utilizan, entre otras, las siguientes empresas de renombre internacional:

American Express,
AT&T,
Boeing,
Duracell U.S.A.,
Glaxo Canada, INC.,
Minolta,
Toyota,
PEMEX,
y la organización UNICEF.

En SQLWindows se pueden crear aplicaciones sin escribir una sola línea de código o con programación mínima. Su lenguaje SAL (SQLWindows Application Language) es parecido a los lenguajes C o C++, por lo que tiene un enorme poder en el desarrollo de aplicaciones fuertes.

Una aplicación creada con SQLWindows puede manejar todos los elementos de la Interfaz Gráfica del Usuario de Windows.

Entre sus diversas características SQLWindows permite:

- Desarrollar aplicaciones de bases de datos sin escribir código utilizando QuickObjects.
- Trabajar en ambiente multiusuario.
- Programación orientada a objetos.
- Creación de reportes.
- Creación de aplicaciones tipo Interfaz de Múltiple Documento.
- Soporte a Librerías de Enlace Dinámico (DLL - Dynamic Link Libraries).
- Soporte a Intercambio Dinámico de Datos (DDE - Dynamic Data Exchange).

- Soporte a Vinculación e Inerustación de Objetos (OLE - Object Link and Embedding).
- Soporte a Controles de Visual Basic (VBX).
- Soporte a Conectividad Abierta a Bases de Datos (ODBC - Open DataBase Connection).
- Conectividad a múltiples Manejadores de Bases de Datos, entre los cuales se tienen:
 - ORACLE,
 - Informix,
 - DB2,
 - Sybase,
 - MS-SQLServer,
 - Gupta SQLBase,
 - Ingress, y
 - Novell B-Trieve.

1.3.2 Lo que se necesita saber para utilizar SQLWindows

Al ser SQLWindows una herramienta muy sofisticada, es necesario contar con conocimientos sobre los siguientes tópicos:

- Manejo de PC's IBM o compatibles.
- PC/MS-DOS (Disk Operating System de IBM o de Microsoft).
- Windows, tanto interfaz de usuario como aplicaciones.
- Uso del Ratón.
- Algún lenguaje de programación tal como COBOL, Pascal, BASIC o C.
- Conceptos de bases de datos relacionales.

Y preferentemente:

- Lenguaje SQL.
- Conceptos de programación orientada a objetos.

Opcionalmente se necesitará lenguaje C y herramientas relacionadas si se desea crear y utilizar Librerías de Enlace Dinámico (D.L.L.'s).

1.3.3 Diseño visual y la paleta de herramientas

En SQLWindows es posible diseñar la interfaz del usuario con sólo unos clics, con operaciones de arrastrar y soltar y de teclear alguna información. Esto es posible gracias a la Paleta de Herramientas, mostrada en la Figura 1-1, ya que contiene todos los elementos u objetos con los cuales el desarrollador trabajará para su diseño.

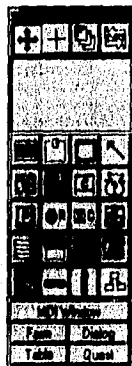


Figura 1-1: Paleta de herramientas

Conforme se coloquen objetos, SQLWindows genera código predefinido en la Ventana de Código Fuente. La conducta de esos objetos se definirá con el lenguaje SAL con ayuda de los mensajes SAM (SQLWindows Application Messages).

El diseño visual es tan fácil como hacer un clic en el objeto deseado sobre la Paleta de Herramientas y enseguida con una operación de arrastrar y soltar sobre un cuadro de diálogo como el de la Figura 1-2, por ejemplo, el objeto se coloca y se redimensiona para ajustarlo a las necesidades particulares en el diseño.

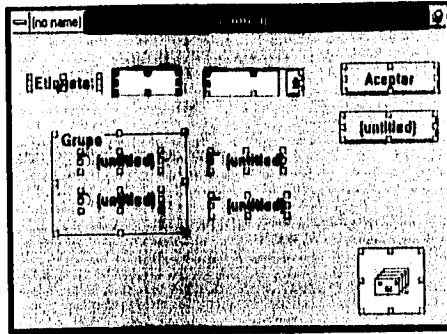


Figura 1-2: Diseño de un cuadro de diálogo con SQLWindows

1.3.4 Personalizador

Al ir colocando objetos en el diseño, se pueden cambiar los atributos de cada uno de ellos, de tal manera que se configuren según una apariencia similar, o para cambiar los colores o el tipo de letra, o para visualizar un gráfico en un botón. Esto se logra con el personalizador, Figura 1-3, que de acuerdo al objeto en cuestión contendrá diferentes atributos.

EdFecha	
Object Name	▶
Visible	▶
Location and Size	▶
Data Type	▶
Max Data Length	▶
Editable	▶
Border	▶
Justify	▶
Format	▶
Input Mask	▶
Country	▶
Background Color	▶
Text Color	▶
Font Name	▶
Font Size	▶
Font Enhancement	▶
Edit Actions	▶

Figura 1-3: Personalizador

1.3.5 Ventana de código fuente

A diferencia de escribir un programa en un lenguaje tal como C o BASIC, donde se tiene un editor de texto, SQLWindows tiene un editor especial llamado Outline Editor, Figura 1-4.

El Outline Editor difiere de los tradicionales editores de texto o procesadores de texto en que mantiene una estructuración muy rígida del código. Por ejemplo, una sentencia IF deberá contener código a ejecutar indentado hacia la derecha y abajo de él.

De acuerdo a las indentaciones de código, que pueden estar bajo un IF, un WHILE, un SELECT CASE, o un ELSE, el código puede estar oculto a la vista para mayor comprensión de la estructura general del programa. Cuando una sentencia contenga código indentado, el diamante a su izquierda será de color negro. Cuando el diamante sea hueco, indicará que no hay sentencias abajo de este nivel, esto también se puede ver en la Figura 1-4. Las sentencias se pueden ocultar o visualizar con un doble clic en el diamante negro.

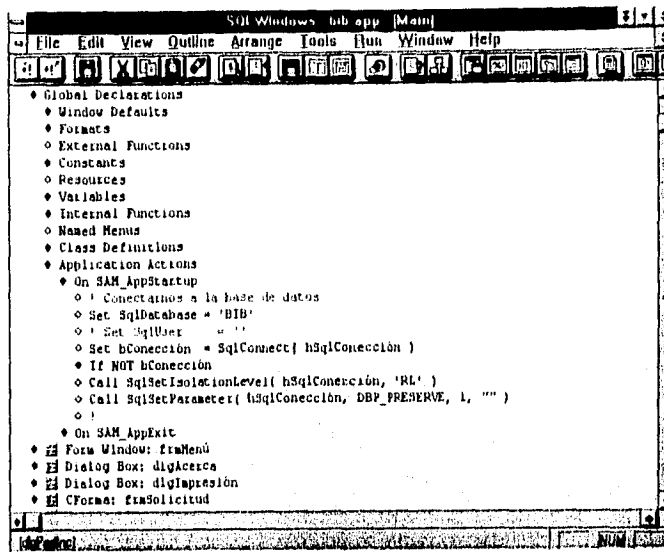


Figura 1-4. Ventana de código fuente

El editor puede tener diferentes vistas de porciones específicas de código, gracias a que SQL Windows es una aplicación del tipo de Interfaz de Múltiple Documento. Así, por ejemplo, si se tienen varios cuadros de diálogo, cada uno de ellos puede tener su propia ventana de código, lo cual agiliza su visualización y edición.

1.3.6 Barra de opciones

La Barra de Opciones provista por el editor de SQL Windows es un asistente para escribir el código de la aplicación.

La Barra de Opciones lista los elementos que se pueden agregar en el editor. Se usa para:

- Definir menús.
- Definir funciones, constantes y variables.
- Codificar sentencias SAL.

La Barra de Opciones, Figura 1-5, es sensible al contexto, al moverse en el editor, sus contenidos cambian indicando qué tipo de elementos pueden agregarse.

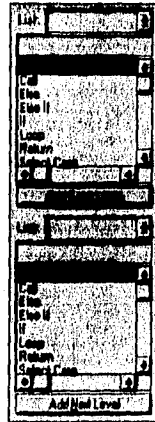


Figura 1-5: Barra de opciones

1.3.7 Lenguaje de aplicación de SQLWindows

Aunque con los QuickObjects es posible construir aplicaciones sin escribir código, las complejas aplicaciones Cliente/Servidor necesariamente demandan que se escriba código.

El lenguaje SAL se encuentra orientado a los objetos, cuenta con decenas de funciones interconstruidas en SQLWindows, o se pueden crear para formar una librería, y aún utilizar las de una librería del tipo DLL.

1.3.8 Facilidad de orientación a los objetos

SQLWindows incorpora la nueva modalidad de programar con orientación a los objetos, aunque no necesariamente se utilice esta característica. Una manera de iniciarse, es el utilizar los QuickObjects, los cuales son como los objetos estándar de la paleta de herramientas, pero con más inteligencia y ligados a tablas y datos de una o varias bases de datos.

Se puede modificar el comportamiento de los QuickObjects ya que se incorporan a SQL.Windows con todos los lineamientos de la programación orientada a objetos. Se podría utilizar la herencia para extender o limitar su conducta. O de la manera más simple, se pueden crear nuevos objetos especiales de ciertos atributos y/o conducta propia para un diseño específico.

1.3.9 Depurador

El depurador de SQL.Windows, Figura 1-6, puede manejar múltiples puntos de ruptura, ejecutar una instrucción a la vez, observar los valores de variables o expresiones en cualquier momento, o simplemente permite ver cómo se ejecutan las instrucciones una a una y sus efectos.

Cuando SQL.Windows suspende la ejecución de la aplicación, se despliega el cuadro de diálogo del depurador. En este punto se puede evaluar una expresión (Eval), continuar la ejecución de la aplicación (Continue), ejecutar la sentencia actual (Step), pasar sobre la sentencia actual (Step Over), detener la aplicación (Halt), o cerrar el cuadro de diálogo del depurador (Close).

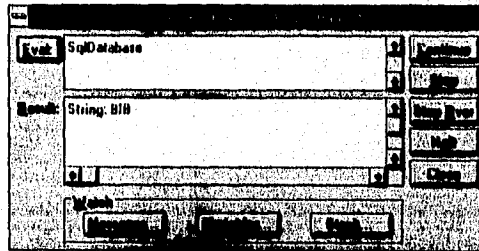


Figura 1-6: Depurador

1.3.10 Compilador

El compilador de SQL.Windows genera código en lenguaje C para portones de la aplicación y así poder elevar el desempeño de la aplicación.

Solamente las funciones internas, variables globales y constantes son convertidas en lenguaje C, con este código se crea un archivo DLL. Futuras versiones de SQL.Windows podrán trabajar sobre otras secciones de la aplicación.

1.3.11 Creación de un archivo ejecutable (.EXE)

Cuando la aplicación se ha finalizado es posible generar un archivo ejecutable e incluso asignarle un icono para incluirla en un grupo en el Administrador de Programas de Windows.

Para distribuir la aplicación también es necesario un conjunto de archivos de soporte para la ejecución de la aplicación, mejor conocido como Run Time.

1.3.12 Requerimientos mínimos

CPU	386 ó superior
RAM	8 MB
Espacio en disco duro	24 MB (mínimo)
Windows 3.1	

Estos requerimientos son para instalar la versión Corporativa, la cual cuenta con:

- QuickObjects que proveen una rápida y poderosa manera de crear aplicaciones.
- Una herramienta gráfica para la creación y utilización de bases de datos.
- Un servidor para el acceso a la base de datos.
- Librerías ODBC que permiten acceder archivos de dBase o Access, por ejemplo.
- Controladores para ORACLE, SQL Server, Informix, Ingres y AS/400, de tal manera que, por ejemplo, se puede acceder a bases de datos en una computadora con Informix y sistema operativo UNIX.
- Un Compilador para agilizar ciertas partes de código de una aplicación.
- Un programa para el monitoreo de servidores SQLBase.
- Un módulo que permite administrar un proyecto, el personal implicado en él y de la seguridad de los mismos.
- Un diseñador de reportes.

- Una interfaz interactiva para administrar y manejar bases de datos.
- Un editor que permite modificar una aplicación compilada para cambiar su apariencia al personalizarla para diferentes grupos de usuarios.

Los 8 MB de memoria RAM se recomiendan para llevar a cabo el desarrollo de un sistema de manera aceptable, sin embargo, 4 MB son suficientes. El producto final se puede utilizar en equipos con al menos 4 MB de memoria RAM.

Capítulo 2 El proceso manual

Objetivo General

- Conocer la operación del Departamento de Adquisiciones de la Biblioteca de la ENEP Aragón

Objetivos Particulares

- Conocer la situación actual del Departamento de Adquisiciones
- Conocer la normatividad a la que se apegará el nuevo Sistema de Control de Adquisiciones

Ya que el Departamento de Adquisiciones es el encargado de acrecentar el acervo de la biblioteca, es de suma importancia que cuente con un sistema que respalde tan crítica tarea, ya sea éste manual o informatizado.

Dado que la información manejada para este proceso es abundante, el proceso manual tiene la desventaja de estar sujeto 100% al factor humano, y es en extremo problemático hacer un seguimiento adecuado en el procedimiento de hacer una nueva adquisición o simplemente la verificación de lo que se ha ejercido del presupuesto en un intervalo fijo o arbitrario de tiempo.

Habiendo hecho un análisis de este proceso manual, se llegó a la conclusión de que es posible, e incluso necesario, automatizar todo el procedimiento. Se logra con esto no solo la exactitud y liberación de un trabajo repetitivo, sino que también se logra ahorrar cantidades considerables de tiempo, factor muy importante en una sociedad moderna como la nuestra.

Primeramente, en la sección 2.1, se describirá lo que es el proceso actual, típico caso donde el auxilio de una computadora con un buen sistema es de gran ayuda.

En la segunda parte, sección 2.2, se detallará la normatividad a la que se apegará el proceso automatizado: La propuesta del Sistema de Control de Adquisiciones, que será el apoyo informático para este departamento.

2.1 Situación actual

Los proveedores entregan las novedades bibliográficas para su revisión física, o simplemente catálogos de su fondo editorial. Cualquiera que sea el caso, se hace del conocimiento de los jefes de carrera y profesores que así lo requieran, para que sugieran la adquisición de nuevos ejemplares según una materia o carrera específica. La sugerencia también puede ser por cuenta propia de los profesores.

2.1.1 Solicitudes

Por cada título solicitado, se llena una forma con los siguientes datos:

- carrera,
- título,

- cantidad de ejemplares,
- autor, y
- editorial.

Y opcionalmente:

- observaciones,
- nombre del profesor solicitante,
- materia, y
- semestre.

La Figura 2-1 muestra una solicitud con algunos datos recabados.

ENEP ARAGÓN
BIBLIOTECA
SOLICITUD DE MATERIAL BIBLIOGRAFICO

CARRERA Industria

TITULO <u>Escritos del Alamo</u>		EJEMPLARES <u>2</u>
AUTOR <u>F. Castro</u>		EDITORIAL <u>W. H. C.</u>
OBSERVACIONES		
PROFESOR SOLICITANTE	MATERIA	SEMESTRE
NOMBRE Y FIRMA DEL COORDINADOR SELLO DE LA COORDINACIÓN FECHA	USO DE LA BIBLIOTECA	SELLO DE LA UNIDAD

Figura 2-1: Solicitud de material bibliográfico

2.1.2 Pedidos

Cuando el Departamento de Adquisiciones cuenta con las solicitudes de material bibliográfico, hace una revisión física para verificar la existencia del material bibliográfico:

- Si existe, se procede a revisar la demanda de préstamo del título, si tiene demanda se solicitan más ejemplares, pero si no tiene demanda y existen ejemplares suficientes (5), la solicitud es rechazada para compra.
- Si no existe, se procede a la compra del material con los proveedores autorizados por la DGB proporcionando listados o copias de las solicitudes de los libros en cuestión. Con sus debidas excepciones también es posible hacer pedidos a proveedores no autorizados, siempre y cuando existan razones suficientes para que así sea, tales como buenos descuentos, por ejemplo.

De acuerdo al tipo de material posiblemente sea necesario hacer más de un pedido a la vez, ya que tal vez un solo proveedor no maneje todos los títulos, ésto de acuerdo a los temas bibliográficos que ellos manejan.

Cuando se recibe el pedido del proveedor, se cateja contra el listado del pedido o los originales de las solicitudes para evitar anomalías, tales como:

- que se intente surtir dos o más veces un pedido,
- que se intente surtir con ejemplares distintos del pedido,
- que se intente surtir con ejemplares extras a los del pedido,
- para controlar cuántos ejemplares quedan pendientes de ciertos títulos, y solicitarlos posteriormente con el mismo proveedor o a uno diferente,
- o incluso, la no llegada de algunos títulos.

Con el pedido llega una o varias facturas de todos los ejemplares surtidos. La Figura 2-2 muestra una factura ejemplo.

Libros Internacionales, S.A. de C.V.

AV. SAN LUIS POTOSÍ No. 50 COL. ROMA CDMX MEXICO D.F.
TELÉFONO: 7-434-23 288-18-20

FACTURA
No 1587

A FAVOR DE: ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
DIRECCION: PLAZA DEL ARAGON, U.N.A.M. MEXICO 20
CLAVE DEL R.F.C. DEL CLIENTE

FECHA
10 DE NOV. 1994
NO SE PUEDE CANCELAR, DEBE SER DEVOLVIDO POR
EL CLIENTE EN UN PLAZO MÁXIMO DE 15 DÍAS
DESDE LA FECHA DE EMISIÓN DEL RECIBO

DESCRIPCIÓN	CANTIDAD	PRECIO UNITARIO	IMPORTE
* XSERIF: MONITOR WILDFIRE 6.0	10	69.80	698.00
* HOUSE: ACCESO RAPIDO A FILE MAGINFISH	10	39.80	398.00
* LEAD: COMO USAR LOTUS 1,2,3. TOMO I	10	49.80	498.00
* LEAD: COMO USAR LOTUS 1,2,3. TOMO II	10	49.80	498.00
* SANCHEZ: SISTEMAS EXPLICITOS-UNA METODOLOGIA	10	49.80	498.00
* QUIROGA: OLIMPIC 5 RESPUESTA RAPIDA	10	69.80	698.00
* DOMINGUEZ: LOS SISTEMAS EXPERTOS EN EMPRESA	10	39.80	398.00
* TOMAS: COMO PARA FRUCCITAMINES	10	69.80	698.00
* ZHANG: ACCESO RAPIDO OLIMPIC 5.0	10	39.80	398.00
* MAI: ACCESO RAPIDO MS DOS 6	10	39.80	398.00
* MAI: ACCESO RAPIDO PC-386/486-Y-PC9000	10	39.80	398.00
* TRUBER: ACCESO RAPIDO LOTUS 1,2,3/WINDOWS	10	39.80	398.00
* TRUBER: HARVARD GRAPHICS/PWINDOWS	10	69.80	698.00
S U M A			MS 6,674.00
DESC 20%			MS 1,334.80
TOTAL			MS 5,339.20

Figura 2-2: Factura de material surtido

2.1.3 Formas F-2


De la factura se extraen los datos más importantes y con ellos se llena una forma, llamada F-2, cuya función es la de servir de resumen. La forma F-2 maneja:

- razón social del proveedor,
- número de factura,
- cantidad de títulos por factura,
- cantidad de ejemplares por factura,
- importe por factura,
- total de títulos,
- total de ejemplares, y
- monto total de las facturas.

La forma F-2 es un documento oficial que debe llevar el visto bueno de la Dirección del plantel para autorizar al proveedor el cobro de su(s) factura(s) en la DGB. Con esta forma también se da a conocer a la Unidad Académica cómo se va ejerciendo el presupuesto. Copias de la forma F-2 y de sus respectivas facturas se archivan en expedientes anuales para respaldar futuras referencias y consultas.

La Figura 2-3 muestra una forma F-2.

FORMA F-2



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

SOLICITUD NUMERO 94/ 03
 CÓDIGO PROGRAMÁTICO: 13 01 444 01 521 06
 DEPENDENCIA: ESCUELA NACIONAL DE ESTUDIOS
 PROFESIONALES "ARAGON"
 CLAVE BIBLIOTECA DEPARTAMENTAL "ARA"
 FECHA: NOVIEMBRE 10 DE 1994

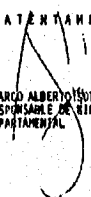
HTO. ADOLFO RODRIGUEZ GALLARDO
 DIRECTOR GENERAL DE BIBLIOTECAS
 Presenta

Agradeceré a usted giro las instrucciones necesarias para que se realice el trámite de pago de las siguientes facturas, así como el proceso técnico de los libros que amparan y de los cuales se anexan fotocopias.

PROVEEDOR	FACT. Nº	Nº. TÍTUL.	Nº. EJEMP.	IMPORTE
LIBROS INTERNACIONALES, S.A.	1578	7	70	\$ 1,213.76
LIBROS INTERNACIONALES, S.A.	1572	11	115	3,232.00
LIBROS INTERNACIONALES, S.A.	1583	10	103	2,024.00
LIBROS INTERNACIONALES, S.A.	1585	6	63	1,721.00
LIBROS INTERNACIONALES, S.A.	1587	13	119	5,323.24
TOTAL		47	670	\$ 13,512.00

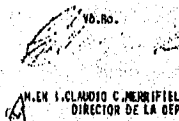
PARA EL USO EXCLUSIVO DEL PROGRAMA DE CONTROL DEL PATRIMONIO BIBLIOGRÁFICO.

ATENTAMENTE



LIC. MARCO ALBERTO VITO VELASCO
 RESPONSABLE DE BIBLIOTECA DEPARTAMENTAL

Nº. NO.



M. EN. CLAUDIO C. MERIFIELB CASTRO
 DIRECTOR DE LA DEPENDENCIA

Figura 2-3: Forma F-2

Los libros se almacenan hasta que la DGB suministra los datos de identidad, los cuales son: la Clasificación, el Número de Adquisición y el Número de Matriz, esta información llega en forma de listados junto con las tarjetas que irán a los ficheros del área de adquisiciones. Existe la posibilidad de enlazar vía correo electrónico a la ENEP Aragón con la DGB para acceder directamente estos datos.

Entonces los libros pasan al Departamento de Procesos Técnicos donde se les inscriben sus datos de identidad, se sellan y protegen magnéticamente. Finalmente pasan al acervo general o al área correspondiente.

Entre los requerimientos más frecuentes de consultas de información se tiene:

- Obtención del ejercicio del presupuesto, ya sea hasta la fecha actual o en un periodo de tiempo, un mes por ejemplo. Esto para indagar la forma en que se está ejerciendo el presupuesto, bien para saber si se ha gastado demasiado, o si hay suficientes fondos para el resto del año. Es información que permite conocer si el presupuesto se va empleando de una manera razonable.
- Un informe por carrera de material adquirido y no adquirido, el cual involucra el material que ha sido surtido y el que está en espera de ser surtido, respectivamente. Aquí se indican los datos principales de cada título, los cuales son:
 - título,
 - autor,
 - editorial,
 - número de ejemplares,
 - observaciones, y
 - por supuesto, la carrera.
- Un concentrado por carrera de material adquirido y no adquirido, el cual sólo contempla los siguientes datos:
 - carrera,
 - cantidad de ejemplares surtidos, y
 - cantidad de ejemplares no surtidos.

La obtención de estas consultas es una labor muy minuciosa que precisa una gran atención por parte del personal encargado de tales tareas.

En la siguiente sección se describe la normatividad a la cual se ajustará todo este proceso manual. Al establecer reglas claras y lineamientos específicos se logrará

que la situación actual sea parte del pasado pues ya será factible introducir un sistema informático.

Todo sistema manual que se automatiza llega a parecer más complejo de lo que aparentaba, ya que es más rígido en sus procesos, por lo que es más difícil que se cometan errores derivados de un mal encauzamiento y/o falta de datos.

Los objetivos del sistema son:

- Controlar la adquisición de nuevos títulos, y
- Hacer el seguimiento del ejercicio del presupuesto.

2.2 Propuesta del Sistema de Control de Adquisiciones

Para lograr sus objetivos, el sistema almacena las Solicitudes, Pedidos y el resumen de las facturas, la Forma F-2. Esta documentación tiene diferentes estados que pueden indicar diversas circunstancias.

Por lo tanto, el sistema se divide en el manejo de estas áreas claves. Se logra crear pequeños módulos de manejo de información, los cuales son:

- Solicitudes,
- Pedidos, y
- Formas F-2.

Por una parte, permite manejar los documentos como tales, es decir, sus altas, bajas y cambios. También permite hacer consultas cuantitativas.

Se le puede indicar al sistema el año lectivo (activo) para sus operaciones, de tal manera que se puedan repetir números de solicitudes, pedidos y formas F-2, distinguiéndolos por el año de la fecha de su captura.

Además, el sistema cuenta con cinco catálogos:

- Proveedores
Lista de Proveedores.

- Temas
Lista de Temas Bibliográficos.
- Carreras
Lista de Carreras de la ENEP Aragón.
- Datos Semifijos
Información que puede variar cada año o menos, tal como el presupuesto o el nombre del director de la ENEP Aragón, por ejemplo.
- Usuarios
Lista de Usuarios autorizados del Sistema.

En los siguientes apartados se analizará el manejo de cada uno de los documentos, de las consultas disponibles y una descripción de los elementos de los catálogos del sistema.

2.2.1 Solicitudes

Una solicitud es el documento que sirve de base para levantar pedidos de nuevos títulos para acrecentar el acervo bibliográfico. Es un documento enviado por profesores o jefes de carrera solicitando la adquisición de bibliografía.

Una solicitud puede tener uno de los tres estados siguientes:

- S1 - Considerada
Para ser considerada en el próximo pedido. Con este estado inicia una solicitud recién capturada e indica que entrará automáticamente en el próximo pedido que se genere.
- S2 - En Espera
Se encuentra en espera de ser incluida en algún pedido. Porque cuando se genere un pedido, el sistema preguntará si se desean incluir todas, algunas o ninguna de las solicitudes en este estado.
- S3 - No Considerada
Ya no es considerada en el próximo pedido. Cuando la solicitud entra en un pedido, ya no es considerada para ningún otro y automáticamente es colocada en este estado. También es posible que el usuario seleccione manualmente esta opción.

Todas las solicitudes que entren al pedido pasan al estado S3 - No Considerada, para que así el sistema no las vuelva a incluir en un pedido. También la solicitud registrará el número de pedido del cual forma parte, por lo que si se quiere cambiar a otro estado o se desea eliminar, es necesario liberarla desde el pedido, es decir, abrir el pedido correspondiente y excluir la solicitud.

2.2.2 Pedidos

Un pedido es la reunión de varias solicitudes que forman una orden de compra para un proveedor.

Un pedido puede estar en uno de cinco estados:

- P1 - No Emitido
Se acaba de generar y no se ha levantado el pedido.
- P2 - En Espera
Se hizo el pedido pero aún no se reciben los ejemplares.
- P3 - Surtido
Se recibieron los ejemplares.
- P4 - Pendiente
Existen problemas en el surtido, por lo que permanecerá pendiente hasta que el proveedor corrija las deficiencias en el pedido.
- P5 - Cancelado
Ya no se desea como pedido.

La única manera de dar de Alta un pedido es a través de su generación. Este proceso toma todas las solicitudes que existan en el estado S1 - Considerada, por lo que en realidad se puede crear más de un pedido a la vez ya que las solicitudes se agrupan por proveedor y por cada uno de ellos se creará un pedido.

Es decir, la generación de un pedido depende de que existan o no solicitudes en el estado apropiado, lo cual significa que será necesario capturar las solicitudes que entrarán en el siguiente pedido porque todas ellas conformarán ese nuevo pedido, por supuesto que opcionalmente podrán ser incluidas las solicitudes En Espera.

Esta característica de tomar de una sola vez todas las solicitudes Consideradas para un nuevo pedido, hace imposible que se pueda capturar material para pedidos que se pretendan generar en diferentes fechas.

Respecto a las solicitudes En Espera, al momento de generar el nuevo pedido se preguntará que si se desean incluir todas ellas, preguntar una a una o simplemente no incluirlas. Si se decide por incluirlas todas, se incluyen transparentemente y ya no se solicitará información extra. Pero si se decide por la opción de preguntar una a una, se visualizarán los datos más importantes de la solicitud (En Espera) en turno y se pedirá confirmación sobre su posible inclusión con un sí o un no, y esta operación se repite por cada solicitud que se encuentre en ese estado.

Al finalizar la generación de los pedidos, el sistema presenta una lista de los pedidos generados con la opción de poder imprimirla. Todos estos pedidos quedan marcados con el estado P1 - No Emitido.

Para que un pedido en estado P1 - No Emitido pase al estado P2 - En Espera será necesario hacer el cambio manualmente, ya que tal vez se imprima el pedido recién elaborado sólo para fines de revisión.

Es posible colocar en cualquier estado a un pedido, excepto cuando ya ha sido surtido.

Es posible que un pedido sea eliminado, pero el sistema preguntará qué se hará con las solicitudes involucradas, ya que es posible regresarlas a su estado original, al S2 - En Espera o al S1 - Considerada, también se podrán eliminar definitivamente de la relación de solicitudes.

Estas mismas alternativas se presentarán cuando se intente eliminar (excluir) alguna solicitud del pedido.

Para poder recibir un pedido es necesario colocarlo en el estado P2 - En Espera o en P4 - Pendiente. Para hacer la recepción, las solicitudes aparecen con sus datos más importantes en forma de tabla donde se deberá indicar una a una la cantidad surtida. También se deberá especificar el Número de la Factura que ampara a las solicitudes.

Con esta información ya es posible indicarle al sistema que verifique posibles diferencias por cada título.

Para comenzar, si el sistema detecta irregularidades en el surtido, pregunta qué se desea hacer con el pedido, ya que se puede continuar con la recepción, o simplemente cancelarlo, dejarlo pendiente o eliminarlo.

Al continuar la recepción de un pedido con problemas pueden suceder tres casos:

1. Que lleguen ejemplares de menos, que a su vez presenta tres opciones:

1. Hacer que la solicitud indique la cantidad surtida, con lo que ya no habría diferencias.
2. Retirar la solicitud del pedido y mantenerla En Espera para que posteriormente pueda ser incluida en otro pedido.
3. Hacer que la solicitud indique la cantidad surtida y con la diferencia crear una solicitud En Espera para que posteriormente pueda ser incluida en otro pedido.

2. Que lleguen ejemplares de más, que a su vez presenta dos opciones:

1. Hacer que la solicitud indique la cantidad surtida, con lo que ya no habría diferencias.
2. Retirar la solicitud del pedido y mantenerla En Espera para que posteriormente pueda ser incluida en otro pedido.

3. Que no se surtan ejemplares, que a su vez presenta dos alternativas:

1. Retirar la solicitud del pedido y mantenerla En Espera para que posteriormente pueda ser incluida en otro pedido.
2. Retirar la solicitud del pedido y eliminarla de la relación de solicitudes ya que, tal vez, no se desea contar ni con su referencia.

Se deberá proseguir así hasta la última solicitud que tenga problemas. Al final, el pedido pasa al estado P3 - Surtido y se almacena con la fecha de recepción. Si se crearon nuevas solicitudes, el sistema las presenta en forma de lista con opción de imprimirla.

2.2.3 Formas F-2

Con las facturas se elaboran las Formas F-2 con las cuales es posible que el proveedor solicite su pago en la Dirección General de Bibliotecas.

Una Forma F-2 tiene dos estados:

- F1 - Aceptada
Es considerada para el ejercicio del presupuesto.
- F2 - Cancelada
No es considerada para el ejercicio del presupuesto.

Manejan el detalle del total de títulos, total de ejemplares, importe total, número de factura o nota de crédito (si hay alguna) y el nombre del proveedor.

Con las Formas F-2 es posible averiguar el ejercicio del presupuesto, ya sea desde el inicio del año o por periodos.

Como máximo cada Forma F-2 puede contar con 7 facturas y una nota de crédito, esta última sólo en caso de requerirse.

2.2.4 Consultas

Las consultas permiten, generalmente, obtener totales sobre el estado o cantidad de documentos, desglosarlos por carrera, obtener el ejercicio del presupuesto, o incluso listados de material adquirido y no adquirido.

En la mayoría de los casos se podrá hacer una consulta en un periodo de fechas. Pero si se desea consultar desde el inicio del año lectivo hasta el día de la consulta, no será necesario indicar ningún periodo. Es decir, el sistema sólo maneja documentos del año lectivo, con sus debidas excepciones, como en el caso de pedidos pendientes de, por ejemplo, un año anterior al lectivo.

Existen consultas para los tres tipos de documentos.

Consultas a solicitudes

Por estado

Se obtienen totales de la cantidad de solicitudes en sus diferentes estados:

Consideradas
En Espera
No Consideradas

Con un total de todas ellas.

Por carrera

Se obtienen totales de la cantidad de solicitudes por carrera en sus diferentes estados:

Consideradas
En Espera
No Consideradas

Con un total por cada carrera y una sumatoria global.

Por número de Factura

Se obtiene una relación de títulos que entran en la factura que se especifique. Los datos desplegados son:

Número de solicitud
Título
Autor
Cantidad de ejemplares

Consultas a pedidos

Por estado

Se obtienen totales de la cantidad de pedidos en sus diferentes estados:

Surtidos
En Espera
Pendientes

No Emitidos
Cancelados

Con un total de todos ellos.

Material solicitado por carrera

Se obtienen totales de la cantidad de títulos y ejemplares por carrera de todas las solicitudes que se encuentran en los pedidos en sus diferentes estados excepto canceladas.

Por Fecha de Pedido

Indica el número de pedido y su estado. La lista se ordena cronológicamente. Incluye todos los de este año lectivo y los de años anteriores en estado de En Espera o Pendientes.

Por Fecha de Surtido

Indica el número de pedido de todos aquellos surtidos en el año lectivo o en el periodo que se indique.

Ejemplares por carrera

Aquí se obtiene básicamente información de Material Adquirido y de Material No Adquirido. El Material Adquirido está conformado por todos los pedidos surtidos en el año lectivo. El Material No Adquirido incluye los pedidos en estado No Emitido, En Espera y Pendientes, todos éstos del año lectivo.

La consulta puede ser de todas las carreras a la vez o alguna en específico. Los datos son:

Carrera
Título
Autor
Editorial
Cantidad de ejemplares
Observaciones

Esta consulta contempla una subconsulta a manera de resumen donde sólo aparecen totales de títulos, cantidad de ejemplares por cada carrera y un total global.

El resumen toma las condiciones establecidas para la consulta normal, por lo que es necesario primero hacer la consulta normal.

Consultas a formas F-2

Por estado

Se obtienen totales de la cantidad de formas F-2 en sus diferentes estados:

Aceptadas
Canceladas

Con un total de los dos estados.

Corte de Ejercicio

Aquí se obtienen dos cifras:

Cantidad ejercida a la fecha
Cantidad por ejercer

La cantidad ejercida se contabiliza al sumar los importes de todas las formas F-2 del año lectivo o de sólo aquellas que se lleguen a indicar en el rango de fechas.

La cantidad por ejercer es la diferencia del total del presupuesto menos lo ejercido.

El monto real por ejercer se observará si el rango se deja abierto, ya que de esta manera se contabiliza el total de lo gastado en el año lectivo.

Por supuesto, sólo entran en la contabilización todas las formas F-2 en estado Aceptadas.

2.2.5 Catálogos

Los catálogos son datos que se mantienen más o menos fijos en el sistema. Se dividen en cinco áreas para un mejor control de la información.

Proveedores

Se asigna una clave y se captura su nombre, RFC, dirección y todos los datos que permiten saber qué tipo de proveedor es y cuáles temas bibliográficos maneja.

Temas bibliográficos

Los temas bibliográficos sólo constan de una clave y una descripción. Es una lista de todos los temas que maneja la UNAM.

Carreras

Es una lista de todas las carreras impartidas en la ENEP Aragón. Se componen de una clave y una descripción.

Datos semifijos

Los datos semifijos indican los datos que pudieran variar año con año como el Código Programático, o de un mes para otro tal como los nombres de los Directores o, incluso, el propio presupuesto. Es aquí donde se indica el año lectivo y el cual será tomado por el sistema para el cálculo de consultas y también en el manejo de la captura de los documentos.

Usuarios del sistema

Aquí se dan de alta a los usuarios autorizados del sistema. Para seguridad del sistema se les asigna una palabra clave secreta que deberán introducir cuando se les solicite, ya sea al inicio del sistema o al intentar entrar en una sección restringida.

El usuario principal será el Jefe de la Biblioteca y es quien tiene prioridad sobre los demás, de tal manera que es el único que puede agregar, modificar o eliminar a un usuario.

Finalmente, la Figura 2-4 muestra el diagrama de flujo de datos según la nueva normatividad.

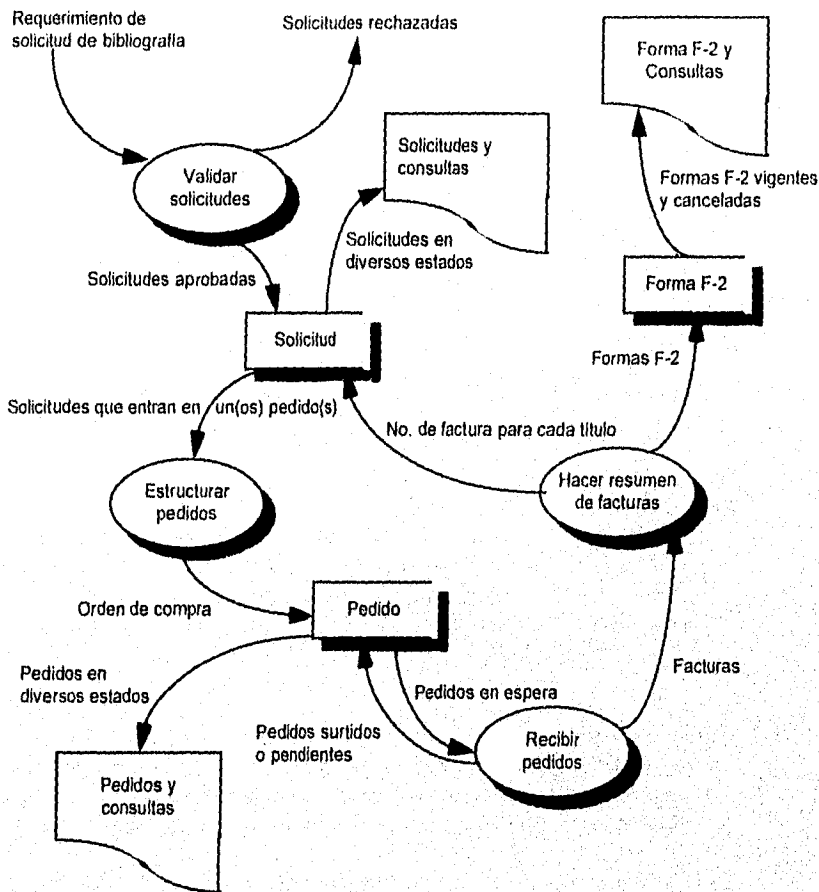


Figura 2-4: Diagrama de flujo de datos para el Sistema de Control de Adquisiciones

Capítulo 3 SQLWindows 5.0

Objetivo General

- Conocer diversos aspectos que conforman a SQLWindows 5.0 y ver cómo está orientado a la filosofía de objetos y comprender que va más allá de un simple lenguaje de programación

Objetivos Particulares

- Entender la teoría de la programación orientada a objetos
- Conocer los objetos con los cuales viene dotado SQLWindows
- Revisar el lenguaje de programación de SQLWindows
- Conocer los diversos tipos de mensajes que puede manejar SQLWindows
- Conocer aspectos fundamentales para el manejo de SQL desde una aplicación de SQLWindows

La programación ha evolucionado desde la programación por cableado al lenguaje ensamblador, pasando después por los lenguajes de alto nivel haciendo uso de la programación estructurada.

Y aunque era un enfoque muy práctico para el diseño, cualquier sistema partía de la nada ya que no existía un enfoque de aprovechamiento de lo desarrollado anteriormente. Casi no había reutilización.

"Los investigadores, conociendo el dilema, comienzan a enfocar esfuerzos a esta problemática a principios de los setentas. La teoría de programación orientada a objetos se desarrolla como respuesta coherente. PARC Xerox, en 1972, presenta un lenguaje de programación al que llamó Smalltalk 72. Diseñado con características especiales en mente, introduce conceptos 'orientados a los objetos' como se ha dado en llamarles..."¹

En las siguientes secciones se hablará de todas las características más sobresalientes de SQL Windows. En la sección 3.1 se analiza la teoría de la programación orientada a objetos, esto como preámbulo a la sección 3.2 donde se hace una revisión de los objetos con los que viene dotado SQL Windows.

En la sección 3.3 se estudia el lenguaje SAL, el lenguaje de programación de SQL Windows. Se observará un estilo muy particular para codificar con este lenguaje.

Como complemento al SAL, en la sección 3.4 se conocen los diversos tipos de mensajes que puede manejar SQL Windows.

En la última sección, 3.5, se hace un estudio de aspectos básicos para la utilización del lenguaje SQL para el manejo de bases de datos desde una aplicación de SQL Windows.

¹ Abel A.P. y Jorge M.M., Programación Orientada a Objetos en un Ambiente Puro, PC/IPS BYTE, México, Mayo 1992, pp 92-97.

3.1 Objetos

3.1.1 ¿Qué son los objetos?

El diseño orientado a objetos descompone un sistema en objetos, los componentes básicos del diseño.

Se intenta manejar la complejidad de los problemas del mundo real por la **abstracción** del conocimiento y **encapsulando** éste dentro de objetos. El encontrar o crear esos objetos es el problema de la estructuración del conocimiento y sus actividades.

Se podría decir que los objetos en un problema son entidades independientes que se relacionan con otras partes del problema. Estas entidades no se eligen o surgen por su fácil o difícil programación, sino porque tienen algún límite físico o conceptual que los separa del resto del problema. Las entidades son representadas como objetos en el programa para computadora.

El objetivo en la programación orientada a los objetos es tener una correspondencia, una a una, entre las entidades en el problema físico y los objetos en el programa.

La información normalmente se maneja en dos tipos distintos: funciones y datos. En la programación procedimental inmediatamente se trabaja sobre las funciones y las particularidades de los datos que van a intervenir. Es decir, se maneja el **¿Cómo...?**

La programación orientada a objetos comienza con una idea más abstracta: el **¿Qué...?**, y no con el **¿Cómo...?**. Primero se averigua sobre lo que intenta el programa. Se buscan los objetos y sus conexiones; para lograrlo se averigua qué operaciones se necesitan desarrollar y qué información resulta de esas operaciones. Cada objeto sabe lo que puede hacer (operaciones) y sobre qué lo puede hacer (datos).

Se llega a descomponer un sistema en entidades que se ajustan a sus propias reglas dentro del sistema porque conocen todo sobre sí mismas.

Abstracción

La gente comprende el mundo al modelar porciones de él en versiones simplificadas. Un modelo debe ser un bosquejo y no debe intentar manejar hasta el último detalle, ya que de lo contrario resulta igual de complejo que lo que se intenta

modelar y entender. Lo que se busca es simplificar, lo cual es crucial para entender al mundo.

Considerar, por ejemplo, un automóvil. Una persona puede conducir un automóvil sin necesidad de comprender muchos de los detalles que implican su funcionamiento; dicho de otro modo, abstrae el conjunto de cientos de piezas distintas a una sola idea: un automóvil.

Es por esto que la abstracción es la clave para diseñar un buen software.

Encapsulación

Dos de las ideas principales son: ¿Qué puede hacer este objeto? y ¿Qué conoce este objeto?. La primera pregunta se refiere a las operaciones o funciones que puede realizar y la segunda pregunta se refiere a los datos que se implican en esas operaciones.

El hecho de reunir en un solo objeto tanto los datos como las operaciones que afectan esos datos se denomina **encapsulación**, Figura 3-1. Se logra así ocultar cierta complejidad hacia otros objetos y hacia el diseñador. Esta complejidad es el **cómo** hace las cosas y esto no importa a otras partes del sistema, lo que les interesa es el **qué** puede hacer. El **qué** es la parte **pública**, lo único que los otros objetos **deben** conocer.

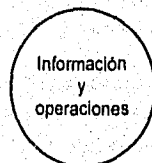


Figura 3-1: Encapsulación

Volviendo al ejemplo del automóvil, al conductor no le interesa cómo funciona, lo importante es que lo puede transportar de un punto a otro en esta gran ciudad.

Los objetos se pueden ver como algo que reúne cierto conocimiento y ciertas operaciones conceptualmente relacionadas a ese conocimiento. Continuando con el ejemplo del automóvil, al agrupar cientos de piezas grandes y pequeñas en una sola unidad se logra que "el mucho" sea visto como una unidad fácilmente manipulada.

Se logra reducir la complejidad y facilitar la conceptualización al hacer innecesario el conocimiento de principios físicos que intervienen en su funcionamiento.

Ocultamiento de la información

Al manejar la complejidad en una caja negra debe haber alguna manera que indique lo que algún objeto en particular puede hacer. Es decir, se está manejando una **representación privada** y una **interfaz pública**, a este efecto se le conoce como **ocultamiento de la información**, y se representa gráficamente en la Figura 3-2.

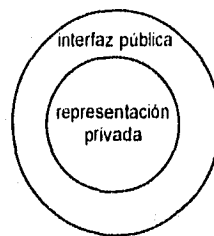


Figura 3-2. Ocultamiento de la información

Esto logra que el diseño sea más fácilmente modificado.

Al ocultar la información, un objeto no dice cómo hace su trabajo, pero públicamente se sabe lo que el objeto puede hacer.

Por otro lado, como un objeto tiene una parte privada, ésta puede ser modificada sin alterar el equilibrio del sistema, ya que si un error es introducido, éste queda enclaustrado en la parte privada del objeto y es más fácil su detección y corrección.

3.1.2 ¿Cómo interactúan los objetos?

Un objeto se puede comunicar con otro mandando un mensaje. A esta acción se le conoce como **transmisión de mensaje**, y es la única manera en que un objeto se puede comunicar con otro. Esto se muestra en la Figura 3-3.

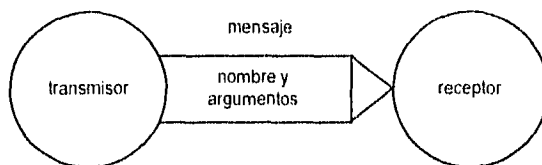


Figura 3-3: Transmisión de un mensaje

Mensaje

Un **mensaje** es el nombre de la operación y tal vez algunos argumentos que necesita la operación.

El conjunto de mensajes a los que un objeto puede responder es conocido como **comportamiento** o **funcionamiento** del objeto. Para abreviar, el nombre de la operación sin considerar los argumentos es conocido como **nombre del mensaje**.

Método

Un **método** es la función ejecutada en respuesta a un mensaje. Dentro del objeto, el nombre del método es el nombre del mensaje.

Entidad

Una **entidad** es la relación del nombre del método, el tipo de argumentos que requiere y el tipo de dato que el método retorna como respuesta. La entidad determina las entradas y salidas de un método.

3.1.3 Clases e instancias

Existen muchos objetos que son bastante similares. Así, por ejemplo, tenemos automóviles, camionetas, camiones, motocicletas, bicicletas, etc. Aunque de diferente manera, pero pueden transportar personas y/o cosas.

Clase

Los objetos que tienen el mismo comportamiento son de la misma **clase**. Están creados en base a un mismo patrón.

Instancia

Cualquier objeto derivado de una clase es llamado *instancia* de esa clase. Es capaz de recibir mensajes y atenderlos si se encuentran dentro de la gama de sus conocimientos.

3.1.4 Polimorfismo

La palabra **polimorfismo** significa "habilidad para asumir varias formas". Aplicando esta idea a los objetos se hablaría de la habilidad de que dos o más métodos de un objeto respondan al mismo mensaje.

Esto indica que las interfaces son similares y que las operaciones (funciones) están implementadas en diferente forma, de tal manera que respondan a un mensaje de un **transmisor** sin que éste se preocupe por el método que ejecutará a ese mensaje.

Regresando de nuevo al ejemplo del transporte, suponer que un cliente establece un contrato con una empresa de paquetería. Si el cliente envía una pequeña caja, la empresa tal vez la remita en una motocicleta. Suponiendo ahora que el cliente envía unas 1,000 cajas, lo más probable es que la empresa utilice camiones de carga.

Este es un claro ejemplo de polimorfismo, al cliente no le interesa el medio por el cual sea enviada su mercancía, lo importante es que llegue a su destino —en el tiempo prometido. La empresa cuenta con un departamento de atención al cliente (**Interfaz pública**) donde se recibe la orden de envío, (**mensaje**), al percatarse de la magnitud del envío la empresa decide utilizar motocicletas o camiones de carga. Dependiendo de la cantidad de mercancía, el **método** de transporte cambia. Figura 3-4.

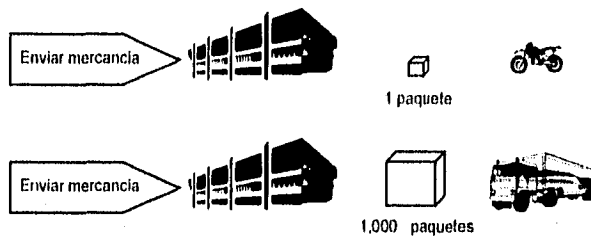


Figura 3-4: Un mensaje polimórfico: enviar mercancia

3.1.5 Herencia

La herencia permite derivar una nueva clase a partir de otra, para así agregar nuevas funciones, hacer algunas modificaciones o imponer algunas restricciones.

La herencia permite reutilizar código. No se necesita volver a escribir todo el código, indirectamente se toma (se hereda) el que ya se escribió y se agrega el nuevo.

Subclase

Una subclase es una clase que hereda las características de otra llamada **superclase** o **clase base**.

Retomando el ejemplo del transporte, se podría decir que la superclase es el automóvil con el equipo de norma: un juego de cuatro llantas, un parabrisas, un volante, un motor, etc. Un camión de carga es en sí un automóvil, pues la mayoría de sus características (comportamiento) ha sido tomado de un automóvil, se han modificado algunos de los componentes, se agregaron o eliminaron otros. Todo es más grande, ya no son cuatro llantas, ahora son seis, diez o más; el motor puede utilizar otro tipo de combustible; se le acondiciona un gran espacio para la carga. Se agregan o eliminan accesorios de acuerdo con el funcionamiento requerido. Figura 3-5.

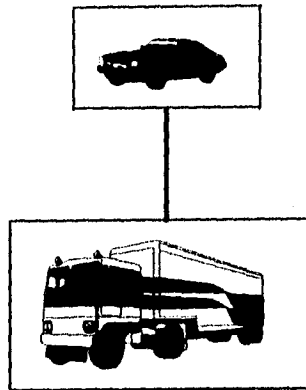


Figura 3-5: Superclase y subclase

Siguiendo con el ejemplo del automóvil, suponer ahora que se tiene la superclase automóvil y la superclase bicicleta, a partir de aquí podemos tener la subclase motocicleta. Figura 3-6.

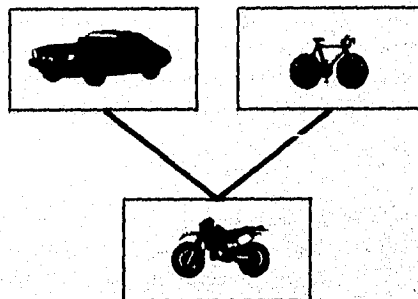


Figura 3-6: Una subclase con dos superclases

Se puede tener otra subclase a partir de la superclase automóvil. Figura 3-7.

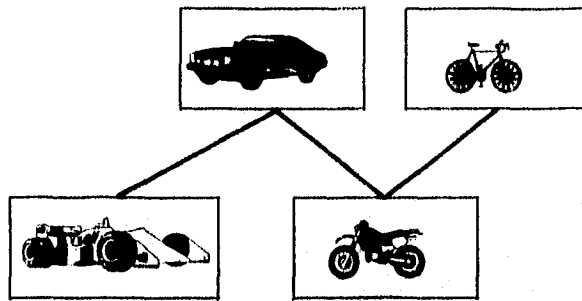


Figura 3-7: Jerarquía de la herencia

Superclase

Una **superclase** es la clase de la cual se hereda su comportamiento. Una clase puede tener una o varias superclases, de esta manera al poner limitaciones y hacer modificaciones en las características heredadas se tiene una nueva clase con vida propia. Una subclase puede ser la superclase para otras subclases.

En el ejemplo anterior, la clase motocicleta tiene dos superclases; automóvil y bicicleta. Por un lado se encuentra la capacidad de transportarse con movimiento automático y con muchas comodidades, por otro lado, se tiene lo práctico de transportarse en un vehículo pequeño aunque se mueva gracias a la acción humana. Al combinar cualidades se obtiene un vehículo impulsado por un motor de combustión interna y se conserva lo práctico al contar con solo dos llantas.

Clase abstracta

Una **clase abstracta** es una clase que sólo sirve para heredar y no para crear instancias. Es una clase que provee de varias funcionalidades en forma de un prototipo, es decir, la subclase que la herede deberá implementar algunos de los métodos para sus específicos propósitos.

Esto logra que el código sea reutilizable. Por otro lado, a las clases que permiten definir instancias de ellas se les conoce como **clases concretas**.

Reutilizando una vez más el ejemplo del transporte, suponer que se ha diseñado la clase abstracta automóvil, a la cual se le han definido todas sus características mínimas, tales como sistema de transmisión, sistema eléctrico, un motor,

suspensión, juego de asientos, chasis, carrocería, entre otras cosas. A partir de esta clase abstracta se obtendrán dos subclases concretas: un automóvil de carreras y un automóvil sedán. Figura 3-8.

Cada una de estas subclases tomará lo que necesite, modificará o añadirá otras características.

El automóvil de carreras sólo tendrá un asiento, la suspensión será especial para las altas velocidades, la carrocería debe ser aerodinámica, el motor de combustión interna de cuando menos 8 cilindros y de un combustible especial, el chasis aunque ligero debe ser muy resistente, esto sin enumerar otras posibilidades.

El automóvil sedán debe tener espacio para cuando menos cinco personas, la suspensión apropiada para la ciudad con sus topes y baches, una carrocería medianamente aerodinámica y muy atractiva a la vista, su motor pudiera ser eléctrico o de combustión interna de unos 4 cilindros.

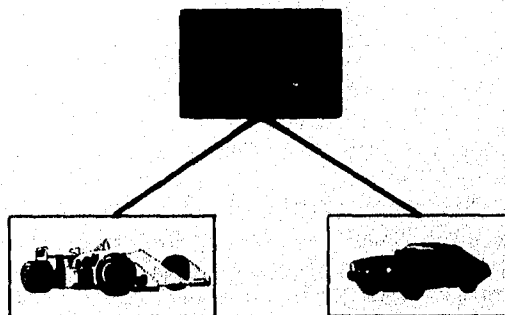


Figura 3-8: Jerarquía de clases de automóviles

3.2 Objetos de SQLWindows

Los objetos de SQLWindows son piezas visuales para componer la interfaz gráfica con el usuario. Los objetos pueden ser tomados directamente de la paleta de herramientas, del menú Tools, de la barra de opciones, o pueden ser cortados,

copiados y pegados desde los objetos existentes en el diseño. También es posible escribirlos directamente en el editor de código.

Al colocar un objeto en el diseño, ya viene con una serie de características estándar, pero es posible modificarlas con el personalizador y así definir nuevos atributos. Para desplegar el personalizador se puede proceder así:

- Con la herramienta Grabber (Sujetar) que parece una cruz con cuatro flechas, hacer doble clic o clic-derecho en el objeto.
- Con un clic en el objeto con el Grabber y presionando Retorno.
- Con un doble clic o un clic-derecho en el icono Personalizador del objeto en el editor de código. El icono Personalizador se encuentra a la izquierda del objeto en el editor de código. Figura 3-9.

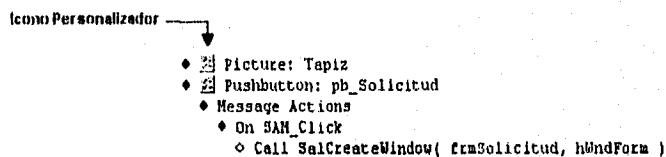


Figura 3-9: Icono personalizador

Se recomienda colocar un prefijo al nombre de los objetos como indicador del tipo de objeto de que se trata. De esta manera se tiene una noción de lo que se espera del objeto, independientemente de lo que su nombre pueda sugerir. Cuando se agrega un objeto al diseño, SQL Windows da un nombre por defecto usando uno de los prefijos listados enseguida. Se utilizará el nombre en inglés pues así es como se le encontrará en todo el diseño y ofrece una idea más precisa de su función, ya que traducidos -según Windows en español- es difícil conectar conceptualmente su nombre en inglés y su significado en español, por ejemplo, un "Push Button" es un "botón de comando" y un "Radio Button" es un "botón de opción".

Objeto	Prefijo	Ejemplo
Form Window	frm	frmNombre
Table Window	tbl	tblNombre
Quest Window	tbl	tblNombre
Dialog Box	dlg	dlgNombre
MDI Window	mdi	mdiNombre
Data Field	df	dfNombre
Multiline Field	ml	mlNombre
Push Button	pb	pbNombre
Radio Button	rb	rbNombre
Check Box	cb	cbNombre
Option Button	ob	obNombre
List Box	lb	lbNombre
Combo Box	cmb	cmbNombre
Picture	pic	picNombre
Horizontal Scroll Bar	hsb	hsbNombre
Vertical Scroll Bar	vsb	vsbNombre
Custom Control	cc	ccNombre
Column	col	colNombre

Existen dos tipos de objetos:

- Alto nivel
- Hijos

Además se pueden utilizar los diálogos estándar de Windows en las aplicaciones de SQL Windows.

3.2.1 Objetos de alto nivel

Hay cuatro objetos de alto nivel:

- Form Window
- Dialog Box
- Table Window
- Quest Window

Se les llama de "alto nivel" por la posición que ocupan en el editor de código. Una aplicación contiene al menos uno de estos objetos. Las Table Windows y QuestWindows también pueden ser objetos hijos.

Todos los objetos de alto nivel tienen:

- Un borde y una barra de título
- Un menú de sistema

y opcionalmente

- Una barra de herramientas
- Una barra de estado

Todos los objetos de alto nivel (excepto los Dialog Box) pueden:

- Tener un menú
- Tener botones para maximizar y minimizar
- Ser creados automáticamente cuando la aplicación inicia o ser creados dinámicamente por la aplicación. Un Dialog Box sólo se crea dinámicamente
- Tener un estado inicial (normal, minimizado, o maximizado)
- Tener un icono que es desplegado cuando la ventana es minimizada
- Cambiar de tamaño en tiempo de ejecución

En el editor de código, todos los objetos de alto nivel tienen:

- Una sección de Contenidos (Contents) donde se pueden agregar objetos hijos (excepto para las QuestWindows)
- Una sección de Acciones de Mensajes (Message Actions) donde se codifican acciones para que sean ejecutadas
- Una sección de Variables de la Ventana (Window Variables) para declarar las variables locales a la ventana
- Una sección de Funciones (Functions) donde se implementarán las diversas funciones que necesite la ventana para llevar a cabo su función. Cada una de las funciones deberá contar con su código en la sección de Actions

Existe un objeto especial que se encuentra en el editor de código a la altura de los objetos de alto nivel, se le conoce como MDI Window y puede contener (en su sección de contenidos) una o más Form Windows o Table Windows de alto nivel.

En seguida se hará una explicación breve de cada uno de los objetos de alto nivel.

Form Window

Las Form Windows se pueden utilizar para introducir y desplegar datos. Se pueden colocar objetos hijos tales como Data Fields, Push Buttons, etiquetas y demás objetos requeridos. Figura 3-10.

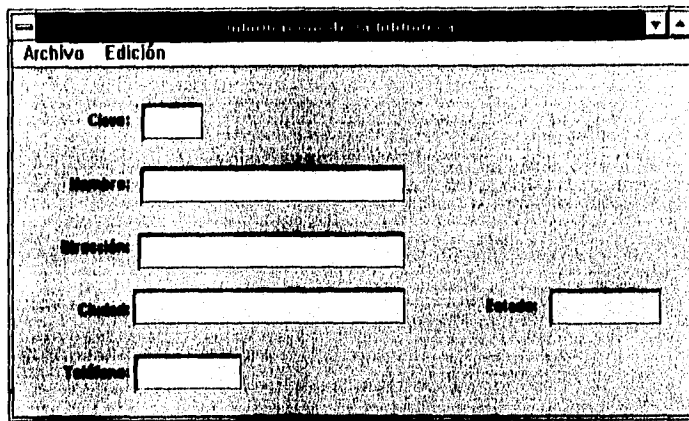


Figura 3-10: Una Form Window con menús, Data Fields y Background Text

Dialog Box

Los Dialog Box se pueden utilizar para permitir que el usuario introduzca datos, o para desplegar alertas o mensajes de error. Se pueden colocar los mismos tipos de objetos hijos que en la Form Window. Figura 3-11.

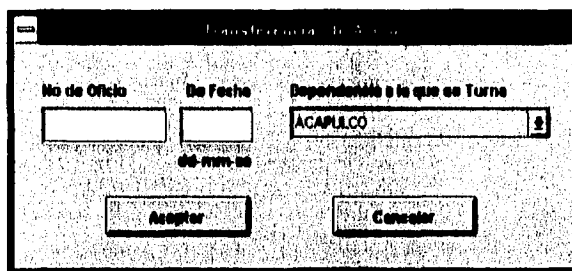


Figura 3-11: Dialog Box

Existen tres tipos de Dialog Box: sin modo, modal y modal del sistema.

Sin modo

Un Dialog Box sin modo no suspende el proceso de la aplicación. El usuario puede moverse entre el Dialog Box y otra ventana en la aplicación o a una ventana en otra aplicación.

Modal

Un Dialog Box modal suspende el proceso de la aplicación hasta que el usuario cierra el Dialog Box. El usuario no puede moverse del Dialog Box a otra ventana en la aplicación. Sin embargo, el usuario sí puede moverse a una ventana en otra aplicación.

Modal del sistema

Un Dialog Box modal del sistema suspende el proceso en todo el sistema hasta que el usuario cierre el Dialog Box. El usuario no se puede mover a otra ventana en la aplicación o a otra ventana en otra aplicación.

Dialog Box con dueño o sin dueño

Se pueden crear Dialog Boxes con o sin un dueño. Cuando un Dialog Box tiene un dueño:

- Siempre permanece sobre la ventana dueña.
- Cuando la ventana dueña se cierra, el Dialog Box se cierra.
- Cuando el usuario minimiza al dueño, el Dialog Box desaparece y cuando el usuario restablece el dueño, el Dialog Box reaparece.

Message Box (Cuadro de Mensaje)

Si la aplicación sólo necesita desplegar un simple mensaje o necesita una entrada simple por parte del usuario, se puede utilizar una Message Box que no requiere más que indicar qué título, mensaje, botones e iconos debe llevar. Figura 3-12.

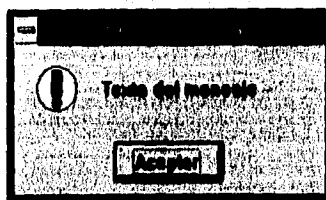


Figura 3-12: Message Box

Table Window

Una Table Window despliega datos en un formato de renglones y columnas, como una hoja de cálculo. A través de una Table Window un usuario puede:

- Desplegar una consulta (de SQL)
- Moverse a través de los renglones de datos
- Insertar, actualizar, o borrar renglones de datos

Las Table Windows tienen muchas características que dan flexibilidad al desarrollador al crear aplicaciones sofisticadas. Además cuenta con las mismas características de una Form Window (título, menú, y un icono).

Una Table Window es una matriz de renglones (líneas) y una o más columnas. La intersección de un renglón y una columna se llama celda. Figura 3-13.

NO. SOLICITUD	FECHA	CLAVE PROV	CLAVE CARRERA
00201	1995-06-09	LIN32	002
00153	1995-05-31	LAR92	012
00050	1995-02-28	LAB62	012
00134	1995-03-07	ELU66	009
00140	1995-05-30	ENT	012
00040	1995-02-28	LAB62	012
00090	1995-03-08	LIN32	005
00124	1995-03-07	ELU66	009
00080	1995-03-08	LIN32	005

Una línea horizontal completa de celdas es un RENGLÓN

Una línea vertical completa de celdas es una COLUMNA

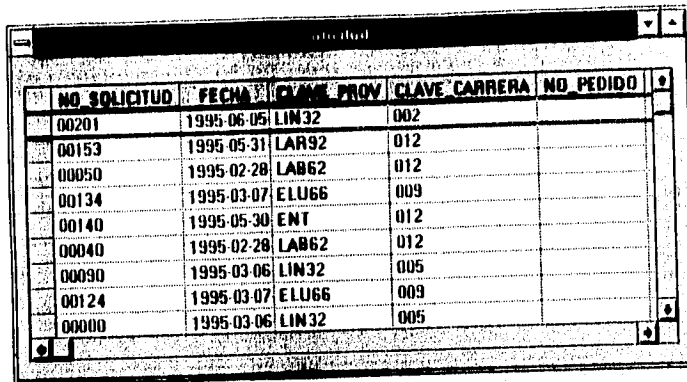
La intersección de cada columna y renglón es una CELLA

Figura 3-13: Table Window

QuestWindow

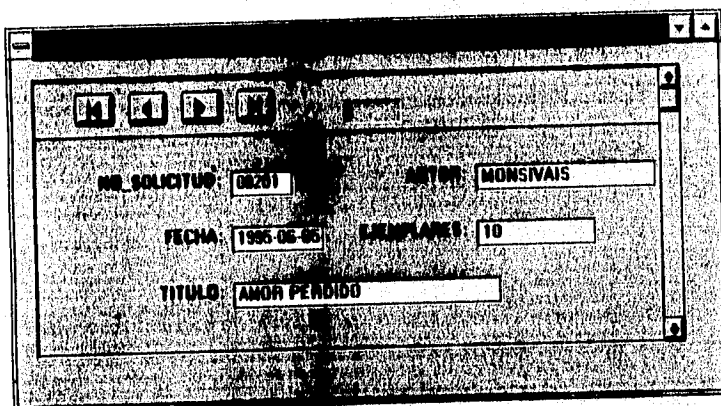
Quest es una herramienta para la creación y utilización de bases de datos. Una QuestWindow combina la facilidad de la interfaz de Quest con la potencia y flexibilidad de una aplicación SQLWindows. El desarrollador crea una consulta o una actividad en una tabla en Quest y entonces la usa en una QuestWindow. Esto permite crear aplicaciones que inserten, actualicen, y seleccionen datos sin escribir sentencias SQL.

Existen dos tipos de QuestWindows, las cuales sólo difieren en la interfaz hacia el usuario, pueden ser en formato de Columnas, Figura 3-14, o en Campos, Figura 3-15.



NO SOLICITUD	FECHA	CLAVE PROV	CLAVE CARRERA	NO PEDIDO
00201	1995-06-05	LIN32	002	
00153	1995-05-31	LAR92	012	
00050	1995-02-28	LAB62	012	
00134	1995-03-07	ELU66	009	
00140	1995-05-30	ENT	012	
00040	1995-02-28	LAB62	012	
00090	1995-03-06	LIN32	005	
00124	1995-03-07	ELU66	009	
00000	1995-03-06	LIN32	005	

Figura 3-14: QuestWindow en formato de columnas



NO SOLICITUD: 00201 AUTOR: MONSIVAIS
FECHA: 1995-06-05 EJEMPLARES: 10
TITULO: AMOR PERDIDO

Figura 3-15: QuestWindow en formato de campos

MDI Window

MDI significa Interfaz de Múltiple Documento (Multiple Document Interface), el cual es un modelo de interfaz para el usuario creado por Microsoft.

Una MDI Window se usa para crear un área de trabajo para una aplicación. Se pueden colocar Form Windows y Table Windows de alto nivel en un área de

trabajo. Las MDI Windows son útiles para manejar un gran número de ventanas de alto nivel que se necesiten desplegar al mismo tiempo.

Una Form o una Table Window dentro de una MDI Window es llamada "ventana hija MDI".

Una ventana hija MDI tiene una barra de título, un menú, un borde para cambiar de tamaño, un menú de sistema, botones para minimizar y maximizar. Sin embargo, no tiene barra de menú ya que todas utilizarán la barra de menú de la MDI Window. Cuando una ventana hija MDI pasa a ser la activa, su menú sustituye al menú de la MDI Window.

Solamente una ventana hija MDI puede estar activa a la vez, lo cual se indica al tener su barra de título destacada y al aparecer sobre todas las otras ventanas hijas MDI. Las ventanas hijas MDI pueden ser minimizadas y aparecerán como un icono en la parte inferior de la MDI Window. Cuando una ventana hija MDI es maximizada, su barra de título desaparece y usa la barra de título de la MDI Window.

En la Figura 3-16 se muestra una MDI Window que tiene una barra de herramientas, una barra de estado y ventanas hijas (una Form Window y dos Table Windows).

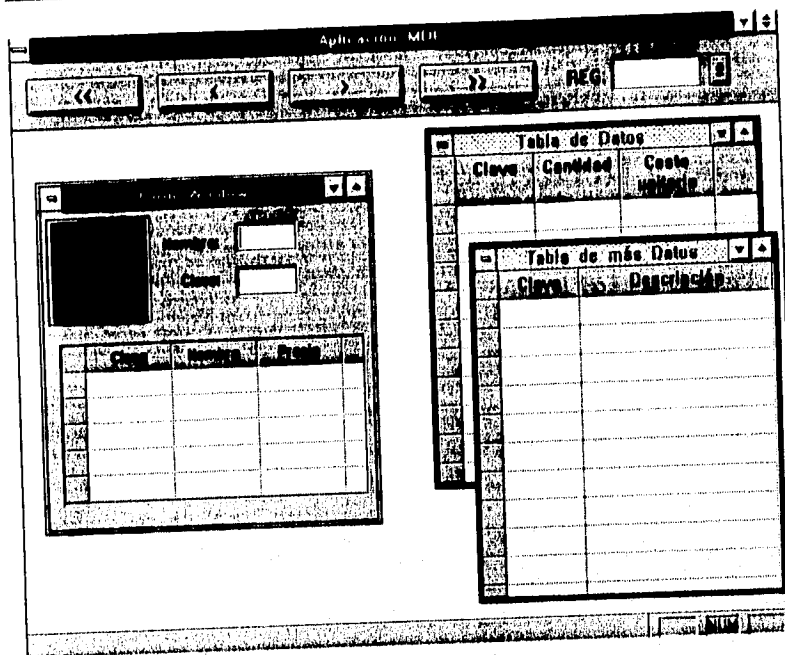


Figura 3-16: Una MDI Window con objetos hijos

Elementos opcionales

Barra de Herramientas

Una barra de herramientas es un área rectangular donde se colocan objetos para las funciones más utilizadas en una aplicación. La Figura 3-17 es un ejemplo de barra de herramientas.



Figura 3-17: Barra de herramientas

Las barras de herramientas se pueden definir en las ventanas de alto nivel y en las MDI Windows. Su posición puede ser la parte superior, izquierda, inferior, o derecha de la MDI.

Barra de Estado

Se puede colocar una barra de estado en la parte inferior de una ventana de alto nivel o en una MDI Window. La barra de estado muestra el estado de las teclas Bloqueo Numérico, Bloqueo Desplazamiento y Bloqueo Mayúsculas. Figura 3-18.

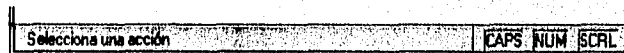


Figura 3-18: Barra de estado

El mensaje a la izquierda de estos tres estados se puede manejar desde la aplicación para dar indicaciones específicas al usuario.

Menús

Se pueden agregar menús a Form Windows, Table Windows de alto nivel, y a MDI Windows.

Los menús son como otros objetos pero se utiliza la barra de opciones para agregar menús emergentes, elementos de menú, y para definir las características de los menús.

Los menús emergentes pueden tener otros menús emergentes o simplemente elementos que al ser elegidos, inmediatamente ejecuten una acción. La acción a ejecutar por un elemento de menú se deberá codificar bajo la sección Menu Actions.

3.2.2 Objetos hijos

Se pueden colocar objetos hijos en los objetos de alto nivel. Cada objeto de alto nivel tiene una sección de Contenidos (Contents) donde se van agregando los objetos hijos. Los objetos hijos son creados y destruidos con su padre (un objeto de alto nivel).

La siguiente tabla lista los objetos hijos que se pueden agregar a los objetos de alto nivel:

Objetos de alto nivel	Objetos hijos que pueden contener
Form Window	Background Text
Dialog Box	Group Box
	Frame
	Line
	Data Field
	Multiline Field
	Push Button
	Radio Button
	Check Box
	Option Button
	List Box
	Combo Box
	Table Window
	QuestWindow
Picture	
Scroll Bar (horizontal y vertical)	
Custom Control	
Table Window	Column

Los objetos hijos no pueden ser padres de otros objetos, con la excepción para las Table Windows hijas que pueden contener columnas como hijas.

Los objetos hijos que pueden procesar acciones por parte del teclado o del ratón dentro de su código predefinido se encontrará la sección de Message Actions, aquí será posible codificar el comportamiento o funcionalidad del objeto en particular.

Background Text

El Background Text (texto de fondo) es utilizado para títulos, etiquetas e instrucciones. Sólo es visual, por lo que no detecta actividad por parte del ratón o del teclado. Figura 3-19.

Teléfono

Figura 3-19: Background Text

Group Box

El Group Box (caja de agrupamiento) se usa para etiquetar un conjunto de objetos relacionados, tales como Radio Buttons. Sólo es visual, por lo que no detecta actividad por parte del ratón o del teclado. Figura 3-20.

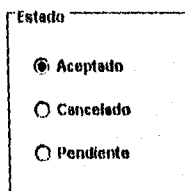
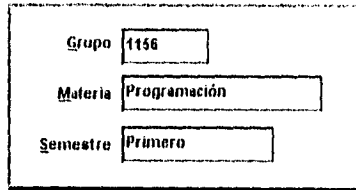


Figura 3-20: Group Box

Frame

Un Frame (marco) es un borde que rodea a uno o varios objetos. Sólo es visual, por lo que no detecta actividad por parte del ratón o del teclado. Figura 3-21.

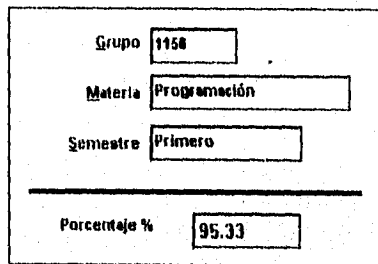


A screenshot of a form window with a black border. It contains three input fields, each with a label to its left: 'Grupo' with the value '1156', 'Materia' with the value 'Programación', and 'Semestre' with the value 'Primero'.

Figura 3-21: Frame

Line

Se puede dibujar un Line (línea) en una Form Window o en un Dialog Box a cualquier ángulo. Sólo es visual, por lo que no detecta actividad por parte del ratón o del teclado. Figura 3-22.



A screenshot of a form window with a black border. It contains four input fields. The first three are 'Grupo' (1156), 'Materia' (Programación), and 'Semestre' (Primero). A thick horizontal line is drawn across the width of the form, positioned below the 'Semestre' field. Below this line is a fourth input field labeled 'Porcentaje %' containing the value '95.33'.

Figura 3-22: Line

Data Field

Un Data Field (literalmente: campo de dato; en Windows: cuadro de texto) despliega información o recibe entradas. Se puede etiquetar un Data Field con un Background Text. Figura 3-23.

Teléfono

Figura 3-23: Data Field

Multiline Field

Un Multiline Field (campo multilinea) acepta y despliega múltiples líneas de datos. El usuario puede presionar Retorno o Ctrl+Retorno para mover el cursor a la siguiente línea cuando introduzca o edite información. Figura 3-24.

Comentarios:

Se necesita capturar todo el documento para conocer los detalles de la solicitud de nuevo material bibliográfico. De esta manera se hará un

Figura 3-24: Multiline Field

Push Button

Cuando el usuario hace clic en un Push Button (literalmente: pulsador o botón; en Windows: botón de comando), la aplicación ejecuta una acción. Figura 3-25.

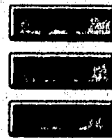


Figura 3-25: Push Button

Radio Button

Cuando el usuario hace clic en un Radio Button (literalmente: botón radio; en Windows: botón de opción), éste se activa o se enciende. Figura 3-26.

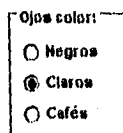


Figura 3-26: Radio Button

Se hace un grupo de Radio Buttons para tener opciones mutuamente excluyentes, es decir, sólo uno de ellos podrá estar activo a la vez, lo cual indica que si se elige un Radio Button no activo, al activarse se desactiva el que previamente era el activo.

Cuando un grupo de Radio Buttons es contiguo, pero procesado como dos diferentes grupos, se debe usar un Group Box.

El usuario puede presionar la tecla Tab para moverse a un Radio Button activo en un grupo y entonces usar las teclas de flecha para mover el foco de entrada a otro Radio Button en el mismo grupo.

Check Box

Cuando el usuario hace clic en un Check Box (literalmente: caja de chequear; en Windows: casilla de verificación), éste turna su opción en activo o desactivo, según su estado anterior. Si ya estaba activo, se desactiva y viceversa.

Más de un Check Box puede estar activo al mismo tiempo. Figura 3-27.

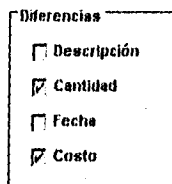


Figura 3-27: Check Box

Option Button

Visualmente es un Push Button, pero funciona como Radio Button o un Check Box, es decir, se activa o se desactiva. Se utiliza cuando se necesitan estas funciones pero con botones, se le puede colocar texto o un gráfico para hacer más intuitiva la función del Option Button.

También funciona como en una paleta de herramientas, es decir, permite seleccionar la función indicada por el Option Button, pero no permite reactivarlo, sino hasta que se selecciona otro Option Button del mismo grupo. Figura 3-28.

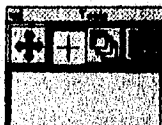


Figura 3-28: Option Button

List Box

Un List Box (literalmente: caja de lista; en Windows: cuadro de lista) despliega una lista de una simple columna que permite al usuario seleccionar uno o más elementos. Un List Box no permite edición, es decir, es de sólo lectura. Figura 3-29.

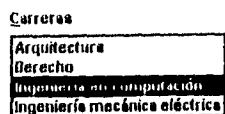


Figura 3-29: List Box

Un List Box se crea con estas características:

- Simple o múltiple selección. Con múltiple selección, más de un elemento puede ser seleccionado
- Barra de desplazamiento vertical y horizontal
- Elementos ordenados

Combo Box

Un Combo Box (en Windows: lista desplegable) contiene un Data Field y un List Box. El List Box contiene elementos predefinidos que el usuario selecciona para llenar el Data Field. Figura 3-30.

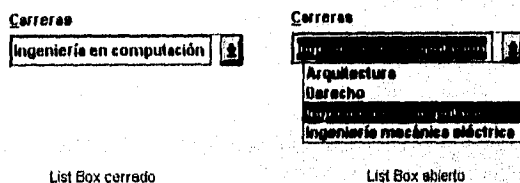


Figura 3-30: List Box

La parte del List Box de un Combo Box puede tener varias características:

- Elementos ordenados
- Barra de desplazamiento horizontal
- Siempre puede estar descolgado

La parte del Data Field de un Combo Box puede ser editable o no editable. Si el Data Field es no editable, no hay un espacio entre el lado derecho del Data Field y el botón de flecha abajo; si el Data Field es editable, existe un espacio entre el lado derecho del Data Field y el botón de flecha abajo.

Uno o ningún elemento puede ser seleccionado en un tiempo dado.

Table Window Column

Las Columns (columnas) son el único contenido en una Table Window. Cada Column es una ventana hija de la Table Window padre.

Un usuario puede cambiar la posición o el tamaño de una Column. Se pueden seleccionar varios renglones o Columns.

Picture

Un Picture (ilustración) despliega una imagen gráfica.

Un objeto Picture puede contener:

- Imágenes gráficas
- Objetos OLE
- Archivos del DOS

Scroll Bars

Se pueden agregar Scroll Bars (barras de desplazamiento) tanto verticales como horizontales en una aplicación. Figura 3-31.

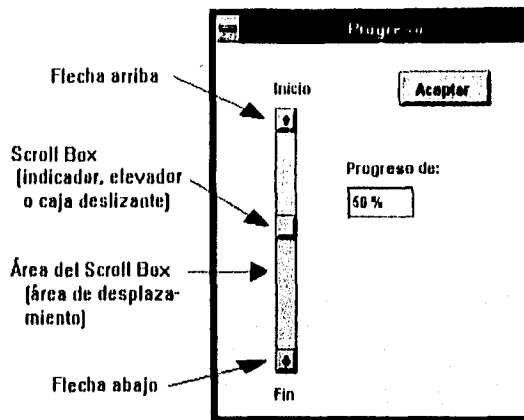


Figura 3-31: Scroll Bar

Una Scroll Bar tiene asociado un valor numérico.

- Cuando el usuario hace un clic en cualquiera de las flechas, el valor entero se incrementa o decrementa en una unidad de "línea", y el Scroll Box se mueve en esa proporción.
- Cuando el usuario hace un clic en el área de desplazamiento de la Scroll Bar, el valor se incrementa o decrementa en una unidad de "página", y el Scroll Box se mueve en esa proporción.
- Cuando el usuario arrastra el Scroll Box, el valor numérico se incrementa o decrementa en la misma proporción.

Cuenta con otros dos valores enteros que representan el valor mínimo y el valor máximo. Cuando el Scroll Box se encuentra en la parte superior (o izquierda) de la Scroll Bar, la posición del Scroll Box se encuentra en el valor mínimo del rango. Cuando el Scroll Box se encuentra en la parte inferior (o derecha) de la Scroll Bar, la posición del Scroll Box se encuentra en el valor máximo del rango.

Custom Controls

Un Custom Control (también conocido como objeto personalizado) extiende la interfaz del usuario en una aplicación SQLWindows. Se pueden agregar dos tipos de Custom Controls a una aplicación:

- VBX (de Visual Basic)
- Microsoft Windows

Un Custom Control puede ser un hijo de una Form Window, de un Dialog Box, o de una barra de herramientas. Por lo regular, los Custom Controls se deberán comprar en forma de archivo tipo DLL, o inclusive se pueden desarrollar con el software apropiado. Figura 3-32.

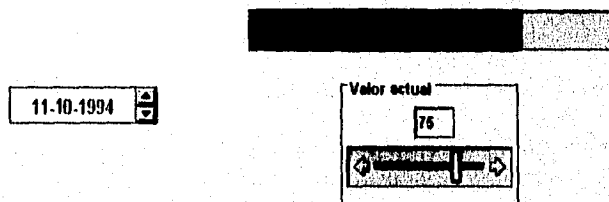


Figura 3-32: Custom Controls

Mnemotécnicos y aceleradores

Aunque no son objetos, permiten manejar de una manera más fácil algunos de los elementos hijos de los objetos de alto nivel.

- Mnemotécnicos:

Cuando un usuario presiona una tecla mnemotécnica al mismo tiempo que la tecla Alt, el foco de entrada se mueve a ese objeto. Los mnemotécnicos se pueden asignar a Background Text, Columns, Radio Buttons, Push Buttons, Option Buttons, Group Boxes, y a Check Boxes.

También pueden ser utilizados en los elementos de los menús. Un mnemotécnico para un elemento de menú no solo mueve el foco de entrada, sino que también ejecuta sus acciones.

Así, por ejemplo, en la Figura 3-33 se tiene un Data Field etiquetado con un Background Text. Basta presionar la tecla de su mnemotécnico junto con la tecla Alt para que el foco de entrada quede situado en ese Data Field.

Teléfono

Figura 3-33: Mnemotécnico

- Aceleradores:

Un acelerador es una tecla que invoca las acciones para un Push Button o un elemento de menú.

3.2.3 Diálogos estándar de Windows

Existen funciones dentro del lenguaje de SQLWindows que permite manipular cualquiera de los siguientes diálogos estándar de Windows (Figura 3-34):

- Color
- Fuentes
- Abrir archivo
- Cerrar archivo

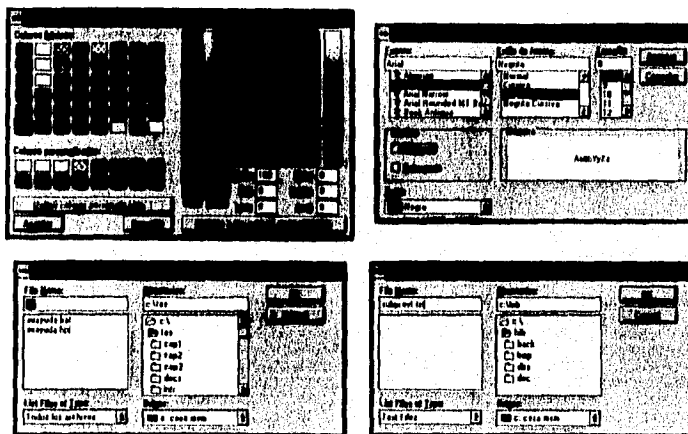


Figura 3-34: Diálogos estándar de Windows

3.3 SAL - Lenguaje de programación

Este apartado explica los elementos del SAL (SQL.Windows Application Language - Lenguaje de Aplicación SQL.Windows):

- Tipos de datos,
- Variables,
- Constantes,
- Arreglos,
- Operadores,
- Expresiones,
- Sentencias,
- Funciones,
- Comentarios, y
- Argumentos de la línea de comandos.

SQLWindows es sensible a mayúsculas y minúsculas. La máxima longitud de un identificador es de 32 caracteres.

3.3.1 Tipos de datos

Un tipo de dato debe ser especificado para una variable o una constante.

Las variables pueden ser de cualquiera de estos tipos de datos:

- Boleano
- Fecha/Hora
- Handle de Archivo
- Cadena Larga
- Número
- Handle Sql
- Cadena
- Handle Ventana

Las constantes pueden ser de cualquiera de estos tipos de datos:

- Boleano
- Fecha/Hora
- Número
- Cadena

Boleano

Boolean: este tipo de dato es para poder indicar sólo dos valores: TRUE o FALSE (Verdad o Falsedad).

Estos valores son constantes del sistema: TRUE es 1, y FALSE es 0.

Fecha/Hora

Date/Time: este tipo de dato es el apropiado para manejar fechas y horas. El formato es AAAA-MM-DD-HH.MM.SS.MSMSMS.

Handle de Archivo

File Handle: con este tipo de dato se identifica un archivo abierto. Cuando se abre o crea un archivo, SQL Windows retorna un file handle, de aquí en adelante se utilizará para identificar ese archivo.

Cadena

String: este tipo de dato es para datos de caracteres. El límite para la longitud de este tipo de dato es la memoria disponible del sistema.

Cadena Larga

Long String: tiene las características del tipo de dato String. Este tipo de dato se debe utilizar para leer y escribir columnas de bases de datos de SQL mayores a 254 bytes.

Handle Ventana

Window Handle: este tipo de dato se utiliza para identificar una instancia simple de una ventana. En tiempo de ejecución, cada instancia de una ventana tiene un handle único.

3.3.2 Variables

Una variable puede contener cualquier valor de su respectiva tipo de dato.

Las variables pueden ser declaradas como globales o locales a un objeto de alto nivel, ya sea como parámetros o en su sección de variables. Dentro de las funciones globales o locales, como parámetros, variables estáticas o en su sección de variables locales.

La sintaxis utilizada para declarar una variable es la siguiente:

Tipo de Dato: NombreVariable

Ámbito

Las variables en la sección Global Declarations son válidas tan pronto como la aplicación inicia y están disponibles en cualquier parte del sistema.

Las variables en la sección Local Variables de la definición de una función son válidas cuando la función es invocada y dejan de existir cuando la función retorna.

Las variables en la sección Window Variables son válidas tan pronto como se crea una instancia de la ventana y dejan de existir cuando la ventana es destruida.

Referencias a variables

Existen dos maneras de referirse a las variables:

- En una referencia *ilimitada* (unqualified), sólo se especifica el nombre de la variable
- En una referencia *limitada* (qualified), hay un prefijo que contiene uno o más nombres y que se agrega al nombre de la variable, los nombres se separan con puntos:

hWnd1.dlSollicitud

Una referencia se limita para:

- Eliminar ambigüedad con nombres duplicados
- Referirse a una ventana que tiene múltiples instancias
- Referirse a una variable de un hijo desde su padre
- Referirse a una variable en una ventana de alto nivel diferente a la actual

3.3.3 Constantes

Una constante contiene un valor simple e inintercambiable.

Las constantes se declaran únicamente en la sección de Global Declarations. De ahí en adelante se podrá referir a una constante como a una variable.

La sintaxis utilizada para declarar una constante es la siguiente:

Tipo de Dato: NombreConstante = expresión

Dentro de las constantes existen algunas que son definidas por el sistema y sirven para tener valores fijos propios del sistema.

3.3.4 Arreglos

Un arreglo es una lista ordenada de variables del mismo tipo, el cual puede ser de cualquier tipo de dato permitido por SQL Windows. Puede ser unidimensional o bidimensional.

Los arreglos inician con base índice igual a cero, pero se puede cambiar la base de inicio. Existen tres maneras para fijar el número de elementos en un arreglo.

Especificando el número de elementos:

Tipo de Dato: NombreVariable[número_de_elementos]

Especificando un rango:

Tipo de Dato: NombreVariable[inicio:término]

Indicando un número dinámico de elementos:

Tipo de Dato: NombreVariable[*]

Tanto para las variables como para las constantes existen convenciones para sus nombres. Se recomienda colocar un prefijo al nombre de la variable de acuerdo a su tipo de dato. De esta manera el programa se autodocumenta para tener una idea más precisa de lo que se espera contenga una variable o una constante.

Tipo de dato	Prefijo
Boolean	b
Date/Time	dt
File Handle	fh
Long String	ls
Number	n
Sql Handle	hSql
String	s o str
Window Handle	hWnd

Para las constantes se recomienda que el prefijo sea en mayúsculas, seguido de un subguión y el nombre en mayúsculas o combinado en mayúsculas y minúsculas.

3.3.5 Operadores

Un operador es un símbolo o palabra que representa una operación a ejecutar sobre uno o más valores. La siguiente tabla muestra los operadores.

Símbolos de operadores	Tipo de operador
+, -, *, /	Númérico
- unario	Unario
=, !=, >, <, >=, <=	Relacional
AND, OR, NOT	Boleano
&	AND para bits
	OR para bits
	Concatenar cadenas

3.3.6 Expresiones

Una expresión es una combinación de constantes, variables y operadores que obtienen un solo valor. Una expresión puede ser:

- El resultado de una función
- Una variable
- Una constante
- Dos o más expresiones conectadas con un operador

SQL Windows utiliza las siguientes reglas de precedencia para evaluar expresiones:

- Las expresiones con AND, OR y NOT se evalúan de izquierda a derecha
- Detención en la evaluación de un AND/OR tan pronto como se conozca el resultado
- Evaluar primero expresiones en paréntesis

3.3.7 Sentencias SAL

Las sentencias SAL se utilizan para codificar acciones que se desee que una aplicación ejecute. Las sentencias SAL son las siguientes:

- **Break**
Como su nombre lo indica, rompe una secuencia o un ciclo.
- **Call**
Ejecuta una función.
- **If, Else, y Else If**
Sentencias para tomar decisiones.
- **Loop**
Para ejecutar un ciclo.
- **On**
Para interceptar un mensaje mandado a la aplicación o a un objeto.
- **Return**
Sirve para terminar una función e incluso retornar un valor.
- **Select Case**
Sirve para evaluar una expresión y seleccionar el caso donde coincida en base a su valor.
- **Set**
Para asignar un valor a una variable.
- **When SqlError**
Para interceptar algún error de SQL.
- **While**
Para ejecutar un ciclo.

Las sentencias SAI, se codifican en las siguientes secciones:

- Message Actions
Sección de todos los objetos de alto nivel y de la mayoría de los objetos hijos.
- Function Actions
Sección dentro de la definición de una función, la función puede ser global o local a una ventana.
- Menu Actions
Sección dentro de un elemento de menú.
- Application Actions
Sección global a la aplicación donde se puede ejecutar código al iniciar o al terminar la aplicación, por ejemplo.

Al hacer la explicación de cada una de las sentencias, la información que va entre corchetes ([]) es opcional.

Break

Termina una sentencia Loop. Si se especifica un nombre de Loop, éste es el que termina; lo cual permite terminar y salir de más de un nivel de Loop. Si no se especifica un nombre, el Loop en el cual está inmerso el Break es el que termina.

Sintaxis

```
Break [Nombre_Loop]
```

Ejemplo

Incrementar una variable en una unidad y terminar cuando el contador llega a 100.

```
Loop  
  If Contador = 100  
    Return  
  Set Contador = Contador + 1  
  ...
```

Call

Ejecuta una función.

Sintaxis

```
Call NombreFunción( [parámetros] )
```

Ejemplo

Ejecutar la función que transforma una variable de cadena a puras mayúsculas.

```
Call SalStrUpper( Nombre, NombreVariable )
```

If, Else, y Else If

Estas sentencias expresan decisiones.

La parte del Else es opcional. Puede haber tantas secciones Else If como se necesiten, pero sólo una sección Else.

La indentación determina la ejecución condicional del código.

Sintaxis

```
If expresión1
  sentencia1
Else If expresión2
  sentencia2
Else
  sentencia3
```

Primero se evalúa la expresión 1, si resulta TRUE (verdadera), entonces se ejecuta la sentencia 1. Si la expresión 1 es FALSE (falsa), entonces se evalúa la expresión 2. Si la expresión 2 es TRUE, entonces se ejecuta la sentencia 2. Si la expresión 2 es FALSE, SQL Windows ejecuta la sentencia 3.

Ejemplo

Para procesar la información de un cliente según su sexo.

```
IF sSexo = 'M'  
  Call ProcesoMasculino( )  
Else IF sSexo = 'F'  
  Call ProcesoFemenino( )  
Else  
  Call ProcesoOtros( )
```

Escribiendo este código con If y Else.

```
IF sSexo = 'M'  
  Call ProcesoMasculino( )  
Else  
  If sSexo = 'F'  
    Call ProcesoFemenino( )  
  Else  
    Call ProcesoOtros( )
```

Loop

Repite las sentencias indentadas bajo de él hasta que el ciclo se rompe con una sentencia Break o una sentencia Return.

Sintaxis

```
Loop {Nombre_Loop}
```

Ejemplo

Incrementar una variable en una unidad y terminar cuando el contador llega a 100.

```
Loop  
  If Contador = 100  
    Return  
  Set Contador = Contador + 1  
  ...
```

On

Atrapa el mensaje que se indique y ejecuta el código indentado bajo de él. Los mensajes se colocan en la sección de Message Actions (acciones de mensajes) o en la de Application Actions (acciones de la aplicación).

Los mensajes son identificados con un número o con una constante que representa ese número de mensaje.

Sintaxis

On mensaje

Ejemplo

Cuando un Push Button recibe un clic del ratón, SQLWindows envía el mensaje SAM_Click, para este ejemplo se supone que es el botón que indica la terminación de la aplicación.

```
Pushbutton: pbTerminar
Message Actions
On SAM_Click
CALL SalQuit()
```

Return

Rompe el flujo de control retornando un valor. Esta sentencia termina:

- Una función
- Acciones de mensajes
- Acciones de menú

Sintaxis

Return expresión

Ejemplo

Validar si un Data Field contiene información.

```
Message Actions
On SAM_Validate
Return SalIsNull( dfcampo )
```


Select Case

Verifica una serie de condiciones. El valor de una expresión es sucesivamente comparada contra constantes. La limitante es que tanto las expresiones como constantes sean tipos de datos numéricos.

La sentencia Break termina la sentencia Select Case, por lo que por cada caso a evaluar se deberá tener un Break, a menos que se desee que varios casos se evalúen en forma continua. El caso Default es opcional y es ejecutado sólo si no hubiera ninguna comparación que coincida con el valor a probar.

Sintaxis

```
Select Case (expresión)
  Case constante1
    sentencia1
    Break
  Case constante2
    sentencia2
    Break
  Case constante...
    sentencia...
    Break
  Default
    sentencia_Default
```

Se evalúa la expresión, si coincide con la constante1, entonces se ejecuta la sentencia1. Si coincide con la constante2, entonces se ejecuta la sentencia2, y así sucesivamente mientras haya más casos. Si la expresión no coincide con ningún caso, entonces se ejecuta la sentencia_Default.

Ejemplo

Expresar con una frase a cuál cuarto del año pertenece una fecha.

```
Select Case ( SalDateQuarter( dFecha_Hoy ) )
  Case 1
    Set sCuarto = 'Primer cuarto'
    Break
  Case 2
    Set sCuarto = 'Segundo cuarto'
    Break
  Case 3
    Set sCuarto = 'Tercer cuarto'
    Break
  Case 4
    Set sCuarto = 'Ultimo cuarto'
    Break
  Default
    Set sCuarto = 'Error'
```

Set

Asigna un valor a una variable.

Sintaxis

```
Set NombreVariable = expresión
```

Ejemplo

```
Set @Mensaje = 'Número no válido'  
Set @Contador = 100
```

When SqlError

Esta sentencia sirve para procesar y manejar un error de una sentencia SQL. Se utiliza en aplicaciones avanzadas de manejo de bases de datos.

While

Esta sentencia repite un grupo de sentencias hasta que la expresión que evalúa resulte FALSE.

Sintaxis

```
While expresión  
  sentencia1  
  sentencia2  
  ...
```

SQLWindows evalúa la expresión, si resulta TRUE, se ejecuta el código indentado. La expresión es reevaluada y si permanece en TRUE, se vuelven a ejecutar las sentencias internas. El ciclo continúa hasta que la expresión sea FALSE.

3.3.8 Comentarios

Una línea que inicia con el signo de admiración derecho (!) se considera un comentario y por lo tanto no tiene efecto para SQLWindows.

Ejemplo

! Averiguar si todo es correcto
Call Verificación(usuario)

3.3.9 Funciones

Una función desarrolla una tarea específica. Existen cinco tipos de funciones:

- System Functions (funciones del sistema)
- External Functions (funciones externas)
- Internal Functions (funciones internas)
- Window Functions (funciones de ventana)
- Class Functions (funciones de clase)

Funciones del sistema

Son las funciones interconstruidas en SQL Windows. Éstas a su vez se dividen en dos tipos:

- Funciones cuyo nombre comienza con el prefijo **Sal** desarrollan tareas tales como el manejo de ventanas, validación de campos, manejo de archivos, manejo de listas, y manipulación de cadenas.
- Funciones cuyo prefijo es **Sql** desarrollan operaciones sobre bases de datos.

Algunos ejemplos de funciones Sal*:

Nombre de la función	Descripción
SalNumberAbs	Cálcula el valor absoluto de un número
SalNumberMax	Retorna el mayor de dos números
SalNumberMin	Retorna el menor de dos números
SalDateDay	Obtiene el día de una fecha
SalDateWeekBegin	Obtiene la fecha del lunes previo
SalModalDialog	Crema una caja de diálogo modal
SalFileOpen	Abre un archivo
SalReportPrint	Imprime un reporte
SalGetDataType	Retorna el tipo de dato de un objeto
SalStrLen	Retorna la longitud de una cadena
SalStrLower	Convierte una cadena a minúsculas
SalStrToDate	Convierte una cadena a un valor de fecha/hora
SalStrToNumber	Convierte una cadena en número
SalQuit	Abandona la aplicación

Algunos ejemplos de funciones Sql*:

Nombre de la función	Descripción
SqlConnect	Conecta un Handle Sql a una base de datos
SqlDisconnect	Desconecta un Handle Sql de una base de datos
SqlError	Obtiene el código de error
SqlOpen	Nombra un cursor y ejecuta una sentencia SQL
SqlSetLockTimeout	Especifica el tiempo de espera para acceso a una base de datos

Funciones externas

Son funciones que se toman de una librería del tipo DLL.

Funciones internas

Son funciones que se escriben con el propio lenguaje SAL. Son funciones internas globales a la aplicación, es decir, se declaran e implementan en la sección Global Declarations.

Funciones de ventana

Son funciones que se escriben con el lenguaje SAL en objetos de alto nivel y en las Table Windows hijas.

Funciones de clase

Al declarar una clase, se puede declarar una función para esa clase.

3.3.10 Argumentos de la línea de comandos

Al iniciar una aplicación, SQLWindows es capaz de recibir argumentos que vengan de la línea de comandos.

3.4 Mensajes

SQLWindows manda un mensaje a un objeto cuando sucede un evento. Una aplicación procesa un mensaje al tomar una acción.

Los mensajes son los que controlan a la aplicación. Las acciones son asociadas con objetos, por ejemplo, Dialog Boxes, Push Buttons, Check Boxes.

Los mensajes son disparados por:

- Acciones del teclado
- Acciones del ratón
- Un reloj
- La aplicación

3.4.1 Tipos de Mensajes

Existen tres tipos de mensajes:

- SQLWindows Application Messages (SAM_*)
- Mensajes definidos en la aplicación
- Mensajes de Windows (WM_*)

Los mensajes son identificados por un nombre y un número. El nombre del mensaje es una constante que representa el número del mensaje. Desde luego que se puede

referir a un mensaje por su nombre o por su número, aunque se recomienda utilizar siempre su nombre para mayor claridad.

SQLWindows Application Messages (SAM_*)

Ejemplos de eventos que causan mensajes del tipo SAM_*:

- Inicio de la aplicación
- Terminación de la aplicación
- Colocarse en un campo
- Abandonar un campo
- Hacer clic en un botón
- Cerrar una ventana

SQLWindows manda mensajes a todos los objetos, excepto a Background Text, Group Boxes, Lines y Frames. La sección Application Actions también recibe mensajes.

Para generalizar, se puede decir que todos los objetos son ventanas y no todos los mensajes se mandan a todo tipo de ventana. Por ejemplo:

- El mensaje SAM_AppStartup es enviado solamente a la sección Application Actions.
- El mensaje SAM_Click no es enviado a un Data Field.

Algunos ejemplos de mensajes SAM_*:

Nombre del mensaje	Descripción
SAM_AppExit	Mandado a la sección Application Actions cuando todas las ventanas de la aplicación son destruidas (cerradas)
SAM_AppStartup	Mandado a la sección Application Actions al momento del arranque, antes de que alguna ventana sea creada
SAM_Click	Mandado a una List Box, Check Box, Radio Button, Push Button, Table Window, Column cuando el usuario hace clic en el objeto con el ratón o su equivalente en el teclado
SAM_Create	Mandado a objetos de alto nivel e hijos cuando son creados pero antes de ser desplegados
SAM_FieldEdit	Mandado a un Data Field, Multiline Field, Table Window, Column cuando el usuario intenta dejar el objeto después de haber cambiado su valor (lo ha editado)
SAM_Help	Mandado a la ventana de alto nivel cuando el usuario presiona la tecla F1
SAM_KillFocus	Mandado a un objeto cuando el foco está saliendo de ese objeto
SAM_SetFocus	Mandado a un Data Field, List Box, Multiline Field, Check Box, Radio Button, Push Button, Scroll Bar, Table Window, Column cuando el objeto recibe el foco de entrada y se hace activo
SAM_Validate	Mandado a un Data Field, List Box, Multiline Field, Column cuando, después de haber cambiado el valor del objeto, el usuario intenta dejar el objeto

Mensajes definidos en la aplicación

Cuando se desee hacer más explícita una acción, se podría definir un mensaje y entonces enviarlo con funciones especiales, donde solo las ventanas que lo procesen realizarán acciones. Se tratan como a los mensajes SAM_*.

Al definirse, los nombres serán lo más posible descriptivos y son totalmente a criterio del desarrollador, se recomienda que tengan un prefijo como AM que indica Application Messages, o alguno otro similar. Algunos ejemplos de mensajes:

Nombre del mensaje	Descripción
AM_Habilitar	Mandado a objetos (botones) para que se habiliten y puedan recibir un clic
AM_Editar	Mandado a objetos para habilitarlos y editarlos
AM_Limpiar	Mandado a un objeto para que se haga nulo
AM_Ocular	Mandado a un objeto para que se haga invisible
AM_Mostrar	Mandado a un objeto para que se haga visible
AM_ChecarContenido	Mandado a un objeto solicitándole se verifique para que no sea nulo y no almacenarlo así

Mensajes de Windows (WM_*)

Los mensajes propios de Windows pueden ser procesados en una aplicación de SQLWindows.

Algunos ejemplos de mensajes WM_*:

Nombre del mensaje	Descripción
WM_CLOSE	Enviado a una ventana cuando se le indica que se cierre
WM_PAINT	Enviado a una ventana cuando debe actualizar la información que despliega sobre ella
WM_CHAR	Enviado a un objeto por cada tecla que se presiona
WM_LBUTTONDOWN	Enviado cuando el usuario presiona el botón izquierdo del ratón
WM_LBUTTONUP	Enviado cuando el usuario libera el botón izquierdo del ratón
WM_MOUSEMOVE	Enviado cuando el usuario mueve el ratón

3.5 SQL - Manejo de bases de datos

Esta sección explica cómo utilizar bases de datos desde SQLWindows. Se habla de las operaciones más comúnmente utilizadas al interactuar con tablas de una base de datos y se describe de forma general las funciones que se usan para llevar a cabo tales operaciones.

En una aplicación que maneja una base de datos, primero que nada se debe conectar a la base de datos, después de esto, se podrán hacer consultas para tomar registros específicos, insertar nuevos, modificar o eliminar información. E incluso aplicar (confirmar) las operaciones que están modificando la base de datos o simplemente descartarlas.

Con SQLWindows es posible desarrollar aplicaciones monousuario y de multiusuario.

Las aplicaciones monousuario son más fáciles de desarrollar. Las aplicaciones multiusuario son más complejas ya que deben vigilarse muchos detalles, tales como la concurrencia de dos o más usuarios a una misma base de datos, una tabla o incluso a un mismo renglón. Un renglón es un conjunto de información relacionada, tal como todos los datos que definen a un proveedor: nombre, RFC, dirección, teléfonos, etc.

Para una aplicación multiusuario es necesario manejar niveles de aislamiento y de bloques, esto permite controlar la integridad de la información cuando, por ejemplo, dos o más usuarios desean modificar o eliminar el mismo registro al mismo tiempo. También se requiere mayor cuidado al ejecutar las sentencias SQL para no monopolizar la base de datos innecesariamente cuando otros usuarios desean también tener acceso a ella.

Hoy en día la mayoría de las aplicaciones se desarrollan para ejecutarse a nivel de red, sin embargo aún se siguen requiriendo aquellas a nivel monousuario. Una buena manera de aprender a desarrollar sistemas multiusuario es comenzando desde una aplicación de base de datos monousuario.

Y aunque el SQL permite manipular una base de datos relacional, no es en sí un lenguaje de programación pues no provee:

- Lógica procedural
- Extensivos tipos de datos
- Variables

Las funciones **Sql*** de SQLWindows son un lenguaje de interfaz que se utiliza para acceder a la base de datos con las sentencias SQL.

SQLWindows puede ser utilizado con varios servidores para acceder la base de datos. SQLBase es un servidor con un conjunto completo de instrucciones del estándar SQL y es compatible con IBM DB2, e inclusive agrega extensiones a DB2. Fue creado por la misma empresa que desarrolló a SQLWindows, por lo que se convierte en el servidor ideal para las aplicaciones. Además, si se adquiere la versión corporativa, SQLBase viene integrado.

Sin embargo, todas las sentencias **Sql*** son independientes del servidor y se pueden utilizar con IBM DB2, ORACLE, Microsoft SQL Server, Sybase, Informix, IBM Database Manager, entre otros.

A las aplicaciones que se desarrollen con SQLWindows se les llama aplicaciones **cliente** (o "front end"). El manejador de la base de datos es conocido como el **servidor** (también llamado "backend" o máquina).

Una base de datos de SQL se puede acceder de dos maneras en una aplicación de SQL Windows:

- Interfaz multi-paso
- Interfaz de un-solo-paso

Estas dos interfaces se pueden combinar en una sola aplicación.

Interfaz multi-paso

Con la interfaz multi-paso el desarrollador se conecta previamente a la base de datos antes de desarrollar cualquier operación. Para cada operación que se ejecute, se deberá compilar y ejecutar una sentencia SQL. Para un SELECT deberá haber además una "transferencia" (fetch) de la información solicitada. Al completar las operaciones con la base de datos, simplemente la aplicación se desconecta de la base de datos. Figura 3-35.

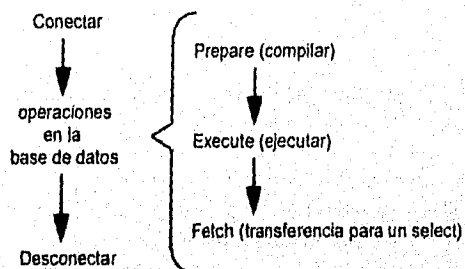


Figura 3-35: Interfaz multi-paso

Cuando se hace un SELECT se crea un conjunto de resultados (result set) del cual se obtienen los registros uno a uno con transferencias (fetching row), es decir, se hacen transferencias de renglones del conjunto de resultados hacia la aplicación. La interfaz multi-paso es más flexible y se obtiene un control más amplio sobre el manejo de los datos que la que se obtiene en la interfaz de un-solo-paso.

Interfaz de un-solo-paso

Con la interfaz de un-solo-paso se pueden desarrollar operaciones de base de datos en una sola llamada de función. Este método es más fácil de utilizar, pero la desventaja es que un SELECT sólo puede retornar un registro.

3.5.1 Interfaz multi-paso

Conexión y desconexión

Para poder ejecutar operaciones sobre una base de datos es necesario que previamente a todas ellas la aplicación se **conecte** a esa base de datos. Al terminar de ejecutarlas, la aplicación se deberá **desconectar** de la base de datos.

Las siguientes dos funciones cumplen con el cometido de conectar y desconectar a una base de datos:

Función	Descripción
SqlConnect	Logra conectar una base de datos
SqlDisconnect	Logra desconectar una base de datos

SqlConnect

Para conectarse a una base de datos se necesita:

- Especificar la base de datos a la cual se desea conectar
- Especificar el nombre de usuario y la palabra de paso
- Llamar a la función de SqlConnect, la cual retorna un Handle Sql

SQL.Windows tiene tres variables de sistema, con valores por omisión, para especificar la base de datos, el nombre de usuario y su palabra de paso:

- SqlDatabase (por omisión: DEMO)
- SqlUser (por omisión: SYSADM)
- SqlPassword (por omisión: SYSADM)

Éstas pueden ser inicializadas en la sección Application Actions. Por ejemplo:

```
On 3AM AppStartup
Set SqlDatabase = 'BOEMDB'
Set SqlUser = 'memo'
Set SqlPassword = 'cv1020'
```

La función `SqlConnection` retorna un `Handle Sql` que identifica una conexión a una base de datos. El `Handle Sql` en realidad es un valor que se maneja simbólicamente por su nombre, pero su nombre se debe utilizar en operaciones subsecuentes en esa base de datos. Un `Handle Sql` se conoce también como *cursor*.

El término `Handle Sql` se refiere a tres cosas en `SQLBase`:

- Un nombre que identifica una conexión a una base de datos
- Una posición de renglón en un conjunto de resultados
- Un área de trabajo en memoria usada para procesar una sentencia SQL.

Antes de utilizarlo, se debe declarar el `Handle Sql` con su tipo de dato `Sql Handle`. Se debe indicar cada uno de ellos antes de utilizarlos:

```
Global Declarations
...
Variables
  Sql Handle: hSqlTransferencia
  Sql Handle: hSqlCambios
```

La manera de conectarse a la base de datos, posterior a la declaración del (los) `Handle Sql` y a la especificación del nombre de la base de datos y del usuario, es como sigue:

```
Set hConexión = SqlConnection( hSqlTransferencia )
If hConexión
  Set hConexión = SqlConnection( hSqlCambios )
```

`SqlConnection` retorna el valor `TRUE` como indicación de éxito en la conexión, de lo contrario el valor será `FALSE`.

Al establecer una conexión se asignan los siguientes recursos para el cliente y el servidor:

- Un área de trabajo para el Handle Sql
- Un buffer de mensajes de entrada y un buffer de mensajes de salida

SqlDisconnect

La función `SqlDisconnect` se llama al terminar de realizar todas las operaciones sobre la base de datos. Siempre se deben desconectar todos los Handles Sql al abandonar la aplicación:

```
on SAM_appexit  
  if bConexión  
    call SqlDisconnect( hSqlTransferencia )  
    call SqlDisconnect( hSqlCambios )
```

Al hacer la desconexión se liberan los recursos asignados al hacer la conexión. Es importante desconectar individualmente cada uno de los Handles Sql utilizados.

Ejecutando operaciones sobre bases de datos

Variables en SQL

Existen dos tipos de variables en SQL: variables de enlace y variables INTO.

Variables de enlace. Las variables de enlace se utilizan para indicar valores de entrada, tales como nombres de columnas en una cláusula WHERE. Se les llama variables de enlace porque enlazan a la aplicación con la base de datos y están ahí cada vez que la sentencia se ejecuta.

En general, las variables de enlace se utilizan para:

- Indicar valores para condiciones en una cláusula WHERE en sentencias SELECT
- Cláusulas VALUES en sentencias INSERT
- Cláusulas SET en sentencias UPDATE

No se pueden utilizar constantes para variables de enlace.

Variables INTO. Estas variables, a diferencia de las variables de enlace, sirven para sacar datos de una sentencia SELECT. La cláusula INTO especifica las variables definidas en la aplicación donde SQL.Windows retorna datos desde la consulta.

Respecto a los nombres tanto de las variables de enlace como de las variables INTO, se deben indicar en la sentencia SQL con dos puntos (:) como prefijo.

Las siguientes tres sentencias SQL muestran cómo codificar variables de enlace y variables INTO:

```
! strSelectPorNombre contiene variables INTO (:dfInvitado, :dfDireccion, y
:dfCrédito) y una variable de enlace (:dfInvitado).
! strInsertar solamente contiene variables de enlace (:dfDireccion, :dfCrédito, y
:dfInvitado).
! strActualizar solamente contiene variables de enlace (:dfDireccion, :dfCrédito, y
:dfInvitado).

Global Declarations
...
Variables
String: strSelectPorNombre
String: strInsertar
String: strActualizar
...
Application Actions
On SAM_AppStartup
Set strSelectPorNombre = 'SELECT nombre, direccion, credito INTO :dfInvitado,
:dfDireccion, :dfCrédito FROM Cartera WHERE nombre = :dfInvitado'
Set strInsertar = 'INSERT INTO Cartera ( direccion, credito, nombre )
VALUES ( :dfDireccion, :dfCrédito, y :dfInvitado )'
Set strActualizar = 'UPDATE Cartera SET direccion = :dfDireccion,
credito = :dfCrédito, nombre = :dfInvitado'
```

Compilando y ejecutando sentencias SQL

Para compilar y ejecutar sentencias SQL se utilizan las siguientes funciones:

Función	Descripción
SqlPrepare	Compila una sentencia SQL.
SqlExecute	Ejecuta una sentencia SQL.
SqlPrepareAndExecute	Compila y ejecuta una sentencia SQL en un paso

La función SqlPrepare compila una sentencia SQL. Al ser invocada, SQL.Windows chequea la cláusula INTO (en una sentencia SELECT) y verifica las variables de enlace, después manda la sentencia a el servidor. Suponiendo SQLBase como el servidor, éste realiza:

1. El análisis gramático, detecta errores de sintaxis y verifica que la base de datos exista.
2. Checa seguridad.

3. Determina cómo acceder los datos. Busca los índices para agilizar el acceso a los datos. Este paso es llamado **plan de ejecución**.
4. Traduce la sentencia en una serie de módulos ejecutables.

SQLBase almacena la sentencia compilada en el área de trabajo del Handle Sql.

Después de compilar una sentencia, se ejecuta con la función `SqlExecute` la cual manda los valores de las variables de enlace al servidor.

También se pueden preparar y ejecutar sentencias SQL en un solo paso al llamar la función `SqlPrepareAndExecute`. Figura 3-36.

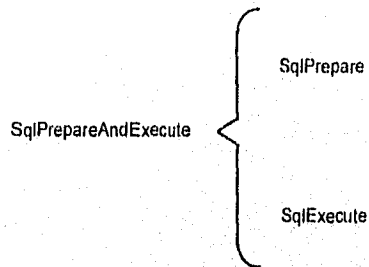


Figura 3-36: `SqlPrepareAndExecute`

Una vez que se prepara una sentencia SQL, se puede llamar a la función `SqlExecute` tantas veces como se requiera. Sin embargo, hay ciertas acciones que destruyen una sentencia compilada:

- Compilar otra sentencia con el mismo Handle Sql
- Hacer un COMMIT o un ROLLBACK si la constante `DBP_PRESERVE` está desactivada

Un COMMIT es una orden que hace que la base de datos se modifique físicamente, es decir, hace que los cambios se asienten definitivamente ya que éstos se encontraban en el área de trabajo del Handle Sql.

Un ROLLBACK es la orden contraria a un COMMIT, descarta todos los cambios almacenados en el área de trabajo del Handle Sql.

SELECT's (consultas)

Esta es la secuencia para ejecutar un SELECT:

1. Preparar (compilar) la sentencia SELECT
2. Ejecutar la sentencia SELECT
3. Llamar a la función SqlFetch* para transferir (fetch) los datos en el área de trabajo hacia las variables INTO

Las siguientes funciones se utilizan para los conjuntos de resultados generados por un SELECT:

Función	Descripción
SqlFetchNext	Transferir el siguiente renglón desde un conjunto de resultados
SqlFetchPrevious	Transferir el renglón previo desde un conjunto de resultados
SqlFetchRow	Transfiere un renglón específico desde un conjunto de resultados
SqlGetResultSetCount	Retorna el número de renglones de un conjunto de resultados después de un SqlExecute
SqlSetResultSet	Alternar el modo de conjunto de resultados en activo o desactivo para el Handle Sql

Conjunto de resultados y Transferencias (proceso de un renglón a la vez). Un conjunto de resultados (o tabla resultado) es un conjunto de renglones de una o más tablas de una base de datos producidos por una consulta (una sentencia SELECT, también conocida como "un query"). Una aplicación puede transferir (tomar) un renglón a la vez desde el conjunto de resultados y procesarlo.

Esto sucede de tal forma que después de compilar y ejecutar una sentencia SELECT es necesario llamar a la función SqlFetchNext para transferir el primer renglón del conjunto de resultados. Después de esto, será necesario seguir llamando a SqlFetchNext para transferir los subsiguientes renglones.

Un conjunto de resultados puede ser destruido por las mismas acciones que destruyen una sentencia compilada.

Modo conjunto de resultados. Los conjuntos de resultados se utilizan para consultas. El modo de conjunto de resultados es activo por omisión y sirve para poder desplazarse por todos los renglones que componen el conjunto de resultados. El modo de conjunto de resultados puede ser desactivado con la función `SqlSetResultSet` o con la variable de sistema `SqlResultSet`.

En el modo activo, después de ejecutar una sentencia `SELECT`, se puede tomar cualquier renglón del conjunto de resultados con la función `SqlFetchRow` sin necesidad de hacer una transferencia secuencial hacia adelante. Una vez posicionado el `Handle Sql`, posteriores transferencias comienzan en ese renglón. Se puede también llamar a la función `SqlFetchPrevious` para hacer transferencias hacia atrás. Figura 3-37.

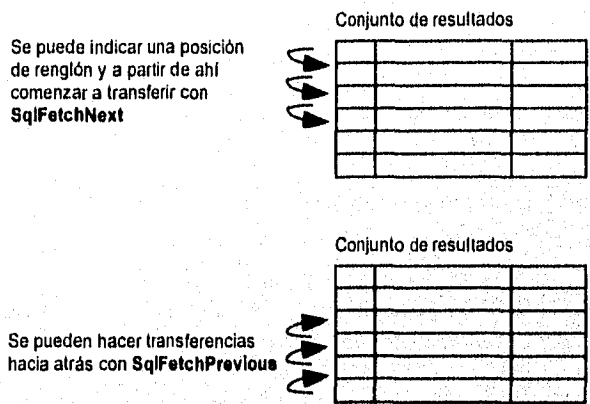


Figura 3-37: Modo activo de conjunto de resultados

Sentencias `INSERT`, `UPDATE` y `DELETE`

El ejemplo abajo muestra como se desarrolla una sentencia de actualización de información, un `UPDATE`. Se podrán manipular datos con otras sentencias `SQL` de la misma manera:

- Compilar la sentencia SQL con SqlPrepare.
- Con SqlExecute ejecutar la sentencia SQL.
- Ejecutar un COMMIT o un ROLLBACK para la transacción (operación que modifica la base de datos). Se puede hacer un COMMIT al llamar a la función SqlCommit o se puede compilar y ejecutar una sentencia cuya cadena sea "COMMIT". Un ROLLBACK puede ser compilado y ejecutado cuando la cadena sea "ROLLBACK".

Otras operaciones

Hasta aquí las operaciones son del tipo DML (Data Manipulation Language), es decir, sólo se han manipulado los datos. Se podría llamar a la función SqlGetModifiedRows para encontrar la cantidad de renglones que han sido cambiados por la ejecución de una de las sentencias INSERT, UPDATE o DELETE.

También se pueden ejecutar operaciones del tipo DDL (Data Definition Language), tales como sentencias que permiten crear (CREATE) o vaciar (DROP) una base de datos.

Y existen otras sentencias que permiten administrar una base de datos.

Múltiples Handles Sql y conexiones a bases de datos

Se puede usar más de un Handle Sql en una aplicación. Cada Handle Sql lleva la pista sobre cada una de las diferentes conexiones a la(s) base(s) de datos.

Si se necesita usar más de una sentencia SQL a la vez, se deberá hacer una segunda conexión con otro Handle Sql, al hacer esto:

- Una aplicación puede tener múltiples conexiones (Handles Sql) a la misma base de datos.
- Una aplicación puede conectarse a múltiples bases de datos. Las bases de datos pueden estar en el mismo o en un diferente servidor.

3.5.2 Interfaz de un-solo-paso

Esta interfaz tiene tan solo dos funciones:

Función	Descripción
SqlImmediate	Conecta un Handle Sql interno, prepara y ejecuta una sentencia SQL.
SqlClearImmediate	Desconecta el Handle Sql creado con SqlImmediate

SqlImmediate ejecuta a las funciones SqlConnect, SqlPrepare, SqlExecute y (sólo para un SELECT) un SqlFetchNext.

La primera vez que se llama a SqlImmediate, SQLWindows llama a todas estas funciones. En posteriores llamadas, SQLWindows no llama a SqlConnect si el Handle Sql permanece conectado.

Para la conexión, SQLWindows utiliza los valores actuales de las variables del sistema SqlDatabase, SqlUser y SqlPassword.

Un SqlImmediate se puede utilizar para:

- Hacer un SELECT de un único renglón de la base de datos
- Ejecutar sentencias del tipo DML, tales como INSERT, UPDATE o DELETE
- Hacer un COMMIT o un ROLLBACK de la transacción actual.
- Ejecutar sentencias del tipo DDL.
- Ejecutar sentencias del tipo DCL (Data Control Language), que permiten controlar el acceso y seguridad. Ejemplos de éstas son ALTER PASSWORD, GRANT y REVOKE.

SQLWindows maneja el Handle Sql internamente, por lo que no podrá ser utilizado por otras funciones Sql*.

Si se llega a llamar un SqlImmediate varias veces para diferentes bases de datos, SQLWindows reutiliza el Handle Sql interno al desconectar y reconectar.

SqlClearImmediate desconecta el Handle Sql interno y libera los recursos ocupados. También ejecuta un COMMIT implícito si éste desconecta el último Handle Sql conectado a la base de datos.

Ejemplo

Seleccionar y actualizar un renglón haciendo dos llamadas a SqlImmediate.

```
Global Declarations
Variables
  String: strSeleccionar
  String: strActualizar
...
Application Actions
On SAM_AppStartup
  Set strSeleccionar = 'SELECT nombre, direccion, credito INTO :dfInvitado,
:dfDireccion, :dfCredito FROM Cartera WHERE nombre = :dfInvitado'
  Set strActualizar = 'UPDATE Cartera SET direccion = :dfDireccion,
credito = :dfCredito, nombre = :dfInvitado'
...
...
Set bSeleccionar = SqlImmediate( strSeleccionar )
...
Set bOk = SqlImmediate( strActualizar )
If bOk
  Call SqlImmediate( 'COMMIT' )
```

3.5.3 List Boxes y Combo Boxes

Existe una función, SallistPopulate, para poblar (llenar) un List Box o un Combo Box con un conjunto de resultados. Si la sentencia SQL ya se encuentra preparada, SallistPopulate solamente la ejecuta y hace la transferencia de los renglones.

Ejemplo

Hacer que un Combo Box contenga la lista de carreras de la ENEP para que el usuario seleccione alguna de ellas.

```
! Poblar el Combo de Carreras.
Call SallistPopulate( cmb_carrera, hSqlConexión,
'SELECT Descripción FROM Carreras')
```

Capítulo 4

Sistema de Control de Adquisiciones

Objetivo General

- Observar en forma práctica cómo es que se hace un diseño orientado a objetos

Objetivos Particulares

- Entender qué es lo que hace cada módulo del Sistema de Control de Adquisiciones
- Visualizar el esquema de la base de datos utilizada
- Pasar de lo abstracto del "qué" al "cómo" definiendo las clases para crear los objetos, recordando que la clase es una especie de plantilla y que el objeto ya es una entidad real que existe basada en esa plantilla

En el Capítulo 2 se analizó el proceso manual del sistema de adquisiciones y se llegó a establecer una normatividad. Ahora esa normatividad se utiliza para establecer la estructura del nuevo sistema, al aplicarla destaca el hecho de que en forma automática el sistema queda dividido por módulos.

En la sección 4.1 se describe cada uno de estos módulos, pero comenzando con una visión en conjunto y después particularizando.

En la sección 4.2 se hace la descripción, obligada, de las entidades y relaciones que conforman la base de datos del sistema de adquisiciones. Se observará que ciertas entidades están estrechamente vinculadas a módulos en el sistema, pues específicamente pueden ser creados para manipular la información de dichas entidades.

En la sección 4.3 se habla sobre la fase posterior al análisis, la cual es el desarrollo. Como es de esperarse, no se describirá el sistema en su totalidad, a cambio se analizará una de las partes más descriptivas por ser la más compleja: El manejo de los datos, es decir, su captura y modificación.

4.1 Descripción de Módulos

Los módulos que componen al sistema de control de adquisiciones son propuestos casi automáticamente por la normatividad establecida.

En la normatividad se menciona que se maneja de una manera especial a las Solicitudes, Pedidos y Formas F-2, esto crea a los tres primeros módulos. También existen Catálogos de datos, los cuales están integrados por información que más o menos permanece constante y que es de apoyo para el resto del sistema.

Otro de los módulos es el de Consultas, disponible sólo para los tres principales documentos. Aquí se permite, generalmente, obtener totales sobre el estado o cantidad de documentos, desglosarlos por carrera, obtener el ejercicio del presupuesto, o incluso, listados de material adquirido y no adquirido. Los catálogos carecen de este tipo de consulta por ser sólo información de apoyo y que, una vez capturada no variará en gran medida dentro del sistema.

También cuenta con un módulo de Ayuda, que funge como manual en línea, donde se hace una descripción total del sistema y sus opciones, se tratan las generalidades

sobre la captura, selección, consulta e impresión. Particularizando, por supuesto, en la ayuda del manejo de cada uno de los documentos.

Por otra parte, no existe el famoso módulo de muchos sistemas, el módulo de Reportes, ya que a lo largo de los módulos de manejo de los documentos, catálogos y consultas, si algo es necesario que se imprima, la opción para ello existirá. Incluso, cuando sea necesario, es posible imprimir la pantalla o habrá diferentes opciones para obtener más de un reporte.

Diseñado de esta manera, el Sistema de Control de Adquisiciones queda integrado como lo muestra la Figura 4-1.

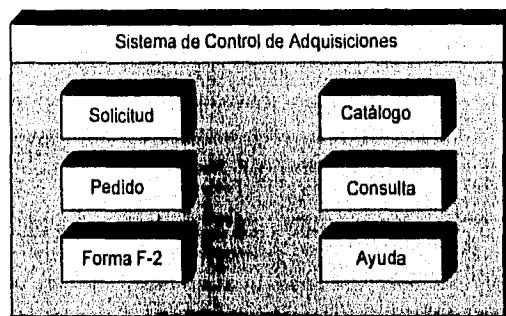


Figura 4-1: El Sistema de Control de Adquisiciones

En los siguientes apartados se revisará cada uno de los módulos que conforman el sistema. Para los fines que persigue esta tesis, sólo se describirán las operaciones que se realizan en cada módulo para proponer una solución orientada a los objetos. Se indicará qué se necesita hacer y enseguida el cómo hacerlo.

4.1.1 Módulo de Solicitudes

Este módulo es el que permite Agregar una nueva solicitud, Modificar, Eliminar o Consultar las ya existentes. Es posible imprimir la pantalla.

Existe una opción que se llama Selección, al elegirla, se presenta otra pantalla donde es posible indicar ciertas condiciones para visualizar las solicitudes que cumplan con el criterio de selección. Las solicitudes se listarán en forma de tabla.

La primera condición puede ser el especificar un par de fechas de un periodo del cual se esté interesado. Las solicitudes deseadas estarán comprendidas en ese lapso de tiempo y se interpretará como el obtener todas aquellas solicitudes desde la primera fecha hasta la segunda fecha, inclusive.

La segunda condición parte de que las solicitudes pueden tener uno de tres estados, Considerada, En Espera y No Considerada, y es posible indicar alguno de ellos o simplemente ninguno, lo cual se interpreta que se desean solicitudes de cualquier estado.

La tercera condición es que si se desea buscar por algún dato en especial, sólo es necesario elegir el dato y especificarlo en forma total o parcial. Es posible buscar una solicitud por su No. de solicitud, Título, Clasificación, No. de adquisición, No. de matriz. O se puede visualizar un grupo de solicitudes al buscar por No. de factura, Tema, Autor, Editorial, Carrera y Proveedor.

Al terminar de hacer la combinación del periodo, estado y dato en particular y se hace clic en el botón de Selección, se visualizarán todas aquellas solicitudes que cumplan con el criterio establecido, o si no existen ese tipo de solicitudes, simplemente no se visualizará información en la tabla.

También es posible tomar la condición inicial de esta pantalla de selección que es la que no maneja un periodo, la que considera las solicitudes de todos los estados y la que no busca por algún dato en particular. Esto logra visualizar todas las solicitudes existentes del año lectivo de que se trate ordenadas por el No. de solicitud.

Es posible imprimir todas las solicitudes visualizadas al elegir el botón Imprimir, para esto se presentan dos opciones, una es el formato que imprime la totalidad de los datos para la revisión de la captura y la otra opción es la que imprime un formato que contiene sólo los datos más importantes a manera de resumen.

Una vez visualizado un conjunto de solicitudes, es posible seleccionar alguna de ellas con el ratón y solicitar su modificación, para lo cual es necesario elegir el botón de Modificar que está sobre la tabla. La ventana de Selección se cierra, y la solicitud elegida queda lista para modificarse en la pantalla previa a la de selección.

4.1.2 Módulo de Pedidos

En este módulo se controlan los pedidos. Es posible Agregar o, dicho de otra manera, generar un nuevo pedido. Se puede Modificar, Eliminar o Consultar los ya existentes. Es posible imprimir la pantalla o el pedido a detalle.

El generar un pedido indica que es el sistema quien buscará todas las solicitudes disponibles para integrarlas en una o varios pedidos agrupándolas por proveedor. Esto evita la recaptura de la información, lo cual es una de las ventajas de un sistema informático. Las solicitudes deberán estar en estado Consideradas y En Espera, según la normatividad.

Por supuesto que para Eliminar se puede referir a una sola solicitud del pedido, o al pedido completo. Para ambas opciones se debe seguir la normatividad respecto a qué hacer con la(s) solicitud(es) que se está(n) descartando del pedido.

También aquí existe la opción de Selección, y funciona de la misma manera que en el módulo de Solicitudes. Sólo que aquí, por supuesto, se refiere a datos del pedido. Cuenta con un periodo de fechas, la selección del estado del pedido que puede ser Pendiente, No Emitido, En Espera, Surtido y Cancelado. También se puede buscar por No. de Pedido, Fecha de Pedido, Fecha de Surtido o Proveedor. La impresión consta de los datos que se visualizan en la tabla. Es posible elegir un pedido y pasar a modificarlo.

Se tendrá que solicitar la modificación de un pedido para poder hacer la Recepción del mismo. Proceso que da por recibidos los títulos solicitados, ésto en el mejor de los casos, porque es posible dejarlo pendiente o cancelarlo por haberse presentado ciertas irregularidades al intentar surtirlo.

4.1.3 Módulo de Formas F-2

Este módulo permite Agregar una nueva forma F-2. También es posible Modificar, Eliminar, Consultar e Imprimir cualquiera de las ya existentes.

Una forma F-2, a diferencia de una solicitud o de un pedido, maneja ciertos datos que se capturan como en su propio módulo de captura, ya que se permite hacer las mismas operaciones de Agregar, Modificar, Eliminar o Consultar sobre ellos. A estos documentos de que está integrada una forma F-2 se les referirá como subdocumentos.

Estos subdocumentos son cada una de las facturas y la posible nota de crédito que puede llegar a tener una forma F-2.

Al modificar una forma F-2, aparecen dos botones, uno que permite manejar las facturas, y el otro que permite manejar la única nota de crédito a que tiene derecho una forma F-2. Al elegir uno u otro botón aparece otra ventana muy similar a la que se está visualizando, sólo que ahí se capturará o modificará el subdocumento elegido.

Por lo tanto, el manejo de las facturas y de la nota de crédito es como tener dos submódulos.

Al igual que en los dos documentos anteriores, aquí también existe la opción de Selección. Cuenta con los mismos elementos dados para solicitudes y pedidos. Se puede indicar un periodo de fechas, seleccionar las formas F-2 Aceptadas o Canceladas, buscar por No. de F-2 o por Proveedor. También es posible imprimir la tabla y elegir una forma F-2 para modificarla en la pantalla inmediata anterior.

4.1.4 Módulo de Catálogos

Los catálogos son datos que se mantienen más o menos fijos en el sistema. Se dividen en cinco áreas o submódulos para un mejor control de la información. Figura 4-2.

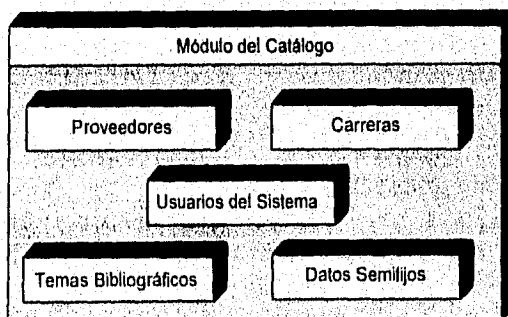


Figura 4-2. Módulo - Catálogo

Para cada uno de los catálogos se permiten ciertas operaciones de actualización, lo cual depende de la información manejada. Ninguno de ellos cuenta con la opción de

Selección, ya que su volumen de información es muy bajo y mas o menos constante, por lo que no amerita tal operación que es para moverse en un gran volumen de datos.

En seguida se describirán las operaciones permitidas para cada uno de los catálogos. Recordando que el objetivo de cada uno de los catálogos está definido en la normatividad.

Proveedores, Temas Bibliográficos y Carreras:

- Agregar
- Modificar
- Eliminar
- Consulta
- Imprimir

Datos Semifijos:

- Agregar
- Modificar
- Consulta

Usuarios del Sistema:

- Agregar
- Modificar
- Eliminar

En la captura de las representaciones exclusivas de un proveedor:

- Agregar
- Modificar
- Eliminar
- Consulta

4.1.5 Módulo de Consultas

Las consultas existen sólo para los tres documentos básicos del Sistema de Control de Adquisiciones.

Las consultas esencialmente despliegan una especie de resumen según el tipo de consulta elegida.

Las consultas son para las solicitudes, pedidos y formas F-2, ya que la cantidad de información requiere monitorear ciertos aspectos cuantitativos para medir aspectos cualitativos, tales como:

- material solicitado por carrera
- material adquirido por carrera
- material pendiente por adquirir
- material pendiente por surtir

Además, es posible monitorear el aspecto más importante:

- el ejercicio del presupuesto

En seguida se listan los diferentes tipos de consulta disponibles para cada documento.

Solicitudes por:

- Estado
- Carrera
- No. de Factura

Pedidos por:

- Estado
- Carrera
- Fecha de Pedido
- Fecha de Surtido
- Ejemplares por Carrera (Material Adquirido y no Adquirido)

Formas F-2 por:

- Estado
- Ejercicio

A excepción de la consulta a solicitudes por No. de Factura, en las demás es posible indicar un rango de fechas para la consulta. Todas ellas pueden ser impresas.

Existen algunas diferencias de consulta a consulta, como por ejemplo en la consulta a solicitudes por No. de factura debe especificarse este dato, o en la consulta a pedidos de ejemplares por carrera se puede hacer la selección de entre material adquirido y el no adquirido, además de elegir todas las carreras o una en especial.

4.1.6 Módulo de Ayuda

La Ayuda es la opción que proporciona un manual de referencia para ejecutar los pasos necesarios para la utilización del Sistema de Control de Adquisiciones.

Este último módulo no se desarrolla con SQLWindows. La herramienta para crearlo es el compilador de ayuda de Windows 3.1, HC31.EXE. Es una utilidad del ambiente DOS que, entre otros, se puede encontrar en el software de Microsoft titulado Software Development Kit, SDK.

Al compilarse el archivo de la ayuda, que debe tener el formato RTF (Rich Text Format) se crea un archivo HLP. El archivo RTF se puede crear con un procesador de textos que permita almacenar un archivo en este formato, como por ejemplo, Word de Microsoft.

SQLWindows sólo provee la interfaz para poder acceder a la ayuda de Windows, lo cual se logra con la función SalWinHelp que con los parámetros adecuados es posible desplegar el índice de la ayuda o un tema en especial.

La ayuda de Windows es manejada por la aplicación WINHELP.EXE, proporciona un modo práctico para presentar información sobre las aplicaciones en un formato en el que los usuarios pueden acceder rápidamente un tema deseado. Es posible moverse en la ayuda como en un libro, hacia atrás o hacia adelante e incluso con saltos hacia subtemas y regresar al mismo contexto.

En la siguiente sección se hablará de la información que manipulará este Sistema de Control de Adquisiciones. La información debe almacenarse de una forma muy organizada dando lugar a entidades, las cuales son las tablas. Por lo tanto, la sección 4.2 describe la base de datos en su totalidad.

4.2 Entidades y relaciones para la base de datos

El conjunto de información de este sistema se almacenará en un único archivo de datos que SQLBase conoce como **Base de Datos**.

Dentro de la base de datos, la información íntimamente relacionada queda almacenada en una tabla, de tal manera que existen varias tablas, una o más por documento.

Recordando que una tabla contiene **columnas y renglones**, cada columna está relacionada de alguna manera con otras columnas en el mismo renglón. Cada columna tiene un nombre, un tipo de dato y una longitud. Cada columna contiene un valor en la intersección de un renglón y una columna.

A continuación se detallarán cada una de las tablas, o entidades como también se les conoce, del Sistema de Control de Adquisiciones y se indicará a qué módulo pertenece, es decir, dónde es que se captura la información.

Tabla:	Solicitud
Propósito:	Almacenar las solicitudes de material bibliográfico
Módulo:	Solicitud

Columna	Tipo de dato	Longitud	Descripción
Fecha	FECHA	8	Fecha en la cual se captura la solicitud
No. Pedido	CARACTER	2	No. de Pedido al cual pertenece
Fecha Pedido	FECHA	8	Fecha del pedido al cual pertenece
Clave Prov	CARACTER	5	Clave del proveedor de este material
Clave Carrera	CARACTER	3	Clave de la carrera a la cual apoya este material
Clave Tema	CARACTER	2	Clave del tema bibliográfico para este material
Título	CARACTER	60	Título del material
Ejemplares	DECIMAL	3	Cantidad de ejemplares a surtir
Autor	CARACTER	60	Autor del material
Editorial	CARACTER	40	Editorial a la que pertenece el material
Profesor	CARACTER	60	Profesor solicitante
Materia	CARACTER	40	Materia a la cual apoya este material
Semestre	CARACTER	2	Semestre al cual apoya este material
Observaciones	CARACTER	80	Comentarios
No. Factura	CARACTER	15	No. de factura en la cual fue surtido este material
Clasificación	CARACTER	25	No. de clasificación que asignó la DGII
Adquisición	CARACTER	10	No. de adquisición que asignó la DGB
Matriz	CARACTER	10	No. de matriz que asignó la DGB
Para Pedido	CARACTER	1	Situación actual de este material
Estado Anterior	CARACTER	1	Situación anterior de este material

Tabla:	Pedido
Propósito:	Almacenar los pedidos de material bibliográfico
Módulo:	Pedido

Columna	Tipo de dato	Longitud	Descripción
No. Pedido	CARACTER	2	No. de pedido
Fecha Pedido	FECHA	8	Fecha en la cual se genera este pedido
Clave Prov	CARACTER	5	Clave del proveedor al cual pertenece este pedido
Fecha Surtido	FECHA	8	Fecha en que se surte el pedido
Observaciones	CARACTER	80	Comentarios
Estado	CARACTER	1	Situación actual de este pedido

Tabla:	FormaF2
Propósito:	Almacenar las Formas F-2
Módulo:	Forma F-2

Columna	Tipo de dato	Longitud	Descripción
No. F2	CARACTER	3	No. de la forma F-2
Fecha	FECHA	8	Fecha de captura de la forma F-2
Clave Prov	CARACTER	5	Clave del proveedor al cual pertenece esta forma F-2
Total Títulos	DECIMAL	10	Cantidad total de títulos de esta forma F-2
Total Ejemplares	DECIMAL	10	Cantidad total de ejemplares de esta forma F-2
Total Importe	DECIMAL	11.2	Cantidad total del importe que ampara esta forma F-2
Estado	CARACTER	1	Situación actual de esta forma F-2

Tabla:	SubFormaF2
Propósito:	Almacenar las facturas y notas de crédito de las Formas F-2
Módulo:	Forma F-2

Columna	Tipo de dato	Longitud	Descripción
No. F2	CARACTER	3	No. de la forma F-2 a la que pertenece
Fecha	FECHA	8	Fecha de captura de la forma F-2
No. Factura	CARACTER	15	No. de factura
No. Títulos	DECIMAL	8	Cantidad de títulos de la factura
No. Ejemplares	DECIMAL	8	Cantidad de ejemplares de la factura
Importe	DECIMAL	9.2	Importe de la factura

Tabla:	Proveedores
Propósito:	Almacenar la relación de los proveedores
Módulo:	Catálogo de Proveedores

Columna	Tipo de dato	Longitud	Descripción
Clave Prov	CHARACTER	5	Clave asignada a cada proveedor
Nombre	CHARACTER	60	Nombre del proveedor
RFC	CHARACTER	13	R.F.C. del contribuyente
Fecha Alta	FECHA	8	Fecha en que se captura a este proveedor
Atencion	CHARACTER	60	Personas que atienden las solicitudes de pedidos
Direccion 1	CHARACTER	40	Dirección del proveedor
Direccion 2	CHARACTER	40	Dirección del proveedor
Direccion 3	CHARACTER	40	Dirección del proveedor
Teléfono	CHARACTER	40	Números de teléfono, FAX y extensiones
Horario	CHARACTER	40	Horario de atención al público
Editor Tipo	CHARACTER	1	¿Será editor?
Editor Descuento	CHARACTER	5	Porcentajes de descuento que hace como editor
Dist Tipo	CHARACTER	1	¿Será distribuidor?
Dist Descuento	CHARACTER	5	Porcentajes de descuento que hace como distribuidor
RepExc Tipo	CHARACTER	1	¿Será representante exclusivo?
Nacional	CHARACTER	1	¿Es este proveedor nacional o extranjero?
Tiempo Entrega	CHARACTER	3	Tiempo mínimo de entrega
Tiempo Descuento	CHARACTER	5	Porcentajes de descuento al surtir en el tiempo mínimo
Autorizado UNAM	CHARACTER	1	¿Es este proveedor autorizado por la UNAM?
Atención Cliente	CHARACTER	1	Calificación otorgada por su atención a clientes

Tabla:	SubProvTemas
Propósito:	Almacenar los temas bibliográficos que manejan los proveedores
Módulo:	Catálogo de Proveedores

Columna	Tipo de dato	Longitud	Descripción
Clave Prov	CHARACTER	5	Clave del proveedor
Clave Tema	CHARACTER	2	Clave del tema bibliográfico que maneja este proveedor

Tabla:	SubProvLista
Propósito:	Almacenar las representaciones exclusivas de cada proveedor
Módulo:	Catálogo de Proveedores

Columna	Tipo de dato	Longitud	Descripción
Clave Prov	CHARACTER	5	Clave del proveedor
Descripción	CHARACTER	30	Descripción del material o editorial representado
Descuento	CHARACTER	5	Porcentajes de descuento por esta representación

Tabla:	Temas
Propósito:	Almacenar la relación de temas bibliográficos que maneja la UNAM
Módulo:	Catálogo de Temas

Columna	Tipo de dato	Longitud	Descripción
Clave_Tema	CARACTER	2	Clave asignada a este tema bibliográfico
Descripcion	CARACTER	30	Nombre del tema bibliográfico

Tabla:	Carreras
Propósito:	Almacenar la relación de carreras de la ENEP Aragón
Módulo:	Catálogo de Carreras

Columna	Tipo de dato	Longitud	Descripción
Clave_Carrera	CARACTER	3	Clave asignada a la carrera
Descripción	CARACTER	30	Nombre de la carrera

Tabla:	DatosSemifijos
Propósito:	Almacenar la relación de los datos semifijos para cada año
Módulo:	Catálogo de Datos Semifijos

Columna	Tipo de dato	Longitud	Descripción
Periodo	CARACTER	2	Año para el cual son estos datos
Codigo	CARACTER	20	Código programático para la biblioteca
Director_DGB	CARACTER	60	Nombre del director de la DGB
Director_ENEP	CARACTER	60	Nombre del director de la ENEP Aragón
Director_Biblio	CARACTER	60	Nombre del jefe de la biblioteca de la ENEP Aragón
Presupuesto	DECIMAL	11	Presupuesto asignado
Efectivo	CARACTER	1	¿Es este año el que controla al sistema?

Tabla:	Usuarios
Propósito:	Almacenar la relación de los usuarios del sistema
Módulo:	Catálogo de Usuarios

Columna	Tipo de dato	Longitud	Descripción
Nombre	CARACTER	60	Nombre del usuario autorizado
Palabra	CARACTER	10	Palabra secreta de acceso
Categoria	CARACTER	1	¿Es el usuario principal o es un usuario normal?

La siguiente lista muestra a cada documento y la relación de dependencia que hay entre ellos.

#	Documento	Depende de
1	Solicitud	
2	Pedido	
3	Forma F-2	
4	Facturas de la Forma F-2	3
5	Nota de crédito de la Forma F-2	3
6	Proveedores	
7	Temas bibliográficos de un proveedor	6
8	Representaciones exclusivas de un proveedor	6
9	Temas bibliográficos de la UNAM	
10	Carreras de la ENEP Aragón	
11	Datos Semifijos	
12	Usuarios del Sistema	

Las tablas en su totalidad se integran en la base de datos. La Figura 4-3 muestra el diagrama de entidad-relación donde se conjunta la totalidad de las tablas.

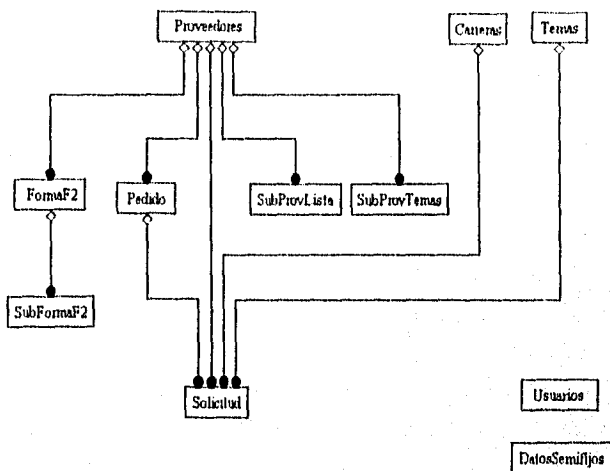


Figura 4-3: Diagrama entidad-relación para la base de datos

Hasta aquí se han descrito las dos partes del sistema: los módulos y la información. En la siguiente sección se describirá el desarrollo, la parte que pasa de lo abstracto a lo concreto, de las ideas en mente a los programas en papel.

4.3 Desarrollo

El desarrollo se dividirá en dos partes:

- Qué se necesita hacer
- Cómo hay que hacerlo

La atención se enfocará a la interfaz del sistema, es decir, a lo que debe suceder al elegir cierta acción. El ejemplo se basará en la edición de cada uno de los documentos.

4.3.1 Qué se necesita hacer

Para hacer la captura y modificación de cualquiera de los documentos se necesitan similares operaciones, tales como agregar, eliminar, etc., sobre los datos de cada uno de ellos.

De la sección 4.1 se tiene que para cada documento se necesitan ciertas operaciones que puedan afectar sus datos. Así, una Solicitud puede aceptar acciones de Agregar, Modificar, Eliminar, Consulta, Selección e Imprimir. Para el documento Usuarios del Sistema, sólo es permitido Agregar, Modificar y Eliminar, ya que se requieren las mínimas operaciones sobre estos datos confidenciales.

De este análisis se desprende la siguiente relación que muestra los documentos y las operaciones permitidas para cada uno de ellos.

	Documento	Operaciones					
		Agregar	Modificar	Eliminar	Consulta	Selección	Imprimir
1	Solicitud	✓	✓	✓	✓	✓	✓
2	Pedido	*	✓	✓	✓	✓	✓
3	Forma F-2	✓	✓	✓	✓	✓	✓
4	Facturas de la Forma F-2	✓	✓	✓	✓		
5	Nota de crédito de la Forma F-2	✓	✓	✓	✓		
6	Proveedores	✓	✓	✓	✓		✓
7	Temas bibliográficos de un prov.	**	✓	✓	✓		
8	Repres. Exclus. de un proveedor	✓	✓	✓	✓		
9	Temas bibliográficos	✓	✓	✓	✓		✓
10	Carreras de la ENEP Aragón	✓	✓	✓	✓		✓
11	Datos Semifijos	✓	✓		✓		
12	Usuarios del Sistema	✓	✓	✓			

* No se captura un pedido. Éste se genera a partir de las solicitudes ya capturadas en estado Consideradas. La confirmación de un pedido es, entonces, automática.

** Se necesita que la lista de temas sea visible todo a la vez para que el usuario simplemente seleccione los temas deseados para un proveedor en particular.

Con estas operaciones se podría integrar una barra de herramientas comunes para todos los documentos. Estas herramientas estarían representadas por botones que permitan realizar estas funciones.

Cada herramienta será representada por la descripción apropiada y reforzada gráficamente. La Figura 4-4 muestra a los botones que equivalen a las operaciones de la tabla anterior.



Figura 4-4: Herramientas de acciones

Por otra parte, suponiendo que se inicia la operación de Agregar, sería conveniente poder cancelar la operación, o en su defecto, confirmar el nuevo documento. Con esto surgen dos nuevas herramientas, la de Aceptar y la de Cancelar, para confirmar o anular una operación, respectivamente.

También será necesaria otra operación, la que permita cerrar la ventana de captura de cualquier documento. Esto para proceder con otra tarea tal como entrar al módulo de consultas o simplemente abandonar el sistema.

Por lo tanto, se tienen otras tres herramientas. Figura 4-5.



Figura 4-5: Herramientas complementarias

Por último, es necesaria una ventana en la cual capturar los datos de cada uno de los documentos. Independientemente de la cantidad de ellos se podría convenir en una ventana con los botones de las herramientas en posiciones fijas. En los documentos donde ciertas herramientas no hagan falta, sus espacios se respetarían acentuando el hecho de que esas operaciones no son permitidas.

El formato estándar para la manipulación de los datos de un documento sería el de la Figura 4-6.

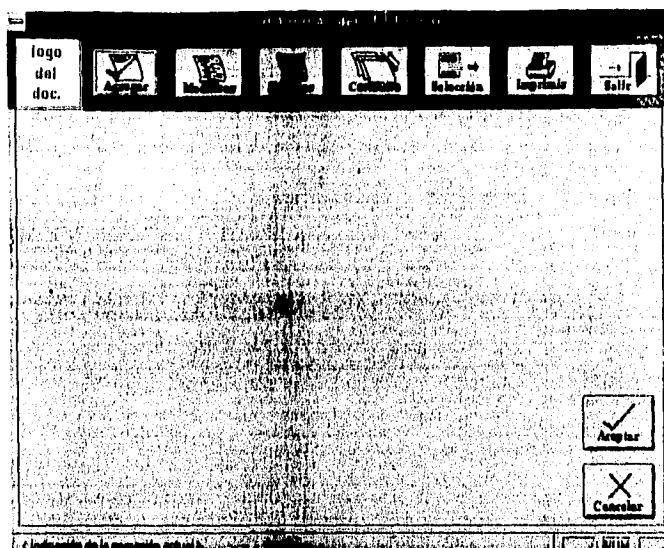


Figura 4-6: Ventana estándar para la captura

4.3.2 Cómo hay que hacerlo

¿Sobre qué actúan los botones de las herramientas?

Las herramientas, para abreviar, deben actuar sobre los datos del documento.

Dentro de un documento existen algunos datos que controlan a los demás. A esos datos se les llamará **campos llave**.

Dentro de una tabla en la base de datos los campos llave son una columna o un grupo de columnas cuyo valor único identifica cada renglón, es decir, cada documento.

Los campos llave permiten tener el control de un documento. En el caso de un proveedor, el campo llave sería su CLAVE, de tal manera que sólo se permitirá que un proveedor tenga esa clave. Al agregar, se verifica que la nueva clave que se

intenta asignar no exista para otro proveedor. Cuando un proveedor tenga su clave, de ahí en adelante por medio de ella se podrá localizar en el catálogo.

Para el Sistema de Control de Adquisiciones cuando mucho se utilizan dos campos tipo llave por documento. La siguiente tabla lista los campos llave según el documento.

Documento	Campo 1	Campo 2
Solicitud	No. de solicitud	Fecha de captura
Pedido	No. de pedido	Fecha de pedido
Forma F-2	No. de forma F-2	Fecha de captura
Proveedores	Clave de proveedor	
Temas Bibliográficos	Clave de tema	
Carreras	Clave de carrera	
Datos Semifijos	Año	
Usuarios del Sistema	Nombre	

Para abreviar, a los campos llave se les llamará simplemente llave, sin importar que se refiera a un solo campo o a varios.

Manejo de las acciones

Las acciones para las herramientas estarán supeditadas al manejo de los campos llave.

Para Agregar se deberán habilitar los Data Field de la llave para mecanografiarla y hasta entonces validarla de tal manera que sea única, de lo contrario el sistema debe rechazarla por no permitir duplicados. Esto de la validación ya es responsabilidad de la llave y será diferente para cada documento.

Para las demás operaciones es necesario trabajar con documentos que se encuentren en la base de datos. Se evitará el trabajo de mecanografiar la clave del documento deseado si en lugar de Data Fields se colocan Combo Boxes.

Un Combo Box podrá ser poblado con todas las claves de documentos disponibles donde la tarea del usuario sea la de localizar y seleccionar el documento deseado.

De cualquier manera, la condición normal de inicio de captura, ya sea porque se acabe de entrar a captura o se haya Aceptado o Cancelado, será el que se visualicen los Data Fields de llave (limpios).

Dado este manejo de objetos -Data Fields y Combo Boxes, es obvio que se involucran operaciones de ocultar y visualizar dichos objetos.

Esto da pauta a un control específico para la herramienta Agregar y para las herramientas Modificar, Eliminar y Consulta. La herramienta Selección no hace más que llamar a otra ventana, por lo que no interviene directamente en las acciones de edición en la pantalla actual de captura.

La herramienta Imprimir inicia el proceso de impresión, lo cual indica que se habilita hasta que la llave haya sido validada y que los datos del documento se hayan desplegado en la pantalla. Antes de eso no hay nada que imprimir.

Igual comportamiento tiene la herramienta Aceptar, pues hasta que no haya datos es que no se podrá aceptar nada. Esto indica que sólo la herramienta Cancelar se habilita de entrada (junto con los campos de la llave) porque posiblemente se desee cancelar la operación sin antes completar o validar la llave.

Enseguida se describirá la acción general para cada herramienta. Se usará pseudocódigo por ser un método conveniente para describir un programa y enseguida se presentará el código fuente para todas las herramientas.

Herramienta Agregar

Indicar en la barra de estado: 'Agregando un registro...'
Deshabilitar herramientas superiores
Habilitar campos de la llave
Habilitar la herramienta de Cancelar
Poner el foco en el primer Data Field de la llave

Herramienta Modificar,

Eliminar,

Consulta

Indicar en la barra de estado: 'Modificando, Eliminando, o Consultando un reg.'
Deshabilitar las herramientas superiores
Ocultar el (los) Data Field de la llave
Visualizar el (los) Combo Box de la llave
Habilitar la herramienta de Cancelar
Habilitar el (los) Combo Box de la llave
Poner el foco en el primer Combo Box de la llave

Herramienta Selección

Abrir la ventana de selección de acuerdo con el documento apropiado

Herramienta Imprimir

Iniciar proceso de impresión de acuerdo al documento

Herramienta Salir

Simplemente destruir la instancia u objeto de esta Ventana

Herramienta Aceptar

Para Agregar

Verificar que no falte un dato importante
Si hace falta un dato requerido
Indicar al usuario y retornar a edición
Si no hace falta un dato requerido
Convertir Radio Buttons y Check Boxes en valores que acepte una tabla
Hacer la inserción en la base de datos

Para Modificar

Verificar que no falte un dato importante
Si hace falta un dato requerido
Indicar al usuario y retornar a edición
Si no hace falta un dato requerido
Convertir Radio Buttons y Check Boxes en valores que acepte una tabla
Hacer la modificación en la base de datos

Para Eliminar

Pedir confirmación al usuario
Si desea eliminar
Hacer la eliminación en la base de datos

! Para Consulta, o independientemente de la operación dejar todo como estaba

Habilitar las herramientas superiores

Deshabilitar los campos de edición

Limpiar los campos de edición

Ocultar el (los) Combo Box de la llave (por si están visibles)

Visualizar el (los) Data Field de la llave (por si están ocultos)

Deshabilitar las herramientas de Aceptar y Cancelar

Indicar en la barra de estado: 'Selecciona una acción'

Herramienta Cancelar

! Dejar todo como estaba

Habilitar las herramientas superiores

Deshabilitar los campos de edición

Limpiar los campos de edición

Ocultar el (los) Combo Box de la llave (por si están visibles)

Visualizar el (los) Data Field de la llave (por si están ocultos)

Deshabilitar las herramientas de Aceptar y Cancelar

Indicar en la barra de estado: 'Selecciona una acción'

Se observa que hay ciertas operaciones repetitivas que se realizan sin importar la herramienta, lo cual da la pista de poder crear subrutinas o funciones donde se concentre una sola vez el código repetido y referirse a él cuantas veces sea necesario. Esas tareas se concentrarán en tres funciones con operaciones específicas.

La siguiente es la relación de operaciones para cada función, ya con un nombre definido:

1. PantallaEdición()

- Habilitar la herramienta Cancelar
- Deshabilitar las herramientas superiores
- Habilitar los campos de la llave

2. PantallaNormal()

- Deshabilitar las herramientas Aceptar y Cancelar
- Habilitar las herramientas superiores
- Deshabilitar los campos de edición

3. EditarClave()

- Coloca el foco en el primer campo de llave

Sin que todavía sea necesario codificar las funciones, el código para las herramientas será el siguiente.

```
Cpb_Stdr: pbAgregar
Message Actions
On SAM_Click
! Quedamos en el modo de Agregar
Call SalStatusSetText( hWndForm, 'Agregando un registro...' )
Set sModo = 'Agregar'

! Habilitar y deshabilitar botones y campos
Call PantallaEdicion()
Call EditarClave()

Cpb_Stdr: pbModificar
Message Actions
On SAM_Click
! Quedamos en el modo de Modificar
Call SalStatusSetText( hWndForm, 'Modificando un registro...' )
Set sModo = 'Modificar'

! Habilitar y deshabilitar botones y campos
Call PantallaEdicion()
```

```
Cpb_Stdr: pbEliminar
Message Actions
On SAM_Click
! Quedamos en el modo de Eliminar
Call SalStatusSetText( hWndForm, 'Eliminado un registro...' )
Set sModo = 'Eliminar'
! Habilitar y deshabilitar botones y campos
Call PantallaEdicion()

Cpb_Stdr: pbConsulta
Message Actions
On SAM_Click
! Quedamos en el modo de Consulta
Call SalStatusSetText( hWndForm, 'Consultando un registro...' )
Set sModo = 'Consulta'
! Habilitar y deshabilitar botones y campos
Call PantallaEdicion()

Cpb_Stdr: pbSelección
Message Actions
On SAM_Click
! Forma para la selección de un registro
Call Seleccion()

Cpb_Stdr: pbImprimir
Message Actions
On SAM_Click
Call Imprimir()

Cpb_Stdr: pbSalir
Message Actions
On SAM_Click
! Simplemente se destruye la instancia u objeto de esta forma.
Call SalDestroyWindow( hWndForm )

Cpb_Stdr: pbAceptar
Message Actions
On SAM_Click
Call SalWaitCursor( VERDAD )
! Entrar cuando hay edición.

If sModo != 'Consulta'
If sModo = 'Agregar'
! Verificar que no falta algún dato importante.
Set bFaltaDato = FALSO
Call SalSendMessageToChildren( hWndForm, WM_ChecarContenido, 0, 0 )
If NOT bFaltaDato
! Preparar los rb, cb y los extras que se necesiten transformar
! antes de poder ser almacenados en la tabla correspondiente.
Call EscribirTipos()

! Insertar en la tabla
Call Agregar()
Else
Call SalWaitCursor( FALSO )
Call SalMessageBox( 'Falta algún dato requerido.',
'Aviso', MB_IconExclamation )
Return VERDAD
```

```

Elsu IF #Modo = 'Modificar'
! Permitir que no haya abono de importación
Set #Faltado = #FALSO
Call SalStatusSetText( #ModoForm, 'Modificación de', # )

! No #Faltado
! Preparar los db, db y los extras que se necesitan transferir
! Antes de poder ser insertados en la tabla correspondiente.
Call EscribirTipos()
! Modificar la tabla
Call Modificar()

Else
Call SalWaitCursor( #FALSO )
Call SalMessageBox( '¿Existe algún dato requerido?',
                    'Aviso', #B_IconExclamation )
Return #FALSO

Else ! Es Eliminar
If #YES = SalMessageBox( '¿Será eliminado el registro?',
                        'Aviso', #B_IconExclamation )
Call Eliminar()

! Dejar todo como estaba.
Call PantallaNormal()
Call SalWaitCursor( #FALSO )
Call SalStatusSetText( #ModoForm, 'Selecciona una acción' )

Cpb_Bldr: pbCancelar
Message Actions
On SAM_Click
! Dejar todo como estaba
Call SalWaitCursor( #FALSO )
Call PantallaNormal()
Call SalWaitCursor( #FALSO )

```

Para codificar las tres funciones que se mencionan, antes será necesario convenir en el comportamiento de la pantalla de edición al elegir determinadas acciones.

Al visualizar por primera vez una pantalla de captura o después de Aceptar o Cancelar una operación, independientemente del documento de que se trate, deberá cumplir con las siguientes reglas:

- Los campos de datos deberán estar en blanco y deshabilitados para edición
- Las herramientas superiores habilitadas listas para elegir una acción, excepto la de Imprimir, pues no hay nada que imprimir
- Las herramientas Aceptar y Cancelar deshabilitadas, ya que no hay nada que aceptar o cancelar.

Se observa que al elegir una acción, unos objetos (ya sea Data Fields, combo Boxes, Push Buttons) se habilitan, otros se deshabilitan, o hasta se ocultan o se visualizan.

Dado el comportamiento de las herramientas, se hace evidente que hay campos que tienen roles específicos en la captura de un documento. Por ejemplo, hay Data Fields para la captura de la llave, existen Combo Boxes para desplegar los campos llave disponibles, hay campos que se deben autoverificar para indicar si su contenido es nulo o no, y existen campos normales para la captura, que bien pueden ser Data Fields, List Boxes, Radio Buttons, etc.

Antes que nada y aprovechando que una aplicación de SQL Windows responde a mensajes, se crearán algunos que puedan ser mandados a los objetos involucrados en la captura de un documento. En seguida se listará la relación de mensajes creados con su respectivo objetivo.

Mensaje	Objetivo
AM_HabilitarEnEdicion	Habilitar cuando hay datos en la ventana
AM_DeshabilitarEnEdicion	Deshabilitar cuando no hay datos en la ventana
AM_Deshabilitar	Deshabilitar el objeto
AM_Habilitar	Habilitar el objeto
AM_EditarLlave	Habilitar el objeto, los relacionados con la llave y la propia llave
AM_Editar	Habilitar el objeto de edición normal
AM_NoEditar	Deshabilitar objeto de edición normal
AM_Limpiar	Limpiar el objeto
AM_Mostrar	Visualizar el objeto de la llave
AM_Ocultar	Ocultar el objeto de la llave
AM_MostrarCombo	Visualizar el Combo Box de la llave
AM_OcultarCombo	Ocultar el Combo Box de la llave
AM_EditarLlaveCombo	Habilitar el Combo Box de la llave para edición
AM_ChecarContenido	Verificar que el objeto no deba estar vacío
AM_ImprimirPantalla	Ocultar objetos no deseados en la impresión
AM_NoImprimirPantalla	Visualizar objetos que se ocultaron para imprimir la ventana

Para cumplir con el cometido de la captura, se crearán clases de campos de captura, desde Data Fields hasta Check Boxes. Al definir estas nuevas clases se integrarán no sólo los mensajes que manejarán cada uno de los distintos campos de edición, sino también los atributos y el tipo de dato a manejar.

La convención para el nombre de estas nuevas clases será el anteponer una letra C mayúscula como indicativo de que es una clase. En seguida, la abreviatura de tipo

de objeto de SQL Windows, tal como df, rb, etc. Por último, se colocará una frase abreviada de lo que intenta el objeto.

Se especificarán los atributos y mensajes que recibirán para poder cumplir con su cometido. Algunos objetos se heredan de otros para combinar características logrando que sean más amplios en sus funciones. La descripción se visualizará tal y como se haría en el Editor de Código.

Data Field Class: CdfFecha
Data Type: Date/Time
Justify: Left
Format: dd-MM-yy
Font Name: Arial
Font Size: 8
Input MASK: 99-99-99
List in Tool Palette? Yes
Description: Data Field con atributo de fecha, tipo de letra y tamaño.

Data Field Class: CdfNúmero
Data Type: Number
Justify: Right
Font Name: Arial
Font Size: 8
Description: Data Field con atributo de número, tipo de letra y tamaño.

Data Field Class: CdfCadena
Data Type: Class Default
Format: Uppercase
Font Name: Arial
Font Size: 8
Description: Data Field tipo cadena con tipo de letra y tamaño.

Data Field Class: CdfNúmeroFormato
Data Type: Number
Justify: Class Default
Format: #,##0
Description: Data Field numérico con formato de miles.

Data Field Class: Cdf_TipoLlave
Data Type: Class Default
Format: Uppercase
Description: Data Field para campos de la llave.
Responde al mensaje AM_EditarLlave, AM_MostrarCombo, AM_OcultarCombo
Message Actions
On **AM_EditarLlave**
Call Sa(EnableWindow(hWndItem)
On **AM_MostrarCombo**
Call Sa(HideWindow(hWndItem)
On **AM_OcultarCombo**
Call Sa>ShowWindow(hWndItem)

Data Field Class: Cdf_Editar
Data Type: Class Default
Justify: Class Default
Description: Data Field para edición.
Responde al mensaje AM_Editar.
Message Actions
On AM_Editar
Call SalEnableWindow(hWndItem)

Data Field Class: Cdf_NoEditar
Data Type: Class Default
Justify: Class Default
Description: Data Field para edición.
Responde al mensaje AM_NoEditar.
Message Actions
On AM_NoEditar
Call SalDisableWindow(hWndItem)

Data Field Class: Cdf_Limpiar
Data Type: Class Default
Justify: Class Default
Description: Data Field para edición.
Responde al mensaje de AM_Limpiar.
Message Actions
On AM_Limpiar
Call SalClearField(hWndItem)

Data Field Class: Cdf_Mostrar
Data Type: Class Default
Justify: Class Default
Description: Data Field para edición de la llave.
Responde al mensaje AM_Mostrar.
Message Actions
On AM_Mostrar
Call SalShowWindow(hWndItem)

Data Field Class: Cdf_Ocultar
Data Type: Class Default
Justify: Class Default
Description: Data Field para la edición de la llave.
Responde al mensaje AM_Ocultar.
Message Actions
On AM_Ocultar
Call SalHideWindow(hWndItem)

Data Field Class: Cdf_CampoCadenaLlave
Data Type: Class Default
Justify: Class Default
List in Tool Palette? Yes
Description: Sirve para editar algún componente de la llave.
Derived From
Class: Cdf_Mostrar
Class: Cdf_Ocultar
Class: Cdf_Limpiar
Class: Cdf_NoEditar
Class: Cdf_TipnLlave
Class: Cdf_Cadena

Data Field Class: Cdf_CampoFechaLlave

Data Type: Class Default
Justify: Class Default

List in Tool Palette? Yes

Description: Sirve para editar algún componente de la llave, por lo tanto, ocultarlo al mostrar un combo y viceversa.

Derived From

Class: Cdf_Mostrar
Class: Cdf_Ocultar
Class: Cdf_NoEditar
Class: Cdf_Limpiar
Class: CdfFecha

Message Actions

On AM_MostrarCombo
Call: SalHideWindow(BndItem)
On AM_OcultarCombo
Call: SalShowWindow(BndItem)

Data Field Class: Cdf_CampoFecha

Data Type: Class Default
Justify: Class Default

List in Tool Palette? Yes

Description: Para editar campos fecha normales.

Derived From

Class: CdfFecha
Class: Cdf_Limpiar
Class: Cdf_NoEditar
Class: Cdf_Editar

Data Field Class: Cdf_CampoNúmeroFormato

Data Type: Class Default
Justify: Class Default

List in Tool Palette? Yes

Description: Para editar campos numéricos normales con formato de miles.

Derived From

Class: Cdf_Limpiar
Class: Cdf_NoEditar
Class: Cdf_Editar
Class: CdfNúmero
Class: CdfNúmeroFormato

Data Field Class: Cdf_CampoNúmero

Data Type: Class Default
Justify: Class Default

List in Tool Palette? Yes

Description: Para editar campos numéricos normales.

Derived From

Class: Cdf_Limpiar
Class: Cdf_NoEditar
Class: Cdf_Editar
Class: CdfNúmero

Data Field Class: Cdf_CampoCadena

Data Type: Class Default
Justify: Class Default

List in Tool Palette? Yes

Description: Para editar campos normales.

Derived From

Class: Cdf_Limpiar
Class: Cdf_NoEditar
Class: Cdf_Editar
Class: CdfCadena

Radio Button Class: Crb_EditarNoEdLimpiar
Font Name: Arial
Font Size: 8
Font Enhancement: Bold
List in Tool Palette? Yes
Description: Radio Botón con atributos de letra y tamaño.
Responde a los mensajes AM_Editar, AM_NoEditar, AM_Limpiar.
Message Actions
On **AM_Limpiar**
Set MyValue = FALSE
On **AM_Editar**
Call SalEnableWindow(hWndItem)
On **AM_NoEditar**
Call SalDisableWindow(hWndItem)

Check Box Class: Ccb_EditarNoEdLimpiar
Font Name: Arial
Font Size: 8
Font Enhancement: Bold
List in Tool Palette? Yes
Description: Check Box con atributos de letra y tamaño.
Responde a los mensajes AM_Editar, AM_NoEditar, AM_Limpiar.
Message Actions
On **AM_Limpiar**
Set MyValue = FALSE
On **AM_Editar**
Call SalEnableWindow(hWndItem)
On **AM_NoEditar**
Call SalDisableWindow(hWndItem)

List Box Class: Clb_EditarNoEdLimpiar
Font Name: Arial
Font Size: 8
List in Tool Palette? Yes
Description: List Box con atributos de letra y tamaño.
Responde a los mensajes AM_Editar, AM_NoEditar, AM_Limpiar.
Message Actions
On **AM_Limpiar**
Call SalListClear(hWndItem)
On **AM_Editar**
Call SalEnableWindow(hWndItem)
On **AM_NoEditar**
Call SalDisableWindow(hWndItem)

Combo Box Class: Ccmb_EditarNoEdLimpiar
Font Name: Arial
Font Size: 8
Text Color: Black
Description: Combo Box con atributos de letra y tamaño.
Responde a los mensajes AM_Editar, AM_NoEditar, AM_Limpiar.
Message Actions
On **AM_Limpiar**
Call SalClearField(hWndItem)
Call SalListClear(hWndItem)
On **AM_Editar**
Call SalEnableWindow(hWndItem)
On **AM_NoEditar**
Call SalDisableWindow(hWndItem)

Combo Box Class: Ccmb_TipoLlave

Font Name: Arial

Font Size: 8

Text Color: Black

List in Tool Palette? Yes

Description: Combo Box tipo llave.

Responde al mensaje AM_MostrarCombo, AM_OcultarCombo,
AM_EditarLlaveCombo, AM_NoEditar, AM_Limpiar, SAM_Create

Message Actions

On **AM_MostrarCombo**

Call SalShowWindow(hWndItem)

On **AM_OcultarCombo**

Call SalHideWindow(hWndItem)

On **AM_NoEditar**

Call SalDisableWindow(hWndItem)

On **AM_EditarLlaveCombo**

Call SalEnableWindow(hWndItem)

On **AM_Limpiar**

Call SalListClear(hWndItem)

On **SAM_Create**

Call SalHideWindow(hWndItem)

Background Text Class: Ctexto

Font Name: Arial

Font Size: 8

Font Enhancement: Bold

Background Color: Gray

List in Tool Palette? Yes

Description: Background estándar para las etiquetas

Background Text Class: CtextoRojo

Font Name: Arial

Font Size: 8

Text Color: Red

List in Tool Palette? Yes

Description: Etiquetas en color rojo.

Combo Box Class: Ccmb_Campo

Font Name: Arial

Font Size: 8

List in Tool Palette? Yes

Description: Combo Box normal para edición.

Derived From

Class: Ccmb_EditarNoEdLimpiar

Group Box Class: Cgbox

Font Name: Arial

Font Size: 8

Font Enhancement: Bold

Text Color: Dark Blue

List in Tool Palette? Yes

Description: Group Box con atributos de letra y tamaño.

Multilino Field Class: Cmlf_CampoMulti
Font Name: Arial
Font Size: 8
List in Tool Palette? Yes
Description: Campo Multilinea con atributos de letra y tamaño.
Responde a los mensajes AM_Editar, AM_NoEditar, AM_Limpiar.
Message Actions
On **AM_Limpiar**
Call SalClearField(hWndItem)
On **AM_Editar**
Call SalEnableWindow(hWndItem)
On **AM_NoEditar**
Call SalDisableWindow(hWndItem)

Pushbutton Class: Cpb_Stdr
Font Name: Arial
Font Size: 8
Font Enhancement: Bold
List in Tool Palette? Yes
Description: Botón con todos los comportamientos necesarios:
Tipo de letra, tamaño, ...
Es para formar la ventana estándar de captura, de esta clase
serán las herramientas.
Message Actions
On **AM_ImprimirPantalla**
Call SalHideWindow(hWndItem)
On **AM_NoImprimirPantalla**
Call SalShowWindow(hWndItem)

Radio Button Class: Crb_Stdr
Font Name: Arial
Font Size: 8
Font Enhancement: Bold
Background Color: Gray
List in Tool Palette? Yes
Description: Radio Botón estándar con tipo de letra y tamaño.

Check Box Class: Ccb_Stdr
Font Name: Arial
Font Size: 8
Font Enhancement: Bold
Background Color: Gray
List in Tool Palette? Yes

Combo Box Class: Ccmb_Stdr
Font Name: Arial
Font Size: 8
List in Tool Palette? Yes
Description: Combo Box con tipo de letra y tamaño.

Data Field Class: Cdf_Stdr
Data Type: Class Default
Format: Uppercase
Font Name: Arial
Font Size: 8
List in Tool Palette? Yes
Description: Data Field con tamaño de letra y tipo.

Todos aquellos objetos con la opción de **List in Tool Palette?** en **Yes**, indican que son los objetos con un comportamiento específico listo para utilizarse en la aplicación. Los demás objetos sólo son de apoyo para crear a los anteriores y por tener su conducta parcialmente definida es que no son aptos para utilizarse directamente en la aplicación.

Todos los anteriores elementos definidos permiten ensamblar las ventanas de captura de cualquiera de los documentos del Sistema de Control de Adquisiciones. Algunas de ellas se muestran en la Figura 4-7.

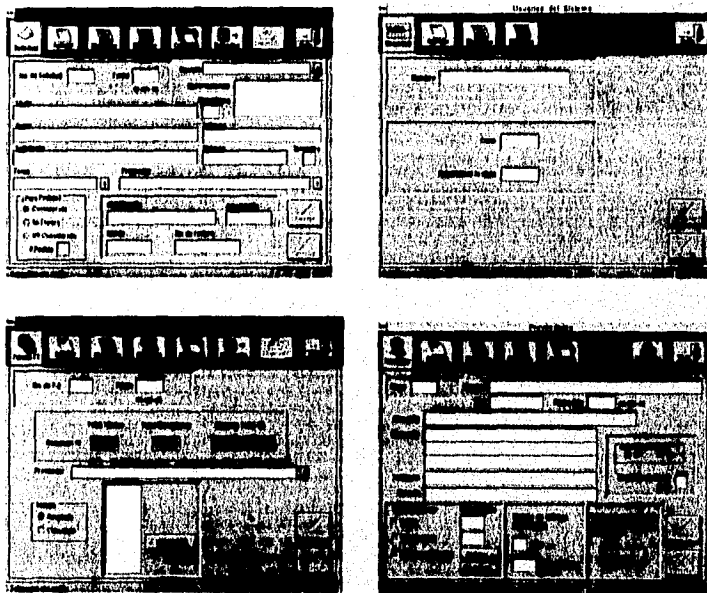


Figura 4-7: Ejemplo de ventanas para la captura de documentos

Como se puede observar, las cuatro ventanas se basan en la misma plantilla y pueden variar ligeramente para ajustarse a las necesidades específicas de cada documento. De un documento a otro la plantilla puede mantenerse idéntica en su apariencia pero con la posibilidad de variar su funcionamiento o comportamiento.

El código comienza con la definición de los mensajes en la sección de Constantes y las variables globales que se necesiten. En seguida se definen las clases y por último se obtienen las instancias de esas clases.

En las instancias es donde se coloca el código que amplía o modifica el comportamiento predefinido en la clase.

En seguida se lista el código de las tres funciones que se definieron después del pseudocódigo de las herramientas. Se listarán dentro del esquema de la ventana estándar de captura que será la clase CForma, la cual a su vez es derivada de la clase estándar de SQL Windows Form Window.

Siempre que se crea una clase a partir de un objeto estándar de SQL Windows en el código se antepone el nombre del objeto seguido de la palabra class, en seguida se coloca el nombre que se selecciona para referirse a la clase de ahí en adelante.

En realidad es una clase ventana que contiene clases de botones que serán las herramientas y otros elementos estéticos. Es en la sección de funciones donde se implementarán las funciones.

También se crearán funciones que necesitarán ser anuladas o redefinidas en las instancias. Por ejemplo, la función Selección() debe ser codificada en la instancia correspondiente porque es hasta entonces que se conoce el nombre de la ventana de selección y será una diferente por instancia.

Es el mismo caso para la función Agregar(), la cual debe ser única para cada documento, ya que debe insertar un documento en una tabla específica de la base de datos.

En seguida se listará el código que implementará la clase CForma, la cual será la plantilla para crear las ventanas de captura de los diferentes documentos.

Form Window Class: CForma
List in Tool Palette? Yes
Description: Forma Base para la interfaz de altas, bajas y cambios.
Contiene 9 botones con su conducta definida.

Contents

Cpb_Stdr: pbAgregar

Message Actions

On SAM_Click

On AM_Habilitar

Call SalEnableWindow(hWndItem)

On AM_Deshabilitar

Call SalDisableWindow(hWndItem)

Cpb_Stdr: pbModificar

Message Actions

On SAM_Click

On AM_Habilitar

Call SalEnableWindow(hWndItem)

On AM_Deshabilitar

Call SalDisableWindow(hWndItem)

Cpb_Stdr: pbEliminar

Message Actions

On SAM_Click

On AM_Habilitar

Call SalEnableWindow(hWndItem)

On AM_Deshabilitar

Call SalDisableWindow(hWndItem)

Cpb_Stdr: pbConsulta

Message Actions

On SAM_Click

On AM_Habilitar

Call SalEnableWindow(hWndItem)

On AM_Deshabilitar

Call SalDisableWindow(hWndItem)

Cpb_Stdr: pbSelección

Message Actions

On SAM_Click

On AM_Habilitar

Call SalEnableWindow(hWndItem)

On AM_Deshabilitar

Call SalDisableWindow(hWndItem)

Cpb_Stdr: pbImprimir

Message Actions

On SAM_Click

On AM_HabilitarEnEdicion

! Habilitar cuando ya hay datos en la forma.

Call SalEnableWindow(hWndItem)

On AM_DeshabilitarEnEdicion

! Deshabilitar cuando ya no hay datos en la forma.

Call SalDisableWindow(hWndItem)

```
Cpb_Std: pbSalir
Message Actions
On SAM_Click

On AM_Habilitar
Call SalEnableWindow( hWndItem )
On AM_Deshabilitar
Call SalDisableWindow( hWndItem )

Cpb_Std: pbAceptar
Message Actions
On SAM_Click

On AM_HabilitarEnEdicion
Call SalEnableWindow( hWndItem )
On AM_DeshabilitarEnEdicion
Call SalDisableWindow( hWndItem )

Cpb_Std: pbCancelar
Message Actions
On SAM_Click

On AM_EditarLlave
! ... Se activa junto con la llave, para poder cancelar
Call SalEnableWindow( hWndItem )
On AM_EditarLlaveCombo
Call SalEnableWindow( hWndItem )
On AM_DeshabilitarEnEdicion
! ... Deshabilitar al terminar la edición
Call SalDisableWindow( hWndItem )
Call SalStatusSetText( hWndForm, 'Selecciona una acción' )

Frame
Frame
Picture: picEtiqueta ! Donde se colocará el logotipo del documento en turno

Class Variables
String: sTemp ! Temporal cadena.
Number: nTemp ! Temporal numérica.
Date/Time: dTemp ! Temporal fecha.

Instance Variables
String: sModo ! Indicará el modo: Agregar, Modificar, Eliminar o Consulta
String: sTemaTemp ! Para averiguar si ha cambiado de tema y deba
! cambiar de proveedores también (con un reQuery).
Boolean: bFaltaData ! Checará si falta un dato importante.
```


Functions

Function: PantallaEdicion

Description: Habilitar el botón de cancelar la edición, deshabilitar los botones de arriba y habilitar los campos de la llave.

Actions

```
! Deshabilitar pbAgregar, ...
Call SalSendMsgToChildren( hWndForm, AM_Deshabilitar, 0, 0 )

! Habilitar campos de la llave, según sea alta
If sModo = 'Agregar'
    Call SalSendMsgToChildren( hWndForm, AM_EditarLlave, 0, 0 )
Else ! 0 ...
    Call SalSendMsgToChildren( hWndForm, AM_MostrarCombo, 0, 0 )
    Call SalSendMsgToChildren( hWndForm, AM_EditarLlaveCombo, 0, 0 )
```

Function: PantallaNormal

Description: Deshabilitar los dos botones de edición, habilitar los botones de arriba y deshabilitar los campos de edición.

Actions

```
! Habilitar pbAgregar, ...
Call SalSendMsgToChildren( hWndForm, AM_Habilitar, 0, 0 )

! Deshabilitar campos de edición
Call SalSendMsgToChildren( hWndForm, AM_NoEditar, 0, 0 )

! Limpiar campos de edición
Call SalSendMsgToChildren( hWndForm, AM_Limpiar, 0, 0 )

! Ocultar Combos de la llave
Call SalSendMsgToChildren( hWndForm, AM_OcultarCombo, 0, 0 )

! Deshabilitar pbAceptar, pbCancelar
Call SalSendMsgToChildren( hWndForm, AM_DeshabilitarEdicion, 0, 0 )
```

Function: EditarClave

Description: Coloca el foco en el campo sClave de la forma.
Rutina sólo para las alias.

Actions

```
Call SalSetFocus( hWndForm.df_sClave )
```

! Las siguientes son las funciones que necesitan ser anuladas:

Function: Seleccion

Description: Debe ser anulada en la instancia.
Sirve para abrir el diálogo donde se pueda buscar un registro en particular dando ciertas condiciones.

Actions

```
Call ..Seleccion()
```

Function: EscribirTipos

Description: Preparar los rb, cb y los extras que se necesitan transformar antes de poder ser almacenados en la tabla correspondiente.

Actions

```
Call ..EscribirTipos()
```

Function: Agregar

Description: Insertar (INSERT) el registro actual en pantalla.

Actions

Call ..Agregar()

Function: Modificar

Description: Modificar (UPDATE) el registro actual en pantalla.

Actions

Call ..Modificar()

Function: Eliminar

Description: Eliminación (DELETE) del registro actual en pantalla.

Actions

Call ..Eliminar()

Function: Imprimir

Description: Habilita un menú de impresión de algún reporte en especial o simplemente imprime la pantalla.

Actions

Call ..Imprimir()

Message Actions

! Acciones para CForma

On SAM_Create

! Esta es la creación de la forma de captura, por lo tanto, preparar
! la interfaz para el usuario:

Call SalStatusSetText(hWndForm, 'Selecciona una acción')

! Deshabilitar los botones no necesarios al inicio.

! Primero: no hay nada que Aceptar

Call SalDisableWindow(pbAceptar)

! Segundo: no hay nada que Cancelar

Call SalDisableWindow(pbCancelar)

! Tercero: no hay nada que Imprimir

Call SalDisableWindow(pbImprimir)

! Finalmente, deshabilitar campos de edición mandándoles el
! mensaje que los deshabilita

Call SalSendMessageToChildren(hWndForm, WM_NoEditar, 0, 0)

Para abreviar, el código de las herramientas ya no es desplegado, sólo se agregan los nuevos mensajes y sus respectivas acciones, pues también deberán responder a las órdenes de habilitarse y deshabilitarse según la operación ejecutada.

La sección Class Variables sirve para definir variables que podrán utilizar todas las instancias de esta clase, es decir son globales a todas las instancias. Es una manera de compartir información en una clase y sus objetos porque cada objeto compartirá la misma copia de una variable de clase.

La sección Instance Variables crea variables únicas para cada instancia de la clase, es decir, son locales a cada instancia. Las variables se replican en cada objeto, por lo que cada objeto recibe una copia privada de tal variable, de tal manera que no entran en conflicto unas con otras.

La sección Message Actions es el área de la forma para ejecutar código en respuesta a algunos mensajes de la aplicación. En este caso se aprovecha el mensaje SAM_Create que es recibido por la forma al momento de crearse pero un momento antes de visualizarse. La idea es dejar la interfaz lista, tal y como el usuario debe verla normalmente cuando no hay una acción seleccionada.

Este código junto con el de las herramientas es la base para comenzar la implementación de una ventana de captura de un documento.

El siguiente código es el que crea la instancia frmSolicitud que se deriva de la clase CForma. Con esto se crea la ventana para capturar el documento Solicitud.

CForma: frmSolicitud

Contants

(untitled) from CForma: phCancelar

Cdf_CampoCadenaLlave: df_sClave

Message Actions

! Cuando se llega a este registro:

On SAM_SetFocus

Call SalWaitCursor(VERDAD)

! Averiguar el ultimo No. consecutivo

set CForma.sTemp = '0'

Call SqlPrepareAndExecute(hSqlConnection,

'SELECT No_Solicitud

INFO :CForma.sTemp

FROM SOLICITUD

WHERE @YEAR(Fecha) = :nAnolectivo

ORDER BY No_Solicitud DESC')

Call SqlFetchNext(hSqlConnection, CForma.nTemp)

! Incrementar en 1.

Set CForma.nTemp = SalStrToNumber(CForma.sTemp 1 + 1

Set CForma.sTemp = SalNumberToStrX(CForma.nTemp, 0)

! Completar con ceros a la izquierda.

Set df_sClave = gCerosLlave(CForma.sTemp, 5)

! Habilitamos los campos para edición.

Call Editar()

! La fecha de este registro será la de hoy.

set df_dFecha = dFechaHoy

Call SalDisableWindow(df_dFecha)

Call SalWaitCursor(FALSO)

Ccmb_TipoLlave: cmb_sClave

Message Actions

On AM_EditarLlaveCombo

Call SalWaitCursor(VERDAD)

set frmSolicitud.sTemp = 'SELECT DISTINCT No_Solicitud

FROM Solicitud

WHERE @YEAR(Fecha) = :nAnolectivo'

Call SqlPrepareAndExecute(hSqlConnection, frmSolicitud.sTemp)

! Si no hay datos, avisar al usuario.

if NOT SqlFetchNext(hSqlConnection, nIndice)

Call SalMessageBox('No existen Solicitudes del año ' ||
SalNumberToStrX(nAnolectivo, 0) || ',
'Aviso', MB_OK|MB_IconExclamation)

! Dejar todo como estaba

Call frmSolicitud.PantallaNormal()

Else

! Poblal el combo con los Nos. de solicitudes.

Call SqlPrepare(hSqlConnection, frmSolicitud.sTemp)

Call SalHistPopulate(hWnditem, hSqlConnection, frmSolicitud.sTemp)

Call SalEnableWindow(hWnditem)

Call SalSetFocus(hWnditem)

Call SalWaitCursor(FALSE)

```

On SAM_Click
If Not SalIsNull(EModifia)
Call SalWaitCursor(VERDAD)
! Cargar los datos.
Call SqlPrepareAndExecute( hSqlConnection,
'SELECT
No_Solicitud, Fecha, Clave_Carrera,
Titulo, Ejemplares, Autor,
Editorial, Profesor, Materia,
Semestre, observaciones, Clave_Prov,
Clave_Tema, No_Factura, Clasificacion,
Adquisicion, Matriz, Para_Pedido,
No_Pedido
FROM Solicitud
INTO
:df_sclave, :df_dfecha, :sCarrera,
:df_sTitulo, :df_nEjemplares, :df_sAutor,
:df_sEditorial, :df_sSolicitante, :df_sMateria,
:df_sSemestre, :df_sObservaciones, :df_sProveedor,
:sTema, :df_sFactura, :df_sClasificacion,
:df_sAdquisicion, :df_sMatriz, :df_sPara_Pedido,
:df_sPedido
WHERE No_Solicitud = :cmb_sclave
AND @YEAR(Fecha) = :mAnolectivo' )

Call SqlFetchNext( hSqlConnection, nIndice )

! Si se trata de Modificar, entonces habilitar para edición.
If frmSolicitud.stModo = 'Modificar'
Call Editor( )
Else
! Habilitar pbAceptar, pbCancelar
Call SalSendMessageToChildren( hWndForm, AM_HabilitarEnEdicion, 0, 0)

Call SalDisableWindow( df_dFecha )
! ... checar tipo de opciones.
Call LeerTipos( )
Call SalWaitCursor( FALSO )
Else
Call SalSetFocus( hWndItem )

cdf_CampoFecha: df_dFecha

Ccmb_Campo: cmb_sCarrera
Message Actions
On SAM_SetFocus
Set CForma.nTemp = 1

cdf_CampoCadena: df_sTitulo
Message Actions
! Este campo no debe estar vacío
On AM_ChecarContenido
If SalIsNull( hWndItem )
Set frmSolicitud.bFaltaDato = VERDAD

cdf_CampoNumero: df_nEjemplares
Message Actions
On AM_ChecarContenido ! Este campo no debe estar vacío
If SalIsNull( hWndItem )
Set frmSolicitud.bFaltaDato = VERDAD

```

```

Cdf_CampoMulti: ml_sObservaciones
Message Actions
On SAM_FieldEdit
Set ml_sObservaciones = SalStdUpperXml_sObservaciones

Cdf_CampoCadena: df_sAutor

Cdf_CampoCadena: df_sEditorial
Message Actions
! Este campo no debe estar vacío
On AM_ChecarContenido
If SalIsNull( hWndItem )
Set frmSolicitud.bFaltaData = VERDAD
Cdf_CampoCadena: df_sSolicitante

Cdf_CampoCadena: df_sMateria

Cdf_CampoCadena: df_sSemestre

Ccmb_Campo: cmb_sTema
Message Actions
On AM_ChecarContenido ! Este campo no debe estar vacío
If SalIsNull( hWndItem )
Set frmSolicitud.bFaltaData = VERDAD

Ccmb_Campo: cmb_sProveedor
Message Actions
! Poblado con los Provs. de acuerdo al tema.
On SAM_SetFocus
! Si aún no hay selección, poblar el combo.
If SalIsNull( hWndItem )
If NOT SalIsNull( cmb_sTema )
Call SalWaitCursor( VERDAD )
Set CForma.nTemp = SalListQuerySelection( cmb_sTema )

Set frmSolicitud.sTemaTemp =
SalListQueryTextX( cmb_sTema, CForma.nTemp )

! Buscar la clave del tema (select 3).
! Buscar las claves de los proveedores que
! manejan ese tema (select 2).
! Buscar los nombres de esas claves de proveedores (select 1).

Call SalListPopulate( cmb_sProveedor,
hSqlConexión,
'SELECT nombre
FROM proveedores
WHERE clave_prov IN
(SELECT clave_prov
FROM subprovtemas
WHERE
clave_tema =
(SELECT clave_tema
FROM temas
WHERE
descripcion = :frmSolicitud.sTemaTemp))' )

Call SalWaitCursor( FALSO )

```

```

Else
! Habrá que actualizar la lista de proveedores por el tema?
! Es decir, ¿se cambió de tema?
If frmSolicitud.sTemaTemp != cmb.sTema
Call SalWaitCursor( VERDAD )
Set frmSolicitud.sTemaTemp = cmb.sTema
Call SalListClear( hWndItem )

! Buscar la clave del tema (select 3).
! Buscar las claves de los proveedores que
! manejan ese tema ( select 2).
! Buscar los nombres de esas claves de proveedores (select 1).

Call SalListPopulate(hWndItem,
hSqlConexión,
'SELECT nombre
FROM proveedores
WHERE clave_prov IN
(SELECT clave_prov
FROM subprovtemas
WHERE clave_tema =
(SELECT Clave_tema
FROM temas
WHERE descripcion = :frmSolicitud.sTemaTemp)')

Call SalWaitCursor( FALSO )

! Este campo no debe estar vacío.
On AM_ChecarContenido
If SalIsNull( hWndItem )
Set frmSolicitud.bFaltaData = VERDAD

Cgbox: ¿Para Pedido?

Crb_EditarNoEdLimpiar: rb_sPara_Pedido1
Message Actions
On AM_Limpiar
Set MyValue = VERDAD
On SAM_Click
If NOT SalIsNull( df_sPedido )
Call SalMessageBox(
'Esta solicitud ya se encuentra en un pedido. No hay cambios.',
'Aviso', MB_IconExclamation )
Set rb_sPara_Pedido3 = VERDAD

Crb_EditarNoEdLimpiar: rb_sPara_Pedido2
Message Actions
On SAM_Click
If NOT SalIsNull( df_sPedido )
Call SalMessageBox(
'Esta solicitud ya se encuentra en un pedido. No hay cambios.',
'Aviso', MB_IconExclamation )
Set rb_sPara_Pedido3 = VERDAD

Crb_EditarNoEdLimpiar: rb_sPara_Pedido3

Data Field: df_sPedido
Message Actions
On AM_Limpiar
Call SalClearField( hWndItem )

```

```

Cdf_CampoCadena: df_sClasificacion

Cdf_CampoCadena: df_sAdquisicion

Cdf_CampoCadena: df_sMatriz

Cdf_CampoCadena: df_sFactura

(untitled) from CForma: pbAceptar
Message Actions
On SAM_Click
If frmSolicitud.sModo = 'Eliminar'
! Si está incluida en un pedido, no se puede borrar.
If NOT SalIsNull( df_sPedido )
Call SalMessageBox( 'Esta Solicitud está incluida en un PEDIDO.'
|| ' Liberarla desde el pedido y después reintentar.'
|| ' eliminarla aquí.', 'Aviso', MB_IconStop )
Return VERDAD

! Si está libre, entonces proceder.
If frmSolicitud.sModo = 'Agregar'
! Si este material se encuentra en otra solicitud,
no se puede agregar de nuevo.

If NOT SalIsNull( df_sTitulo ) AND NOT SalIsNull( df_sEditorial )
Call SqlPrepareAndExecute( hSqlConnection,
'SELECT No_Solicitud INTO :frmSolicitud.sTemp
FROM Solicitud
WHERE Titulo = :df_sTitulo
AND Autor = :df_sAutor
AND Editorial = :df_sEditorial' )

If SqlFetchNext( hSqlConnection, frmSolicitud.nTemp )
Call SalMessageBox( 'Este material existe en Solicitud No: '
|| frmSolicitud.sTemp || '.', 'Aviso', MB_IconStop )
Return VERDAD

! Si no existe este material como solicitud anterior,
! entonces proceder con su alta.
Call SalSendClassMessage( SAM_Click, 0, 0 )

(untitled) from CForma: pbSalir

(untitled) from CForma: pbImprimir

Frame from CForma
Ctexto: No. de Solicitud
Ctexto: Titulo
Ctexto: Fecha
CtextoRojo: dd-mm-aa

Frame
Ctexto: Carrera
Ctexto: Observaciones
Ctexto: Autor
Ctexto: Solicitante
Ctexto: Clasificación

```


Ctexto: Adquisición
Ctexto: Matriz
Ctexto: No. de Factura
Ctexto: Ejemplares
Ctexto: Editorial
Ctexto: Materia
Ctexto: Semestre
Ctexto: Tuma
Ctexto: Proveedor
Frame
Ctexto: # Perdido
(untitled) from CForma: pbSelección
(untitled) from CForma: pbConsulta
(untitled) from CForma: pbEliminar
(untitled) from CForma: pbModificar
(untitled) from CForma: pbAgregar
Frame from CForma
(untitled) from CForma: picEtiqueta

! Funciones globales para la forma:

Functions

Function: Editar

Description: Código para pbAgregar y para pbModificar.
Habilitación de los campos.

Actions

! Deshabilitar campos de la llave
Call SalSendMessageToChildren(hWndForm, AM_NoEditar, 0, 0)
! Habilitar campos de edición restantes
Call SalSendMessageToChildren(hWndForm, AM_Editar, 0, 0)

! Habilitar pbAceptar, pbCancelar
Call SalSendMessageToChildren(hWndForm, AM_HabilitarEnEdicion, 0, 0)
! Poblar los combos de las Carreras y de los Temas.
Call SalListPopulate(cmb_sCarrera, hSqlConexión,
'SELECT Descripción FROM CARRERAS')
Call SalListPopulate(cmb_sTema, hSqlConexión,
'SELECT Descripción FROM TEMAS')
Call SalSetFocus(cmb_sCarrera)

Function: EscribirTipos

Description: ! Preparar los rb, cb y los extras que se necesiten
 Transformar antes de poder ser almacenados en la tabla
 correspondiente.

Actions

```
! Transforma el radio botón en la letra inicial de su opción
If rb_sPara_Pedido1
    Set sPara_Pedido = 'C'
Else If rb_sPara_Pedido2
    Set sPara_Pedido = 'E'
Else ! If rb_sPara_Pedido3
    Set sPara_Pedido = 'H'
```

```
! Tomar la clave del Tema.
Call SqlPrepareAndExecute( hSqlConnection,
    'SELECT Clave_tema
    INTO :sTema
    FROM Temas
    WHERE descripcion = :cmb_sTema' )
Call SqlFetchNext( hSqlConnection, CForma.nTemp )
```

```
! Tomar la clave de la carrera.
Call SqlPrepareAndExecute( hSqlConnection,
    'SELECT Clave_carrera
    INTO :sCarrera
    FROM Carreras
    WHERE descripcion = :cmb_sCarrera' )
Call SqlFetchNext( hSqlConnection, CForma.nTemp )
```

```
! Tomar la clave del Proveedor
Call SqlPrepareAndExecute( hSqlConnection,
    'SELECT Clave_Prov
    INTO :sProveedor
    FROM Proveedores
    WHERE nombre = :cmb_sProveedor' )
Call SqlFetchNext( hSqlConnection, CForma.nTemp )
```

Function: InsertTipos

Description: Preparar los valores almacenados en la tabla para los
 rb, cb y los extras que se necesiten para la pantalla.

Actions

```
! Transforma a radio botón la letra inicial de su opción de la tabla.
Set rb_sPara_Pedido1 = FALSE
Set rb_sPara_Pedido2 = FALSE
Set rb_sPara_Pedido3 = FALSE
```

```
If sPara_Pedido = 'C'
    Set rb_sPara_Pedido1 = VERDAD
Else If sPara_Pedido = 'E'
    Set rb_sPara_Pedido2 = VERDAD
Else ! If sPara_Pedido = 'H'
    Set rb_sPara_Pedido3 = VERDAD
```

```
! Tomar la descripción del tema de acuerdo a su clave.
Call SqlClearField( cmb_sTema )
Call SqlPrepareAndExecute( hSqlConnection,
    'SELECT descripcion
    INTO :cmb_sTema
    FROM temas
    WHERE Clave_tema = :sTema' )
```

```
Call SqlFetchNext( hSqlConexión, CForma.nTemp )

! Tomar la descripción de la carrera de acuerdo a la clave.
Call SqlClearField( cmb_sCarrera )
Call SqlPrepareAndExecute( hSqlConexión,
    'SELECT descripcion
      INTO :cmb_sCarrera
      FROM Carreras
      WHERE Clave_carrera = :sCarrera' )
Call SqlFetchNext( hSqlConexión, CForma.nTemp )

! Tomar el nombre del Proveedor de acuerdo a su clave.
! Poblar también el combo de proveedores.
IF NOT CallIsNull( cmb_sTema ) AND frmSolicitud.sModo = 'Modificar'
    Set frmSolicitud.sTemaTemp = cmb_sTema

! Buscar la clave del tema ( select 3).
! Buscar las claves de los proveedores que
! manejan ese tema ( select 2).
! Buscar los nombres de esas claves de proveedores (select 1).

Call SqlListPopulate( cmb_sProveedor, hSqlConexión,
    'SELECT nombre
      FROM proveedores
      WHERE clave_prov IN
      (SELECT clave_prov
        FROM subprovtemas
        WHERE clave_tema =
        (SELECT Clave_tema
          FROM temas
          WHERE
            descripcion = :frmSolicitud.sTemaTemp))' )

Call SqlPrepareAndExecute( hSqlConexión,
    'SELECT nombre
      INTO :frmSolicitud.sTemp
      FROM Proveedores
      WHERE Clave_Prov = :sProveedor' )
Call SqlFetchNext( hSqlConexión, CForma.nTemp )

Set cmb_sProveedor = frmSolicitud.sTemp
```

Function: Agregar

Description: INSERTA en la Tabla el registro en pantalla.

Actions

```

Call SqlPrepareAndExecute(hSqlConnection,
'INSERT INTO Solicitud
( No_Solicitud, Fecha, Clave_Carrera,
Tiulo, Ejemplares, Autor,
Editorial, Profesor, Materia,
Semestre, Observaciones, Clave_Prov,
Clave_Tema, No_Factura, Clasificacion,
Adquisicion, Matriz, Para_Pedido,
No_Pedido )
VALUES
(:df_sClave, :df_dFecha, :sCarrera,
:df_sTitulo, :df_nEjemplares, :df_sAutor,
:df_sEditorial, :df_sSolicitante, :df_sMateria,
:df_sSemestre, :m_sObservaciones, :sProveedor,
:sTema, :df_sFactura, :df_sClasificacion,
:df_sAdquisicion, :df_sMatriz, :sPara_Pedido,
:df_sPedido) ' )
Call SqlCommit( hSqlConnection )
    
```

Function: Modificar

Description: Modificar el registro actual en pantalla.

Actions

```

Call SqlPrepareAndExecute(hSqlConnection, 'UPDATE Solicitud SET
No_Solicitud = :df_sClave,
Fecha = :df_dFecha,
Clave_Carrera = :sCarrera,
Titulo = :df_sTitulo,
Ejemplares = :df_nEjemplares,
Autor = :df_sAutor,
Editorial = :df_sEditorial,
Profesor = :df_sSolicitante,
Materia = :df_sMateria,
Semestre = :df_sSemestre,
Observaciones = :m_sObservaciones,
Clave_Prov = :sProveedor,
Clave_Tema = :nTema,
No_Factura = :df_sFactura,
Clasificacion = :df_sClasificacion,
Adquisicion = :df_sAdquisicion,
Matriz = :df_sMatriz,
Para_Pedido = :sPara_Pedido,
No_Pedido = :df_sPedido

WHERE
No_Solicitud = :cmb_sClave
AND Fecha = :df_dFecha' )
Call SqlCommit( hSqlConnection )
    
```

Function: Eliminar

Description: Eliminación del registro actual en pantalla.

Actions

```
Call SqlPrepareAndExecute(hSqlConnection,
  'DELETE FROM
      Solicitud
  WHERE
      No_Solicitud = :cmf_sclave
      AND Fecha = :df_fecha')
Call SqlCommit(hSqlConnection)
```

Function: Seleccion

Description: Sirve para abrir una ventana donde se puede buscar un registro en particular dando ciertas condiciones. Llamar a la que pertenece a las solicitudes.

Actions

```
Call SqlCreateWindow(frmSOI_Select, hWindow)
```

Function: Imprimir

Description:

Actions

```
If IRYES = SqlMessageBox(
  '¿Realmente deseas imprimir esta Solicitud?',
  'Impresión', MB_IconQuestion|MB_YesNo|MB_DefButton2)
Then
  ' Esconder todos los botones para imprimir los datos solamente.
  Call SqlSendMessageToChildren(hWindow, AM_NoImprimirPantalla, 0, 0)
  ! Entonces imprimir.
  Call SqlPrintForm(hWindow)

  ! Despues hacerlos visibles.
  Call SqlSendMessageToChildren(hWindow, AM_NoImprimirPantalla, 0, 0)
```

Window Variables

```
String: sCartera ! La Cartera que irá a la tabla e se lee de la tabla.
String: sTema ! Idem para tema.
String: sProveedor ! Idem para proveedor.
String: sPara_Pedido ! Idem para pedido.
```

Message Actions

```
! Ninguna acción para esta instancia.
! Ejecutar las acciones de la clase base (CForm).
```

La Figura 4-8 muestra la ventana frmSolicitud creada según el código anterior.

Figura 4-8: El objeto de captura frmSolicitud

El código de la clase CForma será el mismo para cualquier documento. El comportamiento adicional o modificado se implementa en la instancia de CForma, de tal manera que este código es el que hace la diferencia junto con los objetos hijos que se requieran según el documento.

Al implementar el código se observa que ciertos objetos simplemente ejecutan el código heredado que se define en la clase base CForma. Esto se logra al no redefinir código alguno para el objeto.

En otros casos el código predefinido puede no cumplir con los objetivos deseados, por lo que ese código heredado se anula al reemplazar el nuevo. Tal es el caso de pbAceptar, que reemplaza el código para el mensaje SAM_Click, logrando con esto que el código en la clase base no se ejecute. No se hace lo mismo para sus mensajes AM_Habilitar y AM_Deshabilitar ya que se requiere el comportamiento definido en la clase base.

También fue necesario crear nuevas variables locales a la ventana en la sección Window Variables.

Conclusión

Se vive una época donde el objetivo es mejorar la productividad y seguir siendo competitivo.

Una herramienta software como lo es SQL Windows ha permitido que el Sistema de Control de Adquisiciones se haya concebido e implementado en breve tiempo, gracias a que tiene características que permiten un diseño visual y a sus secciones que indican qué y en dónde codificar.

Como se ha visto, la Programación Orientada a Objetos ha brindado una opción para mejorar el software de hoy en día al hacerla más poderosa y de mejor calidad.

La orientación a objetos logra la reutilización de software, lo cual minimiza el tiempo y esfuerzo en la programación, por lo que se es más productivo al no tener que programar de la nada a cada módulo ya que se parte de bases comunes.

El entorno de Windows ha creado nuevas formas de trabajar al mejorar muchos aspectos del ambiente DOS. Lo más importante es que hoy en día la mayoría de los usuarios pueden utilizar rápidamente una aplicación basada en Windows y se pueden obtener resultados casi inmediatos.

El Sistema de Control de Adquisiciones ha permitido un mejor seguimiento del ejercicio del presupuesto, ya que ahora se puede obtener en un instante la cantidad ejercida y la pendiente por ejercer. Los listados obtenidos en forma automática liberan de una gran cantidad de trabajo al Departamento de Adquisiciones, ya que antes se requerían horas o días enteros de un exhaustivo y concienzudo trabajo para, por ejemplo, saber la cantidad de títulos y ejemplares adquiridos y no adquiridos en

un cierto periodo. El seguimiento de un pedido también se ha facilitado en una gran manera al tener los detalles exactos de todos ellos.

La Programación Orientada a Objetos al facilitar la labor de desarrollo redundante en que el producto final sea de mejor calidad para el usuario, por lo que es una tendencia a seguir hoy en día por la mayoría de los desarrolladores.

Bibliografía

Microsoft Corporation

Guide to programming. Microsoft Windows Software Development KIT

1992, United States of America, 567 págs.

Microsoft Corporation

C++ tutorial. Microsoft C/C++

1991, United States of America, 205 págs.

Wirfs-Brock, Rebeca, Wilkerson, Wiener

Designing Object-Oriented Software

Prentice Hall, Inc., 1990, United States of America, 341 págs.

Eckel, Bruce

Aplique C++

McGraw-Hill, 1990, España, 521 págs.

Bryan Livingston

Secretos de Windows 3.1

LIMUSA, 1993, México, 1147 págs.

Rajesh Lalwani

Power Programming with SQL Windows

Gupta Technologies, Inc., 1994, United States of America, 328 págs.

Bruce Ring

SQL Windows Developer's Reference

Gupta Technologies, Inc., 1994, United States of America, 590 págs.

Bruce Ring, Rick Cumings, Stacia Sambar

Database Administrator's Guide

Gupta Technologies, Inc., 1994, United States of America, 362 págs.

PC/TIPS BYTE

Quina Monroy Baker, Mensual, México, EDICOBHISA, Mayo, 62 págs., 1992