

9

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE CIENCIAS

EMULACION DE CALCULADORA PARA
MICROSOFT ACCESS.

TESIS CON
FALLA DE ORIGEN

T E S I S

QUE PARA OBTENER EL TITULO DE:
A C T U A R I O
P R E S E N T A :
RAFAEL BAHENA ARANDA

DIRECTOR DE TESIS: **DR. MARGARITA CHAVEZ CANO**



MEXICO, D. F.

2002

FACULTAD DE CIENCIAS
SECCION ESCOLAR





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.



ESTADO NACIONAL
UNIVERSIDAD
MATEMÁTICA

EN C. ELENA DE OTEYZA DE OTEYZA
a de la División de Estudios Profesionales de la
ultad de Ciencias
sente

unicamos a usted que hemos revisado el trabajo escrito:

"Emulación de Calculadora para Microsoft Access"

lizado por Rafael Bahena Aranda

i número de cuenta 07202508-2 , quién cubrió los créditos de la carrera de Actuaría

ho trabajo cuenta con nuestro voto aprobatorio.

Atentamente

ctor de Tesis	
pictario	Mat. Margarita Elvira Chávez Cano
pictario	M. en C. Beatriz Eugenia Rodríguez Fernández
pictario	M. en C. Inocencio Rafael Madrid Ríos
ilente	M. en A.P. Ma. del Pilar Alonso Reyes
ilente	Act. José Antonio Climent Hernández

M. E. Oteyza de Oteyza
Beatriz Rodríguez Fernández

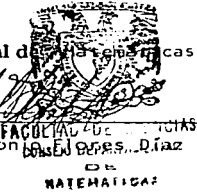
[Signature]
[Signature]

da gaurá

Consejo Departamental de Matemáticas

[Signature]

M. en C. José Antonio Climent Hernández



DE MATEMÁTICAS

A la gente que en una ú otra forma ha contribuido a que yo sea una persona mejor en todos los sentidos. Muchos nombres escapan a mi memoria, pero su obra y su imagen están presentes en mi agradecimiento y en mi corazón:

A mis profesores en todas mis diferentes etapas de estudio quiero mostrar mi afecto y mis respetos por su profesionalismo y su dedicación a la docencia. En especial les agradezco a mis profesores de la Facultad de Ciencias sus enseñanzas.

A Margarita Chávez:

A mi directora de tesis, le estoy profundamente agradecido por su atinada dirección, la enorme ayuda proporcionada y sobretodo, la confianza que hizo renacer en mí de lograr la culminación de un sueño largamente acariciado: el lograr el título de la carrera.

A Margarita Chávez, Beatriz Fernández, Rafael Madrid, José Antonio Climent y Ma. Del Pilar Alonso:

A mis sinodales, les agradezco la inspiración que causaron en mí de seguir su ejemplo, así como su presteza, su tiempo y su dedicación para realizar la revisión de este trabajo.

Al Profesor Víctor Mantilla por su ayuda en un intento anterior de realizar una tesis.

A Gabriela Moreno, Eduardo García (qepd), Jaime González, Víctor Bernal y Max Alvarez.

A mis jefes y compañeros de trabajo les agradezco su liderazgo y su confianza en mí de apoyarlos en la medida que lo necesitaran. A lo largo de mi experiencia laboral han sido una parte esencial para mi desarrollo profesional.

Angel Salazar, Marco Vasconcelos y Héctor Mendoza.

Con especial cariño, a todos mis amigos del departamento de Procesamiento Electrónico de Datos (EDP) de A. C. Nielsen, donde inicié mi carrera en computación.

Una gran nota de agradecimiento a José Luis Lara, amigo y compañero de mi trabajo actual en Modus Media International, por toda la ayuda otorgada en conceptos de Access, Visual Basic y Bases de Datos, el fundamento de programación de esta simulación.

A mis amigos, seres que atesoro y les agradezco el haber entrado y alegrado mi vida con su presencia, con su ayuda, con sus consejos y con incontables momentos felices.

Con especial cariño, afecto y respeto para mis amigos que siempre están cerca y me apoyan:

Malena Álvarez, su esposo Antonio, Nancy y Toñito (y Tory)

Nelly Lozano y su esposo Alejandro

Marivel Reza y familia

Mónica y Collin Stevens

Mercedes Lara, su esposo Eduardo y familia

Marcela Espejel y familia

Rosalinda Munguía y familia

A mis queridísimos amigos: Rocío Cortés y sus hijos Talina y Roberto; Esperanza Cortés y Antonio Cortés.

A mis amigos y compañeros de estudios, por la entrañable amistad que logramos:

Gabriel López, su esposa Patricia y su familia

José González, su esposa Lourdes y su familia

A mi queridísimo amigo Marcos César Tornez, su esposa Margarita y sus hijos; mis amigos del conjunto: Paco, Víctor, Pepe, Beto y los del nuevo conjunto: Roberto y Enrique.

A mis queridos amigos de Simposium del Dr. Rutilio Romero:

Chelita Cruz y su hermana Rosa.

Especial mención a mi queridísima amiga Malena Álvarez por la revisión de ortografía, comentarios, y el enorme apoyo y ayuda que siempre me brinda.

A los seres que más amo y admiro en mi vida: mis padres, les agradezco el haberme prodigado amor y bendiciones.

Ramón Bahena Rodríguez y Ana María Aranda Viveros

A mis hermanos y sus familias, les agradezco su enorme amor y el gran aliento que siempre me brindan:

Mauricio y Betty con Ellis y Gigi

Marcos (qepd) y Mariana con
Marina, Miguel Angel y familia
Marcos, Hortensia y familia
Anita, Arturo y familia
Mauricio, Adriana y familia
Gustavo

Salomón y Mary con Miriam é Israel

Noé y Jossye con Noé, Anhelen y Zoé

Javier y Luz con
Javier, Dulce y familia
Mauricio y Lourdes
Irene, Arturo y familia
Rafael
Roberto

María, mi queridísima hermana y Manuel con Itzi, Itzén é Itzel

Ramón, mi hermano del alma.

A Tere, mi esposa, mi compañera en esta vida y las que estén por venir. Por la enorme confianza depositada en mí de lograr una vida mejor.

A mi familia política, por la confianza, apoyo y comprensión que me han brindado:

Sra. Graciela Muñoz y Dón Jesús Salazar

A mi cuñado Benjamin, su esposa Alma Karina, Benji, Roberto y Eddy , gracias por el aliento.

A mi cuñada Helen, Jeremy y su hija Iris (Por cierto Iris, EmCalc no es una calculadora marciana).

A mi cuñado Francisco, Margarita y sus hijos Eliane, Alan, Bianca, Anaí, Alexis, Michelle y Karen (Gracias Paco por la idea de desarrollar una calculadora).

A John, George, Paul y Ringo por proveer la maravillosa música durante el desarrollo de la tesis

Mi enorme gratitud y amor para todas estas personas y mi gratitud aún mayor a Dios por haberme permitido encontrarlas en mi camino.

EmCalc

Emulación de Calculadora para Access.

1. Introducción	i
1.1. Objetivo.....	i
1.2. Antecedentes.....	i
1.3. Calculadoras de Referencia para la Simulación.....	ii
1.4. Presentación de EmCalc.....	iv
1.5. Terminología Empleada.....	v
2. Manual de Operación de EmCalc	1
2.1. Convenciones y descripción de objetos utilizados.....	1
2.2. Organización de las Pantallas en EmCalc.....	5
2.3. Activación de EmCalc.....	6
2.4. Pantalla principal ó Pantalla de Presentación.....	7
2.5. Módulo 1. Operaciones y Funciones Básicas.....	8
2.6. Módulo 2. Valuación de Funciones.....	12
2.6.1. Módulo 2.0 Consideraciones generales para graficación de funciones.....	17
2.6.2. Módulo 2.1 Graficación de Funciones. Generación de Puntos Manualmente.....	18
2.6.3. Módulo 2.2 Generación de Puntos a Graficar Automáticamente.....	21
2.6.4. Módulo 2.3 Graficación de Múltiples Funciones.....	24
2.7. Módulo 3. Valuación de Funciones F(X,Y).....	27
2.8. Módulo 4. Funciones Estadísticas.....	29
2.8.1. Módulo 4.1 Análisis de datos.....	30
2.8.2. Módulo 4.2 Análisis de Regresión.....	33
2.8.3. Módulo 4.3 Despliegado de Formas con Tablas Estadísticas.....	37
2.8.3.1. Módulo 4.3.1 Generación de Números Aleatorios.....	38
2.8.4. Módulo 4.3 Pre-impresión é Impresión de Tablas Estadísticas.....	40
2.9. Módulo 5. Funciones Financieras.....	41
2.9.1. Módulo 5.0 Anualidades (Características de las pantallas de anualidades).....	42
2.9.2. Módulo 5.1 Valor Presente.....	43
2.9.3. Módulo 5.2 Valor Futuro.....	45
2.9.4. Módulo 5.3 Cálculo de la Tasa de Interés.....	47
2.9.5. Módulo 5.4 Cálculo del Pago Periódico.....	50
2.9.6. Módulo 5.5 Cálculo del Número de Pagos.....	52
2.9.7. Módulo 5.6 Cálculo del Capital é Interés Pagado.....	55
2.9.8. Módulo 5.7. Cálculo del Valor Presente Neto (VPN).....	58
2.9.9. Módulo 5.8. Depreciación.....	60
2.9.10. Módulo 5.9. Tasa Interna de Retorno.....	62
2.10. Módulo 6. Uterías/Solución de Problemas.....	64

3. Diseño y desarrollo de EmCalc.	
3.1. Elementos de Access y Visual Basic Empleados	67
3.1.1.Tablas.....	67
3.1.2.Queries.....	69
3.1.3.Formas ó Pantallas.....	72
3.1.4.Reportes, Macros y Módulos.....	73
3.1.5.Lista de Tablas, Queries y Formas.....	74
3.2. Carga de formas (Transferencia de pantallas)	77
3.2.1.Estructura de organización de las pantallas de EmCalc.....	77
3.2.2.Acceso a las pantallas de trabajo.....	78
3.3. Formas y Subformas	87
3.4. Objetos Empleados para Crear una Forma. Cajas-Combo	90
3.4.1.Cajas-combo ligadas a una tabla ó query.....	91
3.4.2.Cajas-Combo Especificando los Valores al Momento de su Creación.....	98
3.5. Desarrollo de las Pantallas de ayuda	99
4. Código de Programación por Pantalla	104
4.1. Generalidades.....	104
4.2. Operaciones y Funciones Básicas.....	106
4.3. Valuación de Funciones.....	110
4.3.1.Graficación con Captura Manual de Puntos.....	114
4.3.2.Graficación con Generación de Puntos Automáticamente.....	116
4.3.3.Graficación Múltiple.....	119
4.4. Valuación de Funciones F(X,Y).....	122
4.5. Funciones Estadísticas.....	125
4.5.1.Análisis de Datos.....	126
4.5.2.Análisis de Regresión.....	129
4.5.3.Números Aleatorios.....	134
4.5.4.Creación de las Pantallas de Tablas Estadísticas.....	136
4.5.5.Desplegado de los reportes.....	137
4.6. Funciones Financieras.....	139
4.6.1.Anuales: Valor Presente.....	140
4.6.2.Anuales: Valor Futuro.....	143
4.6.3.Anuales: Cálculo del Interés.....	146
4.6.4.Anuales: Cálculo del Pago.....	149
4.6.5.Anuales: Cálculo del Número de Pagos.....	152
4.6.6.Anuales: Cálculo del Capital é Interés Pagado.....	155
4.6.7.Cálculo del Valor Presente Neto.....	158
4.6.8.Depreciación.....	161
4.6.9.Tasa Interna de Retorno.....	165
4.7. Utilerías/Solución de Problemas.....	168
4.7.1.Macros para Importar Archivos.....	169

5. Módulo General de EmCalc (Funciones Creadas)	170
5.1. Operaciones y Funciones Básicas.....	171
5.1.1. Operacion(Op1, Opr, Op2).....	171
5.1.2. PreConst(Funcion, Arg).....	172
5.2. Manejo de archivos	174
5.2.1. TraeTbl(tblName, Tipo).....	174
5.2.2. GuardaTbl(Tipo).....	175
5.3. Valuación de Expresiones para Funciones.....	176
5.3.1. Equation(strMidEquation, dblValue, ArreglaErr, ErrCero, ErrOverFlow, ErrOtro)..	176
5.3.2. StrEcuacion(strEquation, dblValue).....	178
5.4. Creación de Puntos automáticamente para graficar.....	179
5.4.1. CreaSerie(Inicio, Fin, Puntos).....	179
5.4.2. CreaSerie2(Inicio, Fin, Incr).....	180
5.5. Funciones del módulo de Estadística	181
5.5.1. AnDatos().....	181
5.5.2. Regresion(Grado, Desplegar)	184
5.5.3. NAleatorios(TotNos, LInf, LSup).....	190
6. Implementación de Palabra Clave para Acceso a EmCalc.....	191
7. Conclusiones.....	192
Apéndice I. Requerimientos para correr EmCalc.....	I
Apéndice II. Bibliografía.....	II

1. Introducción.

El presente trabajo de tesis es el resultado de una idea sencilla que se fué expandiendo gradualmente hasta convertirse en una aplicación práctica, con una cantidad adecuada de procesos. En su realización se requirió de investigación é implementación de funciones, rutinas y procesos en el software MS-Access y el lenguaje de programación Visual Basic.

A título personal, además de resultar una experiencia interesante, el desarrollo de este trabajo resultó ser de gran utilidad para incrementar mi conocimiento de Access y de Visual basic, los cuáles son parte importante en el desarrollo de mi trabajo actual en un área de programación utilizando bases de datos relacionales.

A lo largo de este capítulo se describe por medio de diferentes aproximaciones en que consiste el trabajo realizado y se dán diferentes motivos por los cuáles se eligió este tema.

1.1 Objetivo.

La tesis desarrollada se puede considerar como un trabajo académico consistente en una simulación ó emulación de calculadora que se asemeja a las aplicaciones disponibles al público actualmente, acompañada de un texto que describe sus funciones y la forma en que fué desarrollado el producto.

La aplicación de computación desarrollada se ofrece como una herramienta de cálculo de operaciones matemáticas que proporciona una alternativa de operación a la que actualmente otorgan las calculadoras y emulaciones que existen de éstas.

El objetivo primordial es mostrar una aplicación amigable que permite realizar procesos avanzados con una capacitación mínima ó nula, la cuál en caso de ser necesario se proporciona por la propia aplicación. En cuánto a "procesos avanzados", se refiere aquí a tareas que un usuario utilizando una calculadora, generalment requiere de un manual para poder desarrollarlos, al menos la primera vez.

Adicionalmente, se puede establecer un objetivo hacia la gente encaminada al área de desarrollo de aplicaciones de computación como "El mostrar el desarrollo de una aplicación general tipo simulación de calculadora, utilizando el software Access".

1.2 Antecedentes.

De los inventos de la era moderna, la calculadora es sin duda una de las herramientas más empleadas, indispensables para la realización de trabajo y otras tareas que requieren cálculos matemáticos, desde operaciones sencillas como suma, resta, multiplicación y división, hasta funciones complejas como logaritmo de un número, exponenciación, funciones trigonométricas, pasando por funciones financieras y funciones estadísticas.

Las calculadoras son parte elemental de la vida moderna y se usan extensamente en los diferentes ámbitos de trabajo, incluido aquí el estudio. Son tán necesarias que en muchas ocasiones, incluso al estar utilizando una computadora, se emplea una calculadora como auxiliar para realizar cálculos rápidamente; en otras ocasiones se abre una hoja de cálculo para realizar operaciones, simulando una calculadora; algunos softwares incluso proveén una simulación de calculadora para usarse en las computadoras.

Una característica muy importante de una calculadora es su sencillez y facilidad de operación. Esto es especialmente cierto para operaciones y funciones sencillas, elementales; para procesos más avanzados lo anteriormente dicho suele ser falso en la mayoría de los casos. Su operación se vuelve sencilla e incluso automática después de seguir repetitivamente los pasos que se muestran en los manuales. Dependiendo de la persona que se trate y factores como su nivel educacional, empeño, paciencia, etc., el aprender a utilizar una calculadora en toda su capacidad puede ser un camino tortuoso. Por supuesto que es una gran recompensa personal el obtener el conocimiento para utilizar una calculadora en toda su extensión.

La solución a este problema que se presenta en la calculadoras es obtener una calculadora que proporcione instrucciones, ayudas y ejemplos al momento de estar trabajando con ella. Esto, presenta el problema de tener que presentar mucha información y tener la capacidad de desplegar diferentes pantallas ó pantallas formatcadas en diferentes maneras. Por supuesto que ésto es imposible con el tamaño de desplegado de una calculadora, sin embargo, es completamente factible utilizando las características de una computadora.

El presente proyecto de tesis presenta una solución al problema mencionado, utilizando los recursos de software actuales y la capacidad de cómputo que una computadora moderna pueden otorgar. El resultado es una emulación de calculadora que facilita el trabajo al tener a la vista las opciones que el usuario desea utilizar, guiándolo en cada paso.

1.3 Calculadoras de Referencia para la Simulación.

Hay varios tipos de calculadoras, con diferentes características y todas ellas proveén funciones de cálculo acorde a su área de especialización.

Los tipos de calculadoras que sirven de referencia para este proyecto son:

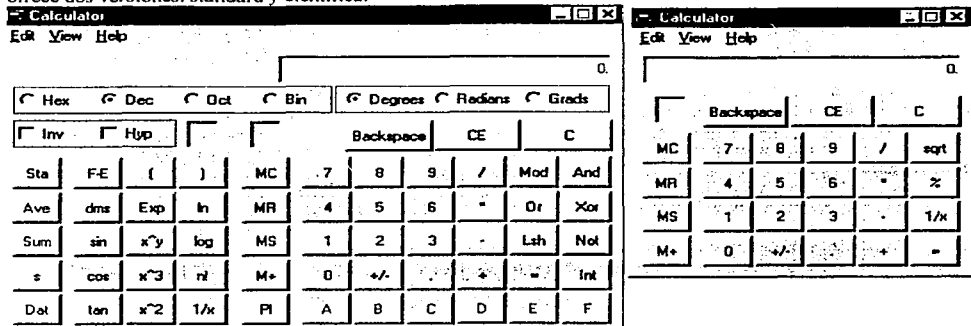
- **Calculadoras científicas.** Cuentan, además de las operaciones básicas (Suma, Resta, Multiplicación y División y uso de una memoria), con un enorme acervo de funciones científicas, entre las cuales se encuentran las funciones trigonométricas, funciones hiperbólicas, logaritmo de un número, exponenciación, etc.
- **Calculadoras financieras.** Incluyen la capacidad de operaciones básicas y adicionalmente proveén funciones especiales diseñadas para el ámbito de finanzas. Las calculadoras financieras generan funciones tales como Tasa Interna de Retorno, Cálculo de Anualidades, Depreciación y otras; adicionalmente proveén las funciones científicas más comunmente usadas.
- **Calculadoras para gráficación.** Permiten realizar gráficas de funciones de una variable.

Adicionalmente, este tipo de calculadoras deben tener las siguientes características:

- **Módulo de estadística.** Es común que las calculadoras científicas y financieras tengan integrado un módulo para el cálculo de funciones estadísticas, como la media, varianza, modelo de ajuste lineal para una serie de puntos dados, etc.
- **Programabilidad.** Cada diferente marca de calculadora utiliza su propia código de programación, el cuál consiste en una serie de instrucciones que el usuario puede dar a la calculadora para lograr un resultado.

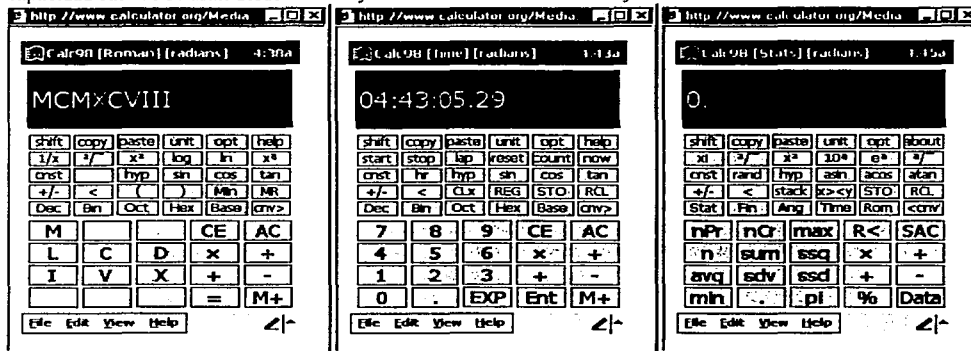
Cabe aclarar que la aplicación desarrollada en esta tesis, no es programable. Se menciona esta característica porque las facilidades que otorga suplen en parte la facilidad de programar, dicho en otra forma, para que una calculadora pueda realizar ciertos procesos que la aplicación en cuestión realiza, necesita ser programable.

En el caso de este proyecto, el software que se emplea es Access de Microsoft y la aplicación desarrollada no emula a ninguna calculadora conocida, sino que es un desarrollo totalmente independiente con características propias de diseño. Aún siendo una emulación de calculadora, difícilmente se puede considerar similar a otras emulaciones que existen actualmente accesibles al público, caso concreto la "calculadora" que Microsoft ofrece a los usuarios de su sistema operativo Windows, la cuál sin duda es la emulación más conocida, la cuál ofrece dos versiones: estándar y científica:



Calculadora de Windows en sus dos versiones: científica y standard.

Otras emulaciones se pueden encontrar gratis ó por un pago, en Internet, algunas ofrecen versiones reducidas en capacidad cuando se usan desde Internet y aumentan sus funciones al bajarlas a un disco duro.



Emulaciones de la página de internet www.calculator.org.

Nótese como generalmente las emulaciones de calculadora intentan parecerse en su presentación y funcionamiento a una calculadora.

La aplicación que se describirá en este trabajo se denominará **EmCalc** por Emulación de Calculadora. Se presentarán ilustraciones de ella a lo largo de este texto.

1.4 Presentación de EmCalc.

Como se estableció en la sección anterior, la aplicación desarrollada se denomina **EmCalc** por Emulación de Calculadora.

Para efectos del presente trabajo una emulación ó simulación de calculadora se define como "una aplicación desarrollada en un software ó en un lenguaje de computación, que permite desarrollar cálculos matemáticos de tipo general, en forma similar a como lo hace una calculadora". Se entiende que la emulación se utiliza en una computadora.

EmCalc es una herramienta que permite el cálculo de operaciones; creación de expresiones para funciones de una y dos variables, valuación y graficación de funciones de una variable; funciones estadísticas, tablas estadísticas y funciones financieras. Brinda una forma sencilla, amigable, propia de una calculadora, el desarrollo del trabajo, a la vez que aprovecha la facilidad y poder que provee una computadora para el despliegado de información graficamente, poder de cálculo y rapidez de respuesta. Adicionalmente, utilizando características del software MS-Access, provee la capacidad de creación, acceso y almacenamiento masivo de información; proporciona ayudas en línea, las cuáles constan de un manual de operación y ayudas contextuales. El lenguaje Visual Basic provee poder adicional y flexibilidad para el desarrollo de funciones preconstruidas, manipulación de la información y chequeos de entrada de información.

Siendo desarrollado en MS-Access y Visual Basic, **EmCalc** es una base de datos; adicionalmente, al igual que una calculadora, EmCalc está orientada al cálculo de operaciones, valuación de funciones y creación/análisis de datos en general, por lo tanto, **EmCalc** es una aplicación de propósito general que aprovecha la base de datos subyacente para proveer facilidades adicionales a las otorgadas por una calculadora. EmCalc también permite importar y exportar datos en el formato de MS-Excel. Ambas clasificaciones son correctas y son complementarias.

EmCalc es una aplicación desarrollada en módulos, enlazados por medio de un módulo principal. Cada módulo permite una serie de funciones ó procesos, acorde a los tipos de calculadoras que se emulan:

- Operaciones y funciones, permite realizar operaciones, valuar funciones y usar 6 memorias.
- Creación y valuación de funciones de una variable, con acceso a tres diferentes módulos de graficación que permiten la especificación manual o creación automática de puntos a graficar, pudiendo graficar desde una, hasta 5 funciones a la vez.
- Creación y valuación de funciones de dos variables.
- Funciones y tablas estadísticas. Datos descriptivos de una serie de observaciones; creación de modelos de ajuste para una variable; puntos porcentuales para las funciones: Normal, t de Student, Ji Cuadrada, F al 5% de significancia, F al 1% de significancia y tablas de números aleatorios generados acorde a parámetros.
- Funciones financieras. Anualidades (valor presente, valor futuro, pago, interés, total de pagos, capital é interés pagado a cierto período), valor presente neto, depreciación y tasa interna de retorno.

- **Utilerías.** Basicamente consiste en la importación de archivos de Excel y recomendaciones en caso que el sistema se haga lento con el paso del tiempo y su utilización.

Cada sección permite el guardado, acceso y borrado de datos en archivos permanentes, según se considero necesario.

1.5 Terminología empleada.

Debido a que es un trabajo desarrollado en el área de computación, se utilizan términos y expresiones propias de computación y programación en inglés, que es el idioma en que estos términos fueron creados. En los casos en que es posible, se utiliza la terminología en español.

Uso como justificación el hecho de que los términos son ampliamente conocidos por la gente del área de computación y programación y permiten una comunicación efectiva con la gente que se dedica a estas áreas. Lamentablemente algunos lectores no estarán familiarizados con este léxico; a ellos principalmente les presento una apología.

Algunos términos, aunque ya se utilizan en español, debido a que su empleo no es del dominio común, crean expresiones extrañas, por ejemplo "bajar un programa a una computadora". Esta expresión surgió del hecho de que en sus inicios, una unidad de cinta para computadora, alojaba la cinta en su parte más alta, quedando colocada en un nivel más alto que las unidades de disco (por ser estas últimas más bajitas), de modo que al hacer transferencias de la cinta hacia el disco, se "bajaban" los archivos de la cinta al disco; a su vez, al transferir del disco hacia la cinta, los archivos se "subían" del disco a la cinta. La idea quedó grabada y ahora se supone que cualquier computadora poderosa o Internet están en un nivel más alto que una computadora de usuario, incluyendo las cintas y otros sistemas de respaldo de información.

En otros casos no definen la amplitud del término, por ejemplo, para la palabra "software" que se puede traducir como "generador de aplicaciones" o "generador de programas", se utilizan las palabras "programa" o "aplicación" como su traducción, las cuales no definen completamente el objeto en cuestión.

Algunas otras palabras crean confusión o expresiones extrañas, por ejemplo la traducción de la palabra "overflow", la cual significa que al realizarse una operación, el resultado rebasó o sobrepasó la capacidad de almacenamiento o de representación de la variable receptora; la palabra utilizada es "desbordamiento", perfectamente adecuada y cada vez más usada, pero, la relacionamos más con el desbordamiento de un río.

Otro ejemplo en que la traducción crea confusión es la palabra "default" usada en expresiones como "valores default", o "valores por default", o "se usa el default..."; que se traducen como "valores por omisión" o "valores por defecto"; "se usan los valores 'X' por omisión" o "se usan los valores 'X' por defecto", realmente son expresiones que no se relacionan directamente con el hecho de que "se usarán ciertos valores 'X' automáticamente en caso que el usuario no especifique esos valores o los especifique erróneamente".

En general se utilizan los términos que se consideran apropiados al área de computación, aceptados en nuestro idioma, en nuestro país.

En el caso de temas en otras áreas se utiliza una comunicación coloquial, mas propia hacia un usuario de la calculadora que el área específica que se trata.

2. Manual de Operación de EmCalc.

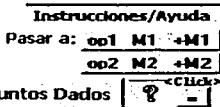
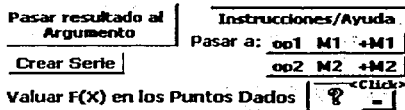
EmCalc fue creada teniendo en mente la facilidad de uso que otorga una calculadora. Aprovechando las facilidades que otorga una computadora y el software Access, se logra crear una "Emulación de calculadora" que permite crear pantallas auto-explicativas, previendo adicionalmente ayudas contextuales en línea.

Para usar EmCalc no existen prerequisites, sin embargo, si es necesario que el usuario conozca la teoría del módulo que está utilizando para que la interpretación de resultados sea la adecuada.

A continuación se presentan los elementos que se utilizan en este manual de operación para describir las tareas que EmCalc puede realizar.

2.1 Convenciones y descripción de objetos utilizados.

Caja-comando. Una caja comando permite realizar una acción al hacer clic en ella. En general las cajas comando aparecen como los ejemplos que se muestran, en formas y tamaños diferentes, pero siempre en un rectángulo o cuadrado.



- En ocasiones mencionan completamente la acción a realizar como <Pasar resultado al Argumento>.
- Otras ocasiones se auxilia de un letrero como "Pasara a:" y enseguida o abajo aparece el receptor de la acción, en este caso Op1, M1, +M1, etc.
- Aparecen siempre en color gris, pero sus letras pueden ser de diferente color.

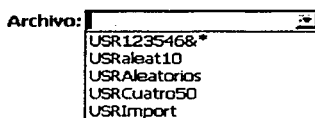
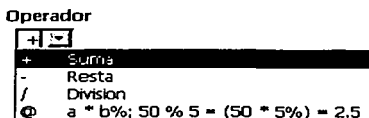
Comando a realizar. Se indicará un comando a utilizar con el nombre que aparece en la pantalla de EmCalc entre signos "<>", ejemplos:

- <Pasar resultado al Argumento>. Se explica por sí mismo.
- <Instrucciones/Ayuda>. Solicita la página de Instrucciones y ayuda general, de la pantalla.
- <Pasara a: +M2>. Nótese que aquí se agrega el letrero auxiliar al receptor de la acción.
- <?>. Indica un comando de ayuda, de alguna sección de la pantalla.

Notas:

- En ocasiones se resalta el comando con negritas, para enfatizarlo: <Comando>.
- "Oprimir" se usará como sinónimo de "hacer clic" al ratón de la computadora.
- Botón se usará como sinónimo de caja-comando. P. Ej. : Oprimir botón <Crear serie>.

Caja-combo. Al oprimir una caja-combo se mostrará una lista de opciones de las cuáles el usuario debe seleccionar una; sólo se puede elegir de las opciones listadas. Una caja-combo siempre tendrá un letrero indicando su nombre. El nombre aparecerá arriba o a la izquierda, según sea necesario colocarlo para mejor identificar la variable u operador de que se trate.



7. Inserción de renglones, sólo se puede hacer al final de la columna (donde está un "*", en la ilustración aparece como "nuevo dato a teclear"). EmCalc ordenará los valores de menor a mayor al valuar F(X) y graficar.

Notas:

Abajo del área de datos existe la palabra "Record" junto con 6 "cajas", cada una representa lo siguiente (de izquierda a derecha), se pone el dibujo aproximado que aparece:

- 1) Ir a primer renglón o inicio de los datos. |<
- 2) Ir a renglón anterior. <
- 3) Caja en blanco. Aquí se indica un renglón al que se desea ir, oprimiendo <enter> después de especificarlo.
- 4) Ir al renglón siguiente. >
- 5) Ir al último renglón (fin de los datos). >|
- 6) Insertar un renglón >* (sólo se pueden escribir nuevos datos donde está el "*").

Tablas de captura con más de una columna.

	Pago	Interes
	150.00	0.5000%
	250.00	0.7000%
	255.55	1.0500%
	244.00	0.6000%
✎	380.00	0.0000%
*	0.00	0.0000%

En este tipo de tablas, el usuario puede teclear ya sea por renglón o por columna.

- El lápiz al inicio del renglón indica que es donde se está editando.
- El "*" indica un nuevo renglón donde se puede insertar datos.

Captura de datos en columnas específicas.

1) Teclear valores para X y Y, ó, traer datos de un archivo

X	Y	TransX	TransY	Yajuste	Residual
13	23	0	0	0	0
5	11	0	0	0	0
0	0	0	0	0	0

En este caso, se requiere que el usuario introduzca datos sólo en las columnas 1 y 2.

EmCalc se asegura de proteger los datos de las columnas adyacentes, son campos no editables. En este caso EmCalc indica donde se deben capturar los datos.

Control de Errores al valuar una función.

En las secciones de Graficación y Análisis de Regresión, que utilizan funciones, se hace necesario controlar los errores que se pueden producir al valuar la expresión de la función.

Control de Errores al Valuar la Función:

Si hay división por cero, hacer F(X) =

Si hay Overflow, hacer F(X) =

Si hay Otro Error, hacer F(X) =

Al detectar errores al valuar F(X), EmCalc hace F(X) = 0 en cada caso, sin embargo, el usuario puede especificar otro valor para detectar de que tipo de error se trata. Los errores detectables son:

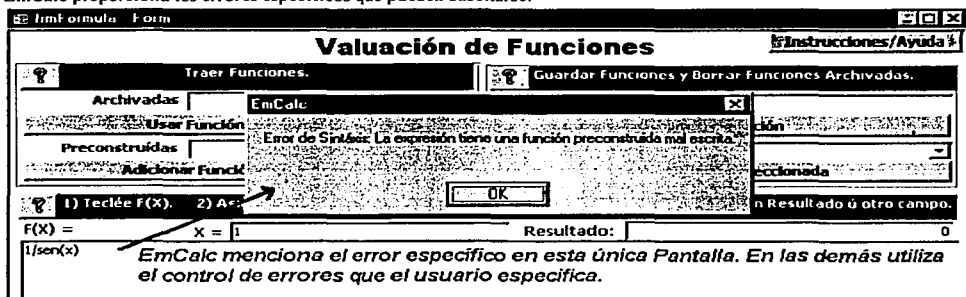
- 1. División por Cero. Use se default (cero) o especifíquese otro valor.

2. **Overflow o Desbordamiento.** Al valuarse la función se genera un número muy grande que no es posible representar por la computadora. Usese default (cero) o especifíquese otro valor.
3. **Otro Error,** úsese default (cero) o especifíquese otro valor. Errores de Otro Tipo (*).
 - Expresión errónea. La expresión tiene sintaxis incorrecta la cual puede ser ocasionada por múltiples causas incluyendo la omisión de operadores, ejemplo: $3x$ debe escribirse $3*x$; los nombres de funciones predefinidas son incorrectos, ejemplo: $\text{Sen}(x)$ debe escribirse $\text{Sin}(x)$; etc. La caja-combo "Preconstruidas" del módulo "Valuación de Funciones" tiene la sintaxis correcta de las funciones.
 - La expresión excede 255 caracteres, ya sea cuando está en términos de X o al sustituir X por su valor.

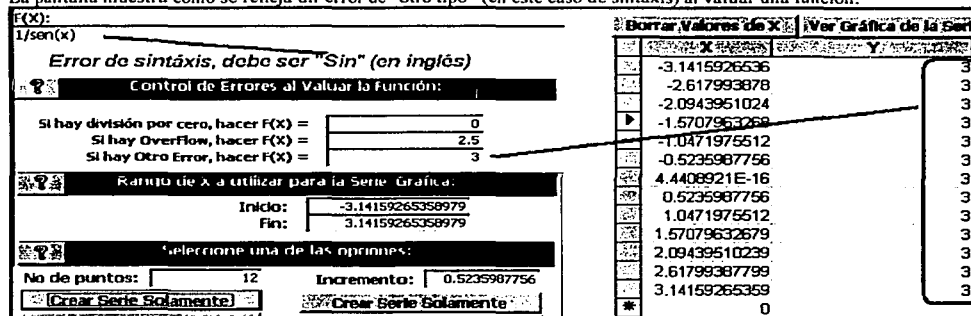
Al encontrar errores, se recomienda acceder al módulo de **Valuación de Funciones** y valorar la función en los puntos en que se produce el error. Este módulo produce el mensaje específico para el error.

A continuación se dan ejemplos de errores detectados, según la sección en que se utilice la función.

Fragmento de la Pantalla **Valuación de Funciones** mostrando el mensaje específico de error. Ninguna otra sección de EmCalc proporciona los errores específicos que pueden suscitarse.



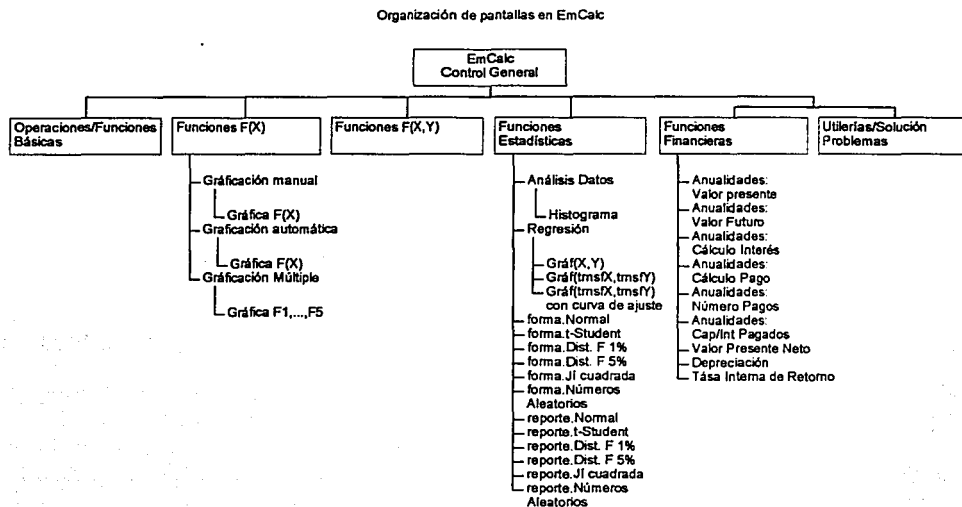
La pantalla muestra como se refleja un error de "otro tipo" (en este caso de sintaxis) al valorar una función.



2.2 Organización de las Pantallas en EmCalc.

Para el desarrollo del trabajo EmCalc provee módulos que se desarrollan en diferentes pantallas. Cada módulo consiste de las pantallas necesarias para proveer información suficiente para el proceso que se esté llevando a cabo.

El siguiente organigrama muestra la estructura en que están organizadas las pantallas de EmCalc. En ella se puede distinguir un máximo de 3 niveles de subordinación.



Estructura jerárquica de las formas o Pantallas.

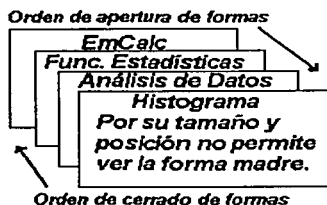
Navegación en las pantallas en EmCalc. El organigrama anterior además de mostrar la estructura jerárquica de las pantallas, muestra la ruta de navegación o acceso a las pantallas.

Apertura y Cerrado de Pantallas. Al irse abriendo las formas, éstas quedan organizadas para cerrarse en el orden en que fueron abiertas, en cada nivel en que se encuentren.

El tamaño y la posición de las pantallas al abrirse, no permite ver las pantallas subyacentes, es por eso que se pueden cerrar en el orden en que fueron abiertas.

Se recomienda al usuario no disminuir la pantalla actual (la que se está viendo), para evitar confusión.

Considérese el ejemplo mostrado en la figura de abajo.



- Inicialmente se abre la forma principal (EmCalc).
- Se usa la opción de Funciones Estadísticas.
- Se selecciona la opción de Análisis de Datos.
- Se genera el Histograma.

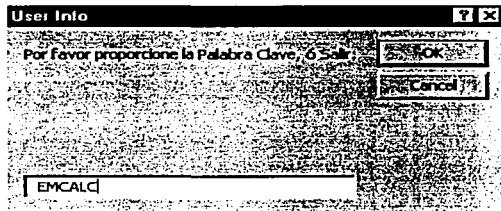
Todas las pantallas quedan abiertas, se recomienda no achicarlas (achicarlas quiere decir que sólo aparece su icono en la barra de aplicaciones de Windows).

Para cerrar las pantallas, hágase en el orden invertido de apertura.

2.3 Activación de EmCalc.

Para activar la aplicación, deben correrse en la computadora los software Access 97 o Access 2000. Al seleccionar la base de datos, el nombre a buscar es **EmCalc.mdb**.

Al ser solicitado proporciónese la palabra clave: **EmCalc**. Esta palabra clave puede ser en mayúsculas, minúsculas o combinación de ambas.



Existe la opción de no activar EmCalc, en cuyo caso hay que teclear **salir**, igual, puede ser en mayúsculas, minúsculas o combinación de ambas. Access será cerrado.

EmCalc dará 10 oportunidades para proporcionar cualquiera de las dos opciones, en caso de no proporcionar ninguna de ambas, Access se cerrará después de emitir un mensaje.

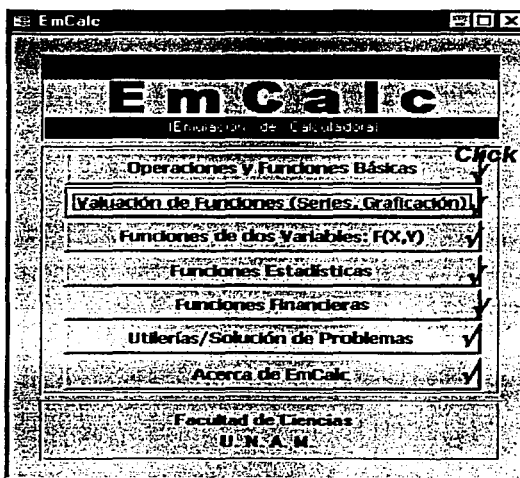
Al proporcionar la palabra clave aparece automáticamente la pantalla de presentación de EmCalc.

2.4. Pantalla principal o Pantalla de Presentación.

La pantalla principal de EmCalc da acceso a los módulos de trabajo y siempre permanece activa. Si se cierra, se pierde acceso a los módulos, si sucediera tal caso, hay que cargar nuevamente EmCalc.

Acceso a los módulos de trabajo.

Para acceder cualquiera de los módulos para realizar una tarea, hágase clic en la caja-comando del módulo respectivo. Esto hará que se cargue la pantalla adecuada a la tarea solicitada.



La figura muestra la pantalla inicial de EmCalc y las opciones disponibles para realizar tareas. Un clic del ratón en cualquier parte de la caja es suficiente para activarlas.

- Para cerrar las pantallas de trabajo, hay que hacer clic en el cuadro marcado con "X" en la parte superior de las pantallas.
- Para terminar EmCalc hay que usar el comando del menú de Access <File> <Close>.

2.5 Módulo 1. Operaciones y Funciones Básicas.

Esta pantalla permite realizar operaciones y valorar funciones preconstruídas. Provee la facilidad de utilizar 5 memorias para almacenar información, más una adicional que permite realizar la suma de las memorias anteriores. Los submódulos de que consta son:

1. Valuación de Funciones Preconstruídas.
2. Realización de Operaciones.
3. Utilización de Memorias.

Pantalla del módulo en su desplegado inicial.

Operaciones y Funciones Básicas

Funciones Preconstruídas

Función	Argumento (arg)	<Clic>	Resultado <Clic>
Log	0	0	0
Pasar resultado al Argumento		Pasar a: (op1) M1 (M1)	
		(op2) M2 (M2)	
		(M3) (M3)	
		(M4) (M4)	
		(M5) (M5)	

Memorias

Borrar	M1	M2	M3	M4	M5	Todas	Suma		
	0	0	0	0	0	0	0		
	op1	op2	arg	op1	op2	arg	op1	op2	arg

Operaciones

Operando 1 (op1)	Operador	Operando 2 (op2)	<Clic>	Resultado <Clic>
0	+	0	0	0
Pasar resultado a Operando 1		Pasar resultado a Operando 2		

BORRAR TODO EN LA PANTALLA

Cómo empezar

Acorde a la tarea a realizar, el usuario puede utilizar inicialmente cualquiera de las secciones **Funciones Preconstruídas** ó **Operaciones**. Las memorias sólo pueden utilizarse después de obtener un resultado de cualquiera de estas secciones.

1. **Funciones Preconstruídas.** Permite la valuación de funciones matemáticas, más un cierto número de funciones derivadas y obtener aproximaciones a los números "Pi" y "e".

Función	Argumento (arg)	<Clic>
Log	0	0
Exp	e^arg {arg<= 709.782712893}	
Sin	Sen(arg) {arg en radianes}	
Cos	Cos(arg)	
Abs	Abs(arg)	

los siguientes controles:

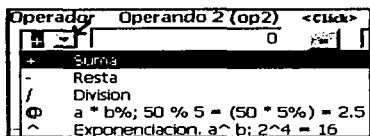
- o Al argumento de la función; clic en el comando <Pasar resultado al argumento>.

Para valorar una función, siganse los siguientes pasos:

- Clic en la caja-combo de **Función**; de la lista que aparece elijase una función;
- Tecléese un valor en **Argumento**, apropiado para la función; excepto para Pi y e;
- Clic en "=" o en **Resultado**. Ambos generan el resultado.
- El resultado de la función se puede transferir a cualquiera de

- o A los operandos 1 o 2 de la sección de Operaciones; clic en **Pasar a:** <op1> ú <op2>.
- o A una memoria; clic a cualquiera de los botones <M1>, <M2>, <M3>, <M4> o <M5>.
- o Sumar a una memoria; clic a cualquiera de los botones <+M1>, <+M2>, <+M3>, <+M4> o <+M5>.

2. **Operaciones.** Se usa para obtener resultados utilizando dos operandos y un operador, se proveen algunas funciones que requieren dos operadores y los números Pi y e.



Para realizar una operación siganse los siguientes pasos:

- Tecléese el **Operando 1**;
- Selecciónese o tecléese un **Operador** en la caja-combo del operador
- Tecléese el **Operando 2**;
- Clic en la caja "=" (signo igual), o clic en la caja de **Resultado**.
- El resultado de la operación se puede transferir a cualquiera de

los siguientes controles:

- o A los operandos 1 o 2; clic en <Pasar Resultado a Operando 1> o <Pasar Resultado a Operando 2>.
- o Al argumento de la función; clic en <Pasar a: Argumen>.
- o A una memoria; clic a cualquiera de los botones <M1>, <M2>, <M3>, <M4> o <M5>.
- o Sumar a una memoria; clic a cualquiera de los botones <+M1>, <+M2>, <+M3>, <+M4> o <+M5>.

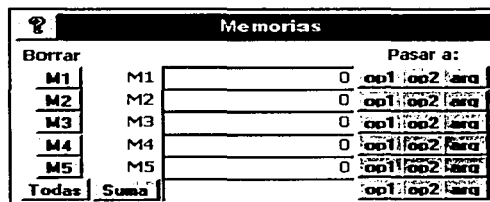
Para obtener el valor de "Pi" y de "e"

- Selecciónese a "Pi" o "e" como operadores,
- Clic en "=" o en "resultado". El valor escogido aparece como resultado.

Función Log_(b),

- Tecléese la base del logaritmo en el operando 1,
- Selecciónese el Operador Lga(b),
- Tecléese en el operando 2 el número del cuál se desea conocer el logaritmo,
- clic en "=" o en **Resultado**

3. **Memorias.** Se dispone de 5 memorias cuyo contenido se visualiza en todo momento, más una adicional que puede generar la suma de las 5 anteriores, también visible en todo momento. Se conocen como M1, M2, M3, M4, M5 y la Suma de Memorias.



El contenido de cualquiera de las memorias (incluida la suma de las memorias) puede ser transferido a los operandos 1 o 2, o al argumento de la función usando los comandos pasar a Op1, OP2, Arg, adyacentes al campo de la memoria.

- Se puede borrar individualmente cada memoria o todas a la vez usando los comandos **borrar: M1, M2, M5,** o <Borrar Todas>.
- <Suma> produce la suma de las memorias M1 a M5

<Borrar Todo en la pantalla>, este comando borra los operandos, el resultado de la operación, el argumento, el resultado de la función y las memorias. Inicializa la función a "Log" y el operador a "+".

Ejemplo: se realiza el cálculo de la función: $3x^2 + \text{Log}(x)$, para $x = 7.7$

Se puede empezar en cualquier sección **Funciones Preconstruidas** ó **Operaciones**, a elección del usuario. En el caso de la función propuesta, existen varias formas de procesar los cálculos. El siguiente orden, por lo tanto, no es el único.

Acción

- 1) Id (7.7) = 7 => M1
- 2) M1 => Arg
- 3) Log(7.7) = 2.041 => M2
- 4) M1 => Op1
- 5) $7.7^2 = 59.29$ => Op1 Opr "A"
- 6) $59.29 * 3$ => 177.87 => M1
- 7) Suma[Memorias] = 179.911

Equivalencia en EmCalc

- Función Id, 7.7 en Arg; clic en "="; <pasar a M1>
 <pasar M1 a Arg>
 función Ln; clic en "="; <pasar a M2>
 <pasar M1 a Op1>
 2 en Op2; clic en "="; <pasar resultado a OPI>
 Opr "A"; 3 en Op2; clic en "="; <pasar a M1>
 Suma [memorias] ← Resultado Final.

Operaciones y Funciones Básicas

Funciones Preconstruidas

Función	Argumento (arg)	<click>	Resultado <click>
Log	7.7	=	2.04122032885964
Pasar resultado al Argumento		Pasar a: op1 M1 +M1	
		op2 M2 +M2	
		M3 +M3	
		M4 *M4	
		M5 +M5	

Memorias

Borrar	M1	M2	M3	M4	M5	Todas	Suma	Pasar a:
		177.87						op1 op2 arg
		2.04122032885964						op1 op2 arg
		0						op1 op2 arg
		0						op1 op2 arg
		0						op1 op2 arg
		179.91122032886						op1 op2 arg

Operaciones

Operando 1 (op1)	Operador	Operando 2 (op2)	<click>	Resultado <click>	Pasar a:
59.29	*	3	=	177.87	M1 =M1
Pasar resultado a Operando 1		Pasar resultado a Operando 2			M2 =M2
					M3 =M3
					M4 =M4
					M5 =M5

BORRAR TODO EN LA PANTALLA

La figura muestra la pantalla después de realizar los cálculos del ejemplo propuesto en esta sección.

Para realizar otros cálculos, por ejemplo valuar la expresión dada, en otro punto, se recomienda "borrar todo en la pantalla" usando el comando que se proporciona, localizado en la parte baja de la pantalla.

Notas:

- Sólo los resultados pueden ser pasados a las memorias, ya sea de una función o de una operación.
- Cada vez que se pasa o se suma un resultado a una memoria, la suma de memorias se hace cero, porque ya no refleja la suma de las memorias.
- Cada vez que se pasa un valor al argumento, el resultado de la función se hace cero, porque ya no refleja el resultado de la función.
- Cada vez que se pasa un valor a los operandos 1 o 2, el resultado de la operación se hace cero, porque ya no refleja el resultado de la operación.

Las siguientes son el acervo de funciones disponibles en la sección de **Funciones Preconstruidas**.

Funciones Preconstruidas. Se incluyen aproximaciones a los números Pi y e:

- | | |
|------------|--|
| 1) Abs(x) | Valor absoluto de un número |
| 2) Atn(x) | Arco Tangente de x. $\{-\pi/2 \leq \text{Atn}(x) \leq \pi/2 \text{ radianes}\}$. Grados = Radianes*180/Pi. |
| 3) Cos(x) | Coseno(x) |
| 4) Exp(x) | Exponencial(x) |
| 5) Fix(x) | Entero(x): Fix(-8.4)=-8; Fix(8.4)=8; véase Int(x). |
| 6) Int(x) | Entero(x): Int(-8.4)=-9; Int(8.4)=8; véase Fix(x). |
| 7) Log(x) | Logaritmo natural(x) |
| 8) Rnd(x) | Número Aleatorio (usar $x >= 0$ para que cambie el número aleatorio; $x < 0$ genera el mismo número, por ejemplo Rand(-3) siempre produce el mismo número. |
| 9) Sgn(x) | Signo(x); +1 si $x > 0$; -1 si $x < 0$; $\text{sgn}(0) = 0$. |
| 10) Sin(x) | Seno(x) |
| 11) Sqr(x) | Raíz Cuadrada(x) |
| 12) Tan(x) | Tangente(x) |

Funciones Matemáticas Derivadas. Se utiliza la expresión derivada para evaluar la función.

- | | |
|------------------------------------|--|
| 13) Secante | $\text{Sec}(X) = 1 / \text{Cos}(X)$ |
| 14) Cosecante | $\text{Cosec}(X) = 1 / \text{Sin}(X)$ |
| 15) Cotangente | $\text{Cotan}(X) = 1 / \text{Tan}(X)$ |
| 16) Arco Seno | $\text{Arcsin}(X) = \text{Atn}(X / \text{Sqr}(-X * X + 1))$ |
| 17) Arco Coseno | $\text{Arccos}(X) = \text{Atn}(-X / \text{Sqr}(-X * X + 1)) + 2 * \text{Atn}(1)$ |
| 18) Arco Secante | $\text{Arcsec}(X) = \text{Atn}(X / \text{Sqr}(X * X - 1)) + \text{Sgn}((X - 1) * (2 * \text{Atn}(1)))$ |
| 19) Arco Cosecante | $\text{Arccosec}(X) = \text{Atn}(X / \text{Sqr}(X * X - 1)) + (\text{Sgn}(X) - 1) * (2 * \text{Atn}(1))$ |
| 20) Arco Tangente | $\text{Arccotan}(X) = \text{Atn}(X) + 2 * \text{Atn}(1)$ |
| 21) Seno Hiperbólico | $\text{HSin}(X) = (\text{Exp}(X) - \text{Exp}(-X)) / 2$ |
| 22) Coseno Hiperbólico | $\text{HCos}(X) = (\text{Exp}(X) + \text{Exp}(-X)) / 2$ |
| 23) Tangente Hiperbólica | $\text{HTan}(X) = (\text{Exp}(X) - \text{Exp}(-X)) / (\text{Exp}(X) + \text{Exp}(-X))$ |
| 24) Secante Hiperbólica | $\text{HSec}(X) = 2 / (\text{Exp}(X) + \text{Exp}(-X))$ |
| 25) Cosecante Hiperbólica | $\text{HCosec}(X) = 2 / (\text{Exp}(X) - \text{Exp}(-X))$ |
| 26) Cotangente Hiperbólica | $\text{HCotan}(X) = (\text{Exp}(X) + \text{Exp}(-X)) / (\text{Exp}(X) - \text{Exp}(-X))$ |
| 26) Inverso de Seno Hiperbólico | $\text{HArcsin}(X) = \text{Log}(X + \text{Sqr}(X * X + 1))$ |
| 27) Inverso Coseno Hiperbólico | $\text{HArccos}(X) = \text{Log}(X + \text{Sqr}(X * X - 1))$ |
| 28) Inverso Tangente Hiperbólica | $\text{HArctan}(X) = \text{Log}((1 + X) / (1 - X)) / 2$ |
| 29) Inverso Secante Hiperbólica | $\text{HArsec}(X) = \text{Log}((\text{Sqr}(-X * X + 1) + 1) / X)$ |
| 30) Inverso Cosecante Hiperbólica | $\text{HArccosec}(X) = \text{Log}((\text{Sgn}(X) * \text{Sqr}(X * X + 1) + 1) / X)$ |
| 31) Inverso Cotangente Hiperbólica | $\text{HArccotan}(X) = \text{Log}((X + 1) / (X - 1)) / 2$ |
| 32) Logaritmo base N | $\text{LogN}(X) = \text{Log}(X) / \text{Log}(N)$ |

Otras funciones y/o constantes.

- | | |
|----------------------|--|
| 33) 1/x = Inverso(x) | Inverso multiplicativo. |
| 34) Id(x) = x | Función Identidad. |
| 35) Pi | Constante utilizada en cálculos que involucran el círculo. |
| 36) e | La base de logaritmos naturales. |

2.6 Módulo 2. Valuación de Funciones.

Esta Sección permite:

- Escribir expresiones en términos de X y valuarlas.
- Guardar permanentemente las funciones (expresiones) creadas, asignándoles un nombre.
- Traer o acceder una función archivada previamente, para valuarla o editarla.
- Traer o acceder funciones preconstruídas para editarlas o combinarlas con Funciones de Usuario.
- Borrar permanentemente funciones que han sido guardadas en el archivo.

Adicionalmente se da acceso a los siguientes módulos de graficación:

- Generación de gráficas dando una función $F(X)$, tecleando los puntos en los cuales se valuará la función.
- Generación de gráficas dando una función $F(X)$, generando los puntos X automáticamente en un intervalo dado, ya sea especificando cuantos puntos se desean, o, el espaciado entre ellos.
- Generación de gráficas de hasta 5 funciones, especificando cada función, generando los puntos X automáticamente en un intervalo dado; ya sea especificando cuántos puntos se desean, o, el espaciado entre ellos.

Pantalla del módulo en su desplegado inicial, excepto por la indicación de "Área de Edición de $F(X)$ ".

Valuación de Funciones Instrucciones/Ayuda

Traer Funciones.

Archivadas

Preconstruídas Adicionar Función al Final de $F(X)$

Guardar Funciones y Borrar Funciones Archivadas.

Nombre (Máx.20 caract.)

Guardar Función

Función a Borrar

Borrar Función Seleccionada

1) Teclee $F(X)$. 2) Asigne valores a X. 3) Enter, Tab ó Click en Resultado u otro campo.

$F(X) =$ $X =$ Resultado: 0

Área de Edición de $F(X)$.

Sustituyendo X por su valor $F(X) =$

BORRAR TODO EN LA PANTALLA

Generación de Series y Graficación de Funciones.

Manualmente Automáticamente Múltiples Funciones

Valuación de una Función. Cómo empezar

Para obtener la valuación de una función siganse los dos siguientes pasos:

1. A elección del usuario, háganse una de las tres siguientes opciones:
 - Construir una función tecleando una expresión en el área de edición de $F(X)$;

- Accesar una Función de Usuario; si existe una expresión para $F(X)$, la nueva función reemplazará la expresión existente. Esta expresión se puede editar.
 - Accesar una Función Preconstruida (FP), si existe una expresión para $F(X)$, la FP se agregará al final de la expresión existente. La expresión resultante se puede editar.
2. Valúese la función tecleando un valor para X.
- Al terminar de escribir el valor de X oprímase <enter>, <tab> o hágase clic en un campo diferente al de captura para X (por ejemplo, en el de resultado).
 - Para valuar en otro punto, reemplácese el valor de X y síganse las instrucciones de (a).

1a) Construcción de una función.

- Posiciónese en el área de edición de $F(X)$,
- Constrúyase la expresión algebraica en términos de X.

Nota: Esta sección acepta expresiones en términos de X y/o de Y, sin embargo, las Y's serán consideradas como X's en todos los casos.

Reglas para crear expresiones.

- Se siguen las reglas que generalmente se usan en los lenguajes de programación y hojas de cálculo, poniendo operadores aritméticos.
- El nombre de las funciones preconstruidas debe estar en inglés. P. Ej. Sin(x).
- Ejemplo, la expresión $3X\text{Sin}(X)$, se debe escribir: $3 * X * \text{Sin}(X)$, se pueden usar espacios: $3 * X * \text{Sin}(X)$
- Se pueden utilizar mayúsculas o minúsculas indistintamente y los espacios necesarios.

El tamaño máximo de una expresión para una función al ser reemplazada por su valor debe ser de un máximo de 255 caracteres.

1b) Traer Funciones de Usuario.

Traer Funciones significa extraer una función del archivo de usuario y hacerla accesible en el área de edición de $F(X)$. Para accesar o traer Funciones de Usuario se requiere que previamente el usuario ya haya creado y guardado funciones en el archivo. Si la caja-combo de funciones está vacía, no existen funciones guardadas.

Para traer una función de usuario hágase lo siguiente:

Funciones de usuario. Nótese que hay funciones $F(X)$ y $F(X,Y)$.

Traer Funciones.		Guardar Funciones y Borrar Funciones	
Archivadas	Expresión	Nombre (Máx.20 caract.)	F
Preconstruidas	f1xy	sin(x)*cos(y)	F(x,y)
	f2XY	5	F(x,y)

- Clic en la caja "Archivadas". Aparecerá una lista de las funciones que el usuario ha guardado.
 - Ejemplo: 2SinCos 2SinCos $\text{Sin}(X) * \text{Cos}(X) * X^2$ $F(x)$
 - Nombre de la función: 2SinCos
 - Expresión: $\text{Sin}(X) * \text{Cos}(X) * X^2$ y adicionalmente,

- Tipo de función: $F(x)$, una variable; también existen $F(x, y)$, de dos variables.
2. Selecciónese la función que se deseé traer,
 3. Clic en comando <Usar Función Seleccionada>. La expresión aparecerá en el área de edición de $F(X)$.

Notas:

- Al transferir una función, ya sea de usuario o preconstruída, los valores que existan para **X** y **resultado** son borrados para evitar que la función se valúe innecesariamente.
- Se recomienda usar en este módulo las funciones de usuario del tipo $F(x)$; si se opta por valuar las funciones $F(x, y)$, se convertirán las **Y's** a **X's** y se procederá a procesarlas en términos de **X**.

1c) Traer Funciones Preconstruídas. EmCalc tiene un archivo de funciones preconstruídas, las cuales pueden ser accedadas y agregadas al final de la expresión existente en el área de edición $F(X)$.

Para traer una función Preconstruída hágase lo siguiente:

Preconstruídas	Función
Cosecante Cosec(x) =	$1 / \text{Sin}(x)$
Cosecante Hiperbolica HCosec(x) =	$2 / (\text{Exp}(x) - \text{Exp}(-x))$
Coseno Cos(x) =	$\text{Cos}(x)$
Coseno Hiperbolico HCos(x) =	$(\text{Exp}(x) + \text{Exp}(-x)) / 2$

Al oprimir la caja-combo "Preconstruídas" aparece una lista de las funciones de las cuáles se proporciona su expresión para ser agregada a la expresión existente en el área de edición de $F(X)$.

1. Clic en la caja "Preconstruídas".
2. Al aparecer la lista de funciones, selecciónese la que se deseé transferir.
3. Clic en comando <Adicionar Función al final de $F(X)$ >. La expresión aparecerá en el área de edición de $F(X)$, a continuación de la expresión que ya exista en esa área.
4. El usuario debe poner el operador adecuado para que la función resultante sea correcta.

Ejemplo de expresión usando una Función Preconstruída Derivada: $3x \text{Cosecante}(x)$

- $\text{Cosecante}(X) = 1/\text{Sin}(x)$;
- Entonces la expresión debe escribirse: $3*X*1/\text{Sin}(x)$
- Para evitar confusión se usan paréntesis $3*X*(1/\text{Sin}(X))$

Guardar Funciones y Borrar Funciones Archivadas.

La figura muestra las secciones para:

- Guardar funciones. Sustitúyase **Nombre función** por el nombre con que se desea guardar la función.
- Borrar funciones. Clic en caja-combo permite ver las funciones a borrar.

Guardar Funciones. La expresión que exista para $F(X)$ se archiva permanentemente como una **Función de Usuario**, con un nombre proporcionado por el usuario, para poder ser accesada posteriormente, indicándose que es del tipo $F(x)$, lo cual significa que está escrita en términos de X .

1. En la caja-texto Asígnese un nombre a la función. No hay restricción en caracteres a usar, pero si existe un límite de 20 caracteres para el nombre de la función.
2. Clic en el botón <Guardar Función>.

Notas:

1. $F(X)$ puede guardarse aún estando incorrecta. Esta facilidad se provee para que el usuario pueda posteriormente arreglar la expresión.
2. Si el nombre ya existe en el archivo de funciones, el sistema **NO** guardará la función. En este caso hay que asignar otro nombre a la función y hacer clic en el botón <Guardar Función>, nuevamente.

Borrado de funciones. Borrado permanente de funciones de usuario guardadas en el archivo. La función borrada ya no será posible accesarla.

1. Clic en caja-combo "Función a Borrar", selecciónese de ahí la función que se desea borrar.
2. Clic en <Borrar Función Seleccionada>.

<Borrar todo en la pantalla>. Al oprimir este comando se borra la información desplegada en la pantalla, pero no la información que ha sido almacenada.

Las Funciones Preconstruidas que EmCalc proporciona no se pueden borrar. Para graficarlas, se recomienda guardarlas como funciones de usuario y posteriormente utilizarlas en los módulos de graficación.

Mensajes de Error:

1. De excederse el límite de 255 caracteres, EmCalc emitirá un mensaje desplegando la longitud de la expresión.
 - El usuario debe corregir la expresión para que no exceda este límite.
2. División por Cero y Overflow. En cada caso, se pide al usuario que se cerciore que la expresión sea válida para los puntos dados.
3. Otro tipo de error. Los errores relativos a "Posible Error de Sintaxis" requieren que el usuario corrija la expresión. Usese la expresión donde se reemplaza X por su valor como auxiliar para encontrar el error.

Ejemplo: Para mostrar el uso de esta pantalla se valúa la función: $3x^2 + \text{Log}(x)$ valuada en $x = 7.7$; aquí la expresión debe tener todos los operadores, en este caso debe ser: $3*x^2 + \text{Log}(x)$. Nótese además que es la misma función utilizada en el módulo anterior **Operaciones y Funciones Básicas**.

La figura muestra la función: $3x^2 + \text{Log}(x)$, para $x = 7.7$. Las líneas indican las instrucciones a seguir.

Valuación de Funciones

Traer funciones.

Archivadas

Preconstruidas

Añadir Función al Final de F(x)

Guardar Funciones y Borrar Funciones Archivadas

Nombre (Máx.20 caract.)

Guardar Función

Borrar Función Seleccionada

1) Teclee f(x). 2) Asigne valores a x. 3) Enter, Tab o Click en Resultado u otro campo.

F(x) = $3x^2 + \text{Log}(x)$ X = 7.7 Resultado: 179.91122032006

Sustituyendo X por su valor F(x)=

$3*(7.7)^2 + \text{LOG}((7.7))$

BORRAR TODO EN LA PANTALLA

Generación de Series y Graficación de funciones.

Manualmente Automáticamente Múltiples Funciones

Generación de Series y Graficación de Funciones. Usando estos comandos se generan series de puntos a graficar, manual o automáticamente, y se crean gráficas de una, o varias funciones a la vez usando un mismo rango de puntos.

1. **Manualmente.** El usuario captura en una tabla los puntos X a graficar.
2. **Automáticamente.** El usuario indica un intervalo a graficar.
3. **Múltiples Funciones.** Permite graficar hasta 5 funciones, sobre los mismos puntos o sobre el mismo intervalo.

2.6.1 Módulo 2.0 Consideraciones generales para graficación de funciones.

1) Especificación de la o las funciones a graficar:

Las pantallas de graficación ofrecen la facilidad de traer una función desde el archivo de funciones de usuario, o de especificar una nueva tecleándola completamente. Para las pantallas de graficación, la función especificada en la pantalla de **Valuación de Funciones**, se transfiere automáticamente, pero se puede reemplazar.

Si se desea traer una función de usuario, hágase lo siguiente:

Función	Click	Expresión	Tipo
2SinCos		$\sin(x) * \cos(x) * x^2$	F(x)
5datos		$57.16409 - 2.81901 * X + .04221 * X^2$	F(x)
Curva		$0.5 * \sin(3 * x) + 0.3 * \cos(1.8 * x) + 0.4 * \cos(0.9 * x + 0.3) + 1.8$	F(x)
Curva1		$57.82738 - 2.92438 * X + .04566 * X^2$	F(x)

1. Clic en la caja-combo "Función". Aparecerá una lista de las funciones que el usuario ha guardado.

- Ejemplo:

2SinCos	Sin(X) * Cos(X) * X^2	F(x)
---------	-----------------------	------

 - Nombre: 2SinCos
 - Expresión: Sin(X) * Cos(X) * X^2 y adicionalmente,
 - Tipo: F(x)

2. Selecciónese la función que se desee graficar,

3. clic en comando <Pasara F(X)>. La expresión aparecerá en el área de F(X) de la pantalla.

Notas:

- Usense en esta pantalla las funciones del tipo F(x); Si se opta por emplear las funciones F(x,y) aquí, se convertirán las Y's a X's y se procederá a procesarlas en términos de X, lo cuál podría ser erróneo, se deja a consideración del usuario su empleo.
- Las funciones preconstruídas de EmCalc no están disponibles en esta pantalla a menos que el usuario ya las haya integrado al archivo de usuario.
- Si se opta por editar o teclear una expresión "desde cero", EmCalc no podrá detectar si hay errores por el "Control de Errores al Valuar la Función" que se especifica en la pantalla, de modo que el usuario debe estar atento a esta situación.

2) Gráficas generadas.

Las gráficas que se producen son todas del tipo "puntos" unidas por líneas. Se especifican en el eje cartesiano donde X es la variable independiente y F(X) es la función subordinada. En general F(X) se referencia como "Y", resultando entonces: $Y = F(X)$.

- La escala en ambos ejes es determinada automáticamente.
- Los colores de las líneas y puntos son predeterminados.

- <Valuar la Función y Graficar> este comando hace ambas tareas a la vez.

Comando <Borrar Valores de X>. Borra la tabla de datos dados. Cualquiera de las dos cajas de este comando borra los valores de X y de F(X).

Notas:

- Al <Valuar F(X) en los puntos dados> se produce una copia de X, la cual aparece en una tabla a la derecha de la pantalla, junto con el valor de F(X): Esta copia de X está ordenada de menor a mayor.
- Se pueden insertar valores de X intermedios a los ya dados, sólo al final de la tabla (Con el título "X"). Al valuar o al graficar F(X), los puntos serán ordenados de menor a mayor.
- Cada vez que se hacen cambios a los valores de X y/o se agregan datos a la tabla, es necesario <Valuar F(X) en los puntos Dados> o <Valuar y Graficar Función> para tener los datos correctos en <Copia de X, Y> y en la gráfica.

Ejemplo. Se desea hacer una grafica con los siguientes datos:

Función: $\text{Sin}(x) * \text{Cos}(x) * X^2$
 Control de Errores:
 División por cero: 5
 Overflow: 7
 Otro Error: 10
 Puntos: (-3,-2,-1,0,1,2)

La figura muestra la pantalla, preparada para el ejemplo que se menciona arriba.

F FormMain1 Serie Form

Función: $\text{Sin}(x) * \text{Cos}(x) * x^2$ **Pasar a F(X)** **Graficación de Funciones**

F(X): $\text{Sin}(x) * \text{Cos}(x) * x^2$ Borrar Expresión de F(X)

Control de Errores al Valuar la función:
 Si hay división por cero, hacer F(X) = 5
 Si hay Overflow, hacer F(X) = 7
 Si hay Otro Error, hacer F(X) = 10

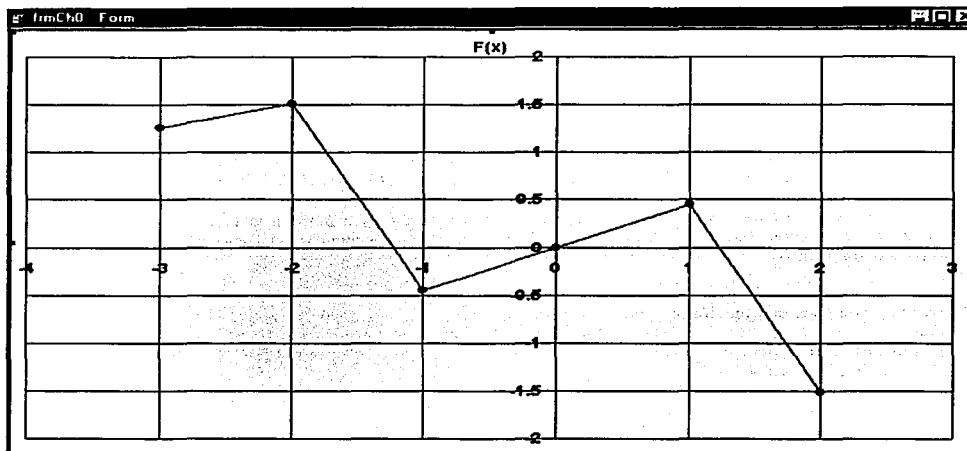
Teclée aquí los valores de X que Desea Valuar Graficar

X	Y
-3	1.25736974189517
-2	1.51360499061586
-1	-0.454648713412841
0	0
1	0.454648713412841
2	-1.51360499061586
0	0

Record: 14 of 6

Record: 14 of 6

La gráfica generada se muestra en la ilustración de abajo. El tipo de gráfica está predeterminado a puntos unidos por líneas, el título y el formato de los ejes también están predeterminados.



Gráfica producida con la función $F(X) = \text{Sin}(X) \cdot \text{Cos}(X) \cdot X^2$.

Notas:

- Se puede especificar una nueva función o nuevos datos en cualquier momento.
- Para generar otra gráfica con puntos generados manualmente, sólo se requiere cerrar la pantalla de la gráfica y poner los nuevos datos.
- Se recomienda usar el comando <Borrar Valores de X> cuando se especifiquen nuevos datos, en lugar de sobrescribir sobre los anteriores (se deja a elección del usuario).
- Para un “Borrado rápido” de todo lo que hay en la pantalla, se recomienda cerrar la pantalla y entrar a ella nuevamente desde la pantalla de Valuación de funciones.
- Para un “Borrado rápido” de todo lo que hay en la pantalla, se recomienda cerrar la pantalla y entrar a ella nuevamente desde la pantalla de Valuación de funciones.

2.6.3 Módulo 2.2 Generación de Puntos a Graficar automáticamente.

El usuario indica un intervalo en el eje X para valuar y graficar a $F(X)$. A su elección debe proveer el número de puntos X a generar, o el espaciado entre los puntos.

Pantalla empleada en su estado inicial. La función se transfiere de la pantalla anterior.

EmCalc: Función Form

Función: **Graficación de Funciones**

f(X):
Sin(x) * Cos(x) ^ 2

La función es transferida automáticamente de la pantalla anterior, pero se puede modificar ó traer de un archivo.

Control de Errores al Valuar la Función:

Si hay división por cero, hacer F(X) =

Si hay Overflow, hacer F(X) =

Si hay Otro Error, hacer F(X) =

Rango de X a utilizar para la Serie Gráfica:

Inicio:

Fin:

Seleccione una de las opciones:

No de puntos: Incremento:

Dar Valores de X

Los puntos X's se pueden modificar por el usuario después de generados, no así "Y".

Nótese que EmCalc asigna datos por default a las cajas-texto. Todas ellas se pueden modificar.

Cada opción "No de Puntos" ó "Incremento" tiene un par de comandos.

Record: 100 1

Pasos para generar una gráfica:

1. Especifíquese la función que se desea valuar y graficar. La función $F(X)$ de la pantalla anterior (la que da origen a esta pantalla) es transferida automáticamente, sin embargo, $F(X)$ se puede editar, teclear enteramente, o traer de un archivo de funciones de usuario (Ver inicio sección para más información).
2. Especifíquense los valores para control de errores. (Ver inicio sección para más información).
 - División por cero, úsese default (cero) o especifíquese otro valor.
 - Overflow, úsese default (cero) o especifíquese otro valor.
 - Otro Error, úsese default (cero) o especifíquese otro valor.
3. Especifíquese el rango de X a graficar. Se debe especificar el inicio y el fin del rango Aparece por default (0,1). El valor de "INICIO" debe ser menor que el de "FIN", de otra forma EmCalc no graficará.
4. Selecciónese "Número de Puntos" o "Incremento" entre los puntos.
 - "Número de Puntos" se dividirá el rango dado en el número de puntos especificado, generando puntos equi-espaciados.
 - "Incremento" entre los puntos, se generan los puntos a graficar a partir del límite inferior, agregando el incremento hasta agotar el rango dado.
 - En caso de valores inválidos para "No. de puntos" o "incremento", EmCalc generará 10 puntos.

5. Dependiendo de la opción tomada en (4) "Número de Puntos" o "Incremento", se utiliza una de las opciones debajo de esa opción, ambas son idénticas, excepto por el parámetro que se utiliza:
- Se puede <Crear Serie Solamente> oprimiendo este comando. Si después de crear la tabla de puntos (X,Y) se desea ver la gráfica, hágase clic en <Ver Gráfica de la Serie>.
 - <Cera Serie y Graficar > este comando crea la tabla y despliega la gráfica, a la vez.

Comando <Borrar Valores de X>. Borra la tabla de puntos generados (X,Y).

Notas:

- Al <Crear Serie Solamente> se producen los puntos X y F(X), referenciados en la tabla como (X,Y).
- Los puntos X generados, pueden ser editados, al cambiar el valor de X, automáticamente se recalcula F(X).
- Se pueden borrar renglones (Puntos(X,Y)) de la tabla creada.
- Al usar la opción "Número de Puntos" EmCalc genera el "Incremento" para que el usuario tenga el dato, conversamente, al utilizar "Incremento" EmCalc calcula "Número de Puntos" para información.
- El máximo número de puntos a crear es 1000. De excederse en cualquier opción ("Incremento" o "Número de Puntos"), el total es ajustado a 1000.
- Se pueden cambiar en cualquier momento: la función, valores para control de errores, rango a utilizar, "Número de Puntos" o "Incremento". De hacer algún cambio, hay que correr nuevamente los comandos que generan los puntos(X,Y) nuevamente.

Ejemplo. Se desea hacer una grafica con los siguientes datos:

Función: $\text{Sin}(x) * \text{Cos}(x) * X^2$
 Control de Errores: División por cero: 5
 Overflow: 7
 Otro Error: 10
 Rango: [-3, 2]
 Número de Puntos: 150

EmCalc2Serie Form

Función: **Función** **Pasar a F00** **Graficación de Funciones** **5**

f(x):
 $\text{Sin}(x) * \text{Cos}(x) * X^2$

- 1) Especificar la función a evaluar/graficar.
- 2) Especificar valores para control de errores.
- 3) Especificar rango a graficar
- 4) Especificar "No. de Puntos" ó "Incremento".
- 5) <Crear Serie Solamente> y <Ver Gráfica de la Serie> ó <Crear serie y Graficar>.

Control de Errores al Valorar la Función:

Si hay división por cero, hacer F(X) =
 Si hay Overflow, hacer F(X) =
 Si hay Otro Error, hacer F(X) =

Rango de X a utilizar para la Serie Gráfica:

Inicio:
 Fin:

Seleccione una de las opciones:

No de puntos: Incremento:

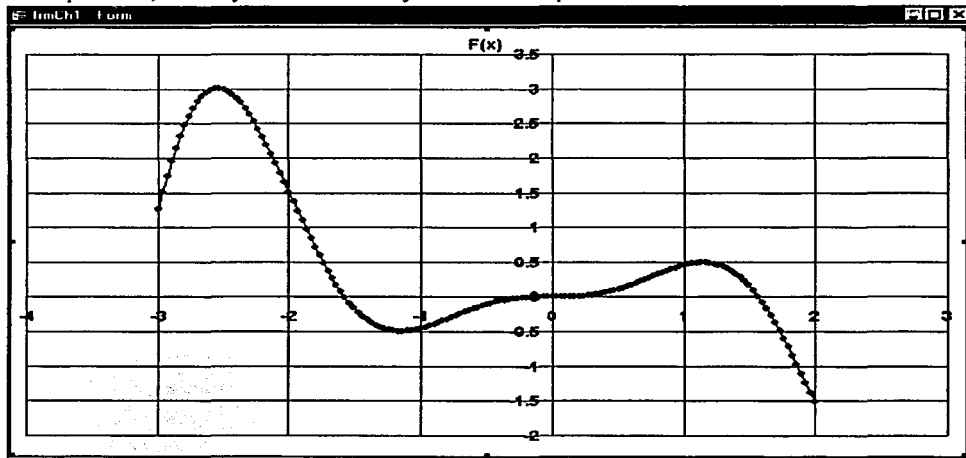
Borrar Valores de X **Ver Gráfica de la Serie**

X	Y
-3	1.25736974189517
-2.9666667	1.50832893019508
-2.9333333	1.74056961472566
-2.9	1.95365216443485
-2.8666667	2.14714519875159
-2.8333333	2.32082349811259
-2.8	2.4745652204595
-2.7666667	2.60836845835703
-2.7333333	2.72234717460418
-2.7	2.81672655714157
-2.6666667	2.89183783668472
-2.6333333	2.94811261262524
-2.6	2.98607673633412
-2.5666667	3.00634379706188
-2.5333333	3.00960626815789
-2.5	2.9963835832231
-2.4666667	2.96826862445796
-2.4333333	2.92539239740745
-2.4	2.86895407344722
-2.3666667	2.80000000000000
-2.3333333	2.71000000000000

Record: 14 of 151

La figura muestra el ejemplo de la gráfica de la misma función que la sección anterior, utilizando 150 puntos.

La gráfica generada se muestra en la ilustración de abajo. El tipo de gráfica está predeterminado a puntos unidos por líneas, el título y el formato de los ejes también están predeterminados.



La gráfica generada con 150 puntos en el mismo rango que la sección anterior. $F(x) = \sin(x) \cdot \cos(x) \cdot x^2$

2.6.4 Módulo 2.3 Graficación de Múltiples Funciones.

Permite graficar de 1 a 5 funciones a la vez, todas sobre un mismo intervalo en el eje X. Los puntos X se generan usando cualquiera de las opciones: manual o automática recién descritas en las secciones anteriores. Esta pantalla no permite ver la valuación de las funciones. Las funciones F1(X), F2(X), F3(X), F4(X), F5(X) se pueden teclear manualmente o se pueden acceder desde el archivo de Funciones de Usuario.

Pantalla empleada, en su estado inicial.

Pasos para generar una gráfica:

- Especifíquense las funciones a graficar. Las funciones F1(X) a F5(X) se pueden traer de un archivo de funciones de usuario, editar o teclear enteramente (Ver inicio del capítulo para más información).
 - Para traer una función hágase lo siguiente:
 - Clic en la caja "Función". Aparecerá una lista de las funciones que el usuario ha guardado.
 - Seleccíonese la función deseada,
 - Clic en caja-combo <Pasar a Fi(X)>, donde $i = 1, \dots, 5$
 - Clic en la "Varita Mágica". La expresión aparecerá en Fi(X) seleccionada, $i = 1, \dots, 5$.
- Especifíquense los valores para control de errores. (Ver inicio sección para más información).
 - División por cero, úsese default (cero) o especifíquese otro valor.
 - Overflow, úsese default (cero) o especifíquese otro valor.
 - Otro Error, úsese default (cero) o especifíquese otro valor.
- Especifíquese el rango de X a graficar. Se debe especificar el inicio y el fin del rango. Aparece por default (0,1). El valor de "INICIO" debe ser menor que el de "FIN", de otra forma EmCalc no graficará.

4. Seleccione una de las tres opciones que se dan

- “Número de Puntos” se dividirá el rango dado en el número de puntos especificado, generando puntos equi-espaciados, después
 - <Crear serie> genera los puntos X, llenando la tabla que aparece bajo la opción 3.
- “Incremento” entre los puntos, se generan los puntos a graficar a partir del límite inferior, agregando el incremento hasta agotar el rango dado, después
 - <Crear serie> genera los puntos X, llenando la tabla que aparece bajo la opción 3.
- En caso de valores inválidos para “No. de puntos” o “incremento”, EmCalc generará 10 puntos.
- “Manualmente”. Esta opción permite teclear los puntos que el usuario desea graficar.

5. <Ver Gráfica de las Funciones>. Este comando despliega la gráfica de las funciones especificadas.

Comando <Borrar Valores de X>. Borra la tabla de puntos generados (X,Y).

Ejemplo. Se desea hacer una grafica con los siguientes datos:

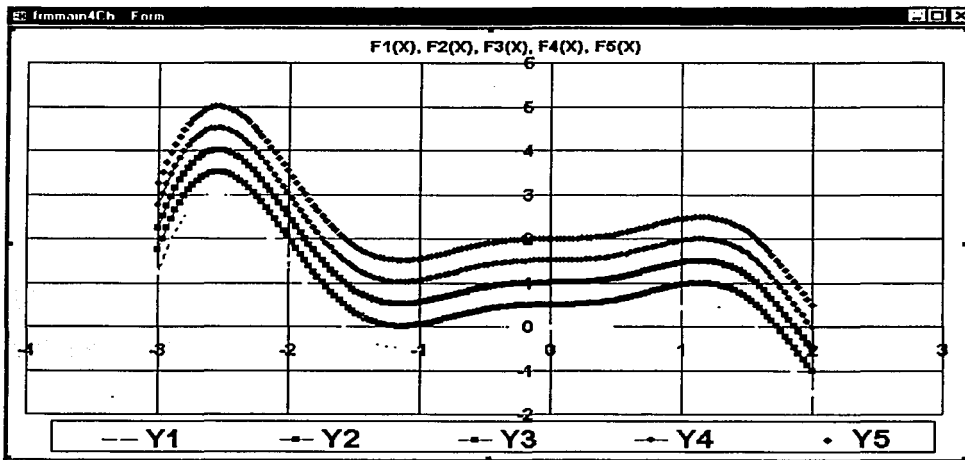
Función 1:	$\text{Sin}(x) * \text{Cos}(x) * X^2$
Función 2:	$\text{Sin}(x) * \text{Cos}(x) * X^2 + 0.5$
Función 3:	$\text{Sin}(x) * \text{Cos}(x) * X^2 + 1.0$
Función 4:	$\text{Sin}(x) * \text{Cos}(x) * X^2 + 1.5$
Función 5:	$\text{Sin}(x) * \text{Cos}(x) * X^2 + 2.0$
Control de Errores.	División por cero: 5
	Overflow: 7
	Otro Error: 10
Rango:	[-3, 2]
Número de Puntos:	152

La ilustración muestra la pantalla con 5 funciones a graficar. Nótese que son iguales, excepto por el 0.5 de diferencia en cada una. Se muestra también la caja-combo para “Pasar a” una función de F1(X) a F5(X).

Procedimiento para especificar las funciones:

- Se transfirió la misma función a cada una de $F1(X)$ a $F5(X)$.
- Se editó cada $F_i(X)$ agregándole 0.5, 1.0, etc a cada una.

La gráfica generada se muestra en la ilustración de abajo. El tipo de gráfica está predeterminado a puntos unidos por líneas, el título, el formato de los ejes también están predeterminados. Las leyendas también están predeterminadas a Y1, Y2, Y3, Y4 y Y5, así como el indicador a usar en el punto (rombo, punto, triángulo, etc).



La gráfica muestra las 5 funciones, todas son idénticas, excepto por el 0.5 que las desplaza, para diferenciarlas.

Notas:

- Al <Crear Serie> se producen los puntos X y se despliegan en la tabla para X.
- Los puntos X generados, pueden ser editados, al cambiar el valor de X, automáticamente se recalculan las funciones $F1(X)$ a $F5(X)$, aunque estas, no son desplegadas.
- Se pueden borrar renglones (puntos X) de la tabla creada.
- Al usar la opción "Número de Puntos" EmCalc genera el "Incremento" para que el usuario tenga el dato, conversamente, al utilizar "Incremento" EmCalc calcula "Número de Puntos" para información.
- El máximo número de puntos a crear es 1000. De excederse en cualquier opción ("Incremento" o "Número de Puntos"), el total es ajustado a 1000.
- Se pueden cambiar en cualquier momento: la función, valores para control de errores, rango a utilizar, "Número de Puntos" o "Incremento". De hacer algún cambio, hay que correr nuevamente los comandos que crean la serie.
- Si se especifican menos de 5 funciones, aparecerá una línea a lo largo del eje X, las funciones no especificadas se hacen $F_i(X) = 0$.

2.7 Módulo 3. Valuación de Funciones F(X, Y).

Nota: debido a la similitud que presentan este módulo y el de Valuación de Funciones F(X), se especificarán aquí las generalidades de éste módulo y se remite al usuario a la lectura del módulo "Valuación de Funciones F(X) para tener mayor información.

Esta Sección permite:

- Escribir expresiones en términos de X y Y, y valuarlas.
- Guardar permanentemente las funciones (expresiones) creadas, asignándoles un nombre.
- Traer o accesar una función archivada previamente, para valuarla o editarla.
- Traer o accesar funciones preconstruídas para editarlas o combinarlas con Funciones de Usuario.
- Borrar permanentemente funciones que han sido guardadas en el archivo.

Pantalla de este módulo en su forma inicial, excepto por la indicación "Área de Edición de F(X, Y)".

Formulador de Form

Valuación de Funciones F(X, Y)

Instrucciones/Ayuda

Traer Funciones.

Archivadas:

Preconstruídas:

Guardar Funciones y Borrar Funciones Archivadas.

Nombre (Máx.20 caract.):

Función a Borrar:

1) Teclée F(X,Y). 2) Asigne valores a X y Y. 3) <Click> en <Resultado>.

X = Y = Resultado:

F(X,Y) =

Área de Edición de F(X, Y)

Sustituyendo valores de X y Y; F(X,Y) =

BORRAR TODO EN LA PANTALLA

Valuación de una Función. Cómo empezar

En el área de edición de F(X, Y), constrúyase una expresión algebraica en términos de X y Y. Se puede escribir la expresión en términos de X solamente (F(X), o de Y solamente (F(Y), sin embargo, si esta expresión es guardada será categorizada como F(X, Y), lo cual puede crear confusión al verla y no encontrar ambos términos; se deja a consideración del usuario el manejo de esta situación.

Para obtener la valuación de una función síganse los dos siguientes pasos:

1. A elección del usuario, háganse una de las tres siguientes opciones:
 - Construir una función tecleando una expresión en el área de edición de F(X, Y);

- Accesar una Función de Usuario. Se pueden accesar funciones categorizadas como $F(x, y)$ y también como $F(x)$; si existe una expresión para $F(X, Y)$, la nueva función reemplazará la expresión existente. Esta nueva expresión se puede editar.
 - Accesar una Función Preconstruida (FP), si existe una expresión para $F(X, Y)$, FP se agregará al final de la expresión existente. La expresión resultante se puede editar. Recuérdese que FP está en términos de X , debe editarse para cambiarse a Y 's en caso necesario.
2. Valúese la función tecleando un valor para X y Y .
- Clic en la caja-comando "Resultado".
 - Para valuar en otro punto, reemplácense los valores de X y/o de Y , clic en "Resultado".

Ejemplo de expresión $F(X, Y)$: $X * \sin(x) + Y * \cos(y)$

La pantalla muestra la expresión $F(X, Y) = X * \sin(x) + Y * \cos(y)$, valuada en los puntos $X = 1$, $Y = 2$.

Notas:

- El tamaño máximo de una expresión para una función $F(X, Y)$ al ser reemplazada por los valores de X y de Y debe ser de un máximo de 255 caracteres.

<Borrar todo en la pantalla>. Al usar este comando, se borra la información desplegada en la pantalla, pero no la información que ha sido almacenada.

Comandos Auxiliares.

Guardar Funciones. Asígnese un nombre y oprímase <Guardar Función>. Al guardar una función desde este módulo serán categorizadas como $F(x, y)$ aún cuando sólo dependan de una variable: X o Y , o sean constantes.

Borrar Funciones. Se selecciona una función del archivo de funciones de usuario haciendo clic en la caja-combo "Función a borrar" y se borran al hacer clic en el comando <Borrar Función Seleccionada>. Se pueden borrar funciones categorizadas como $F(X)$ o como $F(x, y)$.

2.8 Módulo 4. Funciones Estadísticas.

Esta pantalla da acceso a los siguientes módulos que generan:

- Análisis de observaciones que el usuario proporciona:
 - Análisis de datos. Estadística descriptiva.
 - Análisis de Regresión para una variable. Ajuste lineal y ajuste cuadrático.
- Desplegado e impresión de tablas estadísticas:
 - Normal. Tabla de distribución de la función Normal(0,1).
 - T de Student. Muestra puntos porcentuales.
 - Ji-Cuadrada Puntos porcentuales.
 - F 1%. Puntos porcentuales de la distribución F, nivel de significancia del 1%.
 - F 5%. Puntos porcentuales de la distribución F, nivel de significancia del 5%.
 - Números aleatorios. Generador de tablas de números aleatorios acorde a parámetros que el usuario especifica.

Pantalla de Funciones Estadísticas.

Funciones Estadísticas Form

Funciones Estadísticas

Análisis de Datos

Análisis de Regresión

TABLAS

Normal	t Student	Ji ²
F 1%	F 5%	N Aleatorios

IMPRESION DE TABLAS

Normal	t Student	Ji ²
F 1%	F 5%	N Aleatorios

Acceso a las pantallas.

Un clic en la caja-comando deseada dará acceso a la pantalla del módulo o desplegará el reporte de la tabla solicitada en formato de pre-impresión.

2.8.1 Módulo 4.1 Análisis de datos

Correspondiente al área de Estadística Descriptiva, en este módulo se analiza un grupo de datos y se genera su información descriptiva, entre otros: media, varianza, desviación estándar, rango, etc. Se proporciona también un histograma de los datos.

El grupo de datos que el usuario proporciona puede representar una Población o una muestra, por lo cuál se generan ambos tipos de información.

Pantalla del módulo mostrando algunas indicaciones para esta documentación.

Análisis de Datos Instrucciones/Ayuda

Borrar Datos
1) Teclar Datos, ó, traer de archivo.

Datos

El usuario especifica los datos a analizar en la tabla que se proporciona.

Al oprimir <enter> se genera un nuevo renglón.

El límite de datos a analizar está limitado al sistema, 500,000 es un número aún aceptable en la mayoría de las computadoras.

Record: 1 of 1

Archivo:
Archivo:
Guardar Datos en Archivo

2) Generar Información, 3) Ver Histograma.

Generar Información **Hacer Histograma**

Al oprimir <Generar Información>, se analizan los datos y aparece en esta área la información correspondiente a tales cálculos.

Al oprimir <Hacer Histograma> se despliega una nueva pantalla con la gráfica de los datos.

Por dónde empezar

EmCalc analizará los datos que se introduzcan en la tabla de captura que se proporciona. Los siguientes pasos permiten llevar a cabo tal tarea.

1. **Introduzcanse los datos en la tabla que se proporciona.** Manualmente o traídos de un archivo.
 - Capturados (manualmente). Estos datos son la información que será analizada en este módulo. Puede ser teclado en cualquier orden, EmCalc los ordena para analizarlos y para graficarlos.
 - Transferidos desde un archivo creado previamente. (Puede importarse de Excel, ver Utilerías).
 - Hay que cerciorarse que el área de captura de datos esté vacía, de otro modo los datos traídos del archivo se adicionarán a los existentes.

Archivo:

- USR1235468.*
- USR1235468
- USR1235468
- USRAleatorios
- USRCuatro50

Click

- o Clic en caja-combo **Archivo**, junto al comando <Traer Archivo>. De la lista que aparezca, selecciónese el archivo que se desea acceder.
- o Al hacer clic en la caja-comando

<Traer Archivo> se transferirá una copia del archivo hacia el área de Datos. Ahí los datos pueden ser modificados; en esta área además, se pueden agregar o borrar datos. Los datos en el archivo permanecerán intactos.

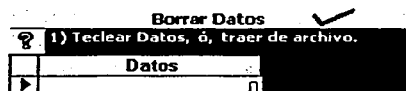
2. **Análisis de los datos.** Use el comando <Generar Información>, el cuál hace que los datos se analicen y se desplieguen en una tabla en la pantalla, indicando las características encontradas.

Nota: el comando <Generar Información> debe ser utilizado cada vez que se modifiquen los datos, ya sea al editar los valores, o al agregar, o borrar datos é introducir nuevos datos.

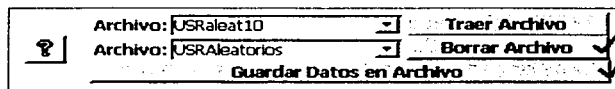
3. **Creación del histograma.** El comando <Hacer Histograma> produce un histograma que permite ver la frecuencia absoluta de los datos.

Nota: <Generar Información> y <Hacer Histograma> son comandos independientes, esto significa que no se requiere correr los comandos en el orden dado arriba. El único requerimiento es que haya datos a analizar.

Los siguientes comandos proporcionan ayuda en el manejo de la información:



- <Borrar Datos>. Borra la tabla de datos que se esté analizando. Sólo borra los datos de la pantalla. Si estos provienen de un archivo, los datos en el archivo quedan intactos.



- <Borrar Archivo>. Borra permanentemente el archivo de datos.
 - o Clic en Archivo. De la lista que aparezca, selecciónese el

- o que se desea borrar.
- o Al hacer clic en <Borrar Archivo> El archivo de datos será borrado permanentemente.
- <Guardar Datos en Archivo>. Los datos Se pueden guardar permanentemente en un archivo.
- o Al oprimir el comando aparece una pantalla preguntando el nombre que se le desea dar al archivo:
 - o Máxima longitud del nombre: 15 caracteres. De excederse, no se guarda el archivo.
 - o EmCalc le agrega el prefijo "USR" al nombre (no cuenta para la longitud de 15 caracteres).
 - o No existe restricción en caracteres a usar para el nombre.

Si se guarda un archivo con un nombre de archivo ya existente, el nuevo archivo sustituirá al viejo perdiéndose los datos del archivo viejo. Una forma de comprobar los nombres existentes de archivos es haciendo clic en cualquiera de las dos cajas-combo de archivos de los comandos <Traer Archivo> y <Borrar Archivo>.

Ejemplo: supóngase que se desea analizar el peso de 75 personas (niños incluidos).

Las mediciones fueron (clasificadas por intervalos):

Peso (en libras): 80 90 102 112 118 130 141 149 160 170 181
 No. de personas: 1 1 2 6 10 14 14 11 8 4 4

Se muestran la pantalla y el histograma que se genera:

Análisis de Datos

Instrucciones/Ayuda

Borrar Datos

1) Teclar Datos, ó, traer de archivo.

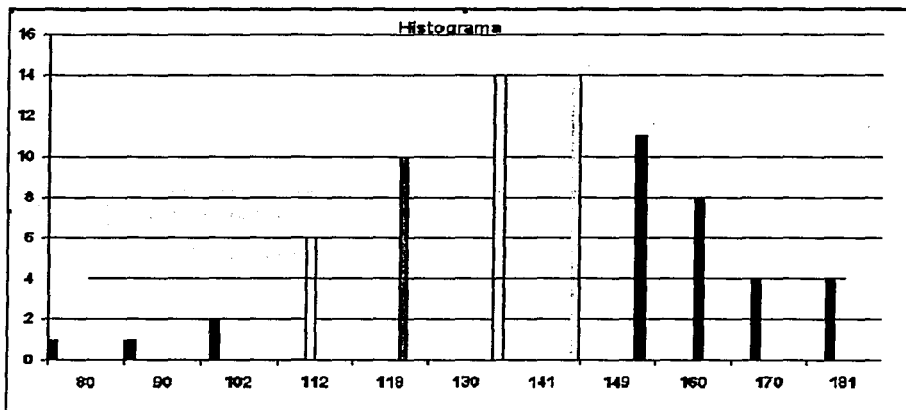
Datos

80	
90	
102	
102	
112	Se
112	especificar
112	las 75
112	observacio
112	nes, una
112	por una.
118	
118	
118	
118	
118	
118	
118	
118	
118	
118	

2) Generar Información. 3) Ver Histograma.

Generar Información	Hacer Histograma
Número de Observaciones	75
Suma	10343
Media	137.906666666667
Varianza Poblacional (N)	453.764622222222
varianza Muestral (n)	459.896576576577
Desv. Standard Pob. (N)	21.3017516233342
Desv. Standard muestra (n)	21.4451993829989
Mínimo	80
Máximo	181
Rango	101
Mediana	141
Mediana 1	118
Mediana 3	149
Rango Intercuartil	31
	0

Record: 14 | 74 | 11 | 181 of: 75



2.8.2 Módulo 4.2 Análisis de Regresión.

Permite crear modelos de ajuste para una variable independiente (X) y una dependiente (Y), por medio de una recta ($Y = a + bX$) o de un modelo cuadrático ($Y = a + bX + cX^2$), por el método de mínimos cuadrados. Se provee adicionalmente la facilidad de hacer transformaciones a los valores de X y/ o de Y independientemente, resultando entonces el modelo lineal $\text{Transf}(Y) = a' + b' \text{Transf}(X)$; y el modelo cuadrático $\text{Transf}(Y) = a' + b' \text{Transf}(X) + c'(\text{Transf}(X))^2$.

Análisis de Regresión

Borrar Todo X Y Transf X Transf Y Yajuste Residual

1) Tecléar valores para X y Y, ó, traer datos de un archivo.

Archivo de Datos Tracer Archivo Guardar Datos en Archivo

2) Opcional: Transformaciones para X y Y.

Función Pasar a: F(X) F(Y) Borrar F(X) F(Y) F(X) F(Y)

Div. por cero? F(X)= 0 F(Y)= 0

Overflow? F(X)= 0 F(Y)= 0

Otro Error? F(X)= 0 F(Y)= 0

Record: 1 of 1

3) Calcular Transformaciones y Ajuste.

Transformaciones Para X y Y

Desplegar cálculos del ajuste.

Ajuste Recta Ajuste Cuadrático

Gráfica de Dispersión (X y Y)

Gráfica de Transformaciones de X y Y

Gráfica Transf. X y Y con Línea de Ajuste

Ajuste Guardar

Area de captura de (X, Y).
TransfX, TransfY, Yajuste y Residual son generados mediante comandos de esta pantalla.

La cantidad de datos está limitada sólo por el sistema: 500,000 es un número aceptable.

La información de la regresión se despliega en esta área.
<Ajuste Recta> y <Ajuste Cuadrático> son los comandos que generan la información.

"Desplegar cálculos del ajuste" permite ver más detalle de los cálculos realizados.

Pasos a seguir para crear un modelo de ajuste.

- Tecléense los pares de valores para X y Y, o transfíranse los valores de un archivo de datos. Los datos a analizar deben aparecer en la tabla para captura de datos con los encabezados X, Y en las columnas a usar. Pueden ser capturados manualmente o traídos de un archivo de datos (También pueden provenir de Excel, véase la pantalla de Utilerías). En caso de duda, véase Manejo del Area de Datos, abajo.
- Opcional: se pueden especificar transformaciones para X y/ o Y. Se pueden especificar expresiones para transformar los datos de X y/ o de Y. En el caso de una transformación para Y, la expresión debe hacerse en términos de Y.
 - Del archivo de funciones de usuario se pueden transferir expresiones para las transformaciones a los campos F(X) y/ o F(Y). En caso que sea para F(Y), la expresión se transformará automáticamente a términos de Y.
 - Se provee la facilidad de valuar F(X) y F(Y) a cierto valor en caso que se ocasione un error al sustituir los valores de X y/ o de Y. EmCalc valúa a Cero por default al haber División por cero, Overflow, ú Otro Error. Véase la sección de **Valuación de Funciones** para mayor información.
 - Si X y/ o Y no requieren transformaciones, F(X) o F(Y) se deja(n) en blanco. EmCalc utilizará los valores de X y Y como sus propias transformaciones.

- En caso que se use una expresión nueva, se recomienda probarla en el área de Valuación de Funciones para verificar que efectivamente sea correcta, de ahí se puede guardar con un nombre y posteriormente ser accedida en esta sección de transformaciones.
3. **Opcional: Calcular Transformaciones.** Al oprimir este comando se pueden calcular las transformaciones para X y Y. No es prerequisite correr este comando para obtener el ajuste correcto. <Transformaciones para X y Y> permite ver los valores de las transformaciones en caso que se especifiquen, de otra forma se replicarán ahí los datos para X y Y.

En los datos traídos de un archivo, pueden aparecer datos para TransfX, TransfY, Y ajuste y Residual, estos valores se recalcularán cada vez que se ejecute el comando <Transformaciones para X, Y> y <Ajuste Recta> o <Ajuste Cuadrático>.

- Si existen valores nulos para X o Y, los datos del ajuste pueden ser inválidos.
- Los valores nulos sólo pueden existir cuando son importados de Excel.
- Si existen valores nulos, los datos aparecerán en blanco.
- Si los nulos representan ceros, deben ser modificados manualmente.

4. **Generar el Modelo de Ajuste.** Los comandos <Ajuste Recta> y <Ajuste Cuadrático> despliegan una tabla con la información del modelo creado.

- Al activar la caja-chequeo "Desplegar cálculos del ajuste" al generar el modelo de regresión la tabla despliega cálculos intermedios, de otro modo se proporcionan sólo datos finales. Se activa haciendo clic en ella, otro clic la desactiva.
- <Ajuste Recta> genera los datos para el ajuste lineal: $Y = a + bX$.
- <Ajuste Cuadrático> genera los datos para el ajuste cuadrático: $Y = a + bX + cX^2$.
- Al generar los modelos de regresión se calculan automáticamente las transformaciones para X y Y, en caso que hayan sido especificadas.

5. **Opcional: Gráficas.** Se proporcionan 3 gráficas, listadas a continuación:
- <Gráfica de Dispersión (X y Y)> crea una gráfica mostrando los puntos (X, Y) originales.
 - <Gráfica de Transformaciones de X y Y> muestra los pares de puntos (transfX, transfY). En caso que no se especifiquen transformaciones, la gráfica es idéntica al punto anterior.
 - <Gráfica TransfX y Y con Línea de Ajuste> muestra los pares de puntos (transfX, transfY) y la estimación realizada utilizando el modelo de regresión calculado. Se muestra la recta o curva calculada.

6. **Opcional: Guardar la función del modelo.** La expresión del modelo puede ser guardado para ser usado posteriormente en cualquier sección o pantalla que accese funciones de usuario.

Ajuste Guardar

- Especifíquese un nombre para la función de un máximo 15 caracteres de longitud.
- Clic en el comando guardar.
- Nota: por su naturaleza, las secciones de regresión generan la expresión del modelo que emplean para el ajuste de los datos, a partir de las observaciones especificadas por el usuario. No es necesario, ni posible el proporcionarle la función a utilizar para ajustar los datos.

Los archivos de datos para Regresión se borran en la sección de Utilerías del Menú Principal.

Manejo del área de entrada de datos. Véase el anexo I: "Captura de datos" para mayor información.

Borrar Todo	X	Y	X y Y				
1) Teclar valores para X y Y, ó, traer datos de un archivo.							
X	Y	TransX	TransY	Yajuste	Residual		
33	62	33	62	65.84094	-3.84094		
50	66	50	66	69.75434	-3.75434		

- **Comando <Borrar Todo>**. Inicializa la pantalla borrando la tabla de captura de datos, el área de delegado de información, y las transformaciones especificadas.
- **Comando <X>**. Borra la serie de valores para X.
- **Comando <Y>**. Borra la serie de valores para Y.
- **Comando <X y Y>**. Borra la tabla de las observaciones (X, Y, TransX, TransY, Yajuste, Residual).
- **Especificación de Datos:** tecleando pares de valores (X, Y).
 - Se provee una área de captura de datos con los encabezados X, Y en las columnas a usar. Las demás columnas no permiten su edición.
 - **TransX, TransY** son las transformaciones que se efectúan de los datos (en caso que se especifiquen), de otro modo contendrán los puntos (X, Y).
 - **Yajuste** es la estimación de Y después de calcular el modelo;
 - **Residual** es la diferencia entre el "ajuste de Y" y Y (o su transformada)
- **Especificación de Datos:** transfiriéndolos de un archivo de datos. Si la caja-combo de archivos está vacía, esto indica que no existen archivos guardados.

Archivo de Datos	UXYabril132002	Traer Archivo
Guardar Datos en Archivo		

- Clic en Archivo de Datos. De la lista que aparezca, selecciónese el que se desea acceder.
- clic en <Traer Archivo> transferirá una copia del archivo al área de Captura de Datos. Ahí los datos pueden ser modificados; en esta área además, se pueden agregar o borrar datos. Los datos en el archivo permanecerán intactos.
- **Guardar datos.** El comando <Guardar Datos en Archivo> permite guardar los datos permanentemente para ser accedidos posteriormente.
 - Clic en <Guardar Datos en Archivo> hace aparecer una pantalla preguntando el nombre que se le desea dar al archivo.
 - El nombre del archivo debe ser de un máximo de 15 caracteres de longitud. De excederse EmCalc no guardará los datos.
 - Al nombre del archivo se le antepondrá "UXY" automáticamente.
 - No existe restricción en caracteres a usar para el nombre.

Nota: al guardar un archivo con un nombre ya existente, el nuevo archivo sustituirá al viejo perdiéndose los datos del archivo viejo. Una forma de comprobar los nombres existentes de archivos es haciendo clic en cualquiera de las dos "cajas" de archivo de los comandos <Traer Archivo>.

Borrado de Archivos de datos. Véase la sección de <Solución de Problemas/Utilerías> del Menú Principal.

Ejemplo. Se desea encontrar el modelo lineal de ajuste para las siguientes observaciones.

Peso (Kg): 80 72 75 65 82 86 90 77
 Estatura (cms): 175 168 177 167 178 181 178 170

Análisis de Regresión

Borrar Todo X Y X y Y Instrucciones/Ayuda

1) Tecliar valores para X y Y, ó, traer datos de un archivo.

X	Y	TransX	TransY	Ajuste	Residual
80	175	80	175	175.145	-0.14495
72	168	72	168	170.741	-2.74095
75	177	75	177	172.3925	4.60765
65	167	65	167	166.8875	0.11255
82	178	82	178	176.246	1.754050
86	181	86	181	178.448	2.552050
90	178	90	178	180.65	-2.64995
77	170	77	170	173.4935	-3.49345

Record: 14

Y = 131.10495 + .5505 * X
 a = 131.10495
 b = 0.55050
 Est. Coef. Corr. = 0.83621652221258
 Est. Var(e) = 9.59868007
 Est. dS(e) = 3.09617366685601

Cálculos: * * * * *

N = 8
 Sum(X) = 627
 Media(X) = 78.375
 Sum(Y) = 1424

2) Opcional. Transformaciones para X y Y.

Función F(X) F(Y) Borrar: F(X) F(Y)
 Div. por Cero? F(X)= 0 F(Y)= 0
 Dverflow? F(X)= 0 F(Y)= 0
 Otro Error? F(X)= 0 F(Y)= 0

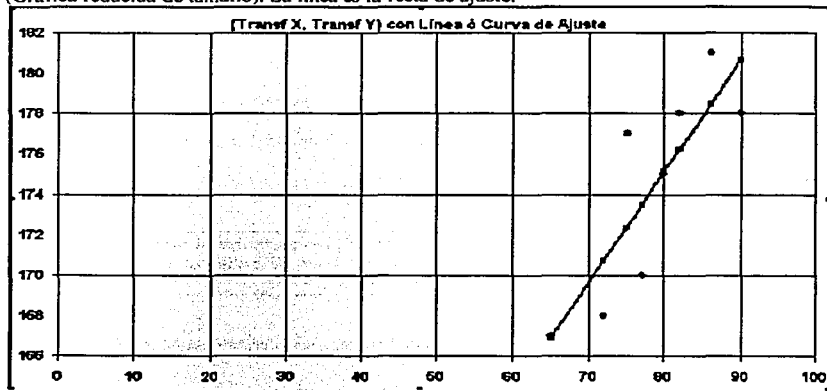
3) Calcular Transformaciones y Ajuste.

Transformaciones para X y Y
 Desplegar cálculos del ajuste.
 Ajuste Rectal Ajuste Cuadrático
 Gráfica de Dispersión (X y Y)
 Gráfica de Transformaciones de X y Y
 Gráfica Trans. X y Y con Línea de Ajuste

Ajuste Guardar

Opciones usadas:

(Gráfica reducida de tamaño). La línea es la recta de ajuste.



Módulo 4.3 Desplegado de Formas con Tablas Estadísticas.

Se muestra un ejemplo del desplgado de las formas.

Para desplegar las pantallas que contienen tablas estadísticas, oprímase la caja-comando correspondiente:

TABLAS

Normal	t Student	χ^2
F 1%	F 5%	N Alcorarios

Ejemplos de desplgado de las formas de tablas estadísticas.

Función Normal										
Tabla de densidad acumulada de la función Normal										
X	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.0	0.5000	0.5040	0.5080	0.5120	0.5160	0.5199	0.5239	0.5279	0.5319	0.5359
0.1	0.5398	0.5438	0.5478	0.5517	0.5557	0.5596	0.5636	0.5675	0.5714	0.5753
0.2	0.5793	0.5832	0.5871	0.5910	0.5948	0.5987	0.6026	0.6064	0.6103	0.6141
0.3	0.6179	0.6217	0.6255	0.6293	0.6331	0.6368	0.6406	0.6443	0.6480	0.6517
0.4	0.6554	0.6591	0.6628	0.6664	0.6700	0.6736	0.6772	0.6808	0.6844	0.6879
0.5	0.6915	0.6950	0.6985	0.7019	0.7054	0.7088	0.7123	0.7157	0.7190	0.7224
0.6	0.7257	0.7291	0.7324	0.7357	0.7389	0.7422	0.7454	0.7486	0.7517	0.7549
0.7	0.7580	0.7611	0.7642	0.7673	0.7704	0.7734	0.7764	0.7794	0.7823	0.7852
0.8	0.7881	0.7910	0.7939	0.7967	0.7995	0.8023	0.8051	0.8078	0.8106	0.8133
0.9	0.8159	0.8186	0.8212	0.8238	0.8264	0.8289	0.8315	0.8340	0.8365	0.8389
1.0	0.8413	0.8438	0.8461	0.8485	0.8508	0.8531	0.8554	0.8577	0.8599	0.8621
1.1	0.8643	0.8665	0.8686	0.8708	0.8729	0.8749	0.8770	0.8790	0.8810	0.8830
1.2	0.8849	0.8869	0.8888	0.8907	0.8925	0.8944	0.8962	0.8980	0.8997	0.9015
1.3	0.9032	0.9049	0.9066	0.9082	0.9099	0.9115	0.9131	0.9147	0.9162	0.9177
1.4	0.9192	0.9207	0.9222	0.9236	0.9251	0.9265	0.9279	0.9292	0.9306	0.9319
1.5	0.9332	0.9345	0.9357	0.9370	0.9382	0.9394	0.9406	0.9418	0.9429	0.9441
1.6	0.9452	0.9463	0.9474	0.9484	0.9495	0.9505	0.9515	0.9525	0.9535	0.9545

Fuente: E. Parzen, *Modern Probability Theory and its applications*, Wiley, 1960. Table I, p.441

2.8.3.1 Módulo 4.3.1 Generación de Números Aleatorios.

Forma de Números Aleatorios. La forma permite crear tablas de números aleatorios de acuerdo a parámetros especificados por el usuario.

La tabla muestra los defaults usados para generar números aleatorios.

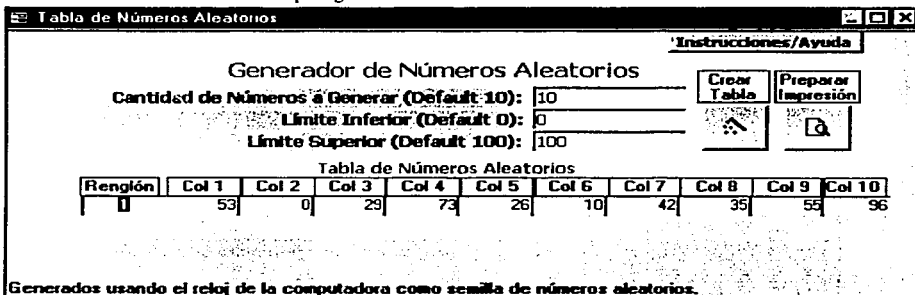


Tabla de Números Aleatorios

Generados usando el reloj de la computadora como semilla de números aleatorios.

Generación de tablas de Números Aleatorios.

Los siguientes son los parámetros que el usuario debe especificar:

- Cantidad de Números a Generar. Se debe especificar un número mayor o igual a 10.
 - El default es 10 cuando hay un valor inválido, o menor a 10.
 - La cantidad de números que se generan es un múltiplo de 10 que contiene al número que se le pide. Por ejemplo, si se especifica "Cantidad de Números = 34", entonces se generarán 40 números aleatorios.
- Rango en el cual se deben producir los números aleatorios:
 - Límite Inferior.
 - Valor mínimo del rango. Puede ser menor que cero, incluso.
 - El default es 0 cuando hay un valor inválido.
 - Límite Superior.
 - Valor máximo del rango. Puede ser menor que cero, incluso.
 - El default es 100 cuando hay un valor inválido.

Para generar la tabla, hágase clic en la "Varita Mágica".

Nota: EmCalc no permite "reversar" los límites. Límite Inferior debe ser menor a Límite Superior, de otra forma usará los defaults.

Si se especifican números decimales, EmCalc los redondeará.

Notas:

- Cada vez que se haga clic en la "Varita Mágica", se generará una nueva tabla de números aleatorios con los mismos parámetros.

- Debido a que son generados por la misma rutina, se les llama Números Pseudo aleatorios, sin embargo, se contrarresta este efecto de pseudo aleatoriedad al inicializar la rutina de generación de números utilizando el reloj de la computadora como semilla.
- La cantidad de números a generar esta limitado sólo por la capacidad del software MS-Access o de la computadora en que se corra el sistema: 1000,000 es un número posible de generar. Véase la sección de <Solución de Problemas/Utilerías> del Menú Principal.

Ejemplo: La siguiente tabla muestra la generación de una tabla con 1,000,000 (un millón) de números aleatorios (Véase nota en Utilerías). El rango utilizado para producir números es (-200, 200).

La tabla muestra los últimos renglones de la tabla creada en el ejemplo.

Tabla de Números Aleatorios

Instrucciones/Avuda

Generador de Números Aleatorios

Cantidad de Números a Generar (Default 10): 1000000

Límite Inferior (Default 0): -200


Límite Superior (Default 100): 200

Tabla de Números Aleatorios

Renglón	Col 1	Col 2	Col 3	Col 4	Col 5	Col 6	Col 7	Col 8	Col 9	Col 10
99989	83	-160	-137	84	27	-35	121	-85	22	13
99990	-21	33	-112	-155	-121	-160	47	-68	37	-188
99991	93	7	76	-194	-117	-175	-44	-137	-85	17
99992	-191	-82	-190	-152	-136	6	92	-127	83	165
99993	67	-108	-109	45	-189	-193	8	-186	-155	178
99994	-23	-36	120	79	-200	17	-105	-145	67	130
99995	121	-20	184	5	-79	8	-90	-111	-26	36
99996	187	90	-14	-149	65	171	-110	200	144	-134
99997	-92	131	95	84	-56	59	-102	10	-135	121
99998	-30	-82	-136	-170	197	76	178	-168	-135	148
99999	32	163	94	-97	-59	-198	-37	24	86	-65
100000	-51	-191	-2	103	-147	183	-72	61	31	-4

Generados usando el reloj de la computadora como semilla de números aleatorios.

El comando <Preparar Impresión> genera la pre-impresión de la tabla, la cuál puede ser impresa o cancelada a elección del usuario.

- Para imprimir oprímase el comando  del menú del sistema Access.
- Para cancelar, oprímase la tecla <Esc> del teclado.

2.8.4 Módulo 4.3 Pre-impresión é Impresión de Tablas Estadísticas.

Se muestra un ejemplo reducido de la impresión de tablas estadísticas

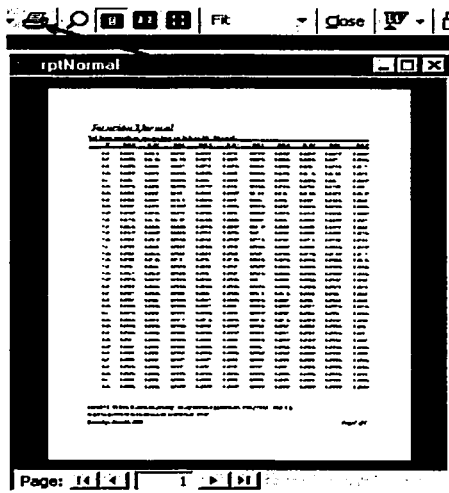
Para desplegar una tabla en formato de impresión, oprímase la caja-comando correspondiente:

IMPRESION DE TABLAS

Normal	t Student	χ^2
F 1%	F 5%	N Aleatorios

Al obtener la pre-impresión, si se desea imprimir, oprímase el comando que se muestra en la ilustración. Si no se desea la impresión, oprímase la tecla <Esc>

Nota: en el caso de la tabla de números aleatorios, se imprime la última que se haya creado.



2.9 Módulo 5. Funciones Financieras. Cálculo de Anualidades, Valor Presente Neto, Depreciación y Tasa Interna de Retorno.

2.9.1 Módulo 5.0 Anualidades y Características de las Pantallas de Anualidades.

Anualidades. Una anualidad es una serie de pagos iguales, realizados periódicamente, durante cierto lapso de tiempo, a una tasa de interés fija, y con periodicidad idéntica. Una anualidad puede representar un préstamo, o una inversión; los pagos pueden ser vencidos o anticipados.

Funciones Financieras Form

Funciones Financieras	
ANUALIDADES	
Valor Presente	Valor Futuro
Interés	Pago
Número de Pagos	Cap é Int Pagado
VPN (Pagos é Interés Variables)	
Depreciación	
Tása Interna de Retorno	

Los siguientes elementos forman parte de una anualidad.

- **El Capital o Principal.** Es el valor presente de un préstamo; de una inversión, es la suma de dinero ahorrado, sin incluir intereses.
- **Pago o Pago periódico.** Los pagos pueden representar salidas de dinero: pagos propiamente, o depósitos; o entradas de dinero: ingresos o recibos de dinero, pero no combinaciones. Todos los pagos son iguales.
- **Periodicidad.** Todos los pagos tienen una periodicidad idéntica. Por ejemplo, todos los pagos son mensuales, o todos son trimestrales o todos usan otra periodicidad idéntica para todos los pagos, pero no combinaciones.
- **Número total de pagos (o Número de Pagos).** Debe haber un número definido de pagos en un lapso de tiempo.

- **Tasa de interés (o interés, o tasa).** Si la anualidad representa un préstamo, la tasa es el porcentaje periódico que hay que pagar sobre el saldo no liquidado de un préstamo; o en su caso si la anualidad representa una inversión, entonces la tasa es el porcentaje de utilidad que se ganará por período sobre el monto acumulado de una inversión.
- **Convertibilidad.** Es una característica de la tasa de interés, relacionada con la periodicidad de los pagos con respecto al período de un año. Ejemplos:
 - o Tasa anual convertible mensualmente. La tasa se divide por 12 para obtener la tasa a aplicar.
 - o Tasa anual convertible trimestralmente, la tasa anual se divide por 4 para obtener la tasa a aplicar.
- **Tipo de Pagos: Vencidos o Anticipados.** Son vencidos cuando el pago se realiza al final del período; son anticipados cuando el pago se realiza al principio del período.
- **Saldos Insolutos.** Es el capital aún no liquidado de una deuda, en un cierto período de tiempo, antes que la deuda expire. Al final de la anualidad, o al expirar la anualidad, el saldo insoluto es cero.

Para los argumentos: Pagos, Valor Presente (Saldo Inicial) o Valor Futuro (Saldo Final), el efectivo pagado, o salidas de dinero, tal como el depósito en ahorros, es representado con números negativos; mientras que, el dinero recibido, o entradas de dinero, tal como cheques de dividendos, es representado con números positivos.

Características de las pantallas de Anualidades.

1. Permiten realizar 4 sensibilidades.

- Cada columna representa una sensibilidad.
- Los datos que el usuario debe proporcionar aparecen en negro.
- El resultado aparece resaltado en rojo.
- Para obtener el resultado se oprime el comando <(Col i) XX Sensibilidad i>, i = 1, 4; XX es alguno de "Valor Presente", "Valor Futuro", "Pago", "Interés", "No. de Períodos", "Capital e Interés Pagado".

Después de leer el dato oprimir aquí =	(col 1) VP Sensibilidad 1	(col 2) VP Sensibilidad 2	(col 3) VP Sensibilidad 3	(col 4) VP Sensibilidad 4
	borrar	borrar	borrar	borrar
Valor Presente	0.00	1464.07	0.00	1171.25
Pago	0	-125	0	-100
No. de Períodos	0	12	0	12
Interés 1	0.00%	0.37%	0.00%	0.37%
Valor Futuro	0	0	0	0
Tipo	0	0	0	0

2. Para copiar de una columna a otra, se especifican la columna origen y la columna destino en las cajas como que se proporcionan. Después se oprime el comando de la "varita mágica". Ejemplo:

Copiar Datos de Columna 2 a Columna 4

- Cualquier columna puede ser el origen y cualquier columna puede ser el destino, para lo cuál las cajas-combo sólo despliegan las columnas 1 a 4.
- El copiar a la misma columna esta inhibido.

3. Se provee una área para realizar cálculos, usando expresiones, cuyo resultado se puede transferir al interés de cualquiera de las cuatro sensibilidades. Por ejemplo:

Expresión: 0.045/12
 Resultado: 0.0037
 Pasar a Interés 1 2 3 4

- Se escribe cualquier expresión válida, utilizando números, operadores y funciones válidas.
- Para obtener el resultado se oprime <enter>, <tab>, clic en cualquier campo.
- Para pasar el Resultado al interés de una sensibilidad, basta oprimir cualquier comando <Pasar a Interés 1, 2, 3 o 4>.

Nota:

En la sección del cálculo del Interés, la transferencia es del interés calculado, hacia la expresión: ejemplo:

Interés 1 2.92% 2 0.00% 3 0.00% 4 0.00%

Pasar Interés 1 2 3 4 a expresión. Usese ratón para editar.

Expresión: 0.0292285407691344 * 12
 Resultado: 0.350742489229613

Nótese que al transferir el resultado se despliegan todos los decimales que el sistema calcula.

Ejemplo 1: Se obtiene un préstamo para un automóvil por el cual se pagarán \$2,600 mensuales, aplicando una tasa de interés del 14% anual, por 48 meses, y al final se hace un pago adicional por \$5,000. ¿Cuál es el precio actual del auto? Esto es equivalente a preguntarse ¿Cuál es el Valor Presente de los pagos que se efectuarán?

Datos (Sensibilidad 1):

Pago:	-2600 (negativo, por ser salida de dinero)
No. de Pagos	48
Interés(*)	0.14/12 = 1.17% (se puede usar área de cálculo y luego pasar a Interés)
Valor Futuro	-5000 (negativo, por ser salida de dinero; es un saldo final a pagar).
Tipo	0: Anticipados. Suponemos que los pagos empiezan al momento de obtener el crédito.
Valor Presente	\$98,011.14 (+ por ser dinero recibido, aunque sea un préstamo). Es el precio del automóvil. La figura abajo lo muestra como "Coll VP Sensibilidad 1".

(*) Se da el interés anual, hay que convertirlo a la periodicidad del pago, que en este caso es mensual.

Sensibilidad 2: ¿Cuál es el valor si al final no se paga nada? (Efecto neto de las mensualidades): \$95,145.82

Sensibilidad 3: ¿Cuál es el efecto del pago de 5,000 al final, traído a valor presente?: \$ 2,865.32.

Nótese que: 95145.82 + 2865.32 = 98011.14. (VP Sensibilidad 1 = VP Sensibilidad 2 + VP Sensibilidad 3)

Sensibilidad 4: ¿Cuál es el efecto de hacer pagos anticipados, en lugar de vencidos? \$ 1,110.03 se estarían "regalando" al acreedor, desde el punto de vista de quien obtiene el préstamo.

La tabla muestra las sensibilidades del ejemplo. Sensibilidad 1 son los datos originales.

Form: **IntAnuaVP : Form**

Anualidades: Valor Presente

	(col 1) VP Sensibilidad 1	(col 2) VP Sensibilidad 2	(col 3) VP Sensibilidad 3	(col 4) VP Sensibilidad 4
	borrar	borrar	borrar	borrar
Valor Presente	98011.14	95145.82	2865.32	99121.17
Pago	-2600	-2600	==> 0	-2600
No. de Periodos	48	48	48	48
Interés	1.17%	1.17%	1.17%	1.17%
Valor Futuro	-5000	==> 0	-5000	-5000
Tipo	0	0	0	==> 1

Copiar Datos de Columna 1 a Columna 4

El interés y los pagos deben tener la misma periodicidad

Para haber calculado el interés y los pagos se debe haber ingresado la aban.

Ejemplo: 0.14/12 1.00/12=1 Enter Tab 0.14/12=0.0117

Expresión: ==>

Resultado:

Pasar a Interés

Se pueden ver las 4 sensibilidades a la vez.

2.9.3 Módulo 5.2 Valor Futuro.

Calcula el capital e intereses acumulados que se obtendrán después de hacer una serie de pagos iguales con periodicidad constante y con una tasa de interés fija; el pago puede ser vencido o anticipado.

Pantalla para cálculo del Valor Futuro de una anualidad.

ImAnualFV - Form
⊞ ⊞ ⊞

? **Anualidades: Valor Futuro**

	(col 1) VF Sensibilidad 1	(col 2) VF Sensibilidad 2	(col 3) VF Sensibilidad 3	(col 4) VF Sensibilidad 4
	<input type="button" value="borrar"/>	<input type="button" value="borrar"/>	<input type="button" value="borrar"/>	<input type="button" value="borrar"/>
Valor Presente	<input type="text" value=""/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Pago	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
No. de Periodos	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Interés	1 <input type="text" value="0.00%"/>	2 <input type="text" value="0.00%"/>	3 <input type="text" value="0.00%"/>	4 <input type="text" value="0.00%"/>
Valor Futuro	<input type="text" value="0.00"/>	<input type="text" value="0.00"/>	<input type="text" value="0.00"/>	<input type="text" value="0.00"/>
Tipo	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>

Copiar Datos de Columna a Columna

El interés y los pagos deben tener la misma periodicidad

Para hacer cálculos, teclee la una expresión algebraica abajo:

Ejemplo: 07/12/107 (1/12)11 Ejemp: Tab. w. 3/12 en otro campo

Expresión:

Resultado:

Pasar a Interés
 1
 2
 3
 4

Especificación de argumentos (los conceptos se definen al inicio del capítulo):

- **Pago.** Es el pago periódico. Puede ser cero si se especifica un Valor Presente (o un Saldo Inicial).
 - Para salidas de dinero usar negativos. Ejemplos: depósitos al banco, pagos a préstamos.
 - Para entradas de dinero usar positivos. Ejemplo: ingresos o recibos de dinero.
- **Interés.** Su periodicidad debe estar acorde al pago periódico. Se pueden hacer cálculos en el área inferior derecha (llamada Expresión), después clic en "pasar a" Interés 1, 2, 3 o 4.
- **Número de Periodos** o **Número de Pagos.** Total de pagos a realizar en la duración de la anualidad.
- **Valor Presente** o **Saldo Inicial** (parámetro opcional). Cantidad Inicial que se deposita junto con el primer depósito a invertir.
 - Si el pago (o depósito) es cero, entonces debe especificarse el Valor Presente (En este caso es un saldo inicial).
- **Tipo de Pago.** Momento en que se hace el pago:
 - **0:** vencidos, pago al final del período;
 - **1:** anticipados, pago al principio del período.

Ejemplo de Valor Futuro: ¿Que cantidad se obtendrá después de 12 depósitos de \$500, empezando con un depósito de \$2,500 adicional en el primer pago, al 6% de interés (anual) que otorga el banco?

Datos (Sensibilidad 1):

Valor Presente: -2500 (negativo, porque es salida de dinero)
 Pago(depósito): -500 (negativo, por ser salida de dinero)
 No.de Períodos: 12
 Interés: 0.06/12 = .005 = .5% (se puede usar el área de cálculo y transferir al interés).
 Tipo: 0: anticipado. (El dinero se retira un mes después de hacer el último pago.)
 Valor Futuro: \$8,821.98 (+ por ser ingreso). Es la cantidad que se acumulará al final de 12 meses.

Sensibilidad 2: ¿Cuánto se acumulará si no se tiene saldo inicial? Efecto neto de las mensualidades: \$6,167.78

Sensibilidad 3: ¿Cuál es el efecto de acumulación del Saldo Inicial, al hacer el último pago?: \$ 2,654.19
 Nótese que: 6167.78 + 2654.19 = 8852.81. (VP Sensibilidad 1 = VP Sensibilidad 2 + VP Sensibilidad 3)

Sensibilidad 4: ¿Cuál es el efecto de hacer depósitos anticipados, en lugar de vencidos? Se obtendrían \$30.83 extras.

La figura muestra las sensibilidades del ejemplo.

IrmAnnualFV : Form

Añualidades: Valor Futuro

Después de teclear datos, oprima aquí =

	(col 1) VF Sensibilidad 1	(col 2) VF Sensibilidad 2	(col 3) VF Sensibilidad 3	(col 4) VF Sensibilidad 4
	borrar	borrar	borrar	borrar
Valor Presente	-2500	==> 0	-2500	-2500
Pago	-500	-500	==> 0	-500
No. de Períodos	12	12	12	12
Interés	1 0.50%	2 0.50%	3 0.50%	4 0.50%
Valor Futuro	8821.98	6167.78	2654.19	8852.81
Tipo	0	0	0	==> 1

Borrar Todo Copiar Datos de Columna [1] a Columna [4]

El interés y los pagos deben tener la misma periodicidad
 Para hacer cálculos de interés compuesto, registre la fecha

Ejemplo: 07/12/10 10/01/11 Enter Tabla Cálculo Períodos

Ejemplo de expresión a evaluar → Expresión: .06/12 Resultado: 0.0050

Passar a Interés 1 2 3 4

2.9.4 Módulo 5.3 Cálculo de la Tasa de Interés.

Calcula el interés para una serie de pagos periódicos iguales, especificando un valor presente o un valor futuro. La tasa de interés calculada tendrá una periodicidad idéntica a la de los pagos especificados.

Pantalla para cálculo de la Tasa de Interés de una anualidad.

FinAnualInt Form
?

Ejemplos Anualidades: Cálculo del Interés

	col 1. Interés Sensibilidad 1	col 2. Interés Sensibilidad 2	col 3. Interés Sensibilidad 3	col 4. Interés Sensibilidad 4
	<input type="button" value="borrar"/>	<input type="button" value="borrar"/>	<input type="button" value="borrar"/>	<input type="button" value="borrar"/>
Valor Presente	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Pago	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
No. de Períodos	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Interés 1	0.00%	0.00%	0.00%	0.00%
Valor Futuro	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Tipo	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>

Copiar Datos de Columna a Columna

El interés y los pagos tienen la misma periodicidad

Para hacer cálculos teclee la siguiente expresión algebraica abajo:

Ejemplos: 07 12 11 07 12 11 Enter Tab o clic en otro grupo.

Pasar Interés a expresión. Usese ratón para editar.

Expresión:

Resultado:

Especificación de argumentos (los conceptos se definen al inicio del capítulo):

- **Valor Presente de la anualidad** o saldo inicial.
 - Si el Valor Presente es cero, entonces deben especificarse Pago y Valor Futuro.
- **Pago.** Es el pago periódico. Puede ser cero si se especifican ambos: Valor Presente y Valor Futuro.
 - Para salidas de dinero usar negativos. Ejemplos: depósitos al banco, pagos a préstamos.
 - Para entradas de dinero usar positivos. Ejemplo: ingresos o recibos de dinero.
- **Número de Períodos** o **Número de Pagos.** Total de pagos a realizar en la duración de la anualidad.
- **Valor Futuro** o **Saldo Final.** Valor futuro de las anualidades o Cantidad a pagar junto con el último pago
 - Si Valor Futuro es cero, entonces deben especificarse Pago y Valor Presente.
- **Tipo de Pago.** Momento en que se hace el pago:
 - 0: vencidos, pago al final del período;
 - 1: anticipados, pago al principio del período.

Al finalizar el cálculo del interés, éste se puede transferir al área de cálculo de expresiones (área inferior derecha), al hacer clic en <pasar Interés> 1, 2, 3 o 4 a expresión; para calcular el interés anual.

- e) VP, Pagos y VF(VP es el total de Anualidad y Saldo Final): Se obtiene un crédito por \$5000 y se deben hacer 24 pagos de \$220 y al final un pago extra de \$500. ¿Qué interés se está aplicando?
 -VP: 5000. Es positivo porque se recibe. -Pago: -220. Salida de dinero.
 -No. de periodos: 24 -VF: -500 (Salida de dinero)
 -Interés: 1.10% mensual.

f) Mismo caso que (e), con pagos anticipados.

g) Mismo caso que e, pero al final en lugar de pagar \$500, regresan \$100.

Mismo valor en los argumentos, excepto:

-VF: 100. Positivo porque es ingreso. Interés = 0.29% mensual.

h) Mismos caso que (f), con pagos anticipados.

La pantalla muestra los ejemplos e-h, dados arriba. Muestra el cálculo del interés anual para la sensibilidad 3.

IrAnualInt Form

Ejemplos	Anualidades: Cálculo del Interés			
	Ejemplo: e			
	col 1. Interés Sensibilidad 1	col 2. Interés Sensibilidad 2	col 3. Interés Sensibilidad 3	col 4. Interés Sensibilidad 4
	borrar			
Valor Presente	5000	5000	5000	5000
Pago	-220	-220	-220	-220
No. de Periodos	24	24	24	24
Interés	1.10%	1.19%	0.29%	0.32%
Valor Futuro	-500	-500	100	100
Tipo	0	1	0	1

Copiar Datos de Columna 3 a Columna 4

El interés y los pagos tienen la misma periodicidad
 Para haber editado, teclee la expresión algebraica sobre
 Ejemplo: 0.0212 1.01 1201 Enter Tablo: 0.0212 1.01 1201
 Pasar Interés 1 2 3 4 a expresión. Use ratón para editar.

Expresión: 0.00290052491729131*12
 Resultado: 0.0348

Borrar Todo

Se calcula interés 3 en forma anual.

2.9.5 Módulo 5.4 Cálculo del Pago Periódico.

Calcula el Pago de una anualidad, la cuál está basada en una serie de pagos iguales (desconocidos) con periodicidad constante y con una tasa de interés fija, el pago puede ser vencido o anticipado.

- Si el pago es negativo representa salida de dinero. Ejemplo: depósitos al banco.
- Si es positivo, representa entradas de dinero. Ejemplo: ingresos o recibos de dinero.

Pantalla para cálculo del Pago Periódico.

ImAnualPago Form				
Ejemplos ? Anualidades: Cálculo del Pago Periódico				
Después de teclear datos presionar aquí				
	(col 1) Pago Sensibilidad 1	(col 2) Pago Sensibilidad 2	(col 3) Pago Sensibilidad 3	(col 4) Pago Sensibilidad 4
	borrar	borrar	borrar	borrar
Valor Presente	0	0	0	0
Pago	0.00	0.00	0.00	0.00
No. de Periodos	0	0	0	0
Interés	0.00%	0.00%	0.00%	0.00%
Valor Futuro	0	0	0	0
Tipo	0	0	0	0
Borrar Todo Copiar Datos de Columna <input type="text" value="1"/> a Columna <input type="text" value="2"/>				
El interés y los pagos deben tener la misma periodicidad Para calcular interés, teclear una expresión algebraica a sí: Ejemplo: $0.07(1 + 0.07)^{12} - 1$ Enter Tab. o clic en otro campo				
Expresión: _____ Resultado: _____ Pasar a Interés <input type="button" value="1"/> <input type="button" value="2"/> <input type="button" value="3"/> <input type="button" value="4"/>				

Especificación de argumentos (los conceptos se definen al inicio del capítulo):

- **Interés.** Su periodicidad debe estar acorde al pago periódico. Se pueden hacer cálculos en el área inferior derecha (llamada **Expresión**), después clic en "pasar a" Interés 1, 2, 3 o 4.
- **Número de Periodos** o **Número de Pagos.** Total de pagos a realizar en la duración de la anualidad.
- **Valor Presente** es el monto inicial del préstamo (Puede ser un **Saldo Inicial**).
 - Si el Valor Presente es cero, entonces debe especificarse el Valor Futuro.
- **Valor Futuro.** Valor futuro de las anualidades o Cantidad a pagar junto con el último pago (según sea el caso).
 - Si Valor Futuro es cero, entonces debe especificarse Valor Presente
- **Tipo de Pago.** Momento en que se hace el pago:
 - **0:** vencidos, pago al final del período;
 - **1:** anticipados, pago al principio del período.

Ejemplos:

Se emplean anualidades anticipadas en todos los ejemplos, a menos que se especifique diferente. Los parámetros no especificados son "Cero".

-VP y VF: ¿Que depósito mensual se necesita si se tienen \$500 y se desea que en 1 año aumente a \$2,500, al 8% anual?

-VP: -500. Depósito: Salida de dinero. -No. de períodos: 12
-Interés: $.08/12 = 0.67\%$ mensual. -VF: 2500. Es positivo porque se recibe
Pago: -157.31. Negativo porque es depósito, salida de dinero.

-VP. Si se obtuvo un crédito por \$5,000 y se pagarán 36 mensualidades, con tasa del 15% anual. ¿Que pago se necesita hacer?

-VP: 5000. Préstamo: entrada de dinero. -No. de períodos: 36
-Interés: $.15/12 = 1.25\%$ mensual. Pago: -173.33 mensual. Salida de dinero.

-VF y Pagos. Se planea lograr \$1,250 al cabo de 12 depósitos mensuales, si el banco otorga una tasa del 7% anual. ¿Que depósito se necesita hacer?

-No. de períodos: 12 -Interés: $0.07/12 = 0.58\%$ mensual.
-VF: 1250. Es positivo porque se recibe. -Pago: -100.87. Es un depósito. Salida de dinero.

-VP, Pagos y VF (VF es el total de Anualidad y Saldo Inicial): Si se inicia o con \$1,000, haciendo 12 depósitos mensuales al banco, el cual otorga o el 8% anual, y se reciben \$2500 al final ¿Cuál es el monto del depósito?

-VP: -1000. Depósito: Salida de dinero. -No. de períodos: 12
-Interés: $0.08/12 = 0.67\%$ -VF: 2500. Es positivo porque se recibe. -
-Pago: -113.82. Es un depósito. Salida de dinero.

-VP, Pagos y VF (VP es el total de Anualidad y Saldo Final): Se obtuvo un crédito al 14% anual, por \$5,000 y se deben hacer 24 pagos, y al final se debe hacer un pago extra de \$500. ¿Qué depósito se debe hacer?

-VP: 5000. Préstamo: entrada de dinero. -No. de períodos: 24
-Interés: $0.14/12 = 1.17\%$ -VF: -500. Es un pago. Salida de dinero.
-Pago: -221.89. Salida de dinero.

---Mismo caso anterior, pero al final en lugar de pagar \$500, regresan \$100.

-Mismo valor en parámetros, excepto VF: 100. Entrada de dinero. Pago = -243.70. Salida de dinero.

2.9.6 Módulo 5.5 Cálculo del Número de Pagos.

Calcula el Total de Pagos Completos más el pago parcial que se hace en un período adicional, que se requieren en una anualidad basada en una serie de pagos iguales con periodicidad constante y con una tasa de interés fija, el pago puede ser vencido o anticipado.

Pantalla para cálculo del Número de Pagos y el pago parcial que se hará en un período posterior.

frmAnualNumPagos Form

Ejemplos
?

Anualidades: Cálculo del Número de Pagos

	col1 No. Pagos Sensibilidad 1	col2 No. Pagos Sensibilidad 2	col3 No. Pagos Sensibilidad 3	col4 No. Pagos Sensibilidad 4
	<input type="button" value="borrar"/>	<input type="button" value="borrar"/>	<input type="button" value="borrar"/>	<input type="button" value="borrar"/>
Valor Presente	0	0	0	0
Pago	0	0	0	0
No. de Períodos	0	0	0	0
Pago Parcial	0.00	0.00	0.00	0.00
Tasa de Interés	1 0.00%	2 0.00%	3 0.00%	4 0.00%
Valor Futuro	0	0	0	0
Tipo	0	0	0	0

Copiar Datos de Columna a Columna

El interés y los pagos deben tener la misma periodicidad

1 Para calcular interés: teclea e ir a este menú algebra: a b c
Ejemplo: 0.07 12 1.07 12 12 1 Enter Tab o clic en otro campo

Expresión:

Resultado:

Pasar a Interés

Especificación de argumentos (los conceptos se definen al inicio del capítulo):

- **Valor Presente** es el monto inicial del préstamo (Puede ser un **Saldo Inicial**).
 - Si Valor Presente es cero, entonces debe especificarse Valor Futuro y Pago.
- **Pago.** Es el pago periódico. Puede ser cero si se especifica un Valor Futuro.
 - Para salidas de dinero usar negativos. Ejemplos: depósitos al banco, pagos a préstamos.
 - Para entradas de dinero usar positivos. Ejemplo: ingresos o recibos de dinero.
- **Interés.** Su periodicidad debe estar acorde al pago periódico. Se pueden hacer cálculos en el área inferior derecha (llamada **Expresión**), después clic en "pasar a" Interés 1, 2, 3 o 4.
- **Valor Futuro** o **Saldo Final** (parámetro opcional). Cantidad a pagar junto con el último pago (en caso de un préstamo).
 - Si Valor Futuro es cero, entonces debe especificarse Valor Presente y Pago.
- **Tipo de Pago.** Momento en que se hace el pago:
 - **0:** vencidos, pago al final del período;
 - **1:** anticipados, pago al principio del período.

Nota: el número de periodos de pago regularmente resulta en un número con decimales. El pago parcial que EmCalc presenta se calcula utilizando esta fracción.

Ejemplos:

En esta ocasión se presenta la pantalla al inicio. Los ejemplos (a a d) se encuentran a continuación

Ejemplos				
Anualidades: Cálculo del Número de Pagos				
col1 No. Pagos Sensibilidad 1	col2 No. Pagos Sensibilidad 2	col3 No. Pagos Sensibilidad 3	col4 No. Pagos Sensibilidad 4	
borrar	borrar	borrar	borrar	
Valor Presente	-2500	5000	0	-1000
Pago	0	-350	-100	-100
No. de Períodos	16.3864968597	15.7185465587	12.2107266597	13.3203155368
Pago Parcial	0.00	-251.49	-21.07	-32.03
Tasa de Interés	0.58%	1.17%	0.42%	0.75%
Valor Futuro	2750	0	1250	2500
Tipo	0	0	0	0
Copiar Datos de Columna 1 a Columna 2				

Sensibilidad 1 no genera pago parcial porque no se especifican pagos

El interés y los pagos deben tener la misma periodicidad
 Para calcular interés, se debe a una expresión algebraica de la forma:
 Ejemplo: .07/12 = .007125

Expresión: .09/12
 Resultado: 0.0075
 Pasar a Interés 1 2 3 4

a) VP y VF: Se depositan \$2500 a una tasa del 7% anual ¿En cuánto tiempo aumentará a \$2,750?
 -VP: -2500. Depósito: salida de dinero. -Pago: 0
 -VF: 2750 (Es positivo porque se recibe) -Interés: .07/12 = 0.58% (se asume convertible mensualmente)
 -R = 16.39 meses ~ 16 meses y 12 días.

Nota: debido a que no se especifica pago, no se genera pago parcial, sino sólo la fracción de periodo que se necesita para acumular al Saldo Final especificado.

b) VP y Pagos. Se obtuvo un crédito por \$5,000 y se harán pagos mensuales de \$350, a una tasa anual del 14%.
 -VP: 5000. Recibo de dinero. -Pago: -350. Negativo, es salida de dinero.
 -Interés: .14/12 = 1.17%
 -R = 15.72 meses ~ 15 pagos de 350 y uno de \$251.49

c) -VF y Pagos. Se planea ahorrar \$100 mensualmente al 5% anual y se desea obtener \$1,250.
 -Pago: -100. Negativo, salida de dinero. -VF: 1250. Es positivo porque se recibe.
 -Interés: .05/12 = 0.42%

-R = 12.21 ~ 12 depósitos de \$100 y uno de \$21.07

d) VP, Pagos y VF (VF es el total de Anualidad y Saldo Inicial): Si se depositan \$1,000 junto con el primero de los depósitos mensuales que se plancan de \$100, y al final se reciben \$2,500; con un rendimiento del 9% anual.

-VP: -1000. Depósito: Salida de dinero.

-Pago: -100. Salida de dinero.

-VF: 2500. Es positivo porque se recibe.

-Interés: .09/12 = 0.75%

R = 13.32 meses ~ 13 depósitos de 100 y uno de \$32.03

Ejemplos en la siguiente pantalla.

e) -VP, Pagos y VF (VP es el total de Anualidad y Saldo Final): Se obtiene un crédito por \$5000 al 15% anual y se deben hacer pagos mensuales de \$220 y al final un pago extra de \$500, junto con el último pago.

-VP: 5000. Es positivo porque se recibe.

-Pago: -220. Salida de dinero.

-VF: -500 (Salida de dinero)

-Interés: .15/12 = 1.25%

R = 24.58 meses ~ 24 pagos de 220 y uno de \$128.24

f) Mismo caso anterior, con pagos anticipados.

R = 24.21 meses ~ 24 pagos de 220 y uno de \$48.08

g) ---Mismo caso anterior, pero al final en lugar de pagar \$500, regresan \$100.

Mismo valor en los argumentos, excepto:

-VF: 100. Positivo porque es ingreso.

R = 27.36 meses ~ 27 pagos de 220 y uno de \$78.98

h) Mismo caso anterior, con pagos anticipados.

R = 26.96 meses ~ 26 pagos de 220 y uno de \$211.20

Se presenta solo la parte de la pantalla que genera los datos. Ejemplos e al h.

frmAnualNumPagos Form				
Ejemplos	Anualidades: Cálculo del Número de Pagos			
Dep. de: de los datos Optimizar datos *	col1 No. Pagos Sensibilidad 1	col2 No. Pagos Sensibilidad 2	col3 No. Pagos Sensibilidad 3	col4 No. Pagos Sensibilidad 4
	borrar	borrar	borrar	borrar
Valor Presente	5000	5000	5000	5000
Pago	-220	-220	-220	-220
No. de Períodos	24.5829066555	24.2185524094	27.3590081289	26.9599852601
Pago Parcial	-128.24	-48.08	-78.98	-211.20
Tasa de Interés	1 1.25%	2 1.25%	3 1.25%	4 1.25%
Valor Futuro	-500	-500	100	100
Tipo	0	1	0	1

2.9.7 Módulo 5.6 Cálculo del Capital e Interés Pagado.

Calcula tanto el pago de Capital e Interés en un periodo dado, como el acumulado pagado de éstos; de una anualidad basada en una serie de pagos iguales con periodicidad constante y con una tasa de interés fija, el pago puede ser vencido o anticipado.

Pantalla para cálculo del Capital e Interés pagados en un cierto periodo y los acumulados hasta ese periodo.

ES FinAnualIntYCap Form

Ejemplos ? **Anualidades: Capital e Interés Pagado**

Después de teclear datos: oprimir "Entrar"	col1.Sensibil.1	col2.Sensibil.2	col3.Sensibil.3	col4.Sensibil.4
Valor Presente	borrar	borrar	borrar	borrar
No. de Periodos	0	0	0	0
Tasa de Interés 1	0.00% 2	0.00% 3	0.00% 4	0.00%
Valor Futuro	0	0	0	0
Tipo	0	0	0	0
Periodo	0	0	0	0
Interés: a pagar en este Periodo	0.00	0.00	0.00	0.00
pagado hasta este Periodo	0.00	0.00	0.00	0.00
Capital: a pagar en este Periodo	0.00	0.00	0.00	0.00
pagado hasta este periodo	0.00	0.00	0.00	0.00

Borrar Todo

Copiar Datos de Columna 1 a Columna 2

El interés y los pagos deben tener la misma periodicidad
 Para calcular interés: reemplazar la expresión algebraica abajo
 Ejemplo: 07/12 1 07 11/12 1 Entrar Tab o clic en otro campo

Expresión: _____
 Resultado: _____

Pasar a Interés 1 2 3 4

Especificación de argumentos (los conceptos se definen al inicio del capítulo):

- **Valor Presente** es el monto inicial del préstamo (Puede ser un **Saldo Inicial**).
 - Si Valor Presente es cero, entonces debe especificarse Valor Futuro.
- **Interés.** Su periodicidad debe estar acorde al pago periódico. Se pueden hacer cálculos en el área inferior derecha (llamada **Expresión**), después clic en "pasar a" Interés 1, 2, 3 o 4.
- **Valor Futuro** o **Saldo Final** (parámetro opcional). Cantidad a pagar junto con el último pago (en caso de un préstamo).
 - Si Valor Futuro es cero, entonces debe especificarse Valor Presente.
- **Tipo de Pago.** Momento en que se hace el pago:
 - 0: vencidos, pago al final del periodo;
 - 1: anticipados, pago al principio del periodo.
- **Periodo.** Es el periodo para el cuál se requiere conocer el Capital e Interés pagados.

Nota: Pago no es un parámetro necesario en este cálculo.

Ejemplos:

1) Se obtiene un préstamo por \$150,000 con una tasa del 14% anual para un automóvil por el cual se pagarán mensualidades por 48 meses. Se desea conocer el total de interés a pagar.

Datos:

VP: 150000. Positivo, entrada de dinero.
 No. de Periodos: 48
 Interés: 0.14/12 = 1.17%
 Período: 48 (En este período se obtiene el total de pagos de Capital y de Interés).
R: Interés este período: 47.27 (Interés a pagar en el período 48)
 hasta este período: 46,750.63 (Total de interés que se paga en toda la vida del préstamo)
 Capital este período: 4,051.70 (los pagos mensuales son de esta magnitud)
 hasta este período: 150,000.00 (Se liquida completamente el préstamo)

2) Mismo que (1), con pagos anticipados.

R: Interés este período: 46.72 (Interés a pagar en el período 48)
 hasta este período: 44,481.68 (Total de interés que se paga en toda la vida del préstamo)
 Capital este período: 4004.98 (los pagos mensuales son de esta magnitud)
 hasta este período: 150,000.00 (Se liquida completamente el préstamo)

La pantalla muestra comparación entre el plan con pagos vencidos y con pagos anticipados.

IrAnualInYCap : Form

Ejemplos ? **Anualidades: Capital e Interés Pagado**

	col1.Sensibil1	col2.Sensibil2	col3.Sensibil3	col4.Sensibil4
	borrar	borrar	borrar	borrar
Valor Presente	150000	150000	0	0
No. de Periodos	48	48	18	18
Tasa de Interés 1	1.17%	1.17%	0.54%	0.54%
Valor Futuro	0	0	175000	175000
Tipo	0	1	0	1
Período	48	48	18	18
Interés: a pagar en este Período	-47.27	-46.72	892.80	887.99
pagado hasta este Período	-46750.63	-44481.68	7919.48	7876.82
Capital: a pagar en este Período	-4051.70	-4004.98	-10175.05	-10120.23
pagado hasta este período	-150000.00	-150000.00	-175000.00	-174057.19

Borrar Todo

Copiar Datos de Columna 3 a Columna 4

El interés y los pagos deben tener la misma periodicidad
 Para calcular interés, tener en cuenta que la periodicidad a utilizar.
 Ejemplos: 0.07/12 = 1.07/12 = 1. Eje: Tasa = 0.07/12 = 0.583%

Expresión: 0.065/12
 Resultado: 0.0054

Passar a Interés 1 2 3 4

Los siguientes ejemplos también se presentan en la pantalla de arriba.

- 3) Se desea conocer el rendimiento que se generará al hacer depósitos mensuales por 18 meses para acumular un Capital de \$175,000; el banco otorga el 6.5% anual sobre el ahorro.

Datos:

No. de Periodos: 18
Interés: $0.065/12 = 0.54\%$
VF: 175000. Positivo, entrada de dinero.
Período: 18.

R: Interés este período: 892.80
hasta este período: 7,919.48 (Total de Interés generado por los depósitos)
Capital este período: 10,175.00 (éste es el monto de cada depósito)
hasta este período: 175,000.00

- 4) Mismo que (3), con pagos anticipados.

R: Interés este período: 887.99
hasta este período: 7,876.82 (Total de Interés generado por los depósitos)
Capital este período: 10,120.23 (éste es el monto de cada depósito)
hasta este período: 175,000.00

2.9.8 Módulo 5.7. Cálculo del Valor Presente Neto (VPN).

Se calcula el valor presente de una serie de pagos periódicos que pueden ser iguales o desiguales; con tasa de interés constante o variable.

Pantalla inicial mostrando notas para el usuario. Las notas no aparecen en la pantalla.

Valor Presente Neto Instrucciones/Ayuda

Pagos Iguales ó Desiguales con Tasa de Interés Constante ó Variable

Teclee los Pagos y el Interés de cada periodo.

Pago	Interés
	0.0000%

En esta tabla se especifican los flujos y la tasa de interés del periodo correspondiente.

El número de pares de datos a introducir está limitado a 1,000.

Existen 4 memorias: M1 a M4 y la suma de ellas.

Adicionalmente se pueden realizar cálculos de expresiones.

Calcular VPN de los Valores en la Tabla

Valor Presente Neto

Posar a: **M1 M2 M3 M4**

Cambiar signo a: Sumar? Borrar M's

M1	<input type="checkbox"/>		M1
M2	<input type="checkbox"/>		M2
M3	<input type="checkbox"/>		M3
M4	<input type="checkbox"/>		M4
Σ	<input type="checkbox"/>		M5

El interés y los pagos deben tener la misma periodicidad.
Ejemplo: 1 año, 2 cuotas por periodo, algebraicamente: 1 año, 2 periodos.
(77) 2 1 03 11 121 Enter Tab o clic en otro grupo

Expresión:

Resultado:

Record: 14 of 1

Pasos a seguir:

1. Especificarse la serie de pagos junto con el interés del periodo, en la tabla que se provee.
 - El interés debe estar en concordancia con la periodicidad del pago: mensual, semestral, anual, etc.
 - Para hacer cálculos se puede utilizar el área inferior derecha de la pantalla (máximo 255 caracteres).
2. Clic en comando <Calcular VPN de los valores en la tabla>.

Uso de las memorias M1, M2, M3, M4 y M5 o Suma.

- Las memorias M1-M4 pueden almacenar datos durante la sesión; M5 sólo almacena sumas.
- El VPN se puede guardar en M1, M2, M3 o M4, al usar el comando <pasar a>: M1, M2, M3 o M4.
- Clic en la caja-chequeo "Sumar" adyacente a la memoria permite que ese valor sea sumado cuando se oprime el comando <Σ>, junto a M5.
- Clic en el comando <Cambiar signo a Mi> i = 1, 5 permite manipular el signo de la cantidad almacenada en la memoria i, para realizar sumas o sustracciones al oprimir el comando <Σ>, junto a M5.
- Borrar M's. Permite borrar la memoria que se deseó: M1, M2, M3, M4 o M5.

Ejemplos usando las memorias:

-3 pagos consecutivos:

per 1: 100 1.667%

per 2: 155 1.667%

per 3: 280 1.667%

VPN: \$514.77

-3 Pagos periódicos, omitiendo pago en un mes, con tasas iguales:

per 1: 100 1.667%

per 2: 155 1.667%

per 3: 0 1.667%

per 4: 280 1.667%

VPN: \$510.40

Calcúlese VPN para ejemplo 1; pasar a M1; clic en el cuadro "sumar";

M1 = 514.77

Ahora VPN para ejemplo 2; pasar a M2; clic en el cuadro "sumar"; cambio de signo a M2;

M2 = -510.40

-Comando sumar (letra sigma junto a M5);

M5 = 4.37

La pantalla muestra el ejemplo descrito. Adicionalmente muestra operaciones que se pueden realizar con los campos Expresión y Resultado.

FormVPvar Form		Valor Presente Neto		Instrucciones/Ayuda	
Pagos Iguales ó Desiguales con Tasa de Interés Constante ó Variable					
Teclee los Pagos y el Interés de cada periodo.					
	Pago	Interés	Datos VPN		
	100.00	1.6670%	ejemplo 2. Los		
	155.00	1.6670%	del ejemplo 1		
	0.00	1.6670%	desaparecen.		
	280.00	1.6670%			
*	0.00	0.0000%			
<p>VPN del ejemplo 1 se pasa a M1. VPN del ejemplo 2 a M2, etc.</p> <p>a M2 se le cambia el signo</p> <p>Se indica que se sumarán M1 y M2 (en M5) (M3 y M4 no son sumadas).</p> <p>Se pueden realizar operaciones usando el campo llamado expresión.</p>					
Record: 11/4			4 of 4		
Calcular VPN de los Valores en la Tabla			Valor Presente Neto 510.40		
Pasará a: M1 M2 M3 M4			cambiar signo a: Sumar? Borrar M's		
M1	<input checked="" type="checkbox"/>	514.77	M1	<input type="checkbox"/>	
M2	<input checked="" type="checkbox"/>	-510.40	M2	<input type="checkbox"/>	
M3	<input type="checkbox"/>	485.86	M3	<input type="checkbox"/>	
M4	<input type="checkbox"/>	501.18	M4	<input type="checkbox"/>	
		4.37	M5	<input type="checkbox"/>	
El interés y los pagos deben tener la misma periodicidad					
Calcule e interese de un ejemplo de pago en un mes. Ejemplo: 100 1.37 12 1. Eder Tabla de 100 1.37 12 1.37					
Expresión: 485.86 - 501.18			Resultado: -15.3200		

Cálculos y/o valuación de expresiones.

Para hacer cálculos utilícese el área inferior derecha de la pantalla.

Reglas para la expresión:

Deben usarse los operadores

Ejemplos:

Expresión = 485.86 - 501.18

Resultado = -15.3200

2.9.9 Módulo 5.8. Depreciación.

Se generan las tablas de depreciación para los métodos de Línea Recta y de Doble Declinación del Saldo (o el factor que se especifique). Tales tablas indican el período; la Depreciación del período; la Depreciación Acumulada a dicho período, y el Costo Remanente, el cual equivale al Costo menos la Depreciación Acumulada hasta ese período.

Form Depreciación

Depreciación Instrucciones/Ayuda

Método de Línea Recta

per	Depr	DeprAcum	CostoRemanente
0	0.00	0.00	0.00

Después de especificar los parámetros adecuados, al oprimir <Calcular> se despliega en esta área la tabla del método depreciación de línea recta.

Record: 1 of 1

Método de Doble Declinación del Saldo (u otro Factor)

per	Depr	DeprAcum	CostoRemanente
0	0.00	0.00	0.00

En esta área aparece la tabla del método de depreciación de declinación acorde al factor especificado.

Record: 1 of 1

Los siguientes parámetros son comunes a ambos métodos:

- **Costo.** Es el valor inicial del bien a depreciar.
- **Valor de Salvamento.** Es el valor del bien después que ha sido completamente depreciado. Puede ser cero.
- **Vida Util.** Es el número de años que un bien está activo. Debe ser un número entero.

Al oprimir el comando <Calcular> se genera la tabla que muestra la depreciación.

- **Método de Línea Recta.** La depreciación es idéntica en cada uno de los períodos. La depreciación es el Costo dividido por el número de años de Vida Util; previa sustracción del Valor de Salvamento:

$$\text{Depreciación} = (\text{Costo} - \text{Valor Salvamento}) / \text{Vida Util}$$

- **Método de Doble Declinación del Saldo (ú otro Factor).** La depreciación es mayor en los períodos iniciales que en los finales.
 - Se utiliza además de los arriba mencionados, el siguiente parámetro
 - **Factor:** indica la velocidad de depreciación del bien.

Se usa la fórmula: Depreciación = Costo(i)/Vida Útil * Factor; donde Costo(i) es el costo remanente en el período i; i = 1, Vida Útil

- Los comandos <Borrar Parámetros y tabla> inicializan el área respectiva en que se encuentra cada uno.

Ejemplos (se comparan ambos métodos en la pantalla):

Costo: 100,000
 Valor de salvamento: 20000
 Vida útil: 7 años

Adicionalmente para método de doble declinación:
 factor de declinación: 2 (valor de default).

La pantalla muestra una comparación de ambos métodos. Nótese que en doble declinación se deja de depreciar antes del tiempo límite debido a que ya llegó a su valor de salvamento.

ImDepreciacion Form
Instrucciones/Ayuda

Depreciación

Método de Línea Recta.

	per	Depr	DeprAcum	CostoRemanente	
Borrar Parámetros y Tabla	1	11428.57	11428.57	88571.43	Se deprecia un valor constante para los años dados.
Costo	2	11428.57	22857.14	77142.66	
Valor de Salvamento	3	11428.57	34285.71	65714.29	
Vida Útil	4	11428.57	45714.29	54285.71	
Factor	5	11428.57	57142.86	42857.14	
Depreciación	6	11428.57	68571.43	31428.57	
Calcular	7	11428.57	80000.00	20000.00	
	*	0	0.00	0.00	

Record: 1 of 7

Método de Doble Declinación del Saldo (ú otro Factor)

	per	Depr	Depr Acum	CostoRemanente	
Borrar Parámetros y Tabla	1	28571.43	28571.43	71428.57	Deja de depreciarse al llegar al límite.
Costo	2	20408.16	48979.59	51020.41	
Valor de Salvamento	3	14577.26	63556.85	36443.15	
Vida Útil	4	10412.33	73969.18	26030.82	
Factor	5	6030.82	80000.00	20000.00	
Depreciación	6	0.00	80000.00	20000.00	
Calcular	7	0.00	80000.00	20000.00	
	*	0	0.00	0.00	

Record: 1 of 7

2.9.10 Módulo 5.9. Tasa Interna de Retorno.

Se calcula la Tasa Interna de Retorno de una serie de flujos de efectivo.

Pantalla inicial.

Tasa Interna de Retorno

Instrucciones/Ayuda

Teclear los flujos de efectivo ó traer un archivo

Valor

Los flujos se capturan en la tabla proporcionada.

El primer flujo debe ser una inversión (negativo), después puede haber inversiones ó retornos de capital.

La T.I.R debe ser positiva, de otro modo los retornos no cubren la inversión.

Record: 1 of 1

Calcular T.I.R.

TIR:

Borrar Pantalla

Archivo: Traer Archivo

Guardar Datos en Archivo

Archivo: Borrar Archivo

Instrucciones:

- Tecléense los flujos en la tabla que se provee.
 - Al menos debe haber una inversión (Las salidas de dinero se especifican con números negativos).
 - Al menos debe haber un retorno (Los ingresos se especifican con números positivos, puede ser cero).
 - Generalmente se empieza con una Inversión y después se pueden tener Retornos o Inversiones; cuando en un determinado período se hace una inversión y se obtiene un retorno, las cantidades se "saldan" y se tecldea el valor resultante, pudiendo ser mayor o menor que cero, según sea el caso.
- <Calcular TIR>. Este comando calcula la Tasa Interna de Retorno a partir de los datos dados.

<Borrar pantalla> Inicializa la pantalla, borrando los datos que se presentan.

Ejemplo:

Se invierten \$1,000 y se obtienen los siguientes retornos: \$600, \$-100, \$600 (esto es, se invierten 1,000 en el periodo 1; se recuperan \$600 en el segundo periodo; se vuelven a invertir \$100 el siguiente periodo y finalmente se recuperan otros \$600 en el periodo 4).

La tabla queda:

-1000
600
-100

600:
TIR. = 4.9476%

La pantalla muestra el ejemplo mencionado.

Valor	Datos del último ejemplo
-1000	
600	
-100	
600	
0	

Los flujos se pueden guardar en un archivo y acceder posteriormente.

También se proporciona un comando para borrar archivos de datos.

Calcular T.I.R.

TIR: 4.9476%

Borrar Pantalla

Archivo: ? Traer Archivo

Guardar Datos en Archivo

Archivo: Borrar Archivo

Record: 14 of 4

Manejo de archivos de datos.

<Traer Archivo>. Transfiere datos guardados en un archivo, al área de captura.

- Clic en "Archivo". De la lista que aparezca, selecciónese el que se desea acceder.
- Clic en <Traer Archivo>. Se transfiere una copia de los datos del archivo al área de captura. Ahí los datos pueden ser modificados y se pueden agregar o borrar datos. Los datos en el archivo permanecerán intactos.

<Guardar Datos en Archivo>. Permite guardar datos permanentemente en un archivo de usuario.

- Al oprimir este comando aparece una pantalla preguntando el nombre para el archivo, el cuál:
 - o Debe ser de un máximo de 15 caracteres de longitud. De excederse, EmCalc no lo guarda.
 - o Al nombre se le agrega el prefijo "TIR", este prefijo no cuenta para el límite de 15 caracteres.
 - o No existe restricción en caracteres a usar para el nombre.

Si se guarda un archivo con un nombre de archivo ya existente, el nuevo archivo sustituirá al viejo perdiéndose los datos del archivo viejo.

<Borrar Archivo> Borra permanentemente los archivos.


- Clic en "Archivo". De la lista que aparece, selecciónese el que se desea borrar.
- Al hacer clic en <Borrar Archivo> El archivo de datos será borrado permanentemente

Nota: Si la "caja" de Archivos esta vacía, esto indica que no hay archivos a borrar.

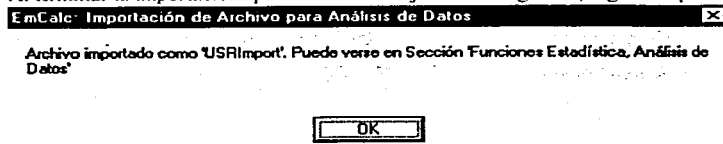
Para archivos de Análisis de Datos (1 columna, llamada "Valor"), guárdese el archivo como:
C:\My Documents\EmCalcDat.XLS

Para archivos de Análisis de Regresión (2 columnas, llamadas "X" y "Y"), guárdese el archivo como:
C:\My Documents\EmCalcREG.XLS

Si el directorio no existe, es necesario crearlo, de otro modo no se podrá realizar la importación.

Para realizar la importación oprímase la caja-comando:  , perteneciente a la opción que desea realizar.

Al terminar la importación aparecerá un mensaje como el siguiente, según la opción tomada.



Los archivos importados podrán accederse como:

- USRImport para la sección de Análisis de datos.
- UXYImport para Análisis de Regresión.

Estos archivos pueden accederse en las secciones mencionadas y pueden ser borrados por el usuario, para crear copias de ellos hágase lo siguiente:

- Accésese el archivo en el área respectiva (como USRImport o UXYImport)
- Guárdese el archivo con un nombre diferente.

Para mayor referencia de como accesarlos/guardarlos, véase la sección correspondiente.

Nota: Sólo puede importarse un archivo de cada sección a la vez. Cada vez que se importa un archivo, se borra el archivo anterior (si acaso existe) y se crea el nuevo. En otras palabras, si se desea importar mas de un archivo, de Análisis de Datos, por ejemplo, el proceso a seguir es:

- Importar primer archivo.
- Crear una copia de él (con el método arriba descrito).
- Repetir este proceso para cada archivo a importar.

Instrucciones para compactar la base de datos (EmCalc).

Después de utilizar repetitivamente EmCalc (o cualquier base de datos de Access), se requiere realizar un proceso llamado Compactación y Reparación.

El tamaño inicial de EmCalc es de 3,200 KB (aproximadamente) el cuál con cada utilización se va incrementando. El compactar hará que regrese a su tamaño original, excepto por los archivos que el usuario haya creado o borrado. El tamaño máximo de crecimiento de esta versión es 500MB, el cuál es, para efectos prácticos, imposible de alcanzar.

En Access 97 hay que emitir cada proceso por separado, primero compactando y después reparando.

Comando de Access 97:

- <Tools>, <Database Utilities>, <Compact Database>
- <Tools>, <Database Utilities>, <Repair Database>

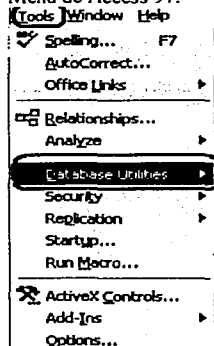
Access 2000 realiza ambos procesos a la vez.

Comando de Access 2000:

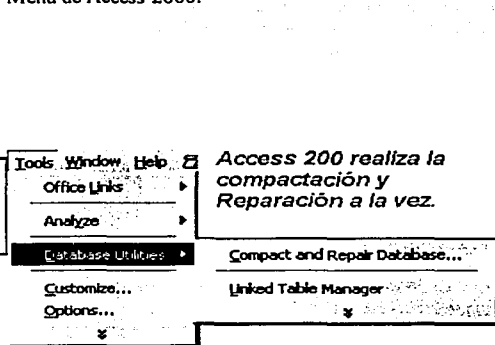
- <Tools>, <Database Utilities>, <Compact and Repair Database>

Nota: En cada caso el usuario debe proporcionar la palabra clave "EmCalc" para volver a conectarse.

Menú de Access 97.



Menú de Access 2000.



3. Diseño y desarrollo de EmCalc.

Se muestra el diseño y desarrollo de EmCalc por medio de los diferentes elementos de Access que se emplean en su creación

3.1 Elementos de Access y Visual Basic Empleados.

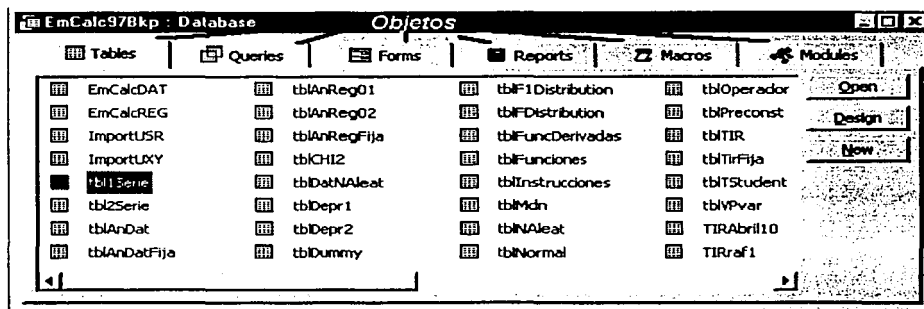
El software empleado para desarrollar EmCalc es Access, auxiliándose del lenguaje de programación Visual Basic que provee capacidad para expandir las funciones de Access; ambos softwares de la compañía Microsoft. El desarrollo se llevó a cabo utilizando Access 97.

Access, es un software para desarrollar aplicaciones para bases de datos, las cuáles proveen la facilidad de la manipulación masiva de información. Tales aplicaciones consisten principalmente en el llenado de tablas de información, por medio de formas (pantallas) que pueden filtrar la información para evitar errores de captura o repeticiones inadvertidas; permite el enlace de información entre distintas bases de datos. La información almacenada o enlazada puede ser extraída selectivamente y manipulada para ser desplegada o transferida a otras bases de datos o archivos.

De Access se enuncian principalmente los elementos empleados para desarrollar EmCalc. Algunos elementos adicionales se mencionan para dar una idea más general de lo que Access y/o Visual Basic pueden realizar. La parte final es complementada por una lista de los elementos empleados por Access.

Para el diseño de bases de datos, Access provee los siguientes objetos, los cuales se definen o se describen ya sea directamente o por medio de sus funciones o sus características., otros elementos inherentes a cada uno de estos objetos serán definidos o descritos posteriormente:

Pantalla de Access mostrando los objetos que se pueden utilizar en el diseño de una base de datos.



3.1.1 Tablas. Las tablas se proveen para el almacenamiento de información. Generalmente se genera una tabla con campos que tienen información relacionada. En EmCalc hay tablas pre-cargadas con información, y hay tablas que se usan para que el usuario capture información, éstas tablas creadas por el usuario pueden ser guardadas permanentemente.

Una tabla puede ser utilizada o vista de las 2 siguientes maneras:

- Diseño. En esta vista se crean los nombres de campos a emplear y el tipo de valores para cada campo (numérico, alfa-numérico, fecha, etc); longitud del campo, valores válidos; etc. También se especifican los campos llave de esa tabla.

Field Name	Data Type	Description
Nombre	Text	Nombre de la función
Funcion	Text	Expresión de la función
Tipo	Text	F(x) ó F(x,y)

Field Properties	
General	Lookup
Field Size	6
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	No
Indexed	No

The field description is optional. It helps you describe the field and is also displayed in the status bar when you select this field on a form. Press F1 for help on descriptions.

- Tabla tblFunciones en vista de diseño. Nótese que al campo Nombre es antecedido por el dibujo de una llave lo cuál indica que es la llave de la tabla, esto produce que el función sea único, es decir, se evita que haya nombres duplicados.

- Captura de datos (ó hoja de cálculo). En esta vista, la tabla aparece agrupada por columnas y renglones. Las columnas corresponden al nombre del campo. Cada renglón define un registro formado por celdas y en ellas se puede almacenar información manualmente. Otras maneras de agregar información a una tabla es por medio de formas (pantallas), queries, código de programación, etc. EmCalc no permite al usuario el llenado de una tabla directamente, lo hace por medio de una forma.

Nombre	Función	Tipo
Espiral1	$3 \cdot \cos(x+4) / \sqrt{1+x^2}$	F(x)
Espiral2	$5 \cdot \sin(x+3) / \sqrt{1+x^2}$	F(x)
f1xy	$\sin(x) \cdot \cos(y)$	F(x, y)
Func11/01 7:57:55 PM7:5	$X^2 + \log(x) / (x+x^2) + 25 \cdot x^4$	F(x)

Fragmento de la tabla tblFunciones en vista "Hoja de Cálculo". Almacena funciones de usuario de EmCalc.

En EmCalc las tablas se utilizan (se mencionan procesos y objetos aún por definir):

- Para captura de datos por el usuario (ligando la tabla a una subforma de captura).
- Para guardar archivos de datos del usuario (creando copias de las tablas en que el usuario captura datos).
- Para guardar cálculos de información a mostrar. Se despliega la información calculada ligando la tabla a una subforma que se despliega en forma continua.
- Para proporcionar las cajas-combo para proporcionar al usuario las opciones que tiene para seleccionar de ahí una.
- Para crear los datos temporales de las secciones de graficación.
- Para importar los archivos de Excel.
- Toda la información permanente (datos), reside en tablas.

Al final de esta sección se listan las tablas que se crearon para EmCalc.

3.1.2 Queries. La palabra "query" puede significar **Vista** o **Búsqueda** en el contexto de bases de datos. Se usará el nombre que se le da al objeto en inglés para evitar confusión.

Un query tiene varias funciones, entre ellas:

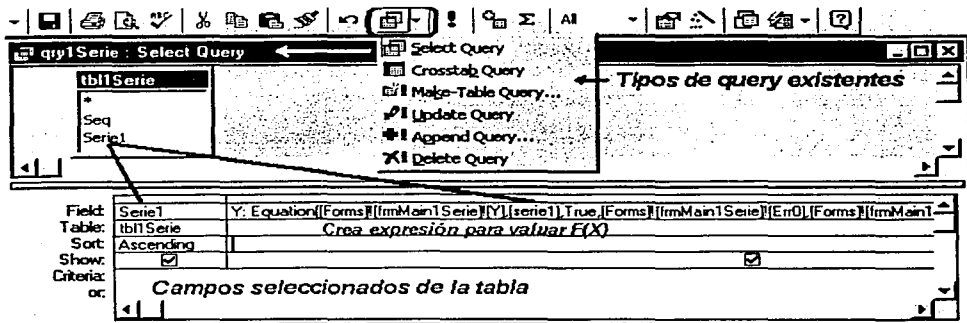
- Proveer medios para ver una tabla (u otro query). Al usar un query se puede ver la misma información de una tabla, se pueden omitir columnas, cambiar su orden de presentación. Se pueden realizar cálculos entre columnas o crear nuevas columnas.
- Relaciona o combina tablas. Dos o más tablas se pueden relacionar en un query por medio de campos comunes para obtener información almacenada en cada una de ellas; para determinar que elementos de cada tabla son iguales, o cuales no ocurren en alguna de las tablas.
- Filtra información. Proporciona la facilidad de extraer de las tablas sólo cierta información de acuerdo a un criterio, por ejemplo, si un campo es fecha, se pueden extraer los registros que se encuentren entre determinadas fechas.

Access define cuatro tipos de queries:

- Para sólo mostrar información (Select queries: queries de selección).
- Para crear tablas de la información extraída (Make-table queries: queries para creación de tablas).
- Para agregar a una tabla la información extraída (Append queries: queries para agregar registros).
- Para actualizar información en una tabla (Update queries: queries de actualización de registros).

Un query puede tener 2 vistas:

- Diseño tipo Access. Aquí se especifica lo siguiente:
 - Tipo de query a emplear: Select, Make-table, Append, Update.
 - Tablas u otros queries de donde se extraerá la información, creando relaciones entre ellos.
 - Campos a extraer y/o cálculos a realizar entre ellos.
 - Criterios que la información debe cumplir para ser extraída.



Query en vista de diseño.

Serial	Y
1	0.45464871341
2	-1.5136049906
3	-1.2573697419
4	7.91486597299
5	-6.8002638861
6	6.6693427741

Record: 1 of 7

Query en vista "Hoja de Cálculo". Este tipo de queries se utilizan para mostrar la información en EmCalc y para el diseño de las gráficas.

Este tipo de queries (de Selección), pueden ser empleados como entrada para otros queries, junto con otros queries ú otras tablas.

Vistas de diseño de los queries para borrado, agregar datos y actualizar. Es la única vista que permite Access para este tipo de queries, por ser queries de acción.

tblAnReg01 **Tipo de Query**

Columnas a borrar: (* = todas)

Field: tblAnReg01.*
Table: tblAnReg01
Delete: From
Criteria:
or:

tblTirFija **Agrega registros al final de esta columna**

Field: Valor
Table: tblTirFija
Sort:
Append To: Valor
Criteria:
or:

tblAnReg01 **Query para actualización.**

Aquí se actualizan todos los registros usando una expresión para TransfX y TransfY; mientras que Yajuste y Residual son inicializados a cero.

Field: TransfX, TransfY, Yajuste, Residual
Table: tblAnReg01, tblAnReg01, tblAnReg01, tblAnReg01
Update To: If([IsNull]([Forms]![frmAn]...), If([IsNull]([Forms]![frmAn]...), 0
Criteria:
or:

3.1.3 Formas. Una forma es el diseño de un formato de desplegado que aparecerá en la pantalla al correrse la aplicación. También se le llama "pantalla" a una forma.

Una forma puede ser utilizada en tres maneras diferentes (y las combinaciones de ellas):

- Captura de información para llenar tablas.
- Presentación de información contenida en tablas; esto incluye gráficas.
- Presentación de ayuda o información fija, incluyendo textos y, dibujos o fotografías.

Una forma tiene 2 vistas:

- Diseño. En esta vista se especifica lo siguiente:
 - Si la forma esta o no ligada a una tabla o un query.
 - Objetos propios de una forma que se crean para manejar la información que se desplegará.

Forma en vista de diseño. La forma que se presenta, muestra todos los objetos que se utilizan en el diseño de EmCalc. Para crear una subforma, se arrastra con el ratón la forma que se quiere insertar. Esta forma no esta ligada a una tabla o a un query, pero, la subforma que contiene sí está ligada.

Forma en vista de diseño. La forma que se presenta, muestra todos los objetos que se utilizan en el diseño de EmCalc. Para crear una subforma, se arrastra con el ratón la forma que se quiere insertar. Esta forma no esta ligada a una tabla o a un query, pero, la subforma que contiene sí está ligada.

Esta es una etiqueta se usa como letrero. A veces da indicaciones, otras indica el nombre de una variable.

Al crear cajas-texto, combo ó chequeo, access les genera también una etiqueta, la cuál se puede borrar.

Etiqueta de Caja-Texto | Caja-Texto Representa una variable.

Caja-Chequeo: equivalente a variable booleana.

Etiqueta de Caja-Combo | Caja-Combo. Se usa como variable.

Un comando puede ser creado conteniendo texto ó un dibujo. Ambos pueden generar la misma acción si así se programan.

Caja-Comando. Genera una Acción.

Una forma insertada se convierte en Subforma

Una subforma se diseña con objetos similares. Si se desea se puede desplegar como hoja de cálculo.

Tables

- EmCalc
- EmCalcAbout
- EmCalcDAT
- EmCalcREG
- Form1
- frm1Serie
- frmAnDat1
- frmAnReg
- frmAnReg01
- frmAnReg02
- frmAnRegCh0
- frmAnRegCh0s
- frmAnRegCh1
- frmAnRegCh2
- frmAnualFV
- frmAnualidade:
- frmAnualInt
- frmAnualIntYC
- frmAnualNump

- Forma. Así se le llama a la vista cuando la aplicación está en producción.

La figura muestra la forma anterior, ahora en la vista de producción. Nótese el error en que se incurre al dejar el texto de la caja-texto. Access trata de evaluarlo y produce un error. Nótese también como la subforma despliega la información tipo Hoja de Cálculo.

Esta es una etiqueta se usa como letrero. A veces da indicaciones, otras indica el nombre de una variable.

Al crear cajas-texto, combo ó chequeo, access les genera también una etiqueta, la cuál se puede borrar.

Etiqueta de Caja-Texto

Caja-Chequeo: equivalente a variable booleana.

Etiqueta de Caja-Combo

Un comando puede ser creado conteniendo texto ó un dibujo. Ambos pueden generar la misma acción si así se programan.

Caja-Comando. Genera una Acción.

Una forma insertada se convierte en Subforma

frmAnReg01

X	Y	TransX	TransY	Yajusto	Residual
1.75	74.5	0	0	0	0
1.71	73.4	0	0	0	0

Record: 1 of 3

Record: 1 of 1

- Adicionalmente una forma puede contener una subforma, la cual puede desplegarse en cualquiera de las vistas permitidas, con información ligada a la forma principal o no. (En el caso de Access ninguna forma se liga o se sincroniza a la forma principal).
- Access provee un módulo de programación local que permite controlar los objetos que una forma tiene, así como las acciones a realizar al ocurrir eventos de la forma (al abrir la forma, al cerrarla, etc.). La serie de módulos de programación se puede ver en la sección Código de Programación por Pantalla.

3.1.4 Reportes. Un reporte es el diseño de un formato de desplegado que aparecerá en la pantalla y de ahí se puede imprimir o cancelar la impresión a solicitud del usuario.

- Un reporte se utiliza para la presentación de información (incluyendo gráficas), generalmente proveniente de tablas, pero, no necesariamente.

Un reporte tiene 2 vistas:

- Diseño. En esta vista se especifica lo siguiente:
 - Si el reporte estará o no ligado a una tabla o un query.
 - Encabezado y pie de reporte y de página; detalle a emplear (los datos repetitivos).
- Pre-impresión. Un reporte se puede ver en formato de pre-impresión, el cuál permite ver el diseño realizado del reporte en la pantalla, sin imprimirlo. Se puede imprimir o cancelar la impresión.

Nota: para el diseño de un reporte se utilizan los mismos elementos que en una forma., excepto que en un reporte, no genera acciones, es decir no utiliza cajas-comando.

Access provee un módulo de programación local que permite controlar los objetos que un reporte tiene, así como las acciones a realizar al ocurrir eventos del reporte (al abrirlo, al cerrarlo, al imprimirlo, etc.). EmCalc no hace uso de estos módulos.

En la sección de **Código de Programación por Pantalla** se muestra el diseño de algunos de los reportes que EmCalc genera.

Macros. Una macro se utiliza para ejecutar varias instrucciones sucesivamente sin intervención del usuario. Es parte de la base para automatizar procesos al permitir utilizar instrucciones de control, correr queries, correr código de programación (subrutinas y funciones escritas por el diseñador de la base de datos), etc.

- Las macros sólo tienen la vista de diseño en que se especifica la serie de comandos a utilizar, los cuales forman parte del acervo de Access o han sido desarrollados previamente por el diseñador de la base de datos.

Módulos. Son módulos de programación. Permiten el desarrollo y almacenamiento de subrutinas y funciones de usuario en el lenguaje de programación Visual Basic.

Access provee dos tipos de módulos de programación. El tipo de programación puede ser idéntico, no así la extensión de su utilización:

- Módulos locales. Estos son utilizados en las formas o en los reportes en forma de subrutinas o funciones. En estos módulos se pueden escribir subrutinas o funciones a realizar en los límites de la forma o reporte en cuestión, si son referenciados desde otras formas o reportes no serán reconocidos.
- Módulos generales. Aquí se pueden escribir subrutinas o funciones que pueden ser empleadas en cualquier parte que puedan ser llamadas: queries, formas, reportes, macros y módulos locales o generales.

Adicionalmente Access y Visual Basic proveen una cantidad de recursos para facilidad del programador como:

- Funciones pre-construidas: matemáticas, para manejo de caracteres, informativas, manejo de mensajes, control de macros, etc.
- Manejo de tablas y queries desde Visual Basic. En caso necesario se puede manipular la información de tablas y queries desde un módulo de programación.
- Importación, exportación y enlace de información de y hacia otro software y otros tipos de formato, entre otros: MS-Excel, otras bases de datos Access, Bases de datos Oracle, SQL, formato de texto, etc.

3.1.5 Tablas, Queries y Formas Empleadas en EMCalc.

TABLAS empleadas en EmCalc y su descripción:

Nombre	Descripción
EmCalcDAT	Ejemplo de hoja de cálculo para importar una variable: Análisis de Datos.
EmCalcREG	Ejemplo de hoja de cálculo para importar dos variables: Regresión.
ImportUSR	Tabla de paso para importar archivos de una variable: Análisis de Datos.
ImportUXY	Tabla de paso para importar archivos de dos variables: Regresión.
MSysObjects	Tabla de Access que permite ver los objetos creados.
tbl1Serie	Almacena puntos (X, Y) para graficación.
tblAnDat	Tabla para almacenar datos capturados para Análisis de Datos.
tblAnDatFija	Tabla auxiliar para traer datos almacenados, para Análisis de Datos.
tblAnReg01	Tabla para almacenar datos capturados para Regresión.
tblAnReg02	Tabla para almacenar datos calculados a desplegar, de Regresión.
tblAnRegFija	Tabla auxiliar para traer datos almacenados, para Regresión.
tblAyudas	Auxiliar para crear documentación para EmCalc.
tblCHI2	Almacena datos de la Ji cuadrada.
tblDatNAleat	Almacena parámetros para crear números aleatorios (Cantidad, Mínimo, Máximo).
tblDepr1	Almacena la tabla de depreciación por el método lineal.
tblDepr2	Almacena la tabla de depreciación por el método "factor".
tblDummy	Auxiliar para crear tablas con un sólo dato.
tblF1Distribution	Almacena puntos de la distribución F, 1% de significancia.
tblFDistribution	Almacena puntos de la distribución F, 5% de significancia.
tblFormas	Auxiliar para crear documentación de EMCalc.
tblFuncDerivadas	Funciones preconstruídas para EmCalc: valuación de funciones.
tblFunciones	Almacena las expresiones de funciones creadas por el usuario: $F(X)$ y $F(X, Y)$.
tblInstrucciones	Contiene las páginas de ayuda y mnemónicos.
tblMdn	Almacena datos generados para Análisis de datos.
tblNAleat	Almacena los números aleatorios.
tblNormal	Datos de la tabla de distribución normal.
tblOperador	Contiene los operadores de la sección de Operaciones y funciones básicas.
tblPreconst	Expresiones de funciones preconstruídas: Operaciones y Funciones Básicas.
tblTIR	Almacena los flujos para calcular la TIR.
tblTirFija	Tabla auxiliar para traer datos almacenados, para TIR.
tblTStudent	Almacena puntos de la función t de Student.
TbIVPvar	Almacena (Pagos, tasa) para Valor Presente Neto.
TIRnombre	Archivo de TIR.: EmCalc agrega el prefijo TIR al nombre dado por el usuario.
USRImport	Archivo de Análisis de datos: EmCalc agrega el prefijo USR al nombre del archivo.
UXYReg01	Archivo de Regresión: EmCalc agrega el prefijo UXY al nombre del archivo.

QUERIES y tipo de query (Muestra indica que sólo muestra o selecciona datos).

Tipo	Nombre	Descripción
Muestra	qry001_01NombresData	Archivos Análisis de Datos.
Muestra	qry01AnDat	Cuenta no. de observaciones y ordena de menor a mayor.
Muestra	qry01AnReg00	Despliega X, Y, TransfX, TransfY, Ajuste(Y), Residuales.
Muestra	qry01AnReg01	Despliega X, Y, TransfX, TransfY, Ajuste(Y), Residuales.
Muestra	qry1Serie	Despliega X y F(X), según la expresión que se tenga para F(X).
Muestra	qry2Serie	Despliega X y F(X), según la expresión que se tenga para F(X).
Muestra	qry3Serie	Despliega X y F(X), según la expresión que se tenga para F(X).
Muestra	qry4Charts	Despliega X y valúa F1(X), F2(X), F3(X), F4(X) y F5(X) para graficar.
Muestra	qryMdn	Cuenta No. de datos y despliega las observaciones agrupadas.
Muestra	qryVPvar	Relaciona subforma de captura de datos con tabla.
	Borra qry01_1RegDelXY	Borra datos de la tabla de captura de regresión
	Borra qry01_8AnRegDel	Borra datos de la tabla de captura de regresión
	Borra qry01DelTir	Borra datos de captura de tabla de TIR.
	Borra qry02AnDat	Borra la tabla de captura de análisis de datos.
	Borra qry02AnRegDel	Borra la tabla de desplegado de Análisis de regresión
	Borra qryBorraFuncion	Borra función de usuario desde F(X) (una a la vez).
	Borra qryBorraFuncionXY	Borra función de usuario desde F(X,Y) (una a la vez).
	Borra qryDel1Serie	Borra datos de captura de graficación de puntos manuales
	Borra qryDel2Serie	Borra datos de captura de graficación de puntos manuales
	Borra qryDelDepr1	Borra datos de captura de la tabla de depreciación lineal.
	Borra qryDelDepr2	Borra datos de captura de la tabla de depreciación Doble declinación.
	Borra qryDelMdn	Borra datos de desplegado de la tabla análisis de datos.
	Borra qryDelNosAleat	Borra la tabla de números aleatorios.
	Borra qryDelSerie1	Borra datos de captura de graficación de puntos manuales
	Borra qryDelTbInoAleat	Borra la tabla de parámetros para números aleatorios.
	Borra qryDelVPvar	Borra datos de captura de Valor Presente Neto.
	Borra qryUXYVacialmport	Borra datos de tabla de importación de Excel para regresión
	Borra qryVacialmportUSR	Borra datos de tabla de importación de Excel para análisis de datos
Actualiza	qry01_2RegDelX	Inicializa valores de X, en la tabla de datos de regresión
Actualiza	qry01_3RegDelY	Inicializa valores de Y, en la tabla de datos de regresión
Actualiza	qry01_4TransfXYUpd	Inicializa valores de X y Y, en la tabla de datos de regresión
Actualiza	qry01_6Yajuste	Valúa el ajuste (X) en la regresión y actualiza su valor en la tabla
	Agrega qry001_02AppendAnDat	Agrega datos de un archivo, a la tabla de captura de análisis de
	Agrega qry01_7AppRegXY	Agrega datos de un archivo, a la tabla de captura de regresión.
	Agrega qry01AppTirFija	Agrega datos de un archivo, a la tabla de captura de TIR
	Agrega QryGuardaAjuste	Guarda el modelo de ajuste en el archivo de funciones de usuario
	Agrega qryGuardaFuncion	Guarda la expresión de F(X) en el archivo de funciones de usuario
	Agrega qryGuardaFuncionXY	Guarda la expresión de F(X, Y) en el archivo de funciones de usuario

FORMAS o PANTALLAS:

Nombre	Descripción
EmCalc	Pantalla Inicial de la aplicación
EmCalcAbout	Pantalla de Información "Acerca de EmCalc"
EmCalcDAT	Ejemplo de como crear el archivo Excel para Análisis de datos.
EmCalcREG	Ejemplo de como crear el archivo Excel para regresión.
frm1Serie	Auxiliar para captura de datos de 1 sola variable
frmAnDat1	Pantalla de Análisis de Datos.
frmAnReg	Pantalla de Análisis de Regresión.
frmAnReg01	Se capturan (X, Y) para regresión, contiene: transfX, transfY, Ajuste, Residual.
frmAnReg02	Contiene cálculos a desplegar de la regresión.
frmAnRegCh0	Pantalla de la gráfica de (X, Y) para Análisis de Regresión.
frmAnRegCh1	Pantalla de la gráfica de (transf(X), transf(Y)) para Análisis de Regresión.
frmAnRegCh2	Pantalla de la gráfica de (transf(X), transf(Y)) y Ajuste de Regresión.
frmAnualFV	Pantalla para Anualidades: Valor Futuro.
frmAnualidades	Pantalla para Anualidades. Controla que proceso de anualidades realizar.
frmAnualInt	Pantalla para Anualidades: Cálculo del Interés.
frmAnualIntYCap	Pantalla para Anualidades: Cálculo del Capital y el interés pagados.
frmAnualNumPagos	Pantalla para Anualidades: Cálculo del número de pagos.
frmAnualPago	Pantalla para Anualidades: Cálculo del pago periódico.
frmAnualVP	Pantalla para Anualidades: Valor Presente.
frmCh0	Pantalla de la gráfica para puntos generados manualmente.
frmCh01AnDat	Pantalla de la gráfica del histograma de Análisis de Datos.
frmCh1	Pantalla de la gráfica para puntos generados automáticamente.
frmDepr1	Forma para presentar tabla de depreciación, método lineal.
frmDepr2	Forma para presentar tabla de depreciación, método "factor".
frmDepreciación	Pantalla general para presentar dos tipos de depreciación.
frmFórmula	Pantalla para crear las expresiones F(X) de una variable.
frmFórmulaXY	Pantalla para crear las expresiones F(X, Y) de dos variables.
frmInstrucciones	Pantalla para presentar las ayudas.
frmMain1Serie	Pantalla para generar puntos a graficar manualmente.
frmMain2Serie	Pantalla para generar automáticamente puntos a graficar.
frmMain4Ch	Pantalla que muestra la gráfica de funciones múltiples.
frmMain4Serie	Pantalla para generar puntos para graficación múltiple.
frmMdn	Forma para mostrar los datos calculados para Análisis de Datos.
frmQry1Serie	Forma para mostrar datos de la tabla tbl1Serie.
frmQry2Serie	Forma para mostrar datos de la tabla tbl1Serie.
frmQry3Serie	Forma para mostrar datos de la tabla tbl1Serie.
frmAnDat	Pantalla general para Análisis de Datos.
frmTir0	Pantalla para presentar la Tasa Interna de Retorno.
frmTir1	Forma para presentar cálculos realizados de la Tasa Interna de Retorno.
frmUtility	Pantalla que presenta las "Utilerías y Solución de Problemas".
frmVPvar	Pantalla que presenta el Valor Presente Neto (pagos, interés variables).
frmVPvar1	Forma para capturar (pagos, interés) para Valor Presente Neto
FuncionesEstadística	Pantalla General de Funciones Estadísticas.
FuncionesFinancieras	Pantalla General de Funciones Financieras.
OpYfuncBasicas	Pantalla de Operaciones y Funciones Básicas
tbl1Normal	Pantalla que muestra la distribución normal.
tbl2TStudent	Pantalla que muestra la t de Student.

tbl3F5Distribution	Pantalla que muestra puntos de la distribución F, 5% de significancia.
tbl4F1Distribution	Pantalla que muestra puntos de la distribución F, 1% de significancia.
tbl5Ji2	Pantalla que muestra puntos de la Ji cuadrada.
tbl6NumAleatorios	Pantalla que muestra los números aleatorios.

REPORTES:

Nombre	Descripción
fmbtlnAleat	Reporte de Números Aleatorios.
RptChi2	Reporte de la distribución Ji Cuadrada.
rptF1Distribution	Reporte de la distribución F al 1% de significancia.
RptF5Distribution	Reporte de la distribución F al 5% de significancia.
rptNormal	Reporte de la distribución normal.
rptTStudent	Reporte de la distribución t de Student.

MACROS:

Nombre	Descripción
Importar.ImportUSR	Importa archivos para la sección Análisis de Datos.
Importar.ImportUXY	Importa archivos para la sección Análisis de Regresión.
mcrCreaSerie1.Grafica0	Crea datos para la gráfica de puntos manuales.
mcrCreaSerie1.Grafica1	Crea datos para la gráfica de puntos creados automáticamente
mcrCreaSerie1.Grafica2	Crea datos para la gráfica de puntos creados automáticamente
mcrCreaSerie1.Grafica3	Crea datos para la gráfica de puntos creados automáticamente
mcrCreaSerie1.Grafica4	Crea datos para la gráfica de puntos creados automáticamente
mcrCreaSerie1.Grafica5	Crea datos para la gráfica de puntos creados automáticamente
mcrCreaSerie1.AnDatos	Calcula la información para la sección de Análisis de datos.

MODULOS:

Nombre	Descripción
EvFórmula	Contiene las funciones generales de EmCalc.

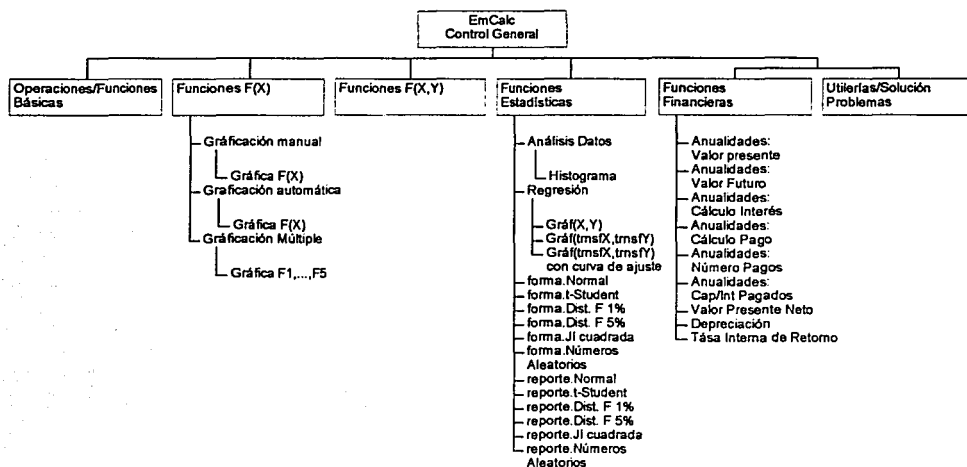
3.2 Carga de formas (Transferencia de pantallas).

Con el objeto de hacer coherente el desarrollo del trabajo para el usuario de EmCalc, los módulos se desarrollan en diferentes pantallas. Con esto en mente, cada módulo consiste en las pantallas necesarias para proveer información suficiente para el proceso que se esté llevando a cabo.

3.2.1 Estructura de organización de las pantallas de EmCalc. Las pantallas tienen una organización jerárquica por módulos, iniciando la jerarquía desde la forma principal, dada funcionalmente por el trabajo que desarrolla cada pantalla en la cuál se programan los comandos de acceso a pantallas subordinadas.

El siguiente organigrama muestra la estructura en que están organizadas las pantallas de EmCalc. En ella se puede distinguir un máximo de 3 niveles de subordinación.

Organización de pantallas en EmCalc

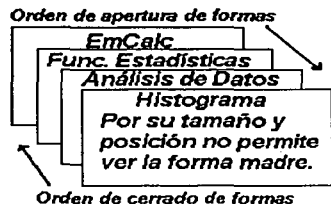


Estructura jerárquica de las formas o Pantallas (este organigrama es idéntico al de Navegación en las Pantallas).

Navegación en las pantallas en EmCalc. Además de ser su organización jerárquica, el organigrama muestra la ruta de navegación o acceso a las pantallas; por ejemplo, al tener a la vista "EmCalc (Control General)", el usuario puede acceder cualquier pantalla de primer nivel; al estar en alguna pantalla de primer nivel (por ejemplo: funciones estadísticas) se pueden acceder cualquiera de sus subordinadas (2o. nivel), proveyendo, que esa pantalla tenga subordinadas.

Nota: a una pantalla de nivel superior se le denominará pantalla-madre y una de nivel inferior pantalla-hija; ejemplo: "Análisis de datos" es una pantalla-hija de "Funciones Estadísticas y a su vez, es una pantalla-madre de "Histograma".

Todas las pantallas-madre de la rama permanecen abiertas al ir accedendo sus pantallas-hija. Por ejemplo, para abrir la pantalla de "Histograma", debió haber estado abierta "Análisis de Datos", a su vez debió haber estado abierta "Funciones Estadísticas" y asimismo, debió haber estado abierta la pantalla "EmCalc".



Apertura y Cerrado de formas. Al irse abriendo las formas, éstas quedan organizadas para cerrarse en el orden en que fueron abiertas, en cada nivel en que se encuentren.

Esta organización es dada en parte por Access y en parte por el diseño. El tamaño y la posición de las pantallas al abrirse, no permite ver las pantallas subyacentes, es por eso que se pueden cerrar en el orden en que fueron abiertas.

3.2.2 Acceso a las pantallas de trabajo. En general, las pantallas o formas son cargadas por medio de una caja de comando la cual hace un llamado a una subrutina que abre la pantalla solicitada, transfiriendo así el control a tal pantalla. En EmCalc se utilizan 3 métodos para hacerlo:

Caso 1) Carga de la forma del menú principal. Realizado con comandos propios de Access. Es la presentación de Emcalc.

Caso 2) Carga Normal o Default. Usando el valor preasignado a las variables al momento de diseñar la forma.

Caso 3) Carga Modificada. La base es la carga normal y se agregan comandos de inicialización de variables y/o subformas.

Caso 1) Carga de la forma del menú principal. Es la única forma que se carga automáticamente al empezar EmCalc. Esta forma no tiene variables definidas, sólo comandos para transferir el control a otras pantallas. La forma del menú principal se llama "EmCalc". Se usa la siguiente sucesión de comandos del menú principal:

- <Tools>,
- <StartUp>, lo cuál da acceso a la siguiente forma a llenar:

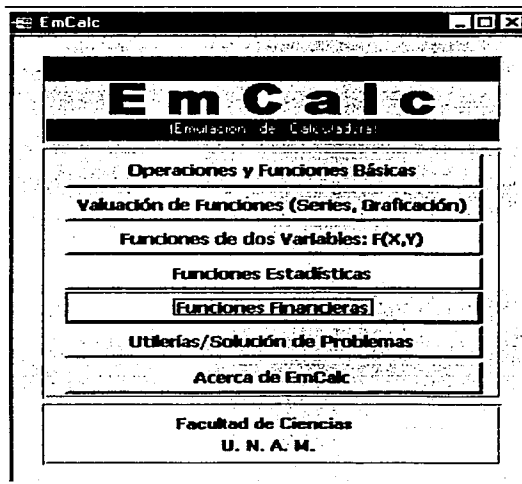
<OK> Almacena los cambios y la siguiente vez que se abra la aplicación, aparecerá la forma EmCalc:

Pantalla de Presentación y Principal de la aplicación.

Esta es la pantalla de presentación de Emcalc, la cuál a su vez permite el acceso a las pantallas de trabajo de la aplicación según el comando que se seleccione.

Esta forma permanece abierta durante toda la sesión de la aplicación. Al transferirse el control a una pantalla de trabajo, esta pantalla principal permanece activa tras la nueva pantalla. Si la nueva pantalla se minimiza o se cierra, aparece esta pantalla para dar acceso a una nueva pantalla o reiniciar en la misma que se estaba usando.

Si por algún motivo esta forma se cierra al estar activa otra pantalla, se recomienda reiniciar la aplicación debido a que al cerrar esta pantalla se pierde la posibilidad de transferencia entre módulos.



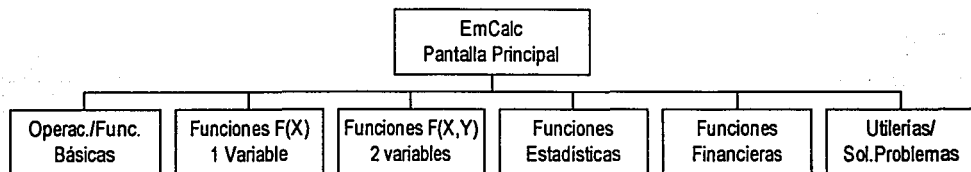
Caso 2) Carga Normal o Default. Usando el valor preasignado a las variables al momento de diseñar la forma. Las formas del **caso (3)** usan este mismo código y se les agregan comandos de inicialización de variables y/o subformas, según el caso específico.

Se ejemplifica con la transferencia de pantallas desde el menú principal.

Comandos que transfieren el control a otras pantallas desde el menú principal de EmCalc.

- Operaciones y Funciones Básicas
- Valuación de Funciones (Series, Graficación)
- Funciones de dos Variables: $F(X, Y)$
- Funciones Estadísticas
- Funciones Financieras
- Utilerías/Solución de Problemas
- Acerca de...

Transferencia de pantallas 1er. Nivel



Organigrama 1. El organigrama muestra las pantallas de trabajo subordinadas a la principal.

Código empleado en los comandos de carga de pantallas. Este ejemplo muestra como abrir la forma Operaciones y Funciones Básicas

```

Private Sub Command1_Click()
On Error GoTo Err_Command1_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "OpYfuncBasicas"
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Command1_Click:
Exit Sub

Err_Command1_Click:
MsgBox Err.Description
Resume Exit_Command1_Click
End Sub
  
```

'Nombre asignado por Access a la subrutina
'transfiere control a esta etiqueta, en caso de error.

'Variable que representa la forma (Pantalla) a abrir.
'Representa parámetros para abrir la forma,
'En este caso, no se requieren y queda en blanco.

'Nombre asignado a la forma a abrir
'Ejecuta comando abrir forma

'terminación normal
'Salida normal

'Access detecta algún error, por ejemplo forma
'dañada, o no existente, error al compilar, etc.
'regresa a esta etiqueta para terminar
'Termina el código de la subrutina.

Código en Visual Basic para abrir una forma.

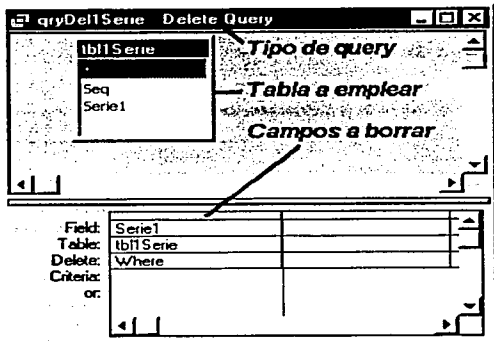
Para abrir las formas de las demás secciones, los comandos empleados utilizan una subrutina con un código similar, simplemente se sustituye el nombre de la forma a abrir:

- Valuación de Funciones (Series, Graficación): **stDocName = "frmFormula"**
- Funciones de dos Variables: F(X, Y): **stDocName = "frmFormulaXY"**
- Y así para cada una de las formas de esta sección, véase la siguiente tabla.

Estructura de acceso a las pantallas de EmCalc. Se indica por medio de (1), (2) y (3) el método empleado. Método 3 aún por definir, se describe por cada una de las pantallas, por ser cada una un caso especial.

Nombre de la sección	Tipo/Comando con nombre de la forma
Menú Principal	1 comando start-up: EmCalc
• Operaciones y Funciones Básicas	2 stDocName = "OpYfuncBasicas"
• Valuación de Funciones (Series, Graficación)	2 stDocName = "frmFormula"
• Graficación. Puntos Manualmente	3 stDocName = "frmMain1Serie"
• Gráfica "manual" F(X)	3 stDocName = "frmCh0"
• Graficación. Puntos automáticamente	3 stDocName = "frmMain2Serie"
• Gráfica "automática" F(X)	3 stDocName = "frmCh1"
• Graficación Múltiple	3 stDocName = "frmMain4Serie"
• Gráfica F1(X),..., F5(X)	3 stDocName = "frmmain4Ch"
• Funciones de dos Variables: F(X, Y)	2 stDocName = "frmFormulaXY"
• Funciones Estadísticas	2 stDocName = "FuncionesEstadisticas"
• Análisis de Datos	3 stDocName = "frmAnDat1"
• Histograma	3 stDocName = "frmCh01 AnDat"
• Análisis de Regresión	3 stDocName = "frmAnReg"
• Gráfica dispersión (X, Y)	3 stDocName = "frmAnRegCh0"
• Gráfica transformadas de (X, Y)	3 stDocName = "frmAnRegCh1"
• Gráfica transformadas c/línea ajuste	3 stDocName = "frmAnRegCh2"
• Distribución Normal	2 stDocName = "tbl1 Normal"
• t de Student	2 stDocName = "tbl2TStudent"
• Ji Cuadrada	2 stDocName = "tbl5J12"
• Distribución F, nivel de significancia 1%	2 stDocName = "tbl4F1 Distribution"
• Distribución F, nivel de significancia 5%	2 stDocName = "tbl3F5 Distribution"
• Generación de Números Aleatorios	2 stDocName = "tbl6NumAleatorios"
• Funciones Financieras	4 stDocName = "FuncionesFinancieras"
• Anualidades: Valor Presente	2 stDocName = "frmAnualVP"
• Anualidades: Valor Futuro	2 stDocName = "frmAnualFV"
• Anualidades: Cálculo de la Tasa de Interés	2 stDocName = "frmAnualInt"
• Anualidades: Cálculo del Pago Periódico	2 stDocName = "frmAnualPago"
• Anualidades: Cálculo del Número de Pagos	2 stDocName = "frmAnualNumPagos"
• Anualidades: Capital e Interés pagados	2 stDocName = "frmAnualIntYCap"
• Valor Presente Neto	4 stDocName = "frmVPvar"
• Depreciación.	4 stDocName = "frmDepreciación"
• Tasa Interna de Retorno	4 stDocName = "frmTir0"
• Utilerías/Solución de Problemas	2 stDocName = "frmUtility"
• Acerca de EmCalc	2 stDocName = "EmCalcAbout"

Caso 3) Inicialización de variables y/o subformas al momento de cargar la forma. Se muestran aquí las subrutinas empleadas para cada una de las formas que requieren inicialización de variables y/o subformas a un valor específico.



A la izquierda se muestra un query en vista de diseño.

En el ejemplo se borra sólo un campo, el cuál es el que alimenta de datos a una subforma que se utiliza para desplegar una tabla de captura de datos.

En las siguientes pantallas se optó por inicializar subformas de las pantallas a desplegar, haciendo un llamado a un query que borra la información de las tablas que alimentan a tales subformas, desde el comando de carga de la forma, es decir, antes de cargar la forma:

Forma: **Valuación de Funciones (Series, Graficación).** Graficación Manual

```
Private Sub Command18_Click()
    Dim stDocName As String
    Dim stLinkCriteria As String
    Dim stDocName1 As String

    stDocName1 = "qryDel1Serie"
    DoCmd.OpenQuery stDocName1, acNormal, acEdit

    stDocName = "frmMain1Serie"
    DoCmd.OpenForm stDocName, stLinkCriteria

End Sub
```

*Nombre asignado por Access a la subrutina
 *representa la forma a abrir
 *criterio (en blanco) para abrir la forma
 *representa el query a correr
 *asigna nombre del query que borra la subforma
 *ejecuta el query
 *Asigna nombre de la forma a abrir
 *abre la forma
 *termina el código

En forma similar a la mostrada, las dos siguientes pantallas son cargadas, previa corrida de un query que borra subformas que contienen:

Valuación de Funciones (Series, Graficación). Graficación Automática

```
stDocName1 = "QryDel2Serie"
stDocName = "frmMain2Serie"
```

*Nombre del query que borra subforma
 *Nombre de la pantalla que carga

Valuación de Funciones (Series, Graficación). Pantalla para **Graficación Múltiple**

```
stDocName1 = "QryDel2Serie"
stDocName = "frmMain4Serie"
```

*Nombre del query que borra subforma
 *Nombre de la pantalla que carga

La siguiente forma utiliza dos subformas, por lo tanto, utiliza dos queries diferentes para borrar cada una de ellas. Aquí, la forma ya ha sido cargada y se corre ésta subrutina al cedérsele control a la forma.

Funciones Estadísticas. Análisis de Datos

```
Private Sub Form_Open(Cancel As Integer)
    Dim stDocName As String
    stDocName = "qry02AnDat"
    DoCmd.OpenQuery stDocName, acNormal, acEdit
    stDocName = "qryDelMdn"
    DoCmd.OpenQuery stDocName, acNormal, acEdit
    Me.Refresh
End Sub
```

```
*Nombre dado por Access
*representa el query a correr
*asigna nombre del 1er. query
*Ejecuta el query
*asigna nombre del 2o. query
*Ejecuta el query
*refresca la forma con los nuevos datos
```

No se presentan las instrucciones para atrapar errores.

La siguiente forma utiliza también dos subformas, por lo tanto, utiliza dos queries diferentes para borrar cada una de ellas, adicionalmente, inicializa otras variables. Aquí, la forma ya ha sido cargada y se corre esta subrutina al cedérsele el control a la forma (se omiten instrucciones de control de errores).

Funciones Estadísticas. Análisis de Regresión

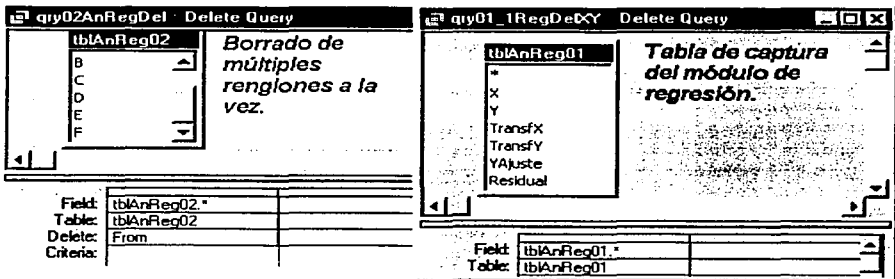
```
Private Sub Form_Open(Cancel As Integer)
    Dim stDocName As String

    stDocName = "qry01_1RegDelXY"
    DoCmd.OpenQuery stDocName, acNormal, acEdit
    stDocName = "qry02AnRegDel"
    DoCmd.OpenQuery stDocName, acNormal, acEdit
    Me.FX = Null
    Me.FY = Null
    Me.Err0 = 0
    Me.Err0X = 0
    Me.Err0v = 0
    Me.Err0vX = 0
    Me.Err0t = 0
    Me.Err0tX = 0
    Me.Arch = Null
    Me.Refresh
End Sub
```

```
*Nombre dado por Access
*Representa query a correr

*Asigna nombre del 1er. query
*El query borra el área de captura de datos
*Asigna nombre del 2o. query
*El query borra la tabla que despliega cálculos
*Función de la transformada de X
*Función de la transformada de Y
*Control error. División por cero FX
*Control error. División por cero FY
*Control error. Overflow FX
*Control error. Overflow FY
*Control error. Otros FX
*Control error. Otros FY
*Nombre Archivo de datos a emplear
*Refresca la pantalla
*Termina la subrutina
```

La figura muestra el diseño de los queries que borran las subformas de la pantalla regresión.



La siguiente forma lee de un archivo de números aleatorios: la cantidad de puntos que hay y los límites inferior y superior empleados para generar esa tabla de números aleatorios. Esto es, después de haberse cargado la forma (antes de cualquier otro evento).

Funciones Estadísticas: Generación de Números aleatorios.

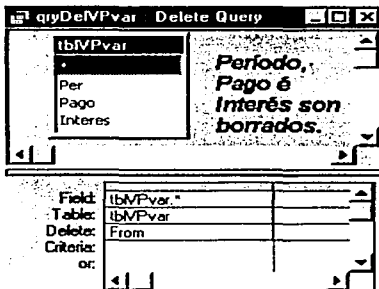
Private Sub Form_Load()	'Nombre dado por Access
Dim rsNAleat As DAO.Recordset	'Define un archivo de datos: rsNAleat
Dim dbs As DAO.Database	'Define una base de datos: dbs
Dim n As Long	'n cuenta la cantidad de números aleatorios
N = DCount("[TotalNos]", "tblDatNAleat")	'guardados en la tabla de números.
If n < 1 Then	'Este evento sólo puede ocurrir al momento
Exit Sub	'de diseño, se deja por posibles cambios
End If	'futuros. n=10 por construcción.
Set dbs = CurrentDb	'Se utiliza a EmCalc como Base de Datos
Set rsNAleat = dbs.OpenRecordset("tblDatNAleat")	'rsNAleat representa la tabla de nos. aleat.
With rsNAleat	'Indica que se utiliza un registro de rsNAleat
.MoveFirst	'Accesa el primer registro del Archivo
Me.TotNums = !TotalNos	'TotNums, Linf, Lsup son variables de la
Me.LInf = !LimInf	'forma: Total de números(aleatorios), Límite
Me.LSup = !LimSup	'inferior y Límite superior empleados.
End With	'Cierra cláusula With
rsNAleat.Close	'Cierra el archivo
End Sub	'termina la subrutina

En las siguientes subrutinas se omite el nombre, las instrucciones para atrapar errores, la instrucción de actualización de la pantalla y la instrucción que indica el fin de la subrutina.

En esta forma, un query borra la información de la subforma y se inicializan algunas variables de la forma.

Funciones Financieras: Valor Presente Neto:

stDocName = "qryDelVPvar"	'Asignación del nombre del query
DoCmd.OpenQuery stDocName, acNormal, acEdit	'Se ejecuta el query
Me.VPresente = Null	'Variables de la pantalla, que se inicializan
Me.IntOrig = Null	
Me.Result1 = Null	



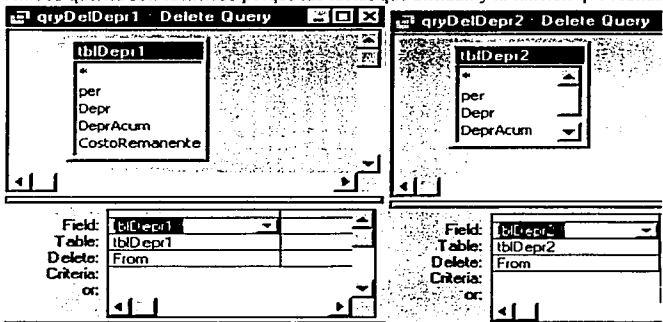
Query para borrar los datos de la subforma de captura de datos de Valor Presente Neto.

La siguiente forma contiene dos subformas, una para cada uno de los métodos de depreciación. Inicializa las subformas utilizando dos queries, después que ha sido cargada la forma.

Funciones Financieras: Depreciación

```
stDocName = "qryDelDepr1" 'Se asigna el 1er. query
DoCmd.OpenQuery stDocName, acNormal, acEdit '...y se ejecuta.
stDocName = "qryDelDepr2" 'Se asigna 2o. query
DoCmd.OpenQuery stDocName, acNormal, acEdit 'Borra la 2a. subforma
```

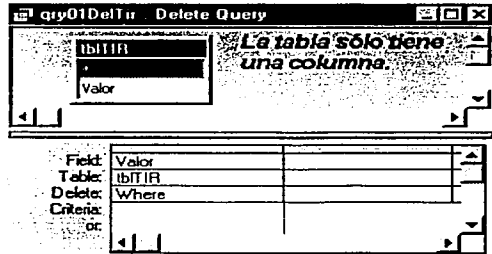
Ambos queries son idénticos porque las tablas que utilizan y la función que realizan es idéntica.



La siguiente forma contiene una subforma, la cual se inicializa con un query, después que ha sido cargada la forma.

Funciones Financieras: Tasa Interna de Retorno

```
stDocName = "qry01DelTir"
DoCmd.OpenQuery stDocName, acNormal, acEdit
```



3.3 Formas y subformas.

Algunas formas en EmCalc contienen una subforma la cual se emplea ya sea para capturar información o representar información que se genera.

Análisis de Datos

1) Traslada Datos, o, traer de archivo.

Datos
4
5
7
6
5
8
7
9

Subforma para captura de datos

Subforma para presentación de información

Record: 11 of 9

Archivo: USRaleat10

Trasladar Datos en Archivo

2) Generar Información. 3) Ver Histograma.

Generar Información	Mostrar Histograma
Número de Observaciones	14
Suma	143
Media	10.2142857142857
Varianza Poblacional (N)	232.882653061224
varianza Muestral (n)	250.796703296703
Desv. Standard Pob. (N)	15.2604932115979
Desv. Standard muestra (n)	15.8365622310116
Mínimo	4
Máximo	65
Rango	61
Mediana	6
Mediana 1	5
Mediana 3	7
Rango Intercuartil	2
	0

La forma contiene dos subformas, una para captura de información y otra para presentar información calculada.

Las formas y subformas en EmCalc son creadas de manera independiente y después se ensamblan, es decir, una subforma es por sí misma una forma, posteriormente al ensamblarse dentro de otra forma se convierte en su subforma. Ejemplo tomado de la sección de Análisis de Datos:

- Forma madre (la forma que contendrá las subformas). Se crea sin ligarse a alguna tabla o query. Se especifica que su vista sea "Forma Normal".
- Subforma para captura de datos (nombre: frmAnDat). Esta subforma se liga a la tabla tblAnDat para que los datos queden grabados. Se liga específicamente al campo "Valor" de la tabla tblAnDat (SELECT tblAnDat.Valor FROM tblAnDat); se especifica que su vista sea "Hoja de Cálculo".

frmAnDat : Form

Format Data Event Other All

Record Source SELECT tblAnDat.Valor FROM tblAnDat;

Filter

Order By

Allow Filters Yes

Allow Edits Yes

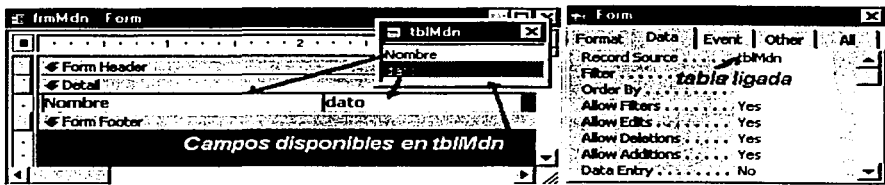
Allow Deletions Yes

Allow Additions Yes

Data Entry No

El campo "Valor" es el único extraído

- Subforma para presentación de información (nombre: frmMdn). Esta subforma se liga a una tabla para mayor facilidad de extracción de datos, la tabla empleada se llama tblMdn. Los campos extraídos son: **Nombre** y **Dato**; se especifica que su vista sea "Forma Continua".



NOTA: las vistas tipo "hoja de cálculo" y "forma continua" pueden ser muy similares, sin embargo tienen características diferentes para navegación en los campos. La vista hoja de cálculo proporciona más facilidades para capturar información. EmCalc usa este formato en todas las tablas de captura, mientras que se usa el de forma continua para todas las subformas de desplegado de información

- Finalmente, Se abre la forma madre **frmAnDat1** y se "arrastran" con el ratón las otras dos formas, colocándolas y haciéndolas del tamaño adecuado.

Análisis de Datos

Adtrac **Borrar Datos**

Clear Datos, ó, traer de archivo.

frmAnDat

Este espacio se asigna a la subforma para capturar datos.

Su espacio se puede controlar, reduciéndolo o ampliándolo. El límite es el tamaño de la forma.

ADguard Unbound **Trasar Archivo**

Archivo: Unbound **Borrar Archivo**

Guardar Datos en Archivo

ADInfo **Ver Historiama**

Generar Información **Hacer Histograma**

frmMdn

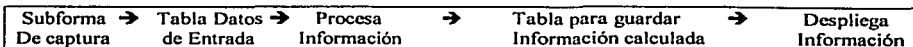
Este espacio se asigna para la subforma que despliega los datos.

Su tamaño se puede controlar, reduciéndolo o ampliándolo según se necesite. El límite es el tamaño de la forma.

Nota: las subformas no se pueden traslapar. Una de ellas no se desplegaría.

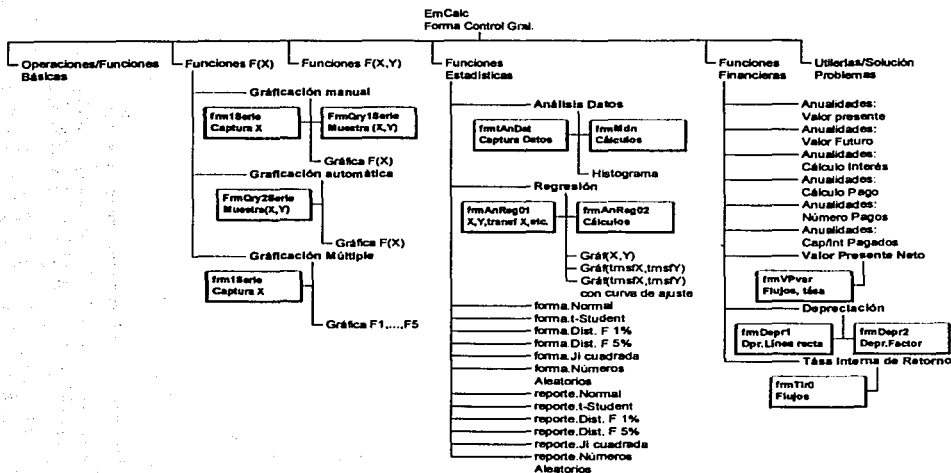
En la vista de diseño de la forma madre, las subformas aparecen como rectángulos, al ser colocadas con el ratón.

La relación entre las subformas es de tipo funcional. Los comandos que se escriben, toman la información de la subforma de captura, procesan la información y la presentan en la subforma de desplegado o en la caja-texto de resultado correspondiente:



Formas y subformas empleadas en EmCalc (los cuadros indican las subformas):

Organización de Formas/Subformas en EmCalc



Formas que utilizan subformas en EmCalc (esta lista corresponde al organigrama):

Módulos de Graficación

FormMain1Serie	frm1Serie(*) frmQry1Serie	Generación manual de puntos. Se captura en ella los datos X. Réplica de X, con $Y = F(X)$: (X, Y).
FormMain2Serie	frmQry2Serie	Generación automática de puntos. Muestra los pares (X, Y)
FormMain4Serie	frm1Serie (*)	Generación manual de puntos. Se captura en ella los datos X. En la parte automática se usa una tabla que no es mostrada; (*) ambas formas utilizan la misma subforma.

Funciones Estadísticas

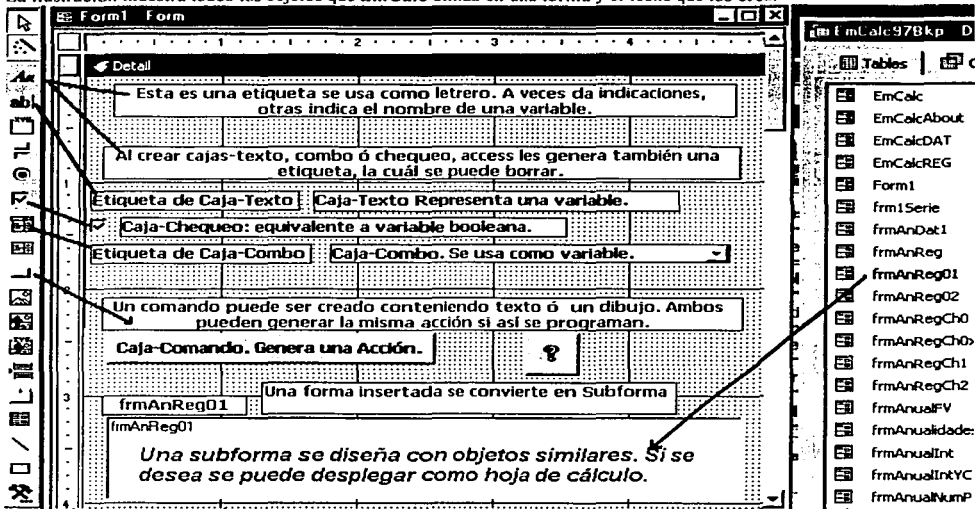
FormAnDat1	frmAnDat, frmMdn	Análisis de Datos. Se capturan en ella los datos. Análisis de Datos. Se muestra información calculada.
FormAnReg	frmAnReg01 frmAnReg02	Análisis Regresión. Se capturan (X, Y) y se muestran transformaciones de (X,Y), el ajuste y el residual. Análisis Regresión. Se muestra información calculada.

Funciones Financieras

frmVPvar	frmVPvar1	VPN. Se capturan los pagos y el interés.
frmDepreciacion	frmDepr1 frmDepr2	Depreciación. Se muestra tabla con información calculada. Depreciación. Se muestra tabla con información calculada.
frmTir0	frmTir1	Tasa Interna de Retorno. Se capturan los flujos.

3.4 Objetos Empleados para Crear una Forma. Cajas-Combo.

La ilustración muestra todos los objetos que EmCalc utiliza en una forma y el icono que los crea.



Etiquetas. Se utilizan para poner letreros o instrucciones. En ocasiones se usa para indicar el nombre de un campo

Caja-Texto. Se utilizan para que el usuario proporcione parámetros o para desplegar información.

Caja-Chequeo. Variables booleanas, el usuario las activa o desactiva.

Caja-Combo. Muestra opciones de las cuáles el usuario debe seleccionar una. Se describen más ampliamente en la sección que sigue.

Caja-Comando. Crea una acción al activar una subrutina, o una macro, tal como: guardar un archivo o extraer datos de un archivo, borrar datos, realizar un cálculo, desplegar una pantalla, etc. La sección de **Código de Programación por Pantalla** presenta todas las subrutinas creadas por las cajas-comando.

Nota:

- En EmCalc se crean subrutinas también al hacer clic en algunas cajas-texto o después de actualizar algunas cajas-texto. Esto se muestra en la sección de **Código de Programación por Pantalla**.
- La creación de subformas se puede ver en la sección de **Formas y Subformas**.

Cajas-Combo. La función principal de una caja-combo en EmCalc es mostrar al usuario las opciones disponibles para un campo y de ahí seleccionar el valor a usar. Posteriormente, al oprimir un comando se genera la acción para la cuál se requiere el valor en la caja-combo.

En EMCalc se utilizan cajas-combo con dos métodos diferentes de ligar la información a mostrar:
Ligando la caja-combo a una tabla o query.

- Indicando los valores en la caja-combo.

3.4.1 Cajas-combo ligadas a una tabla o query.

Para mostrar el diseño de una caja-combo, se utiliza el campo **Opr** de la sección de "Funciones y Operaciones básicas", submódulo "Operaciones".

Opr representa al operador al realizar una operación. La ilustración muestra **Opr** cuando está cerrada y cuando se hace un clic para ver las opciones:

Operando 1 (op1)	Operador	Operando 2 (op2)
0	+ -	0

Operador

+ - ← Click

+ Suma

- Resta

/ División

⊕ a * b%; 50 % 5 = (50 * 5%) = 2.5

^ Exponenciación. a^b: 2^4 = 16

La caja-combo muestra información de la tabla **tblOperador**:

tblOperador = **Opr** | **OprDescr** → **Operador(Caja-combo)**

Se muestra un fragmento de la tabla **tblOperador**

Opr	OprDescr
^	Exponenciación. a^b: 2^4 = 16
e	e {e=base exp.} Pasa directo a Resultado
Lga	Log base a(b); Log 10 (1000) = 3

Propiedades de la caja-combo:

- **Datos.** Proveniencia de los datos y manejo de éstos.

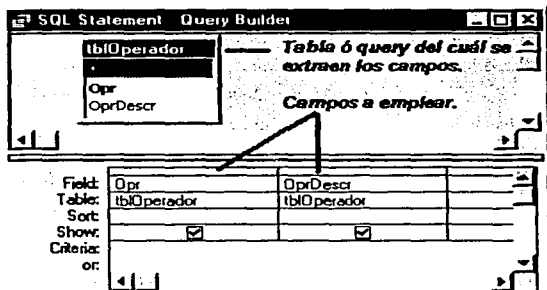
En la creación de la caja-combo se liga a la tabla **tblOperador**.

Combo Box Opr

Format	Data	Event	Other	All	Opciones que se midifican en EmCalc:
Control Source					
Input Mask					
Row Source Type	Table/Query				<i>Para datos numéricos ó fechas. Tipo de renglón fuente:</i>
Row Source	SELECT DISTINCTROW [tblOperador].[Opr], [tblOperador].[OprDescr] FROM [tblOperador];				
Bound Column	1				<i>Guardar columna No... 1</i>
Limit To List	Yes				<i>Limitar a la lista (ó el usuario puede agregar)?</i>
Auto Expand	Yes				<i>Teclas primeras letras y lista opción completa?</i>
Default Value	""				<i>Valor default?</i>
Validation Rule					<i>Regla de validación? P. Ej. >= 10</i>
Validation Text					<i>Mensaje de error en caso de violar regla de validación?</i>
Enabled	Yes				<i>Accesible? (el cursor para aquí?)</i>
Locked	No				<i>Bloqueado? (Puede editarse?)</i>

La figura muestra las opciones empleadas en EmCalc. Row Source se explica en la figura de abajo.

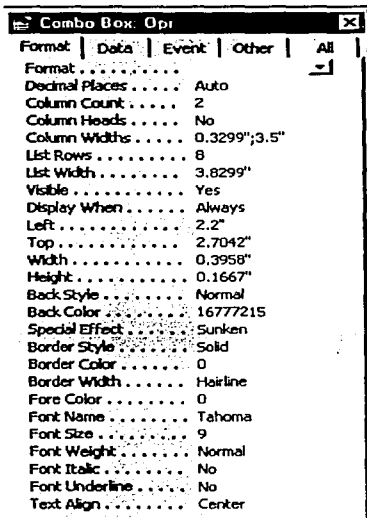
La figura muestra el query que da origen a la expresión de **Row Source** de la figura de arriba. Se asigna una tabla o un query como la fuente de los datos y se selecciona de ahí los campos a emplear:



Este query se genera automáticamente al crear la caja-combo con un "Wizard" (Programa interno de Access y otros Software de Microsoft que guía al diseñador en la creación de objetos).

El query creado no tiene nombre porque está ligado automáticamente por Access, sin embargo se puede editar el diseño del query, é incluso ligar la caja-combo a otro query o a una tabla.

• Formato.



Formato. Como es texto, no se usa en este caso.
Decimales

Columnas a desplegar (al hacer clic)

Usar encabezado de las columnas?

Anchura de las columnas.

Renglones a listar (al hacer clic)

Anchura de la lista.

A desplegar? (Visible?)

Desplegar cuándo?

Margen izquierdo

Margen superior

Anchura de la caja-combo

Altura de la caja-combo

Estilo del fondo

Color del fondo (código asignado al elegir de barra de colores)

Efecto especial (hundido-bajo relieve)

Estilo del marco. Sólido, transparente, línea punteada, etc.

Color del marco (código asignado al elegir de barra de colores)

Anchura del marco

Color de las letras(código asignado al elegir de barra de colores)

Tipo de letra

Tamaño de la letra

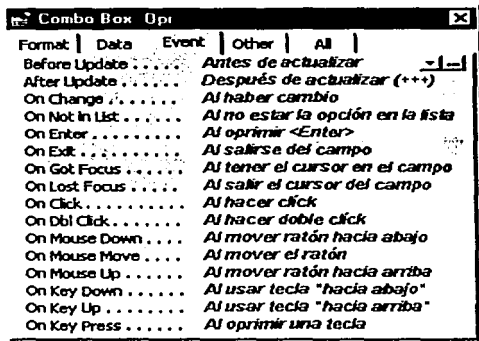
Espesor de la letra

Usar itálicas?

Usar subrayado?

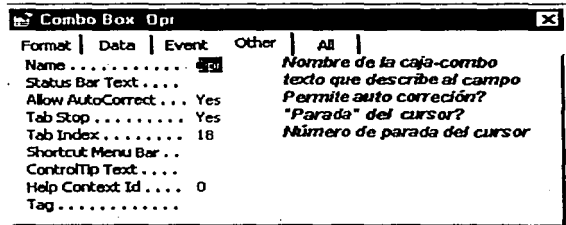
Alineación del texto.

- **Eventos.** En esta serie de propiedades no se especifica ninguna acción.



- La opción más empleada de "Eventos", en EmCalc, es **Después de Actualizar**, sin embargo, en este caso en especial no es necesario.
- Estos eventos, permiten realizar una acción cuando suceden, por ejemplo, correr una subrutina o una macro que causan que se actualice cierto campo.

- **Otras Propiedades.** Se especifica el nombre de la caja-combo y características de navegación.



- Nombre: **Opr**;
- Status Bar text: texto que aparece en la barra de status, abajo, en la pantalla.
- tab stop: al oprimir la tecla Tab se hace parada? Si; de otra forma se accedería sólo con el ratón;
- Tab Index: orden que le toca al oprimir la tecla tab;

- **Todos (All).** Esta opción muestra la lista de las cuatro series de propiedades anteriores.

Otras cajas-combo definidas. Las siguientes cajas-combo también están ligadas a una tabla y se crearon en forma similar a la ya mostrada. Se muestra un fragmento de la tabla a la cual están ligadas.

- Módulo: Operaciones y Funciones Básicas.
Sección: Funciones.

Muestra 2 columnas: la función y su descripción. Se almacena la función que se selecciona, esto, es mostrado en las ilustraciones.

Función	
Log	Log natural (arg)
Exp	E^{arg} {arg <= 709.782712893}
Sin	Sen(arg) {arg en radianes}
Cos	Cos(arg)
Abs	Abs(arg)

Ligada a la tabla tblPreConst:
tblPreConst → Función

Nótese como la información que se despliega coincide con la de la tabla.

Fragmento de la Tabla tblPreConst:

Función	Descripción
Log	Log natural(arg)
Exp	E^{arg} {arg <= 709.782712893}
Sin	Sen(arg) {arg en radianes}
Cos	Cos(arg)

Cajas-Combo ligadas a la tabla tblFunciones (Funciones de Usuario). Debido a su utilización en diferentes secciones de EmCalc, varias cajas-combo se relacionan en forma idéntica a esta tabla.

- Módulo: Valuación de Funciones

Submódulos: Traer Funciones y Guardar/Borrar Archivos. Su función es mostrar las funciones almacenadas, las cuáles están disponibles para que el usuario accese o borre, según la caja-combo y comando empleado.

Estas cajas-combo muestran 3 columnas:

- **Nombre que el usuario le asigna a la función;**
- **Expresión de la función; y,**
- **Tipo de función** (función de una o dos variables). Después de seleccionar la función, la caja-combo muestra el nombre de la función, pero, se transfiere la expresión al área de edición de "F(X)" al oprimir el comando <Usar Función seleccionada>.

Ubicación de las cajas-combo mencionadas.

Valuación de Funciones

[Instrucciones/Ayuda](#)

Traer Funciones.		Guardar Funciones y Borrar Funciones Archivadas.	
Archivadas	<input type="checkbox"/> <input type="checkbox"/>	Nombre (Máx.20 caract.)	<input type="text"/>
Usar Función Seleccionada		Guardar Función	
Preconstruidas	<input type="checkbox"/> <input type="checkbox"/>	Función a Borrar	<input type="checkbox"/> <input type="checkbox"/>
Agregar Función al Final de F(X)		Borrar Función Seleccionada	

Las cajas-combo ligadas a esta tabla pueden mostrar la misma información o inhibir alguna columna.

Archivadas		
2SinCos	$\sin(x) * \cos(x) * x^2$	F(x)
Sdatos	$57.16409 - 2.81901 * X + .04221 * X^2$	F(x)
Curva	$0.5 * \sin(3 * x) + 0.3 * \cos(1.8 * x) + 0.4 * \cos(0.9 * x + 0.3) + 1.8$	F(x)
Curva1	$57.82738 - 2.92438 * X + .04566 * X^2$	F(x)
Descartes	$\sin(x) + \sin(2 * x - 1) + \cos(3.1416 * x)$	F(x)

La caja-combo esta ligada a la tabla tblFunciones: **tblFunciones** → **FunciónAUsar**

Fragmento de la tabla tblFunciones:

Nombre	Función	Tipo
Espiral2	$5 * \sin(x + 3) / \sqrt{1 + x^2}$	F(x)
f1xy	$\sin(x) * \cos(y)$	F(x, y)
Func11/1/01 7:57:55 PM7:5	$X^2 + \log(x) / (x + x^2) + 25 * x^4$	F(x)

Nota: el módulo "Valuación de Funciones F(X, Y) usa réplicas exactas de estas cajas-combo.

• **Módulo: Graficación Manual.**

Submódulo: Acceso de Funciones.

Nombre: Función. Se muestran las funciones disponibles para graficar.

Función	Cambios	Pasar a F(X)	Graficación de Funciones
2SinCos		$\sin(x) * \cos(x) * x^2$	F(x)
Sdatos		$57.16409 - 2.81901 * X + .04221 * X^2$	F(x)
Curva		$0.5 * \sin(3 * x) + 0.3 * \cos(1.8 * x) + 0.4 * \cos(0.9 * x + 0.3) + 1.8$	F(x)
Curva1		$57.82738 - 2.92438 * X + .04566 * X^2$	F(x)
Descartes		$\sin(x) + \sin(2 * x - 1) + \cos(3.1416 * x)$	F(x)
Domingo		$57.16409 - 2.81901 * X + .04221 * X^2 + (\exp(X) + \exp(-X))$	F(x)
Espiral1		$3 * \cos(x + 4) / \sqrt{1 + x^2}$	F(x)
Espiral2		$5 * \sin(x + 3) / \sqrt{1 + x^2}$	F(x)

• **Módulo: Graficación Generación automática de puntos.**

Submódulo: Acceso de Funciones.

Se muestran las funciones disponibles para graficar.

• **Módulo: Graficación Múltiple de Funciones.**

Submódulo: Acceso de Funciones.

Se muestran las funciones disponibles para graficar.

• **Módulo: Análisis de Regresión.**

Submódulo: Transformaciones para X y Y.

Muestra las funciones que se pueden utilizar para transformar a X y/o Y.

2) Opcional. Transformaciones para X y Y.			
Función	Pasar a:	F(X)	F(Y)
F(X)			F(X)
F(Y)			F(Y)

La misma caja-combo muestra la función a transferir a F(X) o a F(Y); los comandos < Pasar a F(X); F(Y) > controlan el crear la función en términos de X o de Y.

- **Módulo: Valuación de Funciones**
Submódulo: Funciones Preconstruidas.

Preconstruidas	
Arco Tangente Atn(X) =	Atn(X)
Cosecante Cosec(X) =	1 / Sin(X)
Cosecante Hiperbólica HCosec(X) =	2 / (Exp(X) - Exp(-X))
Coseno Cos(X) =	Cos(X)
Coseno Hiperbólico HCos(X) =	(Exp(X) + Exp(-X)) / 2

Muestra 2 columnas:
Nombre de la función;
expresión de la función. Después
de seleccionar la función,

Ligada a la tabla tblFuncDerivadas: **tblFuncDerivadas** → **FuncPrec**

Fragmento de la tabla tblFuncDerivadas:

Nombre	Función
Cosecante Hiperbólica HCosec(X)	2 / (Exp(X) - Exp(-X))
Coseno Cos(X) =	Cos(X)
Coseno Hiperbólico HCos(X) =	(Exp(X) + Exp(-X)) / 2
Cotangente Cotan(X) =	1 / Tan(X)


Nota: el módulo "Valuación de Funciones F(X, Y). Usa una réplica exacta de esta caja-combo.

- **Módulos: Análisis de Regresión y Utilerías/Solución de Problemas.**
Submódulo: Manejo de Archivos.

Análisis de Regresión. Muestra los archivos que se pueden cargar: comando <traer archivo>.

Archivo de Datos	▼	Traer Archivo
------------------	---	---------------

Nombre: **ArchivoAborrar** (Utilerías/Solución de Problemas). Mismo caso al anterior, para borrar: comando <borrar archivo: "tirar al bote de basura" >.

Archivo:	▼	
	UXYabril132002	
	UXYE5datos	

Ligados a la tabla **MsysObjects**, la cuál es una tabla del sistema que muestra los objetos creados. En este caso se piden los nombres de las tablas (Type = 1) con el prefijo **UXY**. Por este motivo a todos los archivos de "Análisis de Regresión" se les antepone **UXY** al salvarlos, de esta forma EmCalc reconoce que pertenecen a esta categoría.

MsysObjects → **Arch**
MsysObjects → **ArchivoABorrar**

Fragmento de la tabla **MsysObjects** (se muestran sólo algunos campos):

LvProp	Nombre	Owner	ParentId	RmtInfoLong	RmtInfoShort	Type
	USRSabado1Campo		251658241			1
	UXYabril132002		251658241			1
	UXYEj5datos		251658241			1
	UXYImport		251658241			1

Un método similar al recién mostrado se utiliza en las siguientes cajas-combo:

- **Módulo: Análisis de Datos.**
Submódulo: Manejo de Archivos.

Archivo:	-	Traer Archivo
Archivo:	USR1235468*	Borrar Archivo
	USR1235468*	
	USRAleatorios	

Muestra los archivos que se pueden Accesar: comando <traer archivo>.

Mismo caso al anterior, para borrar: comando <Borrar archivo>.

Ligados a la misma tabla **MsysObjects**. En este caso se pide el nombre de las tablas (**Type = 1**) con el prefijo **USR**. Por este motivo a todos los archivos de "Análisis de Datos" se les antepone **USR** al salvarlos, de esta forma EmCalc reconoce que pertenecen a esta categoría.

MsysObjects	→	Datos
MsysObjects	→	ArchivoABorrar

Fragmento de la tabla **MsysObjects** (se muestran sólo los campos pertinentes y algunos adicionales).

LvProp	Nombre	Owner	Parentid	RmtInfoLong	RmtInfoShort	Type
	USRSabado1Campo		251658241			1
	UXYabril132002		251658241			1
	UXYEj5datos		251658241			1
	UXYimport		251658241			1

3.4.2 Cajas-Combo Especificando los Valores al Momento de su Creación.

Estas cajas combo se crean con el mismo wizard que las ligadas a una tabla, simplemente se especifica que los valores serán proporcionados por el programador.

Se ejemplifica con las cajas-combo del módulo Funciones Financieras; submódulo Anualidades: Valor Presente. Ambas cajas-combo fueron creadas en forma idéntica.

La función de estas caja-combo es mostrar la columna fuente y la columna destino al copiar datos; ambas muestran los valores 1, 2, 3, 4 debido a que existen 4 columnas. Posteriormente el comando representado por la varita mágica ejecuta la copia.



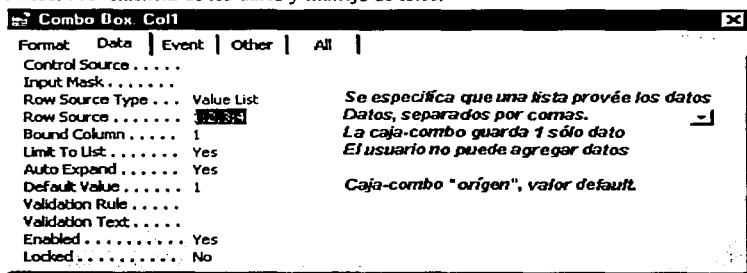
La caja-combo **Columna 1** muestra la columna fuente de los datos a copiar. Valores: 1, 2, 3 y 4.

La caja-combo **Columna 2** muestra la columna destino de los datos a copiar. Valores: 1, 2, 3 y 4.

Propiedades de la caja-combo:

Las propiedades de esta caja-combo son idénticas a las mostradas para las cajas-combos ligadas a una tabla, excepto por la proveniencia de los datos. Se muestra la ilustración con esta sección de propiedades

- **Datos.** Proveniencia de los datos y manejo de éstos.

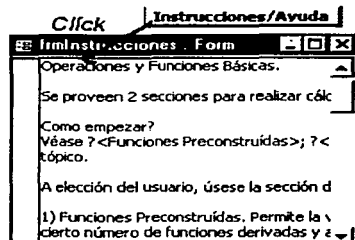


Nota: las cajas-combo de las demás secciones de anualidades se crearon en forma idéntica a éstas aquí descritas, y con nombres idénticos: Col1 y Col2:

- Anualidades: Valor Futuro
- Anualidades: Interés
- Anualidades: Pago
- Anualidades: Número de Pagos
- Anualidades: Capital é Interés Pagados

3.5 Desarrollo de las pantallas de ayuda.

En cada pantalla de trabajo de EmCalc se despliega ayuda en línea. Los comandos de ayuda aparecen en cajas-comando con el texto "Instrucciones/Ayuda" o con el símbolo de interrogación "?".



Al hacer clic en cualquiera de estos comandos se abre una pantalla con la ayuda solicitada

A la izquierda se muestra reducida, una pantalla de ayuda.

La página puede ser adecuada en tamaño por el usuario de EmCalc, ya sea para achicarla, agrandarla o moverla de lugar.

Pasos para la creación de las Páginas de Ayuda.

Se usa como ejemplo la sección de **Funciones y Operaciones Básicas**. Posteriormente se da una tabla que muestra todas las ayudas proporcionadas.

Paso 1)

Creación de la tabla maestra.

La tabla **tblInstrucciones** contiene todas las páginas de ayuda. Tiene 2 campos:

- **Forma.** Contiene los mnemónicos que indican que pantalla de ayuda se debe abrir al oprimir un comando de ayuda. Se define como texto de 30 caracteres.
- **Instrucciones.** Contiene la ayuda correspondiente al mnemónico. Es de tipo memorandum para que pueda crecer hasta 64K (unas 16 páginas de texto).

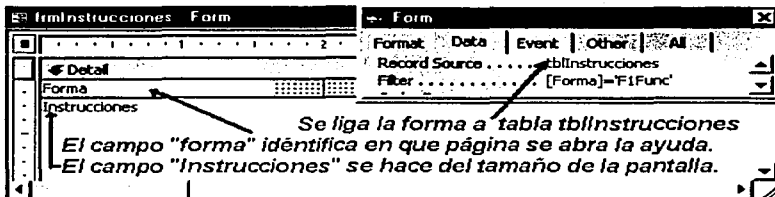
Fragmento de la tabla **tblInstrucciones**. El campo **Instrucciones** (texto de ayuda) solo muestra la primera línea del texto de una página de ayuda:

Forma (Mnemónico)	Instrucciones
Mem	Memorias.
NumAleat	Generador de Números Aleatorios.
Ops	Operaciones (Usando dos operandos).
OpYFuncBasicas	Operaciones y Funciones Básicas.
Rgcalc	3) Calcular Transformaciones y Ajuste.
Rggral	Análisis de Regresión.

El campo **Instrucciones**, por ser muy grande, solo puede ser visto en su totalidad cuando es desplegado en una forma o un reporte.

Paso 2)

Creación de una forma para capturar el texto de ayuda en la tabla, y mostrar las ayudas. La forma se liga a la tabla al asignársele "Record Source (Origen de registros)...tblInstrucciones".



Record Source (fuente de los datos o registros) liga una forma a una tabla.

La forma `frmInstrucciones` se crea con dos variables (cajas-texto) que corresponden a los campos de la tabla. Al estar ligada la forma a la tabla, es usual utilizar los mismos nombres de la tabla, para los objetos de la forma que representan los campos de la tabla:

- **Forma.** Despliega el mnemónico. A este campo se le da la opción `<Visible>=NO` de su propiedad `<Format>` para que al momento de correr la aplicación se oculte.
- **Instrucciones.** Despliega las instrucciones. Este campo se hace del tamaño de una pantalla para que despliegue la más información posible (el usuario de EmCalc le puede cambiar de tamaño).

El campo Instrucciones se protege para que el usuario no pueda actualizarlo. Al usar la opción `<Lock>=Yes` en la propiedad `<Data>`.

Paso 3)

Creación de variables con mnemónicos para las ayudas en las pantallas de EmCalc.

La sección de **Funciones y Operaciones Básicas** provee las siguientes ayudas:

- Instrucciones/Ayuda
- ?(Ayuda) sección de Funciones Preconstruidas
- ?(Ayuda) sección de Operaciones
- ?(Ayuda) sección de Memorias
- Borrar Todo en la Pantalla.

De modo que se definen en esta pantalla **variables** (cajas-texto) a las cuáles se les asignan los siguientes mnemónicos; se opta por llamar a la variable igual al mnemónico que debe contener:

- `OpYFuncBasicas = "OpYFuncBasicas"` -ayuda general (véase fragmento de la tabla).
- `FPCConst = "FPCConst"` -sección de "Funciones Preconstruidas".
- `Ops = "Ops"` -sección "Operaciones" (véase fragmento de la tabla)
- `Mem = "Mem"` -sección "Memorias" (véase fragmento de la tabla).



Cajas- comando y sus variables relacionadas. Se les da la propiedad de "invisibles" para que al correr la aplicación se oculten.

Paso 4) Relación entre las variables de la pantalla, y la página que la forma debe desplegar. La relación se realiza por medio de una subrutina en Visual Basic que abre la forma de instrucciones en el lugar solicitado:

Código del comando <Instrucciones/Ayuda>, la ayuda general de esta sección:

Private Sub Command70_Click()	'Nombre proporcionado por Access
On Error GoTo Err_Command70_Click	'En caso de error, se transfiere el control
Dim stDocName As String	'Variable que representa la forma a abrir.
Dim stLinkCriteria As String	'Variable que representa los parámetros
	'del comando "Abrir forma", véase abajo.
stDocName = "frmInstrucciones"	'La forma ligada a la tabla de Instrucciones.
stLinkCriteria = "[Forma]=" & "" & Me![OpYFunciones] & ""	'Abre la forma en la
	'página que coincide con "OpYFunciones" (*)
DoCmd.OpenForm stDocName, stLinkCriteria	'Ejecuta el comando
	'abrir-forma, con el criterio dado.
Exit_Command70_Click:	'Terminación normal
Exit Sub	'Termina la subrutina
Err_Command70_Click:	'Access detecta algún error, por ejemplo forma
MsgBox Err.Description	'dañada, o no existente; error al compilar; etc.
Resume Exit_Command70_Click	'regresa a esta etiqueta para terminar
End Sub	'termina el código.

(*) stLinkCriteria = "[Forma]=" & "" & Me![OpYFunciones] & ""

Aquí:

[Forma]= es el campo de la forma frmInstrucciones, y
me![OpYFunciones] es la variable (caja-texto)

Al ejecutarse el comando abrir-forma, la forma frmInstrucciones desplegará la página donde **Forma = "OPYFunciones"**; mostrando **Instrucciones = "Operaciones y Funciones Básicas"**.

Las demás ayudas de esta sección.

para abrir la ayuda de **Funciones Preconstruidas** el campo **Forma** de frmInstrucciones debe coincidir con el contenido de la variable **FPConst**, a la cual se le asignó previamente "FPConst":
stLinkCriteria = "[Forma]=" & "" & Me![FPConst] & ""

Para abrir la pantalla de ayuda de **Operaciones**:

stLinkCriteria = "[Forma]=" & "" & Me![Ops] & ""

Para abrir la pantalla de ayuda de **Memorias** (figura 5, mostrada anteriormente):

stLinkCriteria = "[Forma]=" & "" & Me![Mem] & ""

De este mismo modo se abren todas las pantallas de ayuda que EmCalc proporciona. A continuación se da una lista exhaustiva de los mnemónicos, los cuáles coinciden con los nombres de las variables que se utilizan:

Tabla I. Lista de ayudas empleadas y variables relacionadas.

Sección	Nombre Forma	Variable	Ayuda (fragmento primera línea)
Estad:Anals.Datos	frmAnDat1	Adinfo	Generar Información. Hacer Histograma.
Estad:Anals.Datos	frmAnDat1	Adgral	Análisis de Datos. Módulo de Estadística Descriptiva
Estad:Anals.Datos	frmAnDat1	Adguarda	Traer Archivo de Datos; Borrar Archivo de Datos;
Estad:Anals.Datos	frmAnDat1	Adtraer	Especificación de los Datos. Manejo del área de captura.
Estad: Regresión	FrmAnReg	Rgcalc	3) Calcular Transformaciones y Ajuste.
Estad: Regresión	FrmAnReg	Rgral	Análisis de Regresión.
Estad: Regresión	FrmAnReg	Rgtecl	Manejo del área de entrada de datos. Traer datos de archivo.
Estad: Regresión	frmAnReg	Rgtransf	2) Opcional: transformaciones para X y Y.
Anualid: V.Futuro	frmAnualFV	AnVF	Valor Futuro. Proporciona facilidad para desplegar 4 sensib.
Anualid: V.Futuro	frmAnualFV	AnVP	Valor Presente. Proporciona facilidad para desplegar 4 sensib.
Anualid: Interés	frmAnualInt	AnInt	Anualidades. Cálculo de la tasa de Interés.
Anualid: Interés	frmAnualInt	AnIntEjimp	Cálculo del Interés.
Anual:CapInt.Pag	frmAnualIntYCap	AnCapInt	Anualidades. Cálculo del Capital e Interés Pagado.
Anual:CapInt.Pag	frmAnualIntYCap	AnCplntEjim	Cálculo del Capital e Interés Pagado.
Anualid:Num.Pagos	frmAnualNumPagos	AnNPagEjimp	Cálculo del Número de Pagos. Calcula el pago parcial.
Anualid:Num.Pagos	frmAnualNumPagos	anNumPagos	Anualidades. Cálculo del Número de Pagos.
Anualid: Pago	frmAnualPago	AnPagEjimp	Cálculo del Pago Periódico.
Anualid: Pago	frmAnualPago	AnPago	Anualidades. Cálculo del Pago Periódico.
Anualid: V.Present	frmAnualVP	AnVPEjm	Ejemplos:
Depreciación	frmDepreciación	Depr	Depreciación. Método de Línea Recta y Doble Declinación.
Func./Series/Graf	frmFormula	FPConst	Funciones Preconstruidas.
Func./Series/Graf	frmFormula	Fxgral	Valuación de Funciones.
Func./Series/Graf	frmFormula	Fxtraer	Traer Funciones.
Func./Series/Graf	frmFormula	Fxguardar	Guardar Funciones y Borrar Funciones Archivadas.
Func./Series/Graf	frmFormula	Fxgraf	Generación de Series y Graficación de Funciones.
Func./Series/Graf	frmFormula	FIFunc	Función. Expresión en términos de X
Func./Series/Graf	frmFormula	Fxeditar	Valuación de Funciones. "F(X) =", "X =" y "Resultado".
Func.2 Var(FX,Y)	frmFormulaXY	FXYTEcl	Valuación de Funciones. "F(X,Y) =", "X = Y=" y "Resultado"
Func.2 Var(FX,Y)	frmFormulaXY	FXYfunc	Traer Funciones.
Func.2 Var(FX,Y)	frmFormulaXY	FXYguarda	Guardar Funciones y Borrar Funciones Archivadas.
Func.2 Var(FX,Y)	frmFormulaXY	FXYgral	Valuación de Funciones F(X,Y).
Gráficas Manuales	frmMain1Serie	GF1teclear	Especificación de los Datos. Manejo del área de captura.
Gráficas Manuales	frmMain1Serie	GF1ctl	Control de Errores al Valuar la Función.
Gráficas Manuales	frmMain1Serie	GF1gral	Graficación de Funciones. Generación de Puntos Manualmente.
Gráficas Manuales	frmMain1Serie	GF1Func	Funcion F(X).
Graf.Automáticas	frmMain2Serie	GF2Gral	Graficación de Funciones. Generación Puntos automaticamente.
Graf.Automáticas	frmMain2Serie	GF2Sel	Opciones: "No. de puntos" e "Incremento".
Graf.Automáticas	frmMain2Serie	GF2Rng	Especificar Rango (Inicio, Fin) para valores de X.
Gráficas Múltiples	frmMain3Serie	GF3Sel	Opciones: "No. de puntos" e "Incremento".
Gráficas Múltiples	frmMain3Serie	GF3gral	Graficación de Múltiples Funciones (Máximo 5).
Gráficas Múltiples	frmMain3Serie	F3Func	Función. Expresión en términos de X.
Gráficas Múltiples	frmMain3Serie	GF3Opc	Usar una de las 3 opciones para crear valores de X.
Gráficas Múltiples	frmMain3Serie	GF3graf	4) Ver Gráfica de las Funciones.
TIR	frmTir0	TIRgral	Tasa Interna de Retorno.
TIR.	frmTir0	TIRarch	Traer Archivo de Datos; Borrar Archivo de Datos.
Utiliterias	frmUtility	UtilBorrarReg	Borrar Archivos de Datos de Regresiones
Utiliterias	frmUtility	UtilCompact	Compactar EmCalc

Utilerías	frmUtility	UtilImport	Importar Archivos de Excel.
Utilerías	frmUtility	UtilOtros	Utilerías. Misceláneos.
Utilerías	frmUtility	UtilGral	Utilerías/Solución de problemas.
VPN pagos variab	frmVPVar	VPN	Cálculo del Valor Presente Neto (VPN).
Func.Financieras	FuncionesFinancieras	Anualidades	Anualidades.
Oper/Func.Básicas	OpYfuncBásicas	Mem	Memorias.
Oper/Func.Básicas	OpYfuncBásicas	Ops	Operaciones (Usando dos operandos).
Oper/Func.Básicas	OpYfuncBásicas	OpYFuncBasicas	Operaciones y Funciones Básicas.
Oper/Func.Básicas	OpYfuncBásicas	NumAleat	Generador de Números Aleatorios.

Tabla 2. Nombre expandido de la sección.

Sección	Sección Descripción	Forma
Anual:Cap/Int.Pag	Anualidades: Capital é Interés Pagado	frmAnualIntYCap
Anualid: Interés	Anualidades: Interés	frmAnualInt
Anualid: Pago	Anualidades: Pago Periódico	frmAnualPago
Anualid: V.Futuro	Anualidades: Valor Futuro	frmAnualFV
Anualid: V.Present	Anualidades: Valor Presente	frmAnualVP
Anualid:Num.Pagos	Anualidades: Número de Pagos	frmAnualNumPagos
Depreciación	Depreciación	frmDepreciación
Estad: Regresión	Estadística: Análisis de Regresión	frmAnReg
Estad:Anals.Datos	Estadística: Análisis de Datos	frmAnDat1
Func./Series/Graf	Funciones (Series, Graficación)	frmFormula
Func.2 Var(FX,Y)	Funciones de Dos variables: F(X,Y)	frmFormulaXY
Func.Financieras	Funciones Financieras	FuncionesFinancieras
Graf.Automáticas	Graficación Automática: F(X)	frmMain2Serie
Gráficas Manuales	Graficación Manual: F(X)	frmMain1Serie
Gráficas Múltiples	Graficación Múltiple Automática: F(X)	frmMain3Serie
Oper/Func.Básicas	Operaciones y Funciones Básicas	OpYfuncBasicas
TIR	Tasa Interna de Retorno	frmTir0
Utilerías	Utilerías/Solución de Problemas	frmUtility
VPN pagos variab	VPN (Pagos é Interés Variables)	frmVPVar

4. Código de Programación por Pantalla.

4.1 Generalidades .

En cada forma y reporte que se crea, Access genera un **módulo local** de programación, el cuál contiene las subrutinas y funciones que sólo son accesibles en esa forma o reporte. El módulo esta vacío inicialmente.

Al ir creando cajas-comando en una pantalla, tales comandos generan subrutinas, las cuáles se almacenan en el módulo local. Los comandos producen acciones tales como cargar pantallas, generar cálculos, borrar los valores de las subformas, inicializar los valores de una pantalla, etc.

De éste modo, a cada comando o acción que se realiza en una pantalla le corresponde una subrutina, de las cuáles, algunas son creadas automáticamente por Access y otras por el programador. Cabe aclarar que los comandos creados se activan en los siguientes eventos:

- Al hacer click en una caja-comando o en ciertas cajas texto.
- Después de actualizar ciertas cajas-texto, ésto se realiza automáticamente después que el usuario edita una caja-texto (al usar la tecla <tab>, <enter> o click en otra caja-texto, caja-combo o caja-comando).
- Al abrir una forma. Esto se realiza automáticamente al abrir el usuario una pantalla.

Adicionalmente a las subrutinas, se pueden crear funciones, las cuáles sólo pueden ser llamadas en la forma o reporte en que se construya, debido a ésto, las funciones generalmente se desarrollan en módulos generales para que puedan ser accedadas desde cualquier forma, reporte o query.

En la descripción que se hace del módulo de programación de cada una de las pantallas, es necesario remarcar los siguientes puntos:

- En todas las secciones que requieren datos por parte del usuario, se desarrolló una sección de validación de datos. Adicionalmente se escriben instrucciones en Visual Basic para que se atrapen errores de otra índole automáticamente, por ejemplo divisiones por cero, overflows, etc. En cada caso, al ocurrir un error, ya sea la sección de validación de datos o el sistema (Access), emiten un mensaje indicando el problema a lo cuál el usuario de EmCalc debe responder con <OK> para que la aplicación se reanude.
- Por ser el código para atrapar errores repetitivo, se describe al inicio de esta sección, y se omite del código de las subrutinas. Todas las subrutinas contienen instrucciones para atrapar errores.
- Se elimina también de la descripción del código, la instrucción **Me.Refresh** la cuál hace que la pantalla se refresque desplegando los nuevos valores (recién calculados) al correr cada subrutina. En general y sin ninguna excepción, todas las subrutinas de todas las pantallas utilizan esta instrucción, la cuál se coloca inmediatamente después de todos los cálculos o llamadas a funciones que se realizan. Se ejemplifica su uso junto con el control de errores y se muestra en algunas subrutinas para enfatizar su función.
- Access proporciona la facilidad de referenciar las variables definidas en la pantalla, con el prefijo "**Me.**", por ejemplo, la caja-texto **Resultado** se referencia como **Me.Resultado**. En todas las subrutinas, sin excepción alguna, las variables definidas en la forma en cuestión son referenciadas de esta forma.
- Se presentan al inicio de las subrutinas:
 - Las líneas de carga de pantallas de ayuda, para introducir las variables auxiliares declaradas para cargar las pantallas de ayuda , estas variables no son accesibles al usuario de EmCalc.
 - El comando para borrar o inicializar una pantalla, para introducir las variables utilizadas en la pantalla. En estas variables el usuario especifica valores a emplear en algún proceso de EmCalc.

Código para atrapar errores. Se ejemplifica el caso general usando una caja-comando llamada Command7. Debido a que el evento se genera al hacer clic en ella, Access genera automáticamente el nombre Command7_Click para la subrutina, es a discreción del programador el cambiar el nombre o no. Nótese como todas las etiquetas tienen el nombre relacionado con el nombre de la subrutina:

```
Private Sub Command7_Click() 'Nombre dado por Access, generado por el evento de hacer clic.
On Error GoTo Err_Command7_Click 'Si hay un error al ejecutar una instrucción en la subrutina, el control de
ejecución del programa se transfiere a ésta etiqueta.
...
...Líneas de la subrutina 'Las intrucciones de la subrutina siempre aparecen enmedio. Al momento
... 'de ejecución, cada una de estas líneas es checada por Access y en caso
... 'de haber error se emite un mensaje.

Me.Refresh 'Refresca la pantalla desplegando los nuevos valores.

Exit_Command7_Click: 'Ya sea porque hubo error o por el flujo normal del programa, se llega a
Exit Sub 'esta etiqueta para terminar la subrutina

Err_Command7_Click: 'Si hay algún error, el programa llega a esta etiqueta.
MsgBox Err.Description 'Visual Basic prepara el mensaje de error: Err.Description contiene la
'descripción del error que es atrapado automáticamente por Access.

Resume Exit_Command7_Click 'El control del programa se transfiere a esta etiqueta (arriba).
End Sub 'Termina la subrutina.
```

NOTA: en cada sección donde interactúan cajas-combo, por ejemplo, una para mostrar las funciones disponibles y otra para mostrar las funciones a borrar, las cajas-combo deben tener el siguiente código para sincronizar su contenido.

```
Private Sub FuncionAGuardar_AfterUpdate() 'La subrutina es creada para el evento después de actualizar.
Me.Refresh 'Es el único comando necesario.
End Sub
```

Operaciones y Funciones Básicas

Funciones Preconstruidas				Memorias			
Función	Argumento (arg)	Resultado	Unbound	Borrar	M1	M2	M3
Unbound	Unbound	=	Unbound	M1	M2	M3	M4
	Pasar resultado al Argumento	Pasar a:	op1 M1 +M1	M2	M3	M4	M5
			op2 M2 +M2	M3	M4	M5	Todas
			M3 +M3	M4	M5	Suma	
			M4 +M4	M5			
			M5 +M5				

Variables auxiliares invisibles, para cargar las páginas de ayuda.

Operaciones				Pasará a:			
Operando 1 (op1)	Operador	Operando 2 (op2)	Resultado	A	r	u	m
Unbound	+	Unbound	=	M1 +M1	M2 +M2	M3 +M3	M4 +M4
				M2 +M2	M3 +M3	M4 +M4	M5 +M5
				M3 +M3	M4 +M4	M5 +M5	
				M4 +M4	M5 +M5		
				M5 +M5			

BORRAR TODO EN LA PANTALLA

4.2. Operaciones y Funciones Básicas.

Option Compare Database

'Default de Access.

***Carga de las páginas de ayuda. Sólo la primera subrutina se escribe completamente, para las restantes páginas de ayuda se indica solamente la línea que se cambia.**

Private Sub Command70_Click()

On Error GoTo Err_Command70_Click

Dim stDocName As String

Dim stLinkCriteria As String

'Se atrapan los errores.

'Contendrá nombre de la forma a abrir

'Parámetro para abrir la forma.

stDocName = "frmInstrucciones"

'forma a abrir frmInstrucciones está ligada a la tabla

'tblInstrucciones que contiene los campos forma e

'Instrucciones, nombres usados en la forma también.

stLinkCriteria = "[Forma]=" & "" & Me![OpyFunciones] & "" 'La página a abrir es donde el campo forma

DoCmd.OpenForm stDocName, , , stLinkCriteria

'contiene el nombre OpyFunciones

Exit_Command70_Click:

Exit Sub

'Regreso del programa en caso que haya habido error.

'Salida de la subrutina

Err_Command70_Click:

MsgBox Err.Description

Resume Exit_Command70_Click

'En caso de error el programa llega a esta etiqueta

'Access envía mensaje de error.

'Regresa a esta etiqueta para terminar la subrutina

End Sub

'Termina el código.

Otras páginas de ayuda. Instrucción que se cambia en la subrutina de carga de pantalla.

stLinkCriteria = "[Forma]=" & "" & Me![FPCnst] & "" 'Ayuda de la sección de Funciones.

stLinkCriteria = "[Forma]=" & "" & Me![Ops] & "" 'Ayuda de la sección de Operaciones.

stLinkCriteria = "[Forma]=" & "" & Me![Mem] & "" 'Ayuda de la sección de memorias.

OpyFunciones, FPCnst, Ops, Mem son las variables auxiliares definidas para cargar las formas de ayuda. Son definidas como variables de texto y su contenido es idéntico a su nombre.

***Comando "Borrar todo en pantalla". Inicializa las variables definidas en esta forma.**

Private Sub Command66_Click() 'Nombre dado por Access.

'Submódulo de Operaciones:

Me.Op1 = 0 'Operandos 1 y 2.

Me.Op2 = 0

Me.Opr = "+" 'Caja-combo de operador se inicializa a "+"

Me.Result1 = 0 'Resultado de la operación.

'Submódulo de Funciones:

Me.Funcion = "Log" 'Caja-combo de la función se inicializa a Log.

Me.Arg = 0 'Argumento de la función

Me.Result2 = 0 'Resultado de la función.

'Submódulo de Memorias:

Me.M1 = 0 'Memorias M1 a M6

Me.M2 = 0

Me.M3 = 0

Me.M4 = 0

Me.M5 = 0

Me.M6 = null 'M6 es la suma de memorias M1 a M5.

Me.Refresh 'Despliega las variables inicializadas.

End Sub

• Submódulo de Operaciones:

'Resultado de la operación. Se genera con click en caja-comando "=" o click en la caja-texto "Result1"
Private Sub Command51_Click() 'Click en caja-comando "="
Me.Result1 = Operacion(Me.Op1, Me.Opr, Me.Op2) 'Llamado a la función Operación (Módulo General).
Me.Refresh 'Despliega el valor de la operación.
End Sub

'Resultado de la operación. Generado por un click en la caja-texto "Result1"
Private Sub Result1_Click() 'Nótese como el código es idéntico al de la subrutina anterior.
Me.Result1 = Operacion(Me.Op1, Me.Opr, Me.Op2) 'Misma función a la mencionada arriba.
Me.Refresh 'Despliega el valor de la operación.
End Sub

Private Sub Command15_Click() 'Paso del resultado de la operación al operando 1
Me.Op1 = Me.Result1
Me.Result1 = 0 'El resultado se hace cero porque ya no refleja el valor
Me.Refresh 'de la operación con el nuevo operando.
End Sub

'Para pasar el resultado al operando 2 se usa Op2 en la línea: Me.Op1 = Me.Result1

Private Sub Command25_Click() 'Se pasa el resultado de la operación a la memoria 1.
Me.M1 = Me.Result1
Me.M6 = Null 'M6 ya no refleja la suma de las memorias M1 a M5.
Me.Refresh 'Despliega el nuevo valor de M1 y M6
End Sub

'Mismo caso para pasar resultado de la operación a las memorias M2,...,M5

Private Sub Command113_Click() 'Suma el resultado de la operación a M1
Me.M1 = Me.M1 + Me.Result1 'El resultado de la operación se suma a M1
Me.M6 = Null 'M6 ya no refleja la suma de las memorias.
Me.Refresh 'Despliega los nuevos valores de M1 y M6.
End Sub

'Para sumar el resultado de la operación a las memorias M2 a M5, se sustituye M1 por la que se desée: M2,...,M5.

• Submódulo de Funciones:

'Resultado de la función. Se genera con click en caja-comando "=" o click en la caja-texto "Result2"
Private Sub Command97_Click() 'Caja-comando "=" al hacerse click en ella.
Me.Result2 = PreConst(Me.Funcion, Me.Arg) 'Llamada a la función Preconst (Módulo General).
Me.Refresh 'Despliega el valor de la función.
End Sub

'Resultado de la función. generado con click en la caja-texto "Result2"
Private Sub Result2_Click() 'Nótese como el código es idéntico a la subrutina anterior.
Me.Result2 = PreConst(Me.Funcion, Me.Arg) 'Llamada a la función Preconst (Módulo General).
Me.Refresh 'Despliega el valor de la función
End Sub

Private Sub Command145_Click() 'Paso del resultado de la función al argumento.
 Me.Arg = Me.Result2
 Me.Result2 = 0 'El resultado se hace cero porque ya no refleja el valor
 Me.Refresh 'de la función con el nuevo argumento.
 End Sub

Private Sub Command86_Click() 'Se pasa el resultado de la función a la memoria 1.
 Me.M1 = Me.Result2
 Me.M6 = Null 'M6 ya no refleja la suma de las memorias M1 a M5.
 Me.Refresh 'Despliega el nuevo valor de M1 y M6
 End Sub
 'Mismo caso para pasar resultado de la función a operandos 1 y 2; y, a las memorias M2,...,M5

Private Sub Command139_Click() 'Suma el resultado de la operación a M1
 Me.M1 = Me.M1 + Me.Result2 'El resultado de la operación se suma a M1
 Me.M6 = Null 'M6 ya no refleja la suma de las memorias.
 Me.Refresh 'Despliega los nuevos valores de M1 y M6.
 End Sub
 'Para sumar el resultado de la función a las memorias M2 a M5, se sustituye M1 por la que se desee: M2,...,M5.

• Submódulo de Memorias:

Private Sub Command43_Click() 'Pasa memoria M1 a Operando 1
 Me.Op1 = Me.M1 'Operando1 recibe el contenido de M1
 Me.Result1 = 0 'El resultado de la operación se inicializa.
 Me.Refresh 'Despliega el nuevo contenido de M1
 End Sub

El Paso de memorias M2,...,M6 a operando1, operando 2, y argumento de la función son idénticos; en la línea Me.Op1 = Me.M1, se reemplaza Op1 por OP2 o Arg; M1 por M2,...,M6

Private Sub Command144_Click() 'Comando para sumar todas las memorias.
 Me.M6 = Me.M1 + Me.M2 + Me.M3 + Me.M4 + Me.M5 'La suma de las 5 memorias se asigna a M6
 Me.Refresh 'Despliega el nuevo valor de la suma.
 End Sub

Private Sub Command107_Click() 'Borra memoria M1 e inicializa la suma de memorias.
 Me.M1 = 0 'Las memorias se hacen cero
 Me.M6 = Null 'La suma de memorias queda en "blanco"
 Me.Refresh 'Despliega los nuevos valores de M1 y M6
 End Sub
 'Caso idéntico para borrar las memorias M2 a M5.

Private Sub Command42_Click() 'Borrado de todas las memorias (incluida la suma)
 Me.M1 = 0 'Se inicializan todas las memorias a cero.
 Me.M2 = 0
 Me.M3 = 0
 Me.M4 = 0
 Me.M5 = 0
 Me.M6 = Null 'M6 se inicializa al valor nulo.
 Me.Refresh 'Despliega los nuevos valores de las memorias.
 End Sub

'Comando "Borrar todo en pantalla". Inicializa todas las variables definidas en esta forma.

Private Sub Command66_Click()

Me.Op1 = 0

Me.Op2 = 0

Me.Result1 = 0

Me.Result2 = 0

Me.M1 = 0

Me.M2 = 0

Me.M3 = 0

Me.M4 = 0

Me.M5 = 0

Me.M6 = null

Me.Arg = 0

Me.Opr = "+"

Me.Funcion = "Log"

Me.Refresh

End Sub

'Operandos 1 y 2.

'Resultado de la operación.

'Resultado de la función.

'Memorias M1 a M6

'M6 es la suma de memorias M1 a M5.

'Argumento de la función

'Caja-combo de operador se inicializa a "+"

'Caja-combo de la función se inicializa a Log.

'Despliega las variables inicializadas.

4.3 Valuación de Funciones.

Valuación de Funciones

Traer Funciones. Guardar Funciones y Borrar Funciones Archivadas.

 Unbound

 Unbound

1) Teclée F(X). 2) Asigne valores a X. 3) Enter, Tab ó Click en Resultado u otro campo.

Unbound Equation([Y],[X],False,0,0)

Asignación en la pantalla

StrEscuson([Y],[X])

Asignación en la pantalla

BORRAR TODO EN LA PANTALLA

Generación de Series y Graficación de Funciones.

Option Compare Database Default de Access

Carga de pantallas de ayuda. Instrucción que se cambia en la subrutina de carga de pantalla.

```

stLinkCriteria = "[Forma]=" & "" & Me![FXgraf] & "" 'Ayuda General de la pantalla.
stLinkCriteria = "[Forma]=" & "" & Me![FXtraer] & "" 'Ayuda para traer funciones guardadas en archivo
stLinkCriteria = "[Forma]=" & "" & Me![FXguardar] & "" 'Ayuda para guardar funciones en archivo
stLinkCriteria = "[Forma]=" & "" & Me![FXeditar] & "" 'Ayuda para escribir una expresión F(X) válida.
stLinkCriteria = "[Forma]=" & "" & Me![FXgraf] & "" 'Ayuda para transferir a pantallas de graficación
    
```

FXgraf, FXtraer, FXguardar, FXeditar, FXgraf, son las variables auxiliares definidas para cargar las formas de ayuda.

***Comando para "Borrar todo en Pantalla".** Muestra las variables definidas en esta pantalla.

```

Private Sub Command66_Click()
    Me.X = Null 'Valor de X, usado para valuar la expresión dada para F(X).
    Me.Y = Null 'Expresión de F(X) (se supone que Y = F(X), de ahí su nombre).
    Me.FuncionAUsar = Null 'Caja-combo de "Función a traer".
    Me.FuncPreC = Null 'Caja-combo de "Funciones preconstruidas".
    Me.FuncionAGuardar = Null 'Caja-texto de "Función a guardar".
    Me.FuncionABorrar = Null 'Caja-combo de "Función a borrar".
    'Z = Equation([Y],[X],False,0,0) 'El resultado es almacenado en la caja-texto "Z". La fórmula es asignada
    'al momento de creación y es valuada automáticamente al actualizar el valde
    'X o cualquier Me.Refresh en cualquier otro objeto.
    La función Equation se describe en el módulo general.
    Me.Refresh Actualiza los valores de la pantalla.
End Sub
    
```

- **Sección traer funciones:** se encarga de hacer accesibles funciones almacenadas en un archivo de usuario y también las funciones preconstruídas que son almacenadas en un área que el usuario sólo puede leer.

***Transfiere Función guardada en archivo de usuario al área de trabajo "F(X)"**

```
Private Sub Command12_Enter()      'FuncionAUsar es una caja-combo que referencia dos columnas
    Me.Y = Me.FuncionAUsar.Column(1)  'de la tabla tblFunciones. La primera es el nombre y la segunda
    Me.Refresh                       'es la expresión. FuncionaUsar.Column () es una tabla de base 0
End Sub                             'por eso se transfiere Column(1), el segundo elemento.
```

***Transfiere la expresión de función preconstruída al final de la expresión en el área de F(X)**

```
Private Sub Command30_Click()
    Me.Y = Me.Y & Me.FuncPreC          'Usa el operador & (concatenar) para adicionar el texto de la
End Sub                               'función preconstruída a la expresión existente F(X).
```

- **Sección de almacenamiento y borrado de funciones.**

Comando para guardar expresiones de funciones F(X), de una variable.

```
Private Sub Command9_Click()
    Dim stDocName As String          'Se declaran variables auxiliares
    Dim NombreFunc As Variant

    If Not IsNull(Me.FuncionAGuardar) Then      'Se verifica que exista un nombre para F(X).
        If Me.FuncionAGuardar = "" Then        'Al no existir, se emite el mensaje y termina la
            MsgBox ("Asigne un nombre a la función. Máximo 25 caracteres.") 'subrutina.
            GoTo Salida_Command9              'Se sale de la subrutina
        End If
    Else
        MsgBox ("Asigne un nombre a la función. Máximo 25 caracteres.") 'No existe nombre para F(X)
        GoTo Salida_Command9              'Se sale de la subrutina
    End If

    'Se verifica que el nombre de la función no exista para evitar que el sistema termine la aplicación.
    NombreFunc = DLookup("[Nombre]", "tblFunciones", "[Nombre] = " & Me.FuncionAGuardar & """)

    'Si NombreFunc es nulo, quiere decir que el nombre asignado se puede utilizar para guardar la función
    If Not IsNull(NombreFunc) Then
        MsgBox ("Nombre de Funcion existente. Antes de guardar, borre el nombre!: " & NombreFunc)
        GoTo Salida_Command9              'Se sale de la subrutina
    Else
        stDocName = "qryGuardaFuncion"      'Se asigna el query que guarda la función en el
        DoCmd.OpenQuery stDocName, acNormal, acEdit 'archivo de funciones de usuario.
    End If
    Salida_Command9:                    'Etiqueta a la que se llega en caso de error.
End Sub
```

Append ? X

Append To

Table Name:

Current Database

Another Database:

File Name:

qryGuardaFuncion . Append Query

tblDummy
Dummy

Field:	Nombre: [Forms]![frmFormula]!FuncionAGuardar	Funcion: [Forms]![frmFormula]!Y	Tipo: 'Fix'
Table:			
Sort:			
Append To:	Nombre	Funcion	Tipo
Criteria:			
or:			

La figura muestra un query "Append" (para agregar registros), para guardar una función de usuario en un archivo. Nombre ← [Forms]![frmFormula]!FuncionAGuardar es la expresión para referenciar el nombre de la caja-texto que contiene el nombre de la función en la forma; Función ← Nombre: [Forms]![frmFormula]!Y es la referencia completa a la expresión de la función F(X) Tipo ← "F(x)" las funciones guardadas desde esta pantalla son de una sola variable: X.

***Borra función del archivo de funciones de usuario por medio de un query.**

```
Private Sub Command17_Click()
    Dim stDocName As String          'Variable auxiliar
    stDocName = "qryBorraFuncion"    'Nombre del query, se asigna a la variable auxiliar.
    DoCmd.OpenQuery stDocName, acNormal, acEdit 'Se ejecuta el comando de borrado.
End Sub
```

• Sección de transferencia hacia las pantallas de Graficación.

Generación de Series y Graficación de Funciones.

***Carga la pantalla de graficación manual, inicializando la subforma de captura. No requiere Me.Refresh.**

```
Private Sub Command18_Click()
    Dim stDocName As String
    Dim stLinkCriteria As String          'Variables auxiliares.
    Dim stDocName1 As String

    stDocName1 = "qryDellSerie"           'Query que borra la tabla a la cuál está ligada la
    DoCmd.OpenQuery stDocName1, acNormal, acEdit 'subforma de captura, para inicializar la subforma.

    stDocName = "frmMain1Serie"          'Nombre de la forma de graficación.
    DoCmd.OpenForm stDocName, , stLinkCriteria 'Carga la forma y la despliega.
End Sub
```

***Carga la pantalla de graficación automática, inicializando la subforma de datos. No requiere Me.Refresh.**

```

Private Sub Command22_Click()
    Dim stDocName1 As String
    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName1 = "QryDel2Serie"
    DoCmd.OpenQuery stDocName1, acNormal, acEdit

    stDocName = "frmMain2Serie"
    DoCmd.OpenForm stDocName, , , stLinkCriteria

End Sub

```

*Variables auxiliares.

*Query que borra la tabla a la cuál está ligada la subforma de captura, para inicializar la subforma.

*Nombre de la forma de graficación.

*Carga la forma y la despliega.

Comando para abrir la pantalla de graficación múltiple.

```

Private Sub Command34_Click()
    Dim stDocName1 As String
    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName1 = "QryDel2Serie"
    DoCmd.OpenQuery stDocName1, acNormal, acEdit

    stDocName = "frmMain4Serie"
    DoCmd.OpenForm stDocName, , , stLinkCriteria

End Sub

```

*Nombre dado por Access

*Contendrá nombre de query que borrará subforma

*Contendrá nombre de la forma a abrir.

*Es el criterio a emplear, aquí no es empleado, lo crea Access automáticamente.

*Este query borra los datos de captura para que aparezca la subforma vacía.

*Carga pantalla que prepara la graficación múltiple con subforma de captura en blanco.

4.3.1 Graficación con Captura Manual de Puntos.

***F1Func** Unbound **Passar a F(X)** **Graficación de Funciones** **GF1graf**

F(X) **Borrar Expresión de F(X)** **Borrar Valores de X** **Ver Gráfica de la Serie**

Unbound

Esta función es traída automáticamente de la pantalla anterior con una subrutina al cargar esta forma (On Open Form). El usuario la puede traer de un archivo y editar, ó empezar una nueva.

***GF1ctl** **Control de Errores, al Valuar la Función:**

Si hay división por cero, trace F(X) Unbound
 Si hay overflow, trace F(X) Unbound
 Si hay otro error, trace F(X) Unbound

***GF1teclear** **aquí los valores de X que Desea Valuar, utilizar**

frm15 Serie

Aquí se teclean los datos de X, a graficar. Es una subforma que se "encima" a la forma principal. Los datos se guardan en una tabla.

Borrar Valores de X
Valor F(X) en los Puntos Dados
Valor Función y Graficar

Esta subforma esta protegida para que el usuario no pueda modificar los datos. Los cuales sólo pueden ser capturados, editados ó borrados en la subforma de entrada.

Esta subforma, se crea inicialmente ligada a la misma tabla que la subforma de captura, pero el query que la liga crea también el valor de F(X). Después de creada, se "encima" a la forma principal.

Option Compare Database

*Pantallas de ayuda. Instrucción que se cambia en la subrutina de carga de pantalla.
 stLinkCriteria = "[Forma]=" & "" & Me![GF1graf] & "" *Ayuda general de la pantalla.
 stLinkCriteria = "[Forma]=" & "" & Me![F1Func] & "" *Ayuda para traer función de archivo. stLinkCriteria = "[Forma]" & "" & Me![GF1ctl] & "" *Ayuda para control de errores
 stLinkCriteria = "[Forma]" & "" & Me![GF1teclear] & "" *Ayuda para capturar datos.

GF1graf, F1Func, GF1ctl, GF1teclear, son las variables auxiliares definidas para cargar las formas de ayuda.

Variables definidas en esta sección:
 X Caja-texto, valor de X para valuar la expresión de F(X)
 Y Expresión para F(X)
 Funcion Caja-combo de la función a traer del archivo.
 Err0 Valor a emplear en caso de división por cero.
 ErrOv Valor a emplear en caso de Overflow.
 ErrOt Valor a emplear en caso de errores de otra índole

***Transfiere expresión de función de la caja-combo al área de F(X).**
 Private Sub Command45_Click() *la caja-combo función es un arreglo de 2 elementos, con base
 Me.Y = Me.Funcion.Column(1) *cero. Column(0) es el Nombre y Column (1) la Expresión.
 End Sub

***Genera la serie de puntos en el eje X y muestra F(X) en esos puntos.**
 Private Sub Command40_Click()
 If IsNull(Me.Y) Then *verifica que la función exista
 MsgBox "**Error. No existe Función a valuar."
 End If
 Me.Refresh *Después de capturar X's, al refrescar la pantalla se muestran
 End Sub *los valores F(X).

***Genera la serie de puntos en el eje X , F(X) y despliega la gráfica.**

Nota: el comando "Ver gráfica de la serie" tiene código idéntico a éste.

Private Sub Command41_Click()

Dim stDocName As String, Puntos As Long 'Variables auxiliares.

Dim stLinkCriteria As String

If IsNull(Me.Y) Then 'Verifica que haya función a evaluar.

MsgBox "**Error. No existe Función a evaluar."

Exit Sub 'Si hay error termina la subrutina

End If

Puntos = DCount("[Serie1]", "tbl1Serie")

'Verifica que se hayan capturado datos

If Puntos = 0 Then

MsgBox "**Error. No hay datos a graficar. Introduzca datos en la tabla que se provee."

Exit Sub 'Si hay error termina la subrutina

End If

stDocName = "frmCh0"

'Nombre de la forma que contiene la gráfica

DoCmd.OpenForm stDocName, , , stLinkCriteria

'Se carga la pantalla de la gráfica.

End Sub

***Borra los valores capturados para X**

Private Sub Command42_Click()

Dim stDocName As String

'Variables auxiliares.

stDocName = "qryDel1Serie"

'Se borran los datos con un query de borrado.

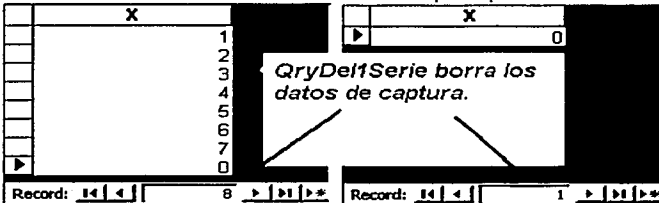
DoCmd.OpenQuery stDocName, acNormal, acEdit

Me.Refresh

'Se refresca la pantalla para que la subforma

End Sub

'de captura aparezca vacía.



Efecto del query de borrado de datos de captura.

***Borra la expresión de F(X)**

Private Sub Command57_Click()

Me.Y = Null

'Inicializa F(X).

End Sub

4.3.2 Graficación con Generación de Puntos Automáticamente.

Unbound **Graficación de Funciones**

Unbound

Esta área se maneja en forma idéntica a la pantalla de graficación manual. Es traída automáticamente de la pantalla anterior, pero el usuario puede traerla de un archivo y modificarla ó empezar una nueva.

Control de Errores al Valorar la Función:

<input type="checkbox"/> Si hay división	por Cero, hacer:	<input type="text" value="00"/>	Unbound
<input type="checkbox"/> Si hay Overflow	hacer:	<input type="text" value="00"/>	Unbound
<input type="checkbox"/> Si hay otro Error	hacer:	<input type="text" value="00"/>	Unbound

Fin de X a utilizar para la Serie Gráfica:

 Unbound

Seleccione una de las opciones:

<input type="button" value="Crear Serie Solamente"/>	<input type="button" value="Crear Serie Solamente"/>
<input type="button" value="Crear Serie y Graficar"/>	<input type="button" value="Crear Serie y Graficar"/>

Option Compare Database

Esta subforma despliega (X,F(X)).
 X se crea con el rango [Inicio, Fin]. F(X) se crea al oprimir cualquiera de las 4 opciones de <crear serie>.

Los puntos X se pueden editar después de creados y F(X) serán recalculados automáticamente debido a la fórmula que radica en el query.

Los puntos Y se protegen para evitar su modificación.

La subforma guarda los valores en la misma tabla que la pantalla de graficación manual.

*Pantallas de ayuda. Instrucción que se cambia en la subrutina de carga de pantalla.
 stLinkCriteria = "[Forma]=" & "" & Me![GF2Gral] & "" *Ayuda General
 stLinkCriteria = "[Forma]=" & "" & Me![F1Func] & "" *Extracción de funciones del archivo
 stLinkCriteria = "[Forma]=" & "" & Me![GF1Ctrl] & "" *Control de errores al valorar F(X)
 stLinkCriteria = "[Forma]=" & "" & Me![GF2Rng] & ""
 stLinkCriteria = "[Forma]=" & "" & Me![GF2Sel] & ""

GF2Gral, F1Func, GF1Ctrl, GF2Rng, GF2Sel, son las variables auxiliares definidas para cargar las formas de ayuda.

Variables definidas en esta forma:

X Caja-texto, valor de X para valorar la expresión de F(X)
 Y Expresión para F(X)
 Funcion Caja-combo de la función a traer del archivo.
 Err0 Valor a emplear en caso de división por cero.
 ErrOv Valor a emplear en caso de Overflow.
 ErrOt Valor a emplear en caso de errores de otra índole
 Inicio Inicio del rango de X a graficar.
 Fin Fin del rango de X a graficar.
 Puntos Se usa para especificar cuantos puntos se desean crear en el rango.
 Incremento Se usa para especificar el incremento a emplear para crear los puntos a partir del rango inicial.

Private Sub Command41_Click()
 Me.Y = Me.Funcion.Column(1)
 End Sub

*Transfiere la expresión de la caja-combo función al área de F(X).
 *Column(0) es el nombre y Column(2) es la expresión de la función.

*Crea los puntos en el eje X y muestra la valuación de F(X) en esos puntos.

```
Private Sub Command31_Click()
Dim stDocName As String
```

*Este bloque valida valores de entrada proporcionados por el usuario:

```
If IsNull(Me.Y) Then 'Verifica que exista F(X)
MsgBox "**Error. No existe Función a evaluar."
Exit Sub
End If
If Not IsNumeric(Me.Inicio) Or Not IsNumeric(Me.Fin) Then 'Inicio y Fin deben ser válidos.
MsgBox ("Error. Especifique valores numéricos en 'Inicio' y 'Fin'.")
Exit Sub
End If
If Val(Me.Inicio) > Val(Me.Fin) Then 'Rango reversado no válido.
MsgBox "**Error. valor de 'Inicio' debe ser mayor que 'Fin'"
Exit Sub
End If
If Not IsNumeric(Me.Puntos) Then 'No. de puntos debe ser válido.
Me.Puntos = 10
Else
If Me.Puntos = 0 Then 'se crean 10 puntos por default
Me.Puntos = 10
End If
End If
```

```
stDocName = "mcrCreaSerie1.Grafica2" 'La macro borra la tabla de captura de datos y
DoCmd.RunMacro stDocName 'después genera los nuevos puntos por medio
Me.Incremento = (Me.Fin - Me.Inicio) / Me.Puntos 'de la función CreaSerie.
Me.Refresh 'Los comandos de la macro aparecen a
End Sub 'continuación.
```

*Instrucciones de la macro mcrCreaSerie1.Grafica2. Borra valores que pudiera haber y genera los propios.

```
OpenQuery qryDel2Serie 'Borra tabla de captura.
RunCode CreaSerie ([Forms]![frmMain2Serie]![Inicio], 'Genera puntos con la función CreaSerie.
[Forms]![frmMain2Serie]![Fin], [Forms]![frmMain2Serie]![Puntos])
```

F(X):	X	Y
Sin(x) * Cos(x)*x^2	0	0.
Inicio: <input type="text" value="0"/>	0.2	0.0078836684617301
Fin: <input type="text" value="1"/>	0.4	0.0573684872719618
No de puntos: <input type="text" value="5"/>	0.6	0.167767035474101
<input type="button" value="Crear Serie Solamente"/>	0.8	0.319863552973282
	1	0.454648713412841

*Crea puntos en eje X, crea la valuación de F(X) y carga la gráfica.

```
Private Sub Command6_Click()
Dim stDocName, stDocName1, stLinkCriteria As String
```

*Esta línea corresponde a código idéntico al de la subrutina anterior para validar valores de entrada.

```
stDocName = "mcrCreaSerie1.Grafica2" 'La macro borra datos de captura y genera la
DoCmd.RunMacro stDocName 'serie de puntos a graficar
stDocName1 = "frmCh1" 'Se asigna el nombre de la forma que
```

```

DoCmd.OpenForm stDocName1, , , stLinkCriteria
Me.Incremento = (Me.Fin - Me.Inicio) / Me.Puntos
End Sub

```

```

'despliega la gráfica y se despliega.
'Valor del incremento.

```

```

'Crea los puntos en el eje X por medio de un incremento y muestra la valuación de F(X) en esos puntos.
Private Sub Command32_Click()
Dim stDocName As String

```

```

'Se verifica que exista función; valores de Inicio y Fin; que no esté
'reversado el rango a graficar; en forma idéntica a las subrutinas anteriores.
If Not IsNumeric(Me.Incremento) Then 'Si el incremento no es válido se crea
Me.Incremento = (Me.Inicio - Me.Fin) / 10 'de tal forma que proporcione 10 pts.
Else
If Me.Incremento <= 0 Then 'El incremento debe ser mayor a cero.
Me.Incremento = (Me.Fin - Me.Inicio) / 10
End If
End If
If Val(Me.Incremento) > (Val(Me.Fin) - Val(Me.Inicio)) Then 'El incremento debe ser menor al rango
Me.Incremento = (Me.Fin - Me.Inicio) / 10 'a graficar.
End If
stDocName = "mcrCreaSerie1.Grafica3" 'La macro borra valores y genera los puntos a
DoCmd.RunMacro stDocName 'graficar.
Me.Puntos = Int((Me.Fin - Me.Inicio) / Me.Incremento) 'No. de puntos creados, informativo.
End Sub

```

```

Macro mcrCreaSerie1.Grafica3. Borra valores previos de la tabla X y genera los nuevos, junto con F(X).
OpenQuery qryDel2Serie 'Query para borrar datos de entrada.
RunCode CreaSerie ([Forms]![frmMain2Serie]![Inicio], 'Genera puntos con la función CreaSerie.
[Forms]![frmMain2Serie]![Fin], [Forms]![frmMain2Serie]![Puntos])

```

```

'Crea los puntos en el eje X por medio de un incremento y muestra F(X) en esos puntos. Despliega la gráfica.
Private Sub Command22_Click()
Dim stDocName, stDocName1, stLinkCriteria As String, Puntos As Long

```

```

'Se verifica que exista función; valores de Inicio y Fin; que no esté
'reversado el rango a graficar; en forma idéntica a la subrutina anterior.
stDocName = "mcrCreaSerie1.Grafica3" 'La macro borra valores y genera los puntos a
DoCmd.RunMacro stDocName 'graficar.
Me.Puntos = Int((Me.Fin - Me.Inicio) / Me.Incremento) 'No. de puntos creados, informativo.
End Sub

stDocName1 = "frmCh1" 'Asigna nombre de la forma que
DoCmd.OpenForm stDocName1, , , stLinkCriteria 'despliega la gráfica y la carga.
End Sub

```

```

Private Sub Command5_Click() 'Borra los valores de los puntos X generados
Dim stDocName As String
stDocName = "qryDel1Serie" 'Query para borrar datos X.
DoCmd.OpenQuery stDocName, acNormal, acEdit 'Ejecuta el query.
Me.Refresh 'Despliega subforma en blanco.
End Sub

```

4.3.3 Graficación Múltiple.

Graficación de Múltiples Funciones

1) Teclar, o Seleccionar Funciones y Transferir a F1(X), ..., F5(X)

2) Control de Errores al Valorar las Funciones.

3) Usar una de las 3 Opciones, de abajo para Errores Valores de X

4) Ver Gráfica de las Funciones

Estos valores se usarán para x_i de las funciones.

Si hay Overflow, teclar F(0) Unbound

Si hay Otro Error, teclar F(0) Unbound

Esta subforma sólo permite la captura de X. También está ligada a la misma tabla que las pantallas anteriores.

También se pueden crear los puntos X's automáticamente.

Borrar valores de X

Option Compare Database

*Pantallas de ayuda. Instrucción que se cambia en la subrutina de carga de pantalla.

```
stLinkCriteria = "[Forma]" & "" & Me![GF3Gra] & ""
stLinkCriteria = "[Forma]" & "" & Me![F3Func] & ""
stLinkCriteria = "[Forma]" & "" & Me![GF3Opc] & ""
stLinkCriteria = "[Forma]" & "" & Me![GF2Rng] & ""
stLinkCriteria = "[Forma]" & "" & Me![GF3Sel] & ""
stLinkCriteria = "[Forma]" & "" & Me![GF1ctl] & ""
stLinkCriteria = "[Forma]" & "" & Me![GF1teclar] & ""
stLinkCriteria = "[Forma]" & "" & Me![GF3Graf] & ""
```

GF3Gra], F3Func, GF3Opc, GF2Rng, GF3Sel, GF1ctl, GF1teclar, GF3Graf, son las variables auxiliares definidas para cargar las formas de ayuda.

Variables definidas en esta forma:

Funcion	Caja-combo de la función a traer del archivo.
X	Caja-combo que especifica a donde transferir la función traída del archivo: F1(X),...,F5(X)
X1	Caja-texto, Expresión de la función a graficar F1(X).
X2	Caja-texto, Expresión de la función a graficar F2(X).
X3	Caja-texto, Expresión de la función a graficar F3(X).
X4	Caja-texto, Expresión de la función a graficar F4(X).
X5	Caja-texto, Expresión de la función a graficar F5(X).
Err0	Valor a emplear en todas las funciones, en caso de división por cero.
ErrOv	Valor a emplear en todas las funciones, en caso de Overflow.
ErrOt	Valor a emplear en todas las funciones, en caso de errores de otra índole
Inicio	Inicio del rango de X a graficar.
Fin	Fin del rango de X a graficar.
Puntos	Se usa para especificar cuantos puntos se descan crear en el rango.
Incremento	Se usa para especificar el incremento a emplear para crear los putos a partir del rango inicial.

Private Sub Command53_Click()

'Transfiere una Funcion a F1, F2, F3, F4, o F5.

Select Case Me.X

'Se analiza el valor de la caja-combo X (que es texto)

Case "F1(X)": Me.X1 = Me.Funcion.Column(1)

'Si es "F1(X)" se transfiere la función a 1 campo X1 y

Case "F2(X)": Me.X2 = Me.Funcion.Column(1)

'sucesivamente para las demás Fi(X).

Case "F3(X)": Me.X3 = Me.Funcion.Column(1)

Case "F4(X)": Me.X4 = Me.Funcion.Column(1)

'Column(0) es el nombre y Column(1) la expresión

Case "F5(X)": Me.X5 = Me.Funcion.Column(1)

'de la función.

End Select

End Sub

Función	2SinCos	▼	Passar a:	▼	⌵	
F1(x)			F1(x)			Borrar
F2(x)	Sin(x) * Cos(x)*x^2		F3(x)			Borrar
F3(x)			F4(x)			Borrar
F4(x)			F5(x)			Borrar
F5(x)						Borrar

Private Sub Command58_Click()

'Borra F1(X).

Me.X1 = ""

F1(X) se inicializa en forma de texto.

End Sub

Para borrar F2, F3, F4 o F5, se sustituye X1 por X2, X3, X4 o X5 respectivamente.

'Genera automaticamente los puntos a graficar. Opción "Número de Puntos".

Private Sub Command31_Click()

Dim stDocName As String

'Verifica que exista al menos una función a graficar.

If IsNull(Me.X1) And IsNull(Me.X2) And IsNull(Me.X3) And IsNull(Me.X4) And IsNull(Me.X5) Then
MsgBox "**Error. No existen Funciones a graficar."

Exit Sub

End If

If Not IsNumeric(Me.Inicio) Or Not IsNumeric(Me.Fin) Then 'Verifica que el Inicio y Fin del
MsgBox ("Error: Especifique valores numéricos en 'Inicio' y 'Fin'.") 'rango a graficar sean válidos.

Exit Sub

End If

If Val(Me.Inicio) >= Val(Me.Fin) Then 'Verifica que el rango no esté reversado

MsgBox "**Error. valor de 'Inicio' debe ser mayor que 'Fin'"

Exit Sub

End If

If Not IsNumeric(Me.Puntos) Then 'Genera 10 puntos por default en caso que sea
Me.Puntos = 10 'inválido el valor.

Else

If Me.Puntos = 0 Then

Me.Puntos = 10

End If

End If

stDocName = "mcrCreaSerie1.Grafica4"

'Nombre de la macro que genera los

DoCmd.RunMacro stDocName

'puntos. Véase abajo instrucciones.

Me.Incremento = (Me.Fin - Me.Inicio) / Me.Puntos

'El incremento es sólo informativo.

End Sub

Instrucciones de la macro: **merCreaSerie1.Grafica4**. Borra los puntos que pudiera haber y crea unos nuevos.
 OpenQuery QryDel2Serie 'Inicializa la subforma de datos.
 RunCode CreaSerie2([Forms]![frmMain4Serie]![Inicio], 'Genera puntos con la función Creaserie2.
 [Forms]![frmMain4Serie]![Fin], [Forms]![frmMain4Serie]![Puntos])

***Genera automáticamente los puntos a graficar. Opción "Número de puntos".**

Private Sub Command32_Click()

Dim stDocName As String

'Los siguientes chequeos los realiza en forma idéntica a la subrutina anterior.

'Verifica que exista al menos una función a graficar;

'Verifica que el rango a graficar exista y tenga valores válidos y que no esté reversado

If Not IsNumeric(Me.Incremento) Then

'Verifica que el Incremento entre puntos sea

Me.Incremento = (Me.Inicio - Me.Fin) / 10

'válido, de ser inválido genera un incremento

Else 'de modo que resulten 11 puntos a graficar.

If Me.Incremento <= 0 Then

Me.Incremento = (Me.Fin - Me.Inicio) / 10

End If

End If

If Val(Me.Incremento) >= (Val(Me.Fin) - Val(Me.Inicio)) Then 'La función "Val" garantiza que se valúen

Me.Incremento = (Val(Me.Fin) - Val(Me.Inicio)) / 10 'las variables en forma numérica.

End If

stDocName = "merCreaSerie1.Grafica5"

'Nombre de la macro que genera los puntos a

DoCmd.RunMacro stDocName

'graficar. Véase abajo instrucciones.

Me.Puntos = Int((Me.Fin - Me.Inicio) / Me.Incremento)

'Genera el número de puntos a graficar.

End Sub

Instrucciones de la macro: **merCreaSerie1.Grafica5**. Borra los puntos que pudiera haber y crea unos nuevos.

OpenQuery QryDel2Serie 'Borra los valores en la subforma.

RunCode CreaSerie2([Forms]![frmMain4Serie]![Inicio],

'Genera puntos con la función Creaserie2.

[Forms]![frmMain4Serie]![Fin],[Forms]![frmMain4Serie]![Incremento])

Private Sub Command5_Click()

'Borra los datos generados para X.

Dim stDocName As String

stDocName = "qryDel1Serie"

'Nombre del query que borra los datos de la tabla de

DoCmd.OpenQuery stDocName, acNormal, acEdit 'de datos de X.

End Sub

Private Sub Command37_Click()

'Genera la gráfica, cargando la pantalla que despliega la gráfica.

Dim stDocName As String

Dim stLinkCriteria As String

'Verifica que exista al menos una función a graficar.

If IsNull(Me.X1) And IsNull(Me.X2) And IsNull(Me.X3) And IsNull(Me.X4) And IsNull(Me.X5) Then

MsgBox "Error. No existen Funciones a graficar."

Exit Sub

End If

Puntos = DCount("[Serie1]", "tbl1Serie")

'Verifica que existan puntos a graficar.

If Puntos = 0 Then

MsgBox "Error. No hay datos a graficar. Especifique valores para 'Inicio' y 'Fin', y <Crear Serie Solamente>."

Exit Sub

End If

stDocName = "frmmain4Ch"

'Nombre de la forma que despliega la gráfica.

DoCmd.OpenForm stDocName, , , stLinkCriteria

'Carga la pantalla de la gráfica.

End Sub

4.4 Valuación de Funciones F(X,Y).

Funciones de dos variables, en términos de X y Y.

Valuación de Funciones F(X,Y)		FXYgral	Instrucciones/Ayuda
Traer Funciones.			
Archivadas: Unbound		Funciones y Borrar Funciones Archivadas.	
Usar Función Seleccionada		Guardar Función	
Preconstruidas: Unbound		Función a Borrar: Unbound	
Adicionar Función al Final de F(X)		Borrar Función Seleccionada	
1) Clic de F(X,Y) 2) Asigne valores a X y Y. 3) Click en Resultado			
F(X,Y):	X: Unbound	Resultado: Unbound	
	Y: Unbound		
Unbound La expresión para F(X,Y) puede ser en términos de X y/o de Y. Para su valuación se hacen dos pasadas a la expresión, sustituyendo primeramente X por su valor y posteriormente Y por X y de ahí, por su valor.			
Sustituyendo valores de X y Y: F(X,Y) =			
Unbound El desplegado de la sustitución de una expresión por sus valores, ayuda a encontrar los errores en una expresión. También esta sustitución se realiza en dos pasadas a la expresión.			
BORRAR TODO EN LA PANTALLA			

Option Compare Database

*Pantallas de ayuda. Instrucción que se cambia en la subrutina de carga de pantalla.

stLinkCriteria = "[Forma]=" & "" & Me![FXygral] & "" *Ayuda general.
 stLinkCriteria = "[Forma]=" & "" & Me![FXyfunc] & "" *Ayuda para crear expresiones válidas.
 stLinkCriteria = "[Forma]=" & "" & Me![FXyguarda] & "" *Ayuda para guardar funciones.
 stLinkCriteria = "[Forma]=" & "" & Me![FXytecl] & "" *Ayuda para extraer funciones del archivo.

FXygral, FXyfunc, FXyguarda, FXytecl, son las variables auxiliares definidas para cargar las formas de ayuda.

Muestra las variables que se definen en esta forma.

*Comando "Borrar todo en la pantalla".

Private Sub Command66_Click()

*Inicializa las siguientes variables:

Me.X = Null	*Valor de X, usado para valuar la expresión dada para F(X,Y).
Me.Y = Null	*Valor de Y, usado para valuar la expresión dada para F(X,Y).
Me.Fxy = Null	*Expresión de F(X,Y).
Me.FuncionAUsar = Null	*caja-combo de función a traer.
Me.FuncPreC = Null	*caja-combo de funciones Preconstruidas
Me.FuncionAGuardar = Null	*caja-texto de función a guardar
Me.FuncionABorrar = Null	*caja-combo de función a borrar
Me.Resultado = 0	*caja-texto de Resultado de la valuación de la función.
Me.TradY = Null	*traducción de F(X,Y); TradY es la expresión de F(X,Y)
End Sub	*con X y Y sustituidos por sus valores.

- **Sección traer funciones:** Se encarga de hacer accesibles funciones almacenadas en un archivo de usuario y también las funciones preconstruidas que son almacenadas en un área que el usuario sólo puede leer.

***Transfiere función de la caja-combo FunciónAUsar (Función de Usuario) al área de F(X,Y)**

```
Private Sub Command12_Click()
    Me.Fxy = Me.FuncionAUsar.Column(1)           'La caja-combo es un arreglo de 2 elementos:
    Exit_Command12_Click:                       'Column(0) = nombre; column(1) = expresión.
End Sub
```

***Transfiere caja-combo función (Función Preconstruida) al área de F(X,Y)**

```
Private Sub Command30_Click()
    Me.Fxy = Me.Fxy & Me.FuncPreC              'La caja-combo FunPreC tiene una sola columna.
End Sub                                       'El comando "&" concatena los caracteres de las expresiones de
                                           'las funciones .
```

- **Sección de almacenamiento y borrado de funciones.**

***Guarda la expresión en un archivo de funciones de usuario.**

```
Private Sub Command9_Click()
    Dim stDocName As String                    'Se declaran variables auxiliares.
    Dim NombreFunc As Variant

    If Not IsNull(Me.FuncionAGuardar) Then    'Verifica que exista nombre de la función,
        If Me.FuncionAGuardar = "" Then      'todos los caracteres son válidos.
            MsgBox ("Asigne un nombre a la funcion. Maximo 25 caracteres.")
            GoTo Salida_Command9
        End If
    Else
        MsgBox ("Asigne un nombre a la funcion. Maximo 25 caracteres.")
        GoTo Salida_Command9
    End If

    'Verifica que el nombre de la función no exista en el archivo.
    NombreFunc = DLookup("[Nombre]", "tblFunciones", "[Nombre] = " & Me.FuncionAGuardar & """)
    If Not IsNull(NombreFunc) Then
        MsgBox ("Nombre de Funcion existente. Antes de guardar, borre el nombre!: " & NombreFunc)
        GoTo Salida_Command9                'Si el nombre existe, no se guarda la función.
    Else
        stDocName = "qryGuardaFuncionXY"    'Usa query para guardar la expresión
        DoCmd.OpenQuery stDocName, acNormal, acEdit 'Ejecuta el query.
    End If
    Salida_Command9:
End Sub
```

***Borra función del archivo de usuario. No se otorga confirmación.**

```
Private Sub Command17_Click()
    Dim stDocName As String

    stDocName = "qryBorraFuncionXY"         'El query de borrar función se emite, dado
    DoCmd.OpenQuery stDocName, acNormal, acEdit 'que la caja-combo verifica que exista la
                                           'función a borrar. El query toma el nombre de
                                           'la función a borrar, de la caja-combo.
End Sub                                     'Utiliza este query para realizar el borrado de
                                           'la función.
```

• Sección de creación y valuación de expresiones.

'Genera la valuación de la función F(X,Y). Se hace en dos pasos:

'1) Se sustituye X por su valor para que la función quede en términos de Y solamente.

'2) Se reemplazan las Y's por X's, posteriormente se sustituye X por el valor dado para Y, y se valúa.

Private Sub Command45_Click()

Dim FX, FY As String, i, lngL As Integer, OKx, OKy As Boolean 'Variables auxiliares.

If IsNull(Me.Fxy) Then Exit Sub 'Verifica que exista una expresión F(X,Y)

If IsNull(Me.X) And IsNull(Me.Y) And Not IsNull(Me.Fxy) Then '

Me.Resultado = 0 'Valúa a cero si X y Y son valores nulos.

Exit Sub 'Se sale de la subrutina.

End If

lngL = Len(Me.Fxy) 'Asigna la longitud de la expresión.

For i = 1 To lngL 'Detecta si existen X's y/o Y's en la expresión

If Mid(Me.Fxy, i, 1) = "X" Then OKx = True 'de F(X,Y) recorriendo cada carácter de la

If Mid(Me.Fxy, i, 1) = "Y" Then OKy = True 'expresión.

Next i

If Not (OKx Or OKy) Then 'Permite valuar expresiones numéricas

Me.TradY = Me.Fxy 'que no involucran ni X's, ni Y's.

Me.Resultado = Eval(Me.Fxy) 'Eval es función preconstruída de Access

Me.Refresh 'Refresca la pantalla con los nuevos datos

Exit Sub 'Se sale de la subrutina (la termina)

End If 'Termina bloque IF.

'1a. parte. Se reemplazan las X's por el valor dado para X.

If OKx And OKy Then 'Si existen X y Y en la expresión,

FY = StrEcuacion([Fxy], [X]) 'X se sustituye por su valor, quedando FY

'en términos de Y.

Else: If OKy Then FY = Fxy

'FXY ya está en términos de Y, se asigna a FY.

End If

'Caso especial en el que el usuario dá la función en términos de X.

If OKx And Not OKy Then 'Si existen X's, pero no Y's en la expresión,

Me.TradY = StrEcuacion([Fxy], [X]) 'Sustituye X por su valor.

Me.Resultado = Equation([Fxy], [X], False, 0, 0) 'Valúa la expresión. [X] es parámetro dummy.

Exit Sub 'Se sale de la subrutina

End If

'2a. parte. Las Y's se transforman a X's y se valúa.

If OKy Then 'Si sólo hay Y's o ya se transformó a Y's,

lngL = Len(FY) 'transforma la expresión para que quede

FX = "" 'en términos de X.

For i = 1 To lngL 'Convierte Y's a X's

If Mid(FY, i, 1) = "Y" Then 'Mid checa el carácter i-ésimo de la expresión

FX = FX + "X" 'Si encuentra Y la convierte a X, si no,

Else: FX = FX + Mid(FY, i, 1) 'FX queda en términos de X 'transfiere el carácter tal cuál.

End If

Next i

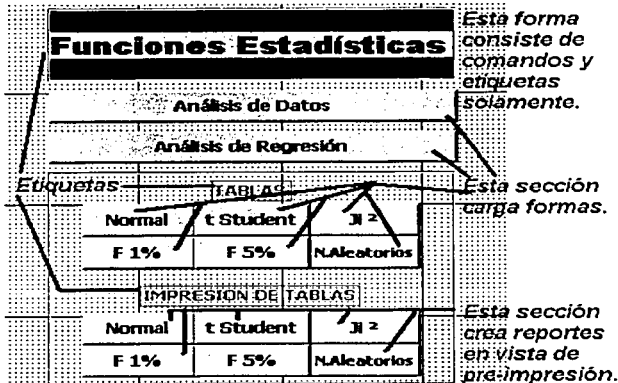
Me.TradY = StrEcuacion([FX], [Y]) 'Sustituye Y por su valor.

Me.Resultado = Equation([FX], [Y], False, 0, 0) 'Valúa la expresión. [Y] es parámetro dummy.

End If

End Sub

4.5 Funciones Estadísticas. Se encarga de cargar formas y Reportes, no hay variables definidas.



Option Compare Database

*Carga de reportes en vista de pre-impresión: ejemplo con Reporte de la Distribución Normal.

```
Private Sub Command13_Click()
```

```
Dim stDocName As String
```

*Variable para poner el nombre del reporte.

```
stDocName = "rptNormal"
```

*Nombre del reporte a desplegar.

```
DoCmd.OpenReport stDocName, acPreview
```

*Despliega el reporte en vista de pre-impresión.

```
End Sub
```

Para cargar los demas reportes se reemplaza la línea `stDocName = "rptNormal"` por:

```
stDocName = "rptTStudent"
```

*t de Student

```
stDocName = "rptCHI2"
```

*Ji cuadrada

```
stDocName = "rptF5Distribution"
```

*Distribución F, 5% de significancia

```
stDocName = "rptF1Distribution"
```

*Distribución F, 1% de significancia

```
stDocName = "frmblNAleat"
```

*Números aleatorios

*Carga de formas. Ejemplo con Análisis de Datos.

```
Private Sub Command2_Click()
```

*Variable para poner el nombre de la forma.

```
Dim stDocName As String
```

*Parámetros a usar (no se usan en este caso)

```
Dim stLinkCriteria As String
```

*Nombre de la forma a desplegar.

```
stDocName = "frmAnDat1"
```

```
DoCmd.OpenForm stDocName, , stLinkCriteria
```

*Despliega la forma.

```
End Sub
```

Para cargar los demas reportes se reemplaza `stDocName = "frmAnDat1"` por:

```
stDocName = "frmAnReg"
```

*Análisis de Regresión

```
stDocName = "tbl1Normal"
```

*Distribución Normal

```
stDocName = "tbl2TStudent"
```

*t de Student

```
stDocName = "tbl5JI2"
```

*Ji cuadrada

```
stDocName = "tbl3F5Distribution"
```

*Distribución F, 5% de significancia

```
stDocName = "tbl4F1Distribution"
```

*Distribución F, 1% de significancia

```
stDocName = "tbl6NumAleatorios"
```

*Números aleatorios

4.5.1 Análisis de Datos.

Análisis de Datos		[-] Adgr	Instrucciones/Ayuda
[-] ADtraer	Borrar Datos	[-] ADguarda	Archivo: Unbound
? Tecler Datos, ó, traer de archivo.		Archivo: Unbound	Traer Archivo
limArDat		Guardar Datos en Archivo	
Esta subforma permite la captura de datos.		[-] ADinfo	2) Generar Información
Se encuentra ligada a la tabla tblAnDat por medio de una query.		3) Ver Historiario	Generar Información
Al hacer click en <Borrar Datos>, un query se encarga de borrar la información de la tabla.		Hacer Histograma	
Al usar el comando <Traer archivo>, un query se encarga de crear una copia del archivo de datos y adicionarlo a la tabla ligada a esta subforma.		limMdn	
Un query se encarga también de guardar los datos en un archivo.		Esta subforma se utiliza para desplegar la información que se calcula para los datos capturados.	
		Al hacer click en <Generar Información> se activa una subrutina que hace un llamado a una función del módulo general, la cuál se encarga de calcular la información y guardarla en una tabla, ligada a esta subforma.	
		la subforma se formatea para que despliegue la información en un formato adecuado (formato de forma continua). Para capturar datos se utiliza formato de Hoja de cálculo	

Option Compare Database

*Pantallas de ayuda. Instrucción que se cambia en la subrutina de carga de pantalla.

stLinkCriteria = "[Forma]=" & "" & Me![ADgral] & ""	*Ayuda general
stLinkCriteria = "[Forma]=" & "" & Me![ADtraer] & ""	*Traer archivos de datos
stLinkCriteria = "[Forma]=" & "" & Me![ADguarda] & ""	*Guardar archivos de datos
stLinkCriteria = "[Forma]=" & "" & Me![ADinfo] & ""	*Generar la información

ADgral, ADtraer, ADguarda, ADinfo, son las variables auxiliares definidas para cargar las formas de ayuda.

Variables definidas:

Datos	Caja-combo de las funciones a traer.
ArchivoABorrar	Caja-combo de las funciones a borrar.

*Subrutina que se corre automaticamente al abrir la hoja de Análisis de Datos. Inicializa subformas para que aparezcan en blanco.

Private Sub Form_Open(Cancel As Integer)

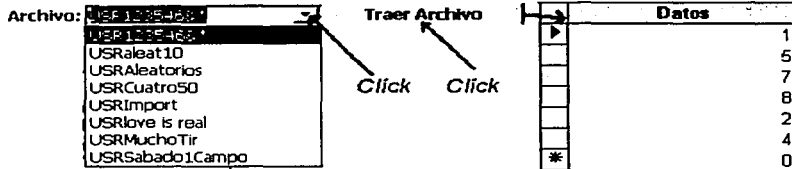
```

Dim stDocName As String
stDocName = "qry02AnDat" 'Inicializa subforma de captura de datos borrando la
DoCmd.OpenQuery stDocName, acNormal, acEdit 'tabla tblAnDat.
stDocName = "qryDeIMdn" 'Inicializa subforma de presentación de información
DoCmd.OpenQuery stDocName, acNormal, acEdit 'calculada borrando tabla tblMdn.
'Datos 'Tiene valor default "Nulo". No se requiere inicializar.
'ArchivoABorrar 'Tiene valor default "Nulo". No se requiere inicializar.
End Sub
    
```

• **Acceso de archivos de datos.**

```
Private Sub TraeDat_Click()
    a = TraeTbl(Me.Datos, 1)
    Me.Refresh
End Sub
```

*Trae archivos de datos de usuario al área de captura de datos.
 *Función creada para extraer archivo, véase sección de Funciones.
 *Permite ver los datos recién traídos.



```
Private Sub Command26_Click()
    Dim stDocName As String
    stDocName = "qry02AnDat"
    DoCmd.OpenQuery stDocName, acNormal, acEdit
End Sub
```

*Borra los datos de captura o traídos de un archivo.
 *El query borra la tabla tblAnDat que es donde se capturan los datos.

```
Private Sub Command25_Click()
    Dim stDocName, stDocName1 As String
    Dim Puntos As Long
    Puntos = DCount("[Valor]", "tblAnDat")
    If Puntos = 0 Then
        MsgBox "**Error. No hay datos a analizar. Click en <Instrucciones/Ayuda> para Información general."
        Exit Sub
    End If
    stDocName = "qryDelMdn"
    DoCmd.OpenQuery stDocName, acNormal, acEdit
    a = AnDatos()
    Me.Refresh
End Sub
```

*Genera la información de los datos.
 *Variable auxiliar.
 *Verifica que haya observaciones a analizar.
 *Borra la subforma que despliega la información por medio de un query.
 *Genera la información. La asignación es dummy, se usa para correr la función que genera los datos.
 *Me.refresh permite ver la nueva información.

```
Private Sub Command24_Click()
    Dim stDocName As String
    Dim stLinkCriteria As String
    Dim Puntos As Long
    Puntos = DCount("[Valor]", "tblAnDat")
    If Puntos = 0 Then
        MsgBox "**Error. No hay datos a graficar. Click en <Instrucciones/Ayuda> para Información general."
        Exit Sub
    End If
    stDocName = "frmCh01AnDat"
    DoCmd.OpenForm stDocName, , stLinkCriteria
End Sub
```

*Genera el histograma de los datos (Carga la forma que contiene la gráfica).
 *Variables auxiliares.
 *Puntos es el número de observaciones.
 *Cuenta el número de datos a graficar.
 *Carga la forma que despliega la gráfica.
 *Ejecuta la acción de cargar gráfica.

```
Private Sub Command28_Click() 'Guarda los datos capturados en un archivo.
    a = GuardaTbl(1) 'Hace una asignación dummy para correr función que guarda los datos.
End Sub
```

```
Private Sub Command48_Click() 'Borra archivo de datos de captura permanente.
If IsNull(Me.ArchivoABorrar) Then 'Verifica que haya un nombre de archivo a
    MsgBox "Acción de borrar archivo cancelada." 'borrar.
Exit Sub
End If
```

```
'Pide confirmación para borrar el archivo.
If MsgBox("Borrar archivo " & Me.ArchivoABorrar & " ?", vbYesNoCancel) <> vbYes Then
    MsgBox "Acción de borrar archivo cancelada."
```

```
Exit Sub
```

```
End If
```

```
DoCmd.DeleteObject acTable, Me.ArchivoABorrar 'Ejecuta el borrado de archivo.
```

```
Me.Refresh
```

```
'Hace que el nombre ya no aparezca en las
```

```
End Sub
```

```
'cajas-combo.
```

4.5.2 Análisis de Regresión.

"RGrcal"

Borrar Todo X Y X Y Y

1) Iniciar valores para X y Y, ó, traer datos de un archivo.

Esta subforma de captura para (X,Y) también contiene datos calculados: *transf(X)*, *transf(Y)*, *Ajuste(Y)*, *Residuales*.

X y Y permiten captura y edición, no así los demás que son protegidos para evitar errores involuntarios.

El borrado de archivos de datos de regresión se realiza en el módulo de *Utilerías/Sol.problemas*.

Si no se especifica *transf(X)*, se usa X para el ajuste, mismo caso para *transf(Y)*.

Los comandos <Ajuste Recta> y <Ajuste Cuadrático> crean las transformaciones en caso que se especifiquen.

La función del ajuste se puede guardar en un archivo y utilizar en cualquier módulo de funciones y de graficación.

Análisis de Regresión | "RGrcal" | Instrucciones/Ayuda

Archivo de Datos: Unbound | Traer Archivo

Guardar Datos en Archivo

"RGtransf" 2) Opcional. Transformaciones para X y Y

Función: Unbound | Pasar a: F(X) F(Y) | Borrar

F(X): Unbound F(Y): Unbound F(X): F(Y):

Div: por cero? F(X): Unbound F(Y): Unbound

GF: Transf. de F(X): Unbound F(Y): Unbound

GF: Transf. de F(Y): Unbound F(X): Unbound

"RGcalc" 3) Opcional. Transformaciones y Ajuste.

Transformaciones para X y Y

Desplegar cálculos del ajuste:

Ajuste Recta Ajuste Cuadrático

Gráfica de Disposición (X y Y)

Gráfica de Transformaciones de X y Y

Gráfica Transf. X y Y con Línea de Ajuste

Ajuste: Unbound Guardar

Option Compare Database

*Pantallas de ayuda. Instrucción que se cambia en la subrutina de carga de pantalla.

stLinkCriteria = "[Forma]" & "" & Me![RGrcal] & "" *Ayuda general
 stLinkCriteria = "[Forma]" & "" & Me![RGrcal] & "" *Ayuda para capturar los datos
 stLinkCriteria = "[Forma]" & "" & Me![RGtransf] & "" *Ayuda para acceder funciones
 stLinkCriteria = "[Forma]" & "" & Me![GF1ctl] & "" *Ayuda para control de errores en las funciones.
 stLinkCriteria = "[Forma]" & "" & Me![RGcalc] & "" *Ayuda para calcular la recta o curva de ajuste.

Rggral, Rgtrcl, Rgtrnsf, GF1ctl, RGcalc, son las variables auxiliares definidas para cargar las formas de ayuda.

*Borra los campos de la pantalla, incluyendo subformas de captura de datos y presentación de información.

```
Private Sub Command50_Click()
    Dim stDocName As String
    stDocName = "qry01_1RegDelXY" 'Variable auxiliar.
    DoCmd.OpenQuery stDocName, acNormal, acEdit 'Query que borra los datos X,Y.
    stDocName = "qry02AnRegDel" 'transformaciones X,Y, ajuste y residuales.
    DoCmd.OpenQuery stDocName, acNormal, acEdit 'Query que borra la subforma de presentación
    'de información.
```

*Variables definidas:

Me.FX = Null 'Función de la transformación de X.
 Me.FY = Null 'Función de la transformación de X.
 Me.Funcion 'Caja-combo de función a traer a F(X) o F(Y).
 Me.Err0 = 0 'Control de error div por cero, transf. Y
 Me.Err0X = 0 'Control de error div por cero, transf. X
 Me.Err0v = 0 'Control de error Overflow, transf. Y
 Me.Err0vX = 0 'Control de error Overflow, transf. X
 Me.Err0t = 0 'Control de "Otro error", transf. Y
 Me.Err0tX = 0 'Control de "Otro error", transf. X
 Me.Arch = Null 'Nombre de archivo de datos a guardar.
 *Modelo es una variable auxiliar que no se despliega. Almacena la expresión de la curva calculada.

End Sub

***Inicialización de la forma. Esta subrutina se corre al cargar la pantalla de regresión.**

Private Sub Form_Open(Cancel As Integer)

 *El cuerpo de esta subrutina es idéntico al de la subrutina recién descrita para borrar los campos de la pantalla.

End Sub

Private Sub Command21_Click()

 *Borra los datos de captura para X.

 Dim stDocName As String

 stDocName = "qry01_2RegDelX"

 *Borra los valores de X por medio de un query

 DoCmd.OpenQuery stDocName, acNormal, acEdit

 *de actualización (Update query).

End Sub

Private Sub Command22_Click()

 *Borra los datos de captura para Y.

 Dim stDocName As String

 stDocName = "qry01_3RegDelY"

 *Borra los valores de Y por medio de un query

 DoCmd.OpenQuery stDocName, acNormal, acEdit

 *de actualización (Update query).

End Sub

Private Sub Command20_Click()

 *Borra los datos de captura para X y Y.

 Dim stDocName As String

 stDocName = "qry01_1RegDelXY"

 *Borra los puntos X y Y por medio de un

 DoCmd.OpenQuery stDocName, acNormal, acEdit

 *Query que borra renglones.

End Sub

***Trae archivo de datos. Previa inicialización de la subforma de captura.**

Private Sub Command41_Click()

 *Variables auxiliares.

 Dim a, stDocName As String

 stDocName = "qry01_8AnRegDel"

 *Inicializa la subforma de captura de datos de X y Y

 DoCmd.OpenQuery stDocName, acNormal, acEdit

 *por medio de un query.

 a = TraeTbl(Me.Arch, 2)

 *Se hace asignación dummy para correr la función que

End Sub

 *copia los datos desde el archivo.

Private Sub Command40_Click()

 *Guarda datos capturados de X y Y en un archivo.

 Dim a As String

 a = GuardaTbl(2)

 *La asignación es dummy, se usa para correr la función que guarda los

End Sub

 *datos. El parámetro 2 indica que pertenece al área de regresión.

***Transfiere función traída de archivo de funciones para transformar a X**

Private Sub Command64_Click()

 Me.FX = Me.Funcion.Column(1)

 *Column(0) es el nombre de la función; Column(1) es la

 Me.Refresh

 *expresión.

End Sub

***Transfiere función de archivo de funciones para transformar a Y. La expresión queda en términos de Y.**

Private Sub Command65_Click()

 Dim i, lngL As Integer, Func As String

 Func = Me.Funcion.Column(1)

 *Asigna la expresión de la función a la variable auxiliar Func.

 lngL = Len(Func)

 *Verifica la longitud de la expresión para posteriormente

 Me.FY = ""

 *recorrer carácter por carácter para cambiar X's por Y's.

 For i = 1 To lngL

 *Barrido de los caracteres de la expresión, uno por uno.

 If Mid(Func, i, 1) = "X" Then

 *FY es la transformada de Y, es llenada carácter por carácter.

```

    Me.FY = Me.FY + "Y"          'Encuentra una X y hace el reemplazo por Y.
Else
    Me.FY = Me.FY + Mid(Func, i, 1) 'Si no hay X, el carácter se "deposita" tal cuál en transf Y.
End If
Next I
Me.Refresh
End Sub

```

***Genera las transformaciones de X y Y, de no especificarse transformaciones, $\text{Transf}(X) = X$ y $\text{transf}(Y) = Y$.**

```

Private Sub Command34_Click()
    Dim stDocName As String
    stDocName = "qry01_4TransfXYUpd" 'Se utiliza un query para realizar las
    DoCmd.OpenQuery stDocName, acNormal, acEdit 'transformaciones de X y Y.
    Me.Refresh
End Sub

```

```

Private Sub Command26_Click() 'Borra expresión para transformar X
    Me.FX = Null
End Sub

```

```

Private Sub Command27_Click() 'Borra expresión para transformar Y
    Me.FY = Null
End Sub

```

***Genera la recta de ajuste ($y = a + bX$)**

```

Private Sub Command6_Click()
    Dim stDocName As String, n As Long
    n = DCount("[X]", "tblAnReg01") 'Verifica que haya datos a analizar.
    If n <= 1 Then
        MsgBox "**Error. Datos insuficientes o no hay datos a analizar."
    Exit Sub
End If

```

```

    stDocName = "qry01_4TransfXYUpd" 'Crea las transformaciones para X y Y, por medio de
    DoCmd.OpenQuery stDocName, acNormal, acEdit 'un query. Si no hay transformaciones se usan (X,Y).

```

```

    stDocName = "qry02AnRegDel" 'Borra la subforma de presentación de información de
    DoCmd.OpenQuery stDocName, acNormal, acEdit 'la regresión con un query.

```

```

    Me.Modelo = Regresion(1, Me.Desplegar) 'Función que calcula la regresión. 1 es el grado;
End Sub 'Me.desplegar indica si se despliegan en la pantalla los
'calculos detallados de la regresión.

```

***Genera la curva de ajuste ($y = a + bX + cX^2$)**

```

Private Sub Command47_Click()
    Dim stDocName As String, n As Long
    n = DCount("[X]", "tblAnReg01") 'Cuenta el número de datos a analizar.
    If n <= 1 Then 'Verifica que haya al menos un punto a graficar
        MsgBox "**Error. Datos insuficientes o no hay datos a analizar."
    Exit Sub
End If

```

```

stDocName = "qry01_4TransfXYUpd" 'Ejecuta las transformaciones especificadas, de otro
DoCmd.OpenQuery stDocName, acNormal, acEdit 'modo se usan los puntos (X,Y).

stDocName = "qry02AnRegDel" 'Borra la subforma que despliega la información de
DoCmd.OpenQuery stDocName, acNormal, acEdit 'la regresión.

Me.Modelo = Regresion(2, Me.Desplegar) 'Genera la información de la regresión; el parámetro 2
'indica el grado; Me.desplegar indica si se despliegan
'los cálculos detallados de la regresión.
Me.Refresh 'Me.Modelo es una variable invisible
End Sub '(no se despliega en la pantalla)

Private Sub Command48_Click() 'Genera la gráfica de dispersión de (X,Y).
Dim stDocName As String
Dim stLinkCriteria As String, n As Long
n = DCount("[X]", "tblAnReg01") 'Verifica que haya puntos a graficar.
If n <= 1 Then
MsgBox "**Error. Datos insuficientes o no hay datos a analizar."
Exit Sub
End If
stDocName = "frmAnRegCh0" 'Nombre de la forma que contiene la gráfica.
DoCmd.OpenForm stDocName, , stLinkCriteria 'Abre la forma de la gráfica.
End Sub

'Genera gráfica de las transformaciones de X y Y, si no se especifican, entonces transf(X)=X, transf(Y)=Y.
Private Sub Command43_Click()
Dim stDocName As String 'Se declaran variables auxiliares.
Dim stLinkCriteria As String, n As Long
n = DCount("[X]", "tblAnReg01") 'Verifica que haya datos a graficar.
If n <= 1 Then
MsgBox "**Error. Datos insuficientes o no hay datos a analizar."
Exit Sub
End If
stDocName = "frmAnRegCh1" 'Carga la forma que despliega la gráfica. La gráfica
DoCmd.OpenForm stDocName, , stLinkCriteria 'está ligada a un query con las transformaciones.
End Sub

'Genera la gráfica de las transformaciones, junto con la recta o la curva ajustada. Si no se especifican
transformaciones, entonces transf(X)=X, transf(Y)=Y.
Private Sub ComLinea_Click()
Dim stDocName As String 'Variables auxiliares
Dim stLinkCriteria As String, n As Long
n = DCount("[X]", "tblAnReg01") 'Verifica que haya datos a graficar.
If n <= 1 Then
MsgBox "**Error. Datos insuficientes o no hay datos a analizar."
Exit Sub
End If
stDocName = "frmAnRegCh2" 'Carga la forma que despliega la gráfica. La gráfica
DoCmd.OpenForm stDocName, , stLinkCriteria 'está ligada a un query con los datos a graficar.
End Sub

```

*Guarda la función de la recta o la curva de regresión calculada.

```
Private Sub Command59_Click()
```

```
Dim stDocName As String
```

```
Dim NombreFunc As Variant
```

```
If Not IsNull(Me.FuncionAGuardar) Then
```

*Verifica que haya un nombre para la función

```
  If Me.FuncionAGuardar = "" Then
```

```
    MsgBox ("Asigne un nombre a la función. Máximo 25 caracteres.")
```

```
    Exit Sub
```

*Valor nulo o texto nulo no son válidos como

```
  End If
```

*nombre.

```
Else
```

```
  MsgBox ("Asigne un nombre a la función. Máximo 25 caracteres.")
```

```
Exit Sub
```

```
End If
```

*Verifica que el nombre no exista en archivo de funciones.

```
NombreFunc = DLookup("[Nombre]", "tblFunciones", "[Nombre] = " & Me.FuncionAGuardar & """)
```

```
If Not IsNull(NombreFunc) Then
```

```
  MsgBox ("Nombre de Funcion existente. Antes de guardar, borre el nombre: " & NombreFunc)
```

```
Exit Sub
```

```
Else
```

```
  stDocName = "qryGuardaAjuste"
```

*Query que guarda la función (esta función

```
  DoCmd.OpenQuery stDocName, acNormal, acEdit
```

*que se guarda, se puede acceder

```
End If
```

*posteriormente en cualquier módulo que

```
End Sub
```

*accese funciones).

Ajuste NombreRecta → Guardar

tblDummy

Dummy

La variable "dummy" ayuda a crear el registro.
Es necesaria porque los datos no provienen de una tabla.

Field:	Nombre: [Forms]![frmAnReg]![FuncionAGuardar]	Funcion: [Forms]![frmAnReg]![Modelo]
Table:		
Sort:		
Append To:	Nombre	Funcion
Criteria:		

4.5.3 Números Aleatorios.

Form Header

Encabezado de la forma										"NumAleat"		Instrucciones/Ayuda		
Generador de Números Aleatorios														
Cantidad de Números a Generar (Default 10):						Unbound		Crear Tabla		Preparar Imprimir		Comandos		
Límite Inferior (Default 0):						Unbound		▲		□				
Límite Superior (Default 100):						Unbound								
Tabla de Números Aleatorios														
Renglon	Col 1	Col 2	Col 3	Col 4	Col 5	Col 6	Col 7	Col 8	Col 9	Col 10				
Renglon	Col01	Col02	Col03	Col04	Col05	Col06	Col07	Col08	Col09	Col10				
Form Footer														
Generados usando el reloj de la computadora como semilla de números aleatorios.														

Pie de Página variables que generan los rengiones
 La forma usa una vista de forma continua, por eso se repiten los rengiones cuando se despliega.

Option Compare Database

*Pantallas de ayuda. Instrucción que se cambia en la subrutina de carga de pantalla.
 stLinkCriteria = "[Forma]=" & "" & Me[NumAleat] & "" *Carga de la forma de ayuda.
 NumAleat es la variable auxiliar definida para cargar la forma de ayuda.

Variables definidas:
 TotNums Cantidad de números a crear
 LInf Límite inferior para el valor de los números.
 Lsup Límite superior para el valor de los números.

```

*Subrutina que se corre automaticamente al abrir lapantalla. Lee los parámetros con que fue creada una tabla anterior, para desplegar esos datos.
Private Sub Form_Load()
    Dim rsNAleat As DAO.Recordset
    Dim dbs As DAO.Database
    Dim n As Long
    n = DCount("[TotalNos]", "tblDatNAleat")

If n < 1 Then
    Exit Sub
End If

'En el siguiente comando, se puede especificar una base de datos remota (fuera de EMCalc) o la actual (EmCalc).
Set dbs = CurrentDb
Set rsNAleat = dbs.OpenRecordset("tblDatNAleat")
With rsNAleat
    .MoveFirst
    Me.TotNums = !TotalNos
    Me.LInf = !LimInf
    Me.LSup = !LimSup
End With
rsNAleat.Close
End Sub

'El evento sucede al momento de cargar la forma.
'Se declara un archivo de datos.
'Se declara una base de datos.
'n es la cantidad de números aleatorios.
'tblDatNAleat es la tabla que contiene los números.

'Este evento es imposible, por construcción. Se utiliza
'para pruebas mientras se diseña.

'Se indica que se utilizará la base de datos "Actual".
'tblDatNAleat se convierte en el archivo a emplear.
'Se usan los campos del archivo rsNAleat
'Se accesa el primer registro,
'Se lee el Total de Números y se asigna a TotNums;
'los campos con el prefijo Me. Pertenecen a la pantalla
'Con el prefijo "!" pertenecen al archivo.
'Se cierra cláusula "With"
'Se cierra el archivo de números aleatorios.
    
```

***Crea la tabla de números aleatorios. La cantidad de números que se crea es múltiplo de 10 , con un mínimo de 10 números y máximo no establecido (limitado solo por capacidad de Access, 100,000 es un número aceptable). Se especifica un rango para el valor de los números.**

Private Sub Command30_Click()

Dim stDocName As String, Tot, Menor, Mayor As Long

*Variables auxiliares.

If Not IsNumeric(Me.TotNums) Then

Me.TotNums = 0

*Verifica que el total de números solicitado
*sea válido, de ser inválido crea 10.

Tot = 10

End If

If Not IsNumeric(Me.LInf) Then

Me.LInf = 0

*Límite inferior del rango en el cuál deben
*crearse los números se hace cero en caso de
*ser inválido.

Menor = 0

End If

If Not IsNumeric(Me.LSup) Then

Me.LSup = 100

*Límite superior del rango en el cuál deben
*crearse los números se hace 100 en caso de
*ser inválido.

Mayor = 100

End If

If Me.TotNums < 10 Then

*Se crea un mínimo de 10 números.

Tot = 10

Else: Tot = Me.TotNums

End If

Menor = Me.LInf

*Variables auxiliares.

Mayor = Me.LSup

If Mayor <= Menor Then

*Si el rango está reversado, se crean los
*números en el rango(0,100).

Menor = 0

Me.LInf = 0

Mayor = 100

Me.LSup = 100

End If

DoCmd.Close acForm, "tbl6NumAleatorios"

*Se cierra la forma de números aleatorios para
*poder borrar la tabla a la cuál está ligada.

stDocName = "qryDelTblNoAleat"

*Se ejecuta el query que borra la tabla.

DoCmd.OpenQuery stDocName, acNormal, acEdit

a = NAleatorios(Tot, Menor, Mayor)

*Se hace una asignación dummy para crear la

*tabla con esta función.

DoCmd.OpenForm "tbl6NumAleatorios"

*Se abre nuevamente la forma.

End Sub

Private Sub Command37_Click()

*Despliega el reporte de números aleatorios.

Dim stDocName As String

stDocName = "frmtblNAleat"

*Nombre del reporte.

DoCmd.OpenReport stDocName, acPreview

*Despliega el reporte.

End Sub

4.5.4 Creación de las Pantallas de la Función Normal, t de student, Ji cuadrada, F 5% de Significancia y F1% de Significancia.

Las formas para desplegar tablas estadísticas utilizan un formato similar. Debido a la magnitud de las cantidades y la cantidad de números, se tiene especial cuidado en su formato para que se desplieguen adecuadamente.

Se dan dos ejemplos para mostrar su diseño: la función normal y la función F.

Form Header

Función Normal										
Tabla de densidad acumulada de la función Normal										
X:	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9

Form Footer

Sección de detalle (Rengiones que se repiten)

Fuente: E. Parzen, Modern Probability Theory and its applications. Wiley, 1960. Table I, p.441

Tabla Fuente	Form				
	Format	Data	Event	Other	All
Forma continua permite desplegar múltiples rengiones.	Record Source	tblF (Distribution)			
	Filter				
	Order By				
	Allow Filters	Yes			
	Caption	Normal			
	Default View	Continuous Forms			

Otro ejemplo del diseño de las formas para mostrar tablas. Función F.

Función F														
Puntos Percentuales de la Distribución F, Nivel de Significancia: 1%														
X:	1	2	3	4	5	6	7	8	9	10	12	15	20	24

Fuente: E. S. Pearson from "Tables of Percentage Points of the Inverted Beta" (Merrington and Catherine M. Thompson)

Forma	Form				
	Format	Data	Event	Other	All
Acceso permite desplegar formas más grandes que la pantalla, en este caso, se fragmenta para mostrar su diseño.	Record Source	tblF (Distribution)			
	Filter				
	Order By				
	Allow Filters	Yes			
	Caption	FDistribution (Pt)			
	Default View	Continuous Forms / Forma continua			

Diseño de los reportes. Para desarrollar los reportes se utiliza un diseño similar para todos, simplemente se cuida que los números se desplieguen adecuadamente.

4.5.5 Despliegado de los reportes.

Los reportes son todos preformateados. Para su diseño se utilizó la vista de diseño. No se modificaron sus subrutinas.

Ejemplo de como se carga un reporte:

```
Private Sub Command13_Click()
    Dim stDocName As String
    stDocName = "rptNormal"
    DoCmd.OpenReport stDocName, acPreview
End Sub
```

Instrucción que se cambia para cargar os demás reportes:

```
stDocName = "rptStudent"      para función t de Student
stDocName = "rptChi2"        Para Ji Cuadrada
stDocName = "rptF1Distribucion" Para Distribución F, 1% de significancia
stDocName = "rptF5Distribucion" Para Distribución F, 5% de significancia
stDocName = "frmtblNormalcat" Para tabla de números aleatorios.
```

Ejemplos del diseño de los reportes.

Report Header **Encabezado del reporte**

Función Normal Tabla ligada al reporte y campos disponibles para reportar

Tabla de densidad acumulada de la función Normal.

Page Header **Encabezado de la página**

X	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
---	------	------	------	------	------	------	------	------	------	------

Detail **Línea de detalle (renglones que se repiten)**

X	Col01	Col02	Col03	Col04	Col05	Col06	Col07	Col08	Col09	Col10
---	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

Page Footer **Plé de página**

Fuente: E. Parzen, Modern Probability theory and its applications, Wiley, 1960.

Impreso por EmCalc.Emulación de cal

Now() **Función preconstruida de Access.**

Report Properties Dialog:

- Format: Data
- Event: Other
- AI: All
- Record Source: tblNormal
- Filter:
- Filter On: No

tblNormal Table:

- X
- Col01
- Col02
- Col03
- Col04
- Col05
- Col06

El reporte para la tabla de la función t de Student.

Función t de Student

Puntos porcentuales para la función t-Student

Page Header

GLP	0.750	0.900	0.950	0.975	0.990	0.995	0.999
-----	-------	-------	-------	-------	-------	-------	-------

Detail

GL	Col01	Col02	Col03	Col04	Col05	Col06
----	-------	-------	-------	-------	-------	-------

Page Footer

[?] el símbolo "as" aparece en lugar d [?] por limitación del tipo de letra cmc]

Report Properties Dialog:

- Format: Data
- Event: Other
- AI: All
- Record Source: tblStudent
- Filter:
- Filter On: No
- Order By:
- Order By On: No

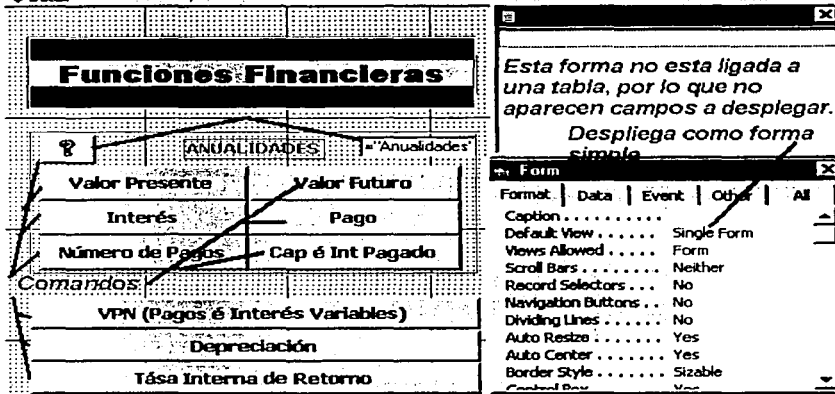
tblStudent Table:

- GL
- Col01
- Col02
- Col03
- Col04
- Col05
- Col06
- Col07

4.6 Funciones Financieras.

Esta pantalla sólo carga otras pantallas, incluida una de ayuda.

◀ Detall



'Carga de pantallas. Instrucción que se cambia en la subrutina de carga de pantalla.

'Forma de ayuda

stDocName = "frmInstrucciones"

stLinkCriteria = "[Forma]= " & "" & McI[Anualidades] & "" 'Criterio utilizado para desplegar la página adecuada.

'Forma que despliega la ayuda.

Formas que calculan las anualidades:

stDocName = "frmAnualVP"

'Anualidades: Valor Presente.

stDocName = "frmAnualFV"

'Anualidades: Valor Futuro.

stDocName = "frmAnualInt"

'Anualidades: Cálculo del Interés.

stDocName = "frmAnualPago"

'Anualidades: Cálculo del pago.

stDocName = "frmAnualNumPagos"

'Anualidades: Cálculo del Número de Pagos.

stDocName = "frmAnualIntYCap"

'Anualidades: Capital é Interés Pagados.

stDocName = "frmVPvar"

'Valor Presente Neto (Pagos é Interés desiguales)

stDocName = "frmDepreciacion"

'Depreciación.

stDocName = "frmTir0"

'Tása Interna de Retorno.

"Anualidades" es la variable auxiliar definida para cargar la forma de ayuda.

Nota:

Las pantallas de Anualidades: Valor Presente, Valor Futuro, Cálculo del Interés, Cálculo del Pago, Cálculo del Número de Pagos y Cálculo del Capital é Interés Pagado utilizan un diseño similar. De modo que al revisar el código de una de ellas, se tendrá la idea de las demás.

4.6.1 Anualidades: Valor Presente.

"AnVP"		Anualidades Valor Presente			
Me.pue. de "lecular de" (punto 3) de		(col 1) VP Sensibilidad 1	(col 2) VP Sensibilidad 2	(col 3) VP Sensibilidad 3	(col 4) VP Sensibilidad 4
		borrar	borrar	borrar	borrar
Valor Presente		Unbound	Unbound	Unbound	Unbound
Pago		Unbound	Unbound	Unbound	Unbound
No. de Periodos		Unbound	Unbound	Unbound	Unbound
Interés 1		Unbound	Unbound	Unbound	Unbound
Valor Futuro		Unbound	Unbound	Unbound	Unbound
Tipo		Unbound	Unbound	Unbound	Unbound
Borrar Todo		Copiar Datos de Columna: Unbo Columna: Unbo			
El interés y los pagos deben tener la misma periodicidad					
Para hacer cálculos, recorra la expresión siguiente a su					
Ejemplos: 07.12.16 11.12.17 Enter Tab o click en otro punto					
Expresión:		Unbound			
Resultado:		Unbound			
Presar & Interés:		1	2	3	4

comandos

Option Compare Database

*Pantallas de ayuda. Instrucción que se cambia en la subrutina de carga de pantalla.

stLinkCriteria = "[Forma] & "" & Me![AnVP] & "" 'Carga la pantalla de ayuda.

*AnVP es la variable auxiliar definida para cargar la forma de ayuda.

*Crea el Valor Presente de la primer sensibilidad. Verifica que los parámetros tengan un valor numérico.

```
Private Sub Command108_Click()
```

*No se provee filtro adicional.

```
    If Not IsNumeric(Me.pmt1) Then
```

```
        MsgBox "'Pago' tiene un valor no numérico."
```

```
        Exit Sub
```

```
    End If
```

```
    If Not IsNumeric(Me.Int1) Then
```

```
        MsgBox "'Interés' tiene un valor no numérico."
```

```
        Exit Sub
```

```
    End If
```

```
    If Not IsNumeric(Me.Nper1) Then
```

```
        MsgBox "'Número de periodos' tiene un valor no numérico."
```

```
        Exit Sub
```

```
    End If
```

```
    If Not IsNumeric(Me.FV1) Then
```

```
        MsgBox "'Saldo Final' tiene un valor no numérico."
```

```
        Exit Sub
```

```
    End If
```

```
    Me.PV1 = PV(Int1, Me.Nper1, Me.pmt1, Me.FV1, Me.Tipo1)
```

*PV es una función preconstruida de Access.

```
    Me.Refresh
```

```
End Sub
```

*El Valor Presente de las sensibilidades 2,3 y 4 se genera en forma idéntica. Se especifica aquí sólo la línea que genera el cálculo:

```
Me.PV2 = PV(Int2, Me.Nper2, Me.pmt2, Me.FV2, Me.Tipo2)
```

```
Me.PV3 = PV(Int3, Me.Nper3, Me.pmt3, Me.FV3, Me.Tipo3)
```

```
Me.PV4 = PV(Int4, Me.Nper4, Me.pmt4, Me.FV4, Me.Tipo4)
```

***Transfiere los datos de la columna i a la j; donde $i < j$; $i = 1..4$; $j = 1..4$**

Private Sub Command120_Click()

Dim i, j As Integer

If Col1 = Col2 Then Exit Sub

i = Col1

j = Col2

Me("PV" + Format(j)) = Me("PV" + Format(i))

Me("Int" + Format(j)) = Me("Int" + Format(i))

Me("Nper" + Format(j)) = Me("Nper" + Format(i))

Me("pmt" + Format(j)) = Me("pmt" + Format(i))

Me("FV" + Format(j)) = Me("FV" + Format(i))

Me("Tipo" + Format(j)) = Me("Tipo" + Format(i))

End Sub

'i y j se emplean como variables auxiliares.

'No es válido copiar una columna de datos a sí misma.

'Se usan i y j por ser más comunes como índices.

'El operador + se usa para concatenar caracteres; por

'ejemplo, si $i=1$ y $j=3$ (copiar col 1 a col 3), hace que

'**Me("PV" + Format(j)) = Me("PV" + Format(i))**

'se convierta en: **Me.PV3 = Me.PV1**

'Caso idéntico para las demás asignaciones.

Copiar Datos de Columna 1 a Columna 2

***Después de escribir la expresión algebraica y moverse a otro campo o comando, se evalúa la expresión apareciendo el resultado en el campo "Resultado".**

Private Sub IntExp_AfterUpdate()

If IsNull(Me.IntExp) Then

Exit Sub

End If

Me.IntResult = Eval(Me.IntExp)

End Sub

'El evento ocurre después de actualizar la expresión.

'Se verifica que exista una expresión.

'Eval es una función predefinida de Access, se usa para

'evaluar expresiones algebraicas.

Expresión: .065/12
Resultado: 0.0054

Private Sub Command160_Click()

Me.Int1 = Me.IntResult

End Sub

Pasar a Interés 1 2 3 4

Se usan subrutinas similares para las sensibilidades 2, 3 y 4. La línea a reemplazar es: **Me.Int1 = Me.IntResult**, por

Me.Int2 = Me.IntResult

Me.Int3 = Me.IntResult

Me.Int4 = Me.IntResult

'Transfiere el resultado de la expresión al Interés de la

'sensibilidad 1.

Private Sub Command121_Click()

Me.PV1 = 0

Me.Int1 = 0

Me.Nper1 = 0

Me.pmt1 = 0

Me.FV1 = 0

Me.Tipo1 = 0

End Sub

'Borra los datos de la sensibilidad 1.

'Valor Presente

'Interés

'Número de Pagos.

'Pago

'Valor Futuro

'Anualidad Vencida o anticipada

(col 1) VP Sensibilidad 1	(col 2) VP Sensibilidad 2	(col 3) VP Sensibilidad 3	(col 4) VP Sensibilidad 4
borrar	borrar	borrar	borrar

Se usan subrutinas idénticas para borrar los datos de las sensibilidades 2, 3 y 4, usando las siguientes variables:

Sensibilidad 2: Me.PV2 = 0 Me.Int2 = 0 Me.Nper2 = 0 Me.pmt2 = 0 Me.FV2 = 0 Me.Tipo2 = 0

Sensibilidad 3: Me.PV3 = 0 Me.Int3 = 0 Me.Nper3 = 0 Me.pmt3 = 0 Me.FV3 = 0 Me.Tipo3 = 0

Sensibilidad 4: Me.PV4 = 0 Me.Int4 = 0 Me.Nper4 = 0 Me.pmt4 = 0 Me.FV4 = 0 Me.Tipo4 = 0

*Se borran los datos de todas las sensibilidades.

Borrar Todo

```
Private Sub Command128_Click()
```

```
Dim i As Integer
```

```
For i = 1 To 4
```

```
Me("PV" + Format(i)) = 0
```

```
Me("Int" + Format(i)) = 0
```

```
Me("Nper" + Format(i)) = 0
```

```
Me("pmt" + Format(i)) = 0
```

```
Me("FV" + Format(i)) = 0
```

```
Me("Tipo" + Format(i)) = 0
```

```
Next i
```

```
Me.IntExp = Null
```

```
Me.IntResult = Null
```

```
Me.Refresh
```

```
End Sub
```

*Se usa i como índice.

*Se recorren las 4 sensibilidades, borrando los datos.

* Si i = 2, la instrucción: Me("PV" + Format(i)) = 0

*equivale a: Me.PV2 = 0

*Caso idéntico para las variables: Int, Nper, pmt, FV y Tipo.

*Se inicializan las variables adicionales. IntExp es el campo en

*que se escribe una expresión algebraica; IntResult es el

*resultado de esa expresión.

4.6.2 Anualidades: Valor Futuro.

Formulario de Anualidades: Valor Futuro. Campos: [AnVF]

	(col 1) VF Sensibilidad 1	(col 2) VF Sensibilidad 2	(col 3) VF Sensibilidad 3	(col 4) VF Sensibilidad 4
Clic para de teclado dato	borrar	borrar	borrar	borrar
Valor Presente	Unbound	Unbound	Unbound	Unbound
[Pago]	Unbound	Unbound	Unbound	Unbound
No. de Periodos	Unbound	Unbound	Unbound	Unbound
Interés	Unbound	Unbound	Unbound	Unbound
Valor Futuro	Unbound	Unbound	Unbound	Unbound
Tipo	Unbound	Unbound	Unbound	Unbound
Borrar Todo	Cajas-combo			
	Copiar Datos de Columna: Unbo -1 & Columna: Unbo -1			
<p>El interés y los pagos deben tener la misma periodicidad</p> <p>Para hacer calculo, teclee e una expresión algebraica abajo</p> <p>Ejemplo: 07.12 107 (1/12)1 Error Tab o clicl en otro campo</p>				
	Expresión:	Unbound		
	Resultado:	Unbound		
	Pasar a Interés	1	2	3
		4		

Este diseño es idéntico al de VP excepto por los comandos y las subrutinas.

Option Compare Database

*Pantallas de ayuda. Instrucción que se cambia en la subrutina de carga de pantalla.
 stLinkCriteria = "[Forma]=" & "" & Me![AnVF] & "" *Carga la pantalla de ayuda.
 AnVF es la variable auxiliar definida para cargar la forma de ayuda.

*Crea el Valor Futuro de la primer sensibilidad. Verifica que los parámetros tengan un valor numérico.

```
Private Sub Command12_Click()
    'No se prevé filtro adicional.
    If Not IsNumeric(Me.pmt1) Then
        'Pago
        MsgBox "'Pago' tiene un valor no numérico."
        Exit Sub
    End If
    If Not IsNumeric(Me.Int1) Then
        'Interés
        MsgBox "'Interés' tiene un valor no numérico."
        Exit Sub
    End If
    If Not IsNumeric(Me.Nper1) Then
        'Número de periodos de pago.
        MsgBox "'Número de periodos' tiene un valor no numérico."
        Exit Sub
    End If
    If Not IsNumeric(Me.PV1) Then
        'Valor Presente o Saldo Inicial.
        MsgBox "'Saldo Inicial' tiene un valor no numérico."
        Exit Sub
    End If
    Me.FV1 = FV(Me.Int1, Me.Nper1, Me.pmt1, Me.PV1, Me.Tipo1) 'FV es una función preconstruída de Access.
End Sub
```

***El Valor Presente de las sensibilidades 2,3 y 4 se genera en forma idéntica. Se especifica aquí sólo la línea que genera el cálculo:**

Me.FV2 = FV(Me.Int2, Me.Nper2, Me.pmt2, Me.PV2, Me.Tipo2)

Me.FV3 = FV(Me.Int3, Me.Nper3, Me.pmt3, Me.PV3, Me.Tipo3)

Me.FV4 = FV(Me.Int4, Me.Nper4, Me.pmt4, Me.PV4, Me.Tipo4)

***Transfiere los datos de la columna i a la j; donde $i < j$; $i = 1..4$; $j = 1..4$**

Private Sub Command120_Click()

Dim i, j As Integer

*i y j se emplean como variables auxiliares.

If Col1 = Col2 Then Exit Sub

*No es válido copiar una columna de datos a sí misma.

i = Col1

*Se usan i y j por ser más comunes como índices.

j = Col2

Me("PV" + Format(j)) = Me("PV" + Format(i))

*El operador + se usa para concatenar caracteres; por

Me("Int" + Format(j)) = Me("Int" + Format(i))

*ejemplo, si i=1 y j=3 (copiar col 1 a col 3), hace que

Me("Nper" + Format(j)) = Me("Nper" + Format(i))

*Me("FV" + Format(j)) = Me("FV" + Format(i))

Me("pmt" + Format(j)) = Me("pmt" + Format(i))

*se convierta en: Me.FV3 = Me.FV1

Me("FV" + Format(j)) = Me("FV" + Format(i))

*Caso idéntico para las demás asignaciones.

Me("Tipo" + Format(j)) = Me("Tipo" + Format(i))

End Sub

***Después de escribir la expresión algebraica y moverse a otro campo o comando, se evalúa la expresión apareciendo el resultado en el campo "Resultado".**

Dim intx As Double

Private Sub IntExp_AfterUpdate()

*El evento ocurre después de actualizar la expresión.

If IsNull(Me.IntExp) Then

*Se verifica que exista una expresión.

Exit Sub

End If

Me.IntResult = Eval(Me.IntExp)

*Eval es una función preconstruida de Access, se usa para

End Sub

*evaluar expresiones algebraicas.

Private Sub Command160_Click()

***Transfiere el resultado de la expresión al Interés de la**

Me.Int1 = Me.IntResult

sensibilidad i.

End Sub

Se usan subrutinas similares para las sensibilidades 2, 3 y 4. La línea a reemplazar es: Me.Int1 = Me.IntResult, por

Me.Int2 = Me.IntResult

Me.Int3 = Me.IntResult

Me.Int4 = Me.IntResult

Private Sub Command125_Click()

***Borra los datos de la sensibilidad 1.**

Me.PV1 = 0

*Valor Presente

Me.Int1 = 0

*Interes

Me.Nper1 = 0

*Número de Pagos.

Me.pmt1 = 0

*Pago

Me.FV1 = 0

*Valor Futuro

Me.Tipo1 = 0

*Anualidad Vencida o anticipada

End Sub

Se usan subrutinas idénticas para borrar los datos de las sensibilidades 2, 3 y 4, usando las siguientes variables:

Sensibilidad 2: Me.PV2 = 0 Me.Int2 = 0 Me.Nper2 = 0 Me.pmt2 = 0 Me.FV2 = 0 Me.Tipo2 = 0

Sensibilidad 3: Me.PV3 = 0 Me.Int3 = 0 Me.Nper3 = 0 Me.pmt3 = 0 Me.FV3 = 0 Me.Tipo3 = 0

Sensibilidad 4: Me.PV4 = 0 Me.Int4 = 0 Me.Nper4 = 0 Me.pmt4 = 0 Me.FV4 = 0 Me.Tipo4 = 0

```
Private Sub Command128_Click()
```

```
Dim i As Integer
```

```
For i = 1 To 4
```

```
Me("PV" + Format(i)) = 0
```

```
Me("Int" + Format(i)) = 0
```

```
Me("Nper" + Format(i)) = 0
```

```
Me("pmt" + Format(i)) = 0
```

```
Me("FV" + Format(i)) = 0
```

```
Me("Tipo" + Format(i)) = 0
```

```
Next i
```

```
Me.IntExp = Null
```

```
Me.IntResult = Null
```

```
Me.Refresh
```

```
End Sub
```

*Se borran los datos de todas las sensibilidades.

*Se usa i como índice.

*Se recorren las 4 sensibilidades, borrando los datos.

* Si i = 2, la instrucción: Me("PV" + Format(i)) = 0

*equivale a: Me.PV2 = 0

*Caso idéntico para las variables: Int, Nper, pmt, FV y Tipo.

*Se inicializan las variables adicionales. IntExp es el campo en

*que se escribe una expresión algebraica; IntResult es el

*resultado de esa expresión.

4.6.3 Anualidades: Cálculo del Interés.

Form

Format	Data	Event	Other	All
Caption
Default View	Single Form
Views Allowed	Form
Record Selectors	Neither
Scroll Bars	Neither
Navigation Buttons	No
Unbinding Lines	No
Auto Resize	Yes
Auto Center	Yes
Border Style	Stable
Control Box	Yes
Min Max Buttons	Both Enabled
Close Button	Yes
Whats This Button	No
Width	6.3479"
Picture	(none)
Picture Type	Embedded
Picture Size Mode	Clip
Picture Alignment	Center
Picture Tiling	No
Grid X	24
Grid Y	24
Layout for Print	No
Palette Source	(Default)

col 1. Interés Sensibilidad 1
 col 2. Interés Sensibilidad 2
 col 3. Interés Sensibilidad 3
 col 4. Interés Sensibilidad 4

Valor Presente (P) Unbound
 Pago Unbound
 No. de Periodos (N) Unbound
 Valor Futuro (F) Unbound
 Tipo Unbound

Borrar Todo
 Copiar Datos de Columna Unbo
 a Columna Unbo

Diseño idéntico a las formas anteriores. Aquí muestra algunas de sus propiedades.

El interés y los pagos tienen la misma periodicidad
 (1) (2) (3) (4) le permiten actualizar para editar.

Expresión: Unbound
 Resultado: Unbound

Option Compare Database

*Pantallas de ayuda. Instrucción que se cambia en la subrutina de carga de pantalla.

```
stLinkCriteria = "[Forma]=" & "" & Me![AnInt] & ""
stLinkCriteria = "[Forma]=" & "" & Me![AnIntEjmp] & ""
```

AnInt, AnIntEjmp, son las variables auxiliares definidas para cargar las formas de ayuda.

*Calcula el Interés de la primer sensibilidad. Verifica que los parámetros tengan un valor numérico.

```
Private Sub Command109_Click()
```

```
If Not IsNumeric(Me.pmt1) Then 'Pago
```

```
MsgBox ""Pago' tiene un valor no numérico."
```

```
Exit Sub
```

```
End If
```

```
If Not IsNumeric(Me.PV1) Then 'Valor Presente o Saldo Inicial.
```

```
MsgBox ""Saldo Inicial' tiene un valor no numérico."
```

```
Exit Sub
```

```
End If
```

```
If Not IsNumeric(Me.Nper1) Then 'Número de pagos
```

```
MsgBox ""Número de periodos' tiene un valor no numérico."
```

```
Exit Sub
```

```
End If
```

```
If Not IsNumeric(Me.FV1) Then 'Valor Futuro o Saldo Final
```

```
MsgBox ""Saldo Final' tiene un valor no numérico."
```

```
Exit Sub
```

```
End If
```

```
Me.Int1 = Rate(Me.Nper1, Me.pmt1, Me.PV1, Me.FV1, Me.Tipo1) 'Rate es una función preconstruida de access.
```

```
If Me.Int1 < 0 Then 'La tasa calculada debe ser mayor que cero.
```

```
MsgBox ""Error. Datos inválidos, se genera tasa de interés menor que cero."
```

```
End If
```

```
End Sub
```

'El Interés de las sensibilidades 2,3 y 4 se genera en forma idéntica. Se especifica aquí sólo la línea que genera el cálculo:

Me.Int2 = Rate(Me.Nper2, Me.pmt2, Me.PV2, Me.FV2, Me.Tipo2)

Me.Int3 = Rate(Me.Nper3, Me.pmt3, Me.PV3, Me.FV3, Me.Tipo3)

Me.Int4 = Rate(Me.Nper4, Me.pmt4, Me.PV4, Me.FV4, Me.Tipo4)

'Transfiere los datos de la columna i a la j; donde i <> j; i = 1..4; j = 1..4

Private Sub Command120_Click()

Dim i, j As Integer

If Col1 = Col2 Then Exit Sub

i = Col1

j = Col2

Me("PV" + Format(j)) = Me("PV" + Format(i))

Me("Int" + Format(j)) = Me("Int" + Format(i))

Me("Nper" + Format(j)) = Me("Nper" + Format(i))

Me("pmt" + Format(j)) = Me("pmt" + Format(i))

Me("FV" + Format(j)) = Me("FV" + Format(i))

Me("Tipo" + Format(j)) = Me("Tipo" + Format(i))

End Sub

'i y j se emplean como variables auxiliares.

'No es válido copiar una columna de datos a sí misma.
'Se usan i y j por ser más comunes como índices.

'El operador + se usa para concatenar caracteres; por

'ejemplo, si i=1 y j=3 (copiar col 1 a col 3), hace que

'Me("Int" + Format(j)) = Me("Int" + Format(i))

'se convierta en Me.Int3 = Me.Int1

'Caso idéntico para las demás asignaciones.

'Después de escribir la expresión algebraica y moverse a otro campo o comando, se evalúa la expresión apareciendo el resultado en el campo "Resultado".

Private Sub IntExp_AfterUpdate()

Dim intExp As Double

If IsNull(Me.IntExp) Then

Exit Sub

End If

Me.IntResult = Eval(Me.IntExp)

End Sub

'El evento ocurre después de actualizar la expresión.

'Se verifica que exista una expresión.

'Eval es una función preconstruída de Access, se usa para
'valuar expresiones algebraicas.

'Transfiere el resultado del Cálculo del Interés a la expresión

Private Sub Command160_Click()

Me.IntResult = Me.Int1

End Sub

'sensibilidad 1.

Se usan subrutinas similares para las sensibilidades 2, 3 y 4. La línea a reemplazar es: **Me.IntResult = Me.Int1**, por

Me.IntResult = Me.Int2

Me.IntResult = Me.Int3

Me.IntResult = Me.Int4

Private Sub Command125_Click()

Me.PV1 = 0

Me.Int1 = 0

Me.Nper1 = 0

Me.pmt1 = 0

Me.FV1 = 0

Me.Tipo1 = 0

End Sub

'Borra los datos de la sensibilidad 1.

'Valor Presente

'Interes

'Número de Pagos.

'Pago

'Valor Futuro

'Anualidad Vencida o anticipada

Se usan subrutinas idénticas para borrar los datos de las sensibilidades 2, 3 y 4, usando las siguientes variables:

Sensibilidad 2: Me.PV2 = 0 Me.Int2 = 0 Me.Nper2 = 0 Me.pmt2 = 0 Me.FV2 = 0 Me.Tipo2 = 0

Sensibilidad 3: Me.PV3 = 0 Me.Int3 = 0 Me.Nper3 = 0 Me.pmt3 = 0 Me.FV3 = 0 Me.Tipo3 = 0

Sensibilidad 4: Me.PV4 = 0 Me.Int4 = 0 Me.Nper4 = 0 Me.pmt4 = 0 Me.FV4 = 0 Me.Tipo4 = 0

```
Private Sub Command128_Click()
```

```
Dim i As Integer
```

```
For i = 1 To 4
```

```
Me("PV" + Format(i)) = 0
```

```
Me("Int" + Format(i)) = 0
```

```
Me("Nper" + Format(i)) = 0
```

```
Me("pmt" + Format(i)) = 0
```

```
Me("FV" + Format(i)) = 0
```

```
Me("Tipo" + Format(i)) = 0
```

```
Next i
```

```
Me.IntExp = Null
```

```
Me.IntResult = Null
```

```
Me.Refresh
```

```
End Sub
```

*Se borran los datos de todas las sensibilidades.

*Se usa i como índice.

*Se recorren las 4 sensibilidades, borrando los datos.

* Si i = 2, la instrucción: Me("PV" + Format(i)) = 0

*equivale a: Me.PV2 = 0

*Caso idéntico para las variables: Int, Nper, pmt, FV y Tipo.

*Se inicializan las variables adicionales. IntExp es el campo en

*que se escribe una expresión algebraica; IntResult es el

*resultado de esa expresión.

4.6.4 Anualidades: Cálculo del Pago.

Formulario de datos: Cálculo del Pago Periódico

	(col 1) Pago Sensibilidad 1	(col 2) Pago Sensibilidad 2	(col 3) Pago Sensibilidad 3	(col 4) Pago Sensibilidad 4
borrar	borrar	borrar	borrar	borrar
Valor Presente	Unbound	Unbound	Unbound	Unbound
Pago	Unbound	Unbound	Unbound	Unbound
No. de Períodos	Unbound	Unbound	Unbound	Unbound
Interés	Unbound	Unbound	Unbound	Unbound
Valor Futuro	Unbound	Unbound	Unbound	Unbound
Tipo	Unbound	Unbound	Unbound	Unbound
Borrar Todo	Copiar Datos de Columna: [Unb] a Columna: [Unb]			

El interés y los pagos deben tener la misma periodicidad.

Las cajas de texto permiten realizar operaciones en ellas

Expresión: Unbound
Resultado: Unbound
Pagos a Interés: 1 2 3 4

Format | Data | Event | Other | All

- Name Detail
- Force New Page None
- Now Row Or Col None
- Keep Together No
- Visible Yes
- Display When Always
- Can Grow No
- Can Shrink No
- Height 1.1993"
- BackColor 16777215
- Special Effect Flat
- Tag On Click
- On DB Click On Mouse Down
- On Mouse Move On Mouse Up

Características del detalle de las formas de anualidades.
Todas las formas de anualidades permiten 4 sensibilidades

Option Compare Database

*Pantallas de ayuda. Instrucción que se cambia en la subrutina de carga de pantalla.

```
stLinkCriteria = "[Forma]=" & "" & "" & Me![AnPago] & ""
```

```
stLinkCriteria = "[Forma]=" & "" & "" & Me![AnPagEjmp] & ""
```

AnPago, AnPagEjmp, son las variables auxiliares definidas para cargar las formas de ayuda.

*Calcula el Pago de la primer sensibilidad. Verifica que los parámetros tengan un valor numérico.

```
Private Sub Command109_Click()
```

```
    If Not IsNumeric(Me.PV1) Then                'Valor presente o Saldo Inicial
```

```
        MsgBox "'Valor Presente (ó Saldo Inicial)' tiene un valor no numérico."
```

```
    Exit Sub
```

```
End If
```

```
    If Not IsNumeric(Me.Int1) Then              'Interés
```

```
        MsgBox "'Interés' tiene un valor no numérico."
```

```
    Exit Sub
```

```
End If
```

```
    If Not IsNumeric(Me.Nper1) Then            'Número de Pagos.
```

```
        MsgBox "'Número de períodos' tiene un valor no numérico."
```

```
    Exit Sub
```

```
End If
```

```
    If Not IsNumeric(Me.FV1) Then              'Valor Futuro o Saldo Final
```

```
        MsgBox "'Valor Futuro (ó Saldo Final)' tiene un valor no numérico."
```

```
    Exit Sub
```

```
End If
```

```
    Me.pmt1 = Pmt(Me.Int1, Me.Nper1, Me.PV1, Me.FV1, Me.Tipo1)'Pmt es una función preconstruída de Access.
```

```
    If Me.Int1 < 0 Then                          'El Interés calculado debe ser mayor que cero.
```

```
        MsgBox "'Error. Datos inválidos, se genera tasa de interés menor que cero."
```

```
    End If
```

```
End Sub
```

'El Pago de las sensibilidades 2,3 y 4 se genera en forma idéntica. Se especifica aquí sólo la línea que genera el cálculo:

Sensibilidad 1: Me.pmt2 = Pmt(Me.Int2, Me.Nper2, Me.PV2, Me.FV2, Me.Tipo2)

Sensibilidad 2: Me.pmt3 = Pmt(Me.Int3, Me.Nper3, Me.PV3, Me.FV3, Me.Tipo3)

Sensibilidad 3: Me.pmt4 = Pmt(Me.Int4, Me.Nper4, Me.PV4, Me.FV4, Me.Tipo4)

'Transfiere los datos de la columna i a la j; donde $i < j$; $i = 1..4$; $j = 1..4$

Private Sub Command120_Click()

Dim i, j As Integer

'i y j se emplean como variables auxiliares.

If Col1 = Col2 Then Exit Sub

'No es válido copiar una columna de datos a sí misma.

i = Col1

'Se usan i y j por ser más comunes como índices.

j = Col2

Me("PV" + Format(j)) = Me("PV" + Format(i))

'El operador + se usa para concatenar caracteres; por

Me("Int" + Format(j)) = Me("Int" + Format(i))

'ejemplo, si i=1 y j=3 (copiar col 1 a col 3), hace que

Me("Nper" + Format(j)) = Me("Nper" + Format(i))

'Me("Int" + Format(j)) = Me("Int" + Format(i))

Me("pmt" + Format(j)) = Me("pmt" + Format(i))

'se convierta en: Me.Int3 = Me.Int1

Me("FV" + Format(j)) = Me("FV" + Format(i))

'Caso idéntico para las demás asignaciones.

Me("Tipo" + Format(j)) = Me("Tipo" + Format(i))

End Sub

'Después de escribir la expresión algebraica y moverse a otro campo o comando, se evalúa la expresión apareciendo el resultado en el campo "Resultado".

Private Sub IntExp_AfterUpdate()

'El evento ocurre después de actualizar la expresión.

Dim intExp As Double

If IsNull(Me.IntExp) Then

'Se verifica que exista una expresión.

Exit Sub

End If

Me.IntResult = Eval(Me.IntExp)

'Eval es una función preconstruida de Access, se usa para

End Sub

'valuar expresiones algebraicas.

Private Sub Command160_Click()

'Transfiere el resultado de la expresión al Interés de la

Me.Int1 = Me.IntResult

'sensibilidad 1.

End Sub

Private Sub Command160_Click()

'Transfiere el resultado de la expresión al Interés de la

Me.Int1 = Me.IntResult

'sensibilidad 1.

End Sub

Se usan subrutinas similares para las sensibilidades 2, 3 y 4. La línea a reemplazar es: Me.Int1 = Me.IntResult, por

Me.Int2 = Me.IntResult

Me.Int3 = Me.IntResult

Me.Int4 = Me.IntResult

Private Sub Command125_Click()

'Borra los datos de la sensibilidad 1.

Me.PV1 = 0

'Valor Presente

Me.Int1 = 0

'Interes

Me.Nper1 = 0

'Número de Pagos.

```
Me.pmt1 = 0
Me.FV1 = 0
Me.Tipo1 = 0
End Sub
```

```
'Pago
'Valor Futuro
'Anualidad Vencida o anticipada
```

Se usan subrutinas idénticas para borrar los datos de las sensibilidades 2, 3 y 4, usando los siguientes variables:

Sensibilidad 2: Me.PV2 = 0 Me.Int2 = 0 Me.Nper2 = 0 Me.pmt2 = 0 Me.FV2 = 0 Me.Tipo2 = 0

Sensibilidad 3: Me.PV3 = 0 Me.Int3 = 0 Me.Nper3 = 0 Me.pmt3 = 0 Me.FV3 = 0 Me.Tipo3 = 0

Sensibilidad 4: Me.PV4 = 0 Me.Int4 = 0 Me.Nper4 = 0 Me.pmt4 = 0 Me.FV4 = 0 Me.Tipo4 = 0

```
Private Sub Command128_Click()
```

```
Dim i As Integer
```

```
For i = 1 To 4
```

```
Me("PV" + Format(i)) = 0
```

```
Me("Int" + Format(i)) = 0
```

```
Me("Nper" + Format(i)) = 0
```

```
Me("pmt" + Format(i)) = 0
```

```
Me("FV" + Format(i)) = 0
```

```
Me("Tipo" + Format(i)) = 0
```

```
Next i
```

```
Me.IntExp = Null
```

```
Me.IntResult = Null
```

```
Me.Refresh
```

```
End Sub
```

'Se borran los datos de todas las sensibilidades.

'Se usa i como índice.

'Se recorren las 4 sensibilidades, borrando los datos.

' Si i = 2, la instrucción: Me("PV" + Format(i)) = 0

'equivale a: Me.PV2 = 0

'Caso idéntico para las variables: Int, Nper, pmt, FV y Tipo.

'Se inicializan las variables adicionales. IntExp es el campo en

'que se escribe una expresión algebraica; IntResult es el

'resultado de esa expresión.

4.6.5 Anualidades: Cálculo del Número de Pagos.

Option Compare Database

Ejemplos ? = "AnNumPagos" des: Cálculo del Número de Pagos

= "AnNPagEjmp"

Una opción de teclado para optimizar el flujo

	col1 No. Pagos Sensibilidad 1	col2 No. Pagos Sensibilidad 2	col3 No. Pagos Sensibilidad 3	col4 No. Pagos Sensibilidad 4
	borrar	borrar	borrar	borrar
Valor Presente (Pago)	Unbound	Unbound	Unbound	Unbound
No. de Periodos (Pago Parcial)	Unbound	Unbound	Unbound	Unbound
Tasa de Interés (Tipo)	Unbound	Unbound	Unbound	Unbound
Valor Futuro (Tipo)	Unbound	Unbound	Unbound	Unbound
Borrar Todo	Copiar Datos de Columna: Unbo a Columna: Unbo			

Aquí se realiza el cálculo de 2 resultados.

El interés y los pagos deben tener la misma periodicidad

Para calcular interés, teclee una expresión algebraica a continuación

Ejemplo: 07/12 * 1.07 / (1+12/1) Enter Tab o clic en otro campo

Expresión: Unbound
Resultado: Unbound

Presar a Interés: 1 2 3 4

*Pantallas de ayuda. Instrucción que se cambia en la subrutina de carga de pantalla.

```
stLinkCriteria = "[Forma]=" & "" & Me![AnNPagEjmp] & ""
stLinkCriteria = "[Forma]=" & "" & Me![AnNumPagos] & ""
```

AnNPagEjmp, AnNumPagos, son las variables auxiliares definidas para cargar las formas de ayuda.

*Calcula el Número de Pagos de la primer sensibilidad. Verifica que los parámetros tengan un valor numérico.

```
Private Sub Command108_Click()
    If Not IsNumeric(Me.Pmt1) Then 'Pago
        MsgBox "'Pago' tiene un valor no numérico."
        Exit Sub
    End If
    If Not IsNumeric(Me.Int1) Then 'Interés
        MsgBox "'Interés' tiene un valor no numérico."
        Exit Sub
    End If
    If Not IsNumeric(Me.PV1) Then 'Valor Presente o Valor Inicial
        MsgBox "'Valor Presente (ó Saldo Inicial)' tiene un valor no numérico."
        Exit Sub
    End If
    If Not IsNumeric(Me.FV1) Then 'Valor Futuro o Saldo Final
        MsgBox "'Valor Futuro (ó Saldo Final)' tiene un valor no numérico."
        Exit Sub
    End If
    Me.Nper1 = nper(Me.Int1, Me.pmt1, Me.PV1, Me.FV1, Me.Tipo1) 'Nper es función Preconstruida
    If Me.Nper1 < 0 Then 'de Access. El Número de Pagos
```

```

MsgBox "**Error. Datos inválidos, se genera No. de períodos menor que cero." 'debe ser mayor que cero.
End If
Me.PParcial1 = (Me.Nper1 - Int(Me.Nper1)) * Me.pmt1 'Calcula el pago parcial.
End Sub

```

***El Número de Pagos de las sensibilidades 2,3 y 4 se genera en forma idéntica. Se especifica aquí sólo la línea que genera el cálculo:**

```

Sensibilidad 2: Me.Nper2 = nper(Me.Int2, Me.pmt2, Me.PV2, Me.FV2, Me.Tipo2)
Sensibilidad 3: Me.Nper3 = nper(Me.Int3, Me.pmt3, Me.PV3, Me.FV3, Me.Tipo3)
Sensibilidad 4: Me.Nper4 = nper(Me.Int4, Me.pmt4, Me.PV4, Me.FV4, Me.Tipo4)

```

***Transfiere los datos de la columna i a la j; donde $i < j$; $i = 1..4$; $j = 1..4$**

```

Private Sub Command120_Click()
Dim i, j As Integer 'i y j se emplean como variables auxiliares.
If Col1 = Col2 Then Exit Sub 'No es válido copiar una columna de datos a sí misma.
i = Col1 'Se usan i y j por ser más comunes como índices.
j = Col2
Me("PV" + Format(j)) = Me("PV" + Format(i)) 'El operador + se usa para concatenar caracteres; por
Me("Int" + Format(j)) = Me("Int" + Format(i)) 'ejemplo, si i=1 y j=3 (copiar col 1 a col 3), hace que
Me("Nper" + Format(j)) = Me("Nper" + Format(i)) 'Me("Int" + Format(j)) = Me("Int" + Format(i))
Me("pmt" + Format(j)) = Me("pmt" + Format(i)) 'se convierta en: Me.Int3 = Me.Int1
Me("FV" + Format(j)) = Me("FV" + Format(i)) 'Caso idéntico para las demás asignaciones.
Me("Tipo" + Format(j)) = Me("Tipo" + Format(i))
Me("PParcial" + Format(j)) = Me("PParcial" + Format(i))
End Sub

```

***Después de escribir la expresión algebraica y moverse a otro campo o comando, se evalúa la expresión apareciendo el resultado en el campo "Resultado".**

```

Private Sub IntExp_AfterUpdate() 'El evento ocurre después de actualizar la expresión.
Dim intExp As Double
If IsNull(Me.IntExp) Then 'Se verifica que exista una expresión.
Exit Sub
End If
Me.IntResult = Eval(Me.IntExp) 'Eval es una función preconstruída de Access, se usa para
End Sub 'valuar expresiones algebraicas.

```

```

Private Sub Command160_Click() 'Transfiere el resultado de la expresión al Interés de la
Me.Int1 = Me.IntResult 'sensibilidad 1.
End Sub

```

Se usan subrutinas similares para las sensibilidades 2, 3 y 4. La línea a reemplazar es: Me.Int1 = Me.IntResult, por
Me.Int2 = Me.IntResult
Me.Int3 = Me.IntResult
Me.Int4 = Me.IntResult

Private Sub Command125_Click()

Me.PV1 = 0
Me.Int1 = 0
Me.Nper1 = 0
Me.pmt1 = 0
Me.FV1 = 0
Me.Tipo1 = 0
Me.PParcial = 0
End Sub

***Borra los datos de la sensibilidad 1.**
*Valor Presente
*Interes
*Número de Pagos.
*Pago
*Valor Futuro
*Anualidad Vencida o anticipada
*Pago Parcial.

Se usan subrutinas idénticas para borrar los datos de las sensibilidades 2, 3 y 4, usando las siguientes variables:

Sensibilidad 2: Me.PV2 = 0 Me.Int2 = 0 Me.Nper2 = 0 Me.pmt2 = 0 Me.FV2 = 0 Me.Tipo2 = 0;

Me.PParcial = 0

Sensibilidad 3: Me.PV3 = 0 Me.Int3 = 0 Me.Nper3 = 0 Me.pmt3 = 0 Me.FV3 = 0 Me.Tipo3 = 0; Me.PParcial = 0

Sensibilidad 4: Me.PV4 = 0 Me.Int4 = 0 Me.Nper4 = 0 Me.pmt4 = 0 Me.FV4 = 0 Me.Tipo4 = 0; Me.PParcial = 0

Private Sub Command128_Click()

Dim i As Integer

For i = 1 To 4

Me("PV" + Format(i)) = 0

Me("Int" + Format(i)) = 0

Me("Nper" + Format(i)) = 0

Me("pmt" + Format(i)) = 0

Me("FV" + Format(i)) = 0

Me("Tipo" + Format(i)) = 0

Me("PParcial" + Format(i)) = 0

Next i

Me.IntExp = Null

Me.IntResult = Null

Me.Refresh

End Sub

***Se borran los datos de todas las sensibilidades.**

*Se usa i como índice.

*Se recorren las 4 sensibilidades, borrando los datos.

* Si i = 2, la instrucción: Me("PV" + Format(i)) = 0

*equivale a: Me.PV2 = 0

*Caso idéntico para las variables: Int, Nper, pmt, FV y Tipo.

***Se inicializan las variables adicionales. IntExp es el campo en que se escribe una expresión algebraica; IntResult es el resultado de esa expresión.**

4.6.6 Anualidades: Cálculo del Capital e Interés Pagado.

Ejemplos		Capital e Interés Pagado			
"AnCpIntEjm"		col1.Sensibil1	col2.Sensibil2	col3.Sensibil3	col4.Sensibil4
borrar		borrar		borrar	
Valor Presente		Unbound	Unbound	Unbound	Unbound
No. de Periodos		Unbound	Unbound	Unbound	Unbound
Tasa de Interés	1	Unbound	Unbound	Unbound	Unbound
Valor Futuro		Unbound	Unbound	Unbound	Unbound
Tipo		Unbound	Unbound	Unbound	Unbound
Periodo		Unbound	Unbound	Unbound	Unbound
Interés a pagar en este periodo		Unbound	Unbound	Unbound	Unbound
Pagado hasta este periodo		Unbound	Unbound	Unbound	Unbound
Capital a pagar en este periodo		Unbound	Unbound	Unbound	Unbound
Pagado hasta este periodo		Unbound	Unbound	Unbound	Unbound
Borrar Todo		Copiar Datos de Columna: Unbo		a Columna: Unbo	

Aquí se calculan 4 resultados por columna.

El interés y los pagos deben tener la misma periodicidad
 Para calcular interés, teclee e una expresión algebraica abaj.
 Ejemplo: 0.07/12 * 1.07 * (1/12)^1 Enter Tab o clic en otro campo

Expresión: Unbound
 Resultado: Unbound

Passar a Interés: 1 2 3 4

Option Compare Database
 Private Sub Command108_Click()

'Pantallas de ayuda. Instrucción que se cambia en la subrutina de carga de pantalla.
 stLinkCriteria = "[Forma]=" & "" & Me![AnCapInt] & "" 'Carga la pantalla de ayuda.
 stLinkCriteria = "[Forma]=" & "" & Me![AnCpIntEjm] & "" 'Carga la pantalla que proporciona ejemplos.

AnCapInt, AnCpIntEjm, son las variables auxiliares definidas para cargar las formas de ayuda.

'Calcula el Capital e Interés Pagados para un periodo y también el acumulado hasta ese periodo. Verifica que los parámetros tengan un valor numérico.

```
Private Sub Command113_Click()
Dim i As Integer, IJ, IntPagado, IntMes, CapPagado, CapMes As Double 'Variables auxiliares
If Not IsNumeric(Me.Int1) Then 'Interés
MsgBox "'Interés' tiene un valor no numérico."
Exit Sub
End If
If Not IsNumeric(Me.PV1) Then 'Valor Presente o Saldo Inicial' tiene un valor no numérico."
MsgBox "'Valor Presente (ó Saldo Inicial)' tiene un valor no numérico."
Exit Sub
End If
If Not IsNumeric(Me.FV1) Then 'Valor Futuro o Saldo Final
MsgBox "'Valor Futuro (ó Saldo Final)' tiene un valor no numérico."
Exit Sub
End If
If Not IsNumeric(Me.Per1) Then 'Período especificado
MsgBox "'Período en que se desea conocer Interés Pagado y Capital Pagado' tiene un valor no numérico."
Exit Sub
End If
```

```

End If
If Not IsNumeric(Me.Nper1) Then          'Número de Pagos
MsgBox ""Número de Pagos" tiene un valor no numérico."
Exit Sub
End If
If Val(Me.Nper1) < Val(Me.Per1) Then    'Período especificado debe ser menor que no. de pagos.
MsgBox ""Error. Período en que se desea conocer Interés Pagado y Capital Pagado" es mayor que el "Total de
Periodos."
Exit Sub
End If

For i = 1 To Me.Per1                    'Calcula Cap é Int pagados para
IntMes = IPmt(Me.Int1, i, Me.Nper1, Me.PV1, Me.FV1, Me.Tipo1) 'períodos desde el inicio hasta el
IntPagado = IntPagado + IntMes          'especificado.
CapMes = PPmt(Me.Int1, i, Me.Nper1, Me.PV1, Me.FV1, Me.Tipo1)'ipmt y PPmt son funciones
CapPagado = CapPagado + CapMes        'Preconstruidas de Access
Next i

Me.IntPag1 = IntPagado                 'Se asignan los cálculos a las
Me.IntMes1 = IntMes                   'variables de la pantalla.
Me.CapPag1 = CapPagado
Me.CapMes1 = CapMes
End Sub

```

*El Número de Pagos de las sensibilidades 2,3 y 4 se genera en forma idéntica. Se especifica aquí sólo las líneas que generan los cálculos:

```

IntMes = IPmt(Me.Int2, IJ, Me.Nper2, Me.PV2, Me.FV2, Me.Tipo2)
CapMes = PPmt(Me.Int2, IJ, Me.Nper2, Me.PV2, Me.FV2, Me.Tipo2)

IntMes = IPmt(Me.Int3, IJ, Me.Nper3, Me.PV3, Me.FV3, Me.Tipo3)
CapMes = PPmt(Me.Int3, IJ, Me.Nper3, Me.PV3, Me.FV3, Me.Tipo3)

IntMes = IPmt(Me.Int4, IJ, Me.Nper4, Me.PV4, Me.FV4, Me.Tipo4)
CapMes = PPmt(Me.Int4, IJ, Me.Nper4, Me.PV4, Me.FV4, Me.Tipo4)

```

*Transfiere los datos de la columna i a la j; donde $i < j$; $i = 1..4$; $j = 1..4$

```

Private Sub Command120_Click()
Dim i, j As Integer                    'i y j se emplean como variables auxiliares.
If Col1 = Col2 Then Exit Sub          'No es válido copiar una columna de datos a sí misma.
i = Col1                               'Se usan i y j por ser más comunes como índices.
j = Col2
Me("PV" + Format(j)) = Me("PV" + Format(i))
Me("Int" + Format(j)) = Me("Int" + Format(i))
Me("Nper" + Format(j)) = Me("Nper" + Format(i))
Me("pmt" + Format(j)) = Me("pmt" + Format(i))
Me("FV" + Format(j)) = Me("FV" + Format(i))
Me("Tipo" + Format(j)) = Me("Tipo" + Format(i))
Me("PParcial" + Format(j)) = Me("PParcial" + Format(i))
Me("IntMes" + Format(j)) = Me("IntMes" + Format(i))
Me("IntPag" + Format(j)) = Me("IntPag" + Format(i))
Me("CapMes" + Format(j)) = Me("CapMes" + Format(i))
Me("CapPag" + Format(j)) = Me("CapPag" + Format(i))
End Sub

```

'El operador + se usa para concatenar caracteres; por ejemplo, si i=1 y j=3 (copiar col 1 a col 3), hace que
'Me("Int" + Format(j)) = Me("Int" + Format(i))
'se convierta en: Me.Int3 = Me.Int1
' Caso idéntico para las demás asignaciones.

```

'Después de escribir una expresión algebraica, se evalúa la expresión en el campo "Resultado".
Private Sub IntExp_AfterUpdate()
    'El evento ocurre después de actualizar la expresión.
    Dim intExp As Double
    If IsNull(Me.IntExp) Then
        'Se verifica que exista una expresión.
        Exit Sub
    End If
    Me.IntResult = Eval(Me.IntExp)
    'Eval es una función preconstruída de Access, se usa para
    'valuar expresiones algebraicas.
End Sub

Private Sub Command160_Click()
    'Transfiere el resultado de la expresión al Interés de la
    'sensibilidad 1.
    Me.Int1 = Me.IntResult
End Sub

```

Se usan subrutinas similares para las sensibilidades 2, 3 y 4. La línea a reemplazar es: **Me.Int1 = Me.IntResult**, por

```

Me.Int2 = Me.IntResult
Me.Int3 = Me.IntResult
Me.Int4 = Me.IntResult

```

```

Private Sub Command125_Click()
    'Borra los datos de la sensibilidad 1.
    Me.PV1 = 0
    'Valor Presente
    Me.Int1 = 0
    'Interes
    Me.Nper1 = 0
    'Número de Pagos.
    Me.pmt1 = 0
    'Pago
    Me.FV1 = 0
    'Valor Futuro
    Me.Tipo1 = 0
    'Anualidad Vencida o anticipada
    Me.IntMes1 = 0
    'Interés generado durante el periodo
    Me.IntPag1 = 0
    'Interés acumulado pagado.
    Me.CapMes1 = 0
    'Capital pagado durante el periodo
    Me.CapPag1 = 0
    'Capital acumulado pagado.
End Sub

```

Se usan subrutinas idénticas para borrar los datos de las sensibilidades 2, 3 y 4, usando las siguientes variables:

```

Sensibilidad 2: Me.PV2 = 0 Me.Int2 = 0 Me.Nper2 = 0 Me.pmt2 = 0 Me.FV2 = 0 Me.Tipo2 = 0;
                Me.IntMes1 = 0 Me.IntPag1 = 0 Me.CapMes1 = 0 Me.CapPag1 = 0
Sensibilidad 3: Me.PV3 = 0 Me.Int3 = 0 Me.Nper3 = 0 Me.pmt3 = 0 Me.FV3 = 0 Me.Tipo3 = 0; Me.IntMes1 =
                0 Me.IntPag1 = 0 Me.CapMes1 = 0 Me.CapPag1 = 0
Sensibilidad 4: Me.PV4 = 0 Me.Int4 = 0 Me.Nper4 = 0 Me.pmt4 = 0 Me.FV4 = 0 Me.Tipo4 = 0; Me.IntMes1 =
                0 Me.IntPag1 = 0 Me.CapMes1 = 0 Me.CapPag1 = 0

```

```

Private Sub Command128_Click()
    'Se borran los datos de todas las sensibilidades.
    Dim i As Integer
    'Se usa i como índice.
    For i = 1 To 4
        'Se recorren las 4 sensibilidades, borrando los datos.
        Me("PV" + Format(i)) = 0
        ' Si i = 2, la instrucción: Me("PV" + Format(i)) = 0
        'equivale a: Me.PV2 = 0
        'Caso idéntico para las variables: Int, Nper, pmt, FV y Tipo.
        Me("Int" + Format(i)) = 0
        Me("Nper" + Format(i)) = 0
        Me("pmt" + Format(i)) = 0
        Me("FV" + Format(i)) = 0
        Me("Tipo" + Format(i)) = 0
        Me("PParcial" + Format(i)) = 0
        Me("IntMes" + Format(i)) = 0
        Me("IntPag" + Format(i)) = 0
        Me("CapMes" + Format(i)) = 0
        Me("CapPag" + Format(i)) = 0
    Next i
    Me.IntExp = Null
    'Se inicializan las variables adicionales. IntExp es el campo en
    Me.IntResult = Null
    'que se escribe una expresión algebraica; IntResult es el
    Me.Refresh
    'resultado de esa expresión.
End Sub

```

4.6.7 Cálculo del Valor Presente Neto.

Option Compare Database

Valor Presente Neto		-VPN		Instrucciones/Ayuda	
Pagos Iguales ó Desiguales con Tasa de Interés Constante ó Variable					
Teclee los Pagos y el Interés de cada periodo.					
limVPvar					
Esta subforma permite la captura de flujos é interés para cada periodo. Está ligada a la tabla "tblVPvar".					
Al calcular VPN's se pueden pasar a las memorias, donde se los puede cambiar el signo é indicar si se desean sumar en M5 al hacer click en comando suma (letra Sigma).					
Adicionalmente se pueden realizar operaciones en las cajas-texto "Expresión" y "Resultado".					
Esta forma no ofrece guardado/acceso de datos a archivos. La forma se limpia automaticamente cada ocasión que se carga.					
Valor Presente Neto		Unbound		Calcular VPN de los Valores en la Tabla	
Cambiar signo:		Sumar?		Borrar M's	
M1	<input checked="" type="checkbox"/>	Unbound		M1	
M2	<input checked="" type="checkbox"/>	Unbound		M2	
M3	<input checked="" type="checkbox"/>	Unbound		M3	
M4	<input checked="" type="checkbox"/>	Unbound		M4	
	<input type="checkbox"/>	Unbound		M5	
El interés y los pagos deben tener la misma periodicidad					
El tipo de letra, tamaño y color son iguales a los de la tabla					
Expresión: Unbound					
Resultado: Unbound					

Option Explicit

*Pantallas de ayuda. Instrucción que se cambia en la subrutina de carga de pantalla.

stLinkCriteria = "[Forma]=" & "" & Me![VPN] & ""

VPN es la variable auxiliar definida para cargar la forma de ayuda.

Private Sub Command63_Click()

*Inicializa las variables empleadas en la pantalla.

Me.M1 = Null

*Memoria 1 a 4 almacenan el VP neto, M5 es la suma de M1 a M4.

Me.M2 = Null

Me.M3 = Null

Me.M4 = Null

Me.M5 = Null

Me.M1Ch = False *Caja-chequeo indica si M1 se suma o no al usar el comando suma de memorias.

Me.M2Ch = False *Cajas chequeo M2 a M4, mismo caso que M1.

Me.M3Ch = False

Me.M4Ch = False

*Vpresente

No se inicializa.

End Sub

Private Sub Form_Open(Cancel As Integer)

*Inicializa valores de la forma al cargarse.

Dim stDocName As String

stDocName = "qryDelVPvar"

*Query que borra datos de la subforma.

DoCmd.OpenQuery stDocName, acNormal, acEdit

*Se ejecuta el query.

Me.VPresente = Null

*Variable en que se calcula el Valor Presente

Me.IntOrig = Null

*Variable en que se escribe la expresión algebraica.

Me.Result1 = Null

*Variable en que se calcula la expresión algebraica.

End Sub

*Crea el valor presente neto recorriendo todos los flujos de entrada en orden reversado.

Private Sub Command9_Click()

Dim dbs As DAO.Database

'Se declara un base de datos

Dim rs As DAO.Recordset

'Se declara un archivo de datos

Dim i, j, nper As Integer, Flujo(360), Tasa(360), ValPres As Double

'variables auxiliares.

Dim stDocName As String

nper = DCount("[Pago]", "tblVPvar")

'Se verifica cuantos flujos hay, equivale al número de pagos.

If nper = 0 Then

MsgBox "Error. No existen pagos."

Exit Sub

End If

ContNulls = 0

'Se verifica que no haya datos omitidos (que aparezcan en blanco).

Set dbs = CurrentDb

'Se conecta a EmCalc

Set rs = dbs.OpenRecordset("tblVPvar")

'Abre la tabla de datos como archivo.

rs.MoveFirst

'Lee el primer registro.

Do While Not rs.EOF

'Ejecuta el ciclo mientras haya registros en el archivo.

If IsNull(rs!Pago) Or IsNull(rs!Interes) Then ContNulls = ContNulls + 1

'Encuentra valores omitidos.

rs.MoveNext

'Lee el siguiente registro del archivo.

Loop

'Cierra la estructura Do While.

rs.Close

'Cierra el archivo.

If ContNulls > 0 Then

'Verifica si encontró valores omitidos.

MsgBox "****ERROR***. Existen: " & ContNulls & " valores en blanco. Borre los renglones o reemplácelos por ceros"

Exit Sub

'termina la subrutina por haber encontrado errores.

End If

'Cierra la estructura IF.

Set dbs = CurrentDb

'Se indica que se usa la base de datos actual (EmCalc)

Set rs = dbs.OpenRecordset("tblVPvar")

'La tabla tblVPvar se usa como archivo. Contiene los campos

'Pago é Interés.

'Se llenan los arreglos con los flujos (pagos) y la tasa de interés de ese período.

rs.MoveFirst

'Se accesa el primer registro (el primer flujo capturado).

With rs

'La instrucción permite referenciar los campos con "!"

Do While Not .EOF

'Recorre todos los flujos.

i = i + 1

Flujo(i) = !Pago

'Guarda en un arreglo los flujos y el interés

Tasa(i) = !Interes

'Accesa el siguiente registro.

.MoveNext

'Termina el ciclo "Do while".

Loop

'Termina la instrucción "With rs"

End With

'Se crea el valor presente acumulando los flujos en forma reversada (primero el más reciente).

For i = 1 To nper

'Se recorren todos los períodos.

j = (nper + 1) - i

'Se crea el nuevo índice para accesar primero el más reciente.

ValPres = (ValPres + Flujo(j)) / (1 + Tasa(j))

'Se suman los flujos, descontando el interés.

Next

'Cierra ciclo "For".

Me.VPresente = ValPres

'Asigna el cálculo a la variable de la pantalla.

rs.Close

'Cierra archivo de flujos é interés.

dbs.Close

'Cierra la base de datos.

Set dbs = Nothing

'Asigna valor nulo a la base de datos.

End Sub

Private Sub Command52_Click() *Se pasa el cálculo del Valor Presente a la memoria 1.

Me.M1 = Me.Vpresente

End Sub

Mismo caso para las memorias M2, M3 y M4, reemplazando M1.

Private Sub Command73_Click() *Se cambia de signo a M1.

If IsNull(Me.M1) Then *Verifica que M1 tenga datos.

Exit Sub

Else: Me.M1 = -Me.M1 *Cambia el signo.

End If

End Sub

Mismo caso para M2, M3 y M4, reemplazando M1.

Private Sub Command64_Click() *Genera la suma de memorias acumulando una por una, verificando

Me.M5 = 0 *una por una que esté indicado que se sume.

If Me.M1Ch Then Me.M5 = Me.M1

If Me.M2Ch Then Me.M5 = Me.M5 + Me.M2

If Me.M3Ch Then Me.M5 = Me.M5 + Me.M3

If Me.M4Ch Then Me.M5 = Me.M5 + Me.M4

End Sub

cambiar

signo a:

	Sumar?		Borrar M's
M1	<input checked="" type="checkbox"/>	335.78	M1
M2	<input checked="" type="checkbox"/>	-382.12	M2
M3	<input type="checkbox"/>		M3
M4	<input type="checkbox"/>		M4
	<input checked="" type="checkbox"/>	-46.34	M5

Private Sub Command58_Click() *Borra memoria M1, a la vez que indica que no se suma. M5 es

Me.M1 = Null

*borrada porque ya no refleja la suma de las memorias M1 a M4.

Me.M5 = Null

Me.M1Ch = False

Me.Refresh

End Sub

Mismo caso para borrar las memorias M2 a M5. M5 es la suma de las memorias M2 a M4.

Private Sub IntOrig_AfterUpdate() *Valúa la expresión algebraica después de editar la expresión.

Dim intx As Double

If IsNull(Me.IntOrig) Then *Verifica que exista una expresión

Exit Sub

End If

intx = Eval(Me.IntOrig)

*Eval es una función preconstruída de Access.

Me.Result1 = intx

*El resultado se asigna a la variable de la pantalla.

End Sub

4.6.8 Depreciación.

Option Compare Database

Depreciación		-="Depr"	Instrucciones/Ayuda																				
Método de Línea Recta.																							
frmDepr1																							
Aquí se utilizan dos subformas que despliegan la información calculada en forma idéntica.																							
Abajo se puede ver la subforma en vista de diseño.																							
El usuario puede posicionarse sobre los campos de las subformas para copiar (de ser necesario) con las teclas <Ctl> + C, pero no se permite su edición.																							
Método de Doble Declinación del Saldo (ú otro factor)																							
frmDepr2																							
Esta es una sobreposición.																							
<table border="1"> <thead> <tr> <th colspan="2">Borrar Parámetros y Tabla</th> </tr> </thead> <tbody> <tr> <td>Costo:</td> <td>Unbound</td> </tr> <tr> <td>Valor de Salvamento:</td> <td>Unbound</td> </tr> <tr> <td>Vida Útil:</td> <td>Unbound</td> </tr> <tr> <td colspan="2" style="text-align: center;">Calcular</td> </tr> <tr> <td>Depreciación:</td> <td>Unbound</td> </tr> </tbody> </table>				Borrar Parámetros y Tabla		Costo:	Unbound	Valor de Salvamento:	Unbound	Vida Útil:	Unbound	Calcular		Depreciación:	Unbound								
Borrar Parámetros y Tabla																							
Costo:	Unbound																						
Valor de Salvamento:	Unbound																						
Vida Útil:	Unbound																						
Calcular																							
Depreciación:	Unbound																						
<table border="1"> <thead> <tr> <th colspan="2">Borrar Parámetros y Tabla</th> </tr> </thead> <tbody> <tr> <td>Costo:</td> <td>Unbound</td> </tr> <tr> <td>Valor de Salvamento:</td> <td>Unbound</td> </tr> <tr> <td>Vida Útil:</td> <td>Unbound</td> </tr> <tr> <td>Factor:</td> <td>Unbound</td> </tr> <tr> <td colspan="2" style="text-align: center;">Calcular</td> </tr> <tr> <td>Depreciación:</td> <td>Unbound</td> </tr> </tbody> </table>				Borrar Parámetros y Tabla		Costo:	Unbound	Valor de Salvamento:	Unbound	Vida Útil:	Unbound	Factor:	Unbound	Calcular		Depreciación:	Unbound						
Borrar Parámetros y Tabla																							
Costo:	Unbound																						
Valor de Salvamento:	Unbound																						
Vida Útil:	Unbound																						
Factor:	Unbound																						
Calcular																							
Depreciación:	Unbound																						
<table border="1"> <thead> <tr> <th colspan="4">Form Header</th> </tr> </thead> <tbody> <tr> <td>Per</td> <td>Depr</td> <td>DeprAcum</td> <td>CostoRemanente</td> </tr> <tr> <th colspan="4">Detail</th> </tr> <tr> <td>per</td> <td>Depr</td> <td>DeprAcum</td> <td>CostoRemanente</td> </tr> <tr> <th colspan="4">Form Footer</th> </tr> </tbody> </table>				Form Header				Per	Depr	DeprAcum	CostoRemanente	Detail				per	Depr	DeprAcum	CostoRemanente	Form Footer			
Form Header																							
Per	Depr	DeprAcum	CostoRemanente																				
Detail																							
per	Depr	DeprAcum	CostoRemanente																				
Form Footer																							
La subforma, es una forma en su diseño; su uso dentro de otra forma es la que la hace "Subforma".																							

Option Explicit

*Pantallas de ayuda. Instrucción que se cambia en la subrutina de carga de pantalla.

stLinkCriteria = "[Forma]=" & "" & Me![Depr] & ""

Depr es la variable auxiliar definida para cargar la forma de ayuda.

Se genera la tabla de depreciación del método de doble declinación ú otro factor.

```
Private Sub Command19_Click()
```

```
Dim dbs As DAO.Database
```

*Definición de EmCalc como base de datos.

```
Dim rs As DAO.Recordset
```

*Se define un archivo a emplear

```
Dim i, Vutil As Integer, Depr, DeprAcum As Double
```

*variables auxiliares.

```
Dim stDocName As String
```

```
If Not IsNumeric(Me.Costo2) Then
```

*Validación de datos de entrada.

```
MsgBox "**Error. Costo no numérico."
```

*Costo del bien a depreciar.

```
Exit Sub
```

```
Else
```

```
If Val(Me.Costo2) < 1 Then
```

```
MsgBox "**Error. Costo erróneo. Debe ser mayor que 0"
```

```
Exit Sub
```

```
End If
```

```
End If
```

```
If Not IsNumeric(Me.Salv2) Then
```

*Validación del Valor de Salvamento

```
MsgBox "**Error. Valor de Salvamento no numérico."
```



```

Exit Sub
Else
  If Val(Me.Salv2) < 0 Or Val(Me.Salv2) >= Val(Me.Costo2) Then
    MsgBox "Error. Valor de Salvamento inválido. Debe ser mayor que 0 y menor que el Costo."
  Exit Sub
End If
End If

If Not IsNumeric(Me.VUtil2) Then
  MsgBox "Error. Años de Vida Util no numérico."
Exit Sub
Else
  If Val(Me.VUtil2) < 1 Then
    MsgBox "Error. Años de Vida Util inválido. Debe ser mayor que 0."
  Exit Sub
End If
End If

If Not IsNumeric(Me.fact2) Then
  MsgBox "Error. Factor de depreciación no numérico."
Me.fact2 = 2
Else
  If Val(Me.fact2) < 0 Then
    MsgBox "Error. Factor de depreciación inválido. Debe ser mayor que 0."
  Exit Sub
End If
End If

Vutil = Me.VUtil2
If Me.fact2 = 0 Then Me.fact2 = 2

Vutil = Me.VUtil2
stDocName = "qryDelDepr2"
DoCmd.OpenQuery stDocName, acNormal, acEdit

Me.Deprec2 = "Véase Tabla"
Set dbs = CurrentDb
Set rs = dbs.OpenRecordset("tblDepr2")

For i = 1 To Vutil
  With rs
    .AddNew
    !Per = i
    Depr = DDB(Me.Costo2, Me.Salv2, Vutil, i, Me.fact2)
    DeprAcum = DeprAcum + Depr
    !Depr = Depr
    !DeprAcum = DeprAcum
    !CostoRemanente = Me.Costo2 - DeprAcum
    rs.Update
  End With
Next

rs.Close
dbs.Close
Set dbs = Nothing
End Sub

```

Genera la depreciación por el método lineal.

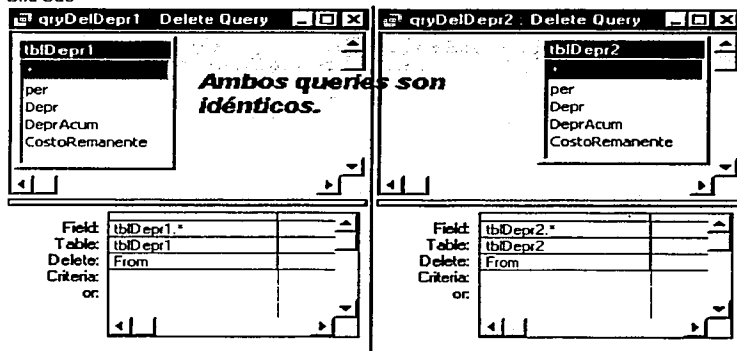
```
Private Sub Command9_Click()  
Dim dbs As DAO.Database 'Definición de EmCalc como base de datos.  
Dim rs As DAO.Recordset 'Se define un archivo a emplear  
Dim i, Vutil As Integer, Depr, DeprAcum As Double 'variables auxiliares.  
Dim stDocName As String  
If Not IsNumeric(Me.Costo1) Then 'Validación de datos de entrada.  
MsgBox "**Error. Costo no numérico." 'Costo del bien a depreciar.  
Exit Sub  
Else  
If Val(Me.Costo1) < 1 Then  
MsgBox "**Error. Costo erróneo. Debe ser mayor que 0"  
Exit Sub  
End If  
End If  
If Not IsNumeric(Me.Salv1) Then 'Validación del Valor de Salvamento  
MsgBox "**Error. Valor de Salvamento no numérico."  
Exit Sub  
Else  
If Val(Me.Salv1) < 0 Or Val(Me.Salv1) >= Val(Me.Costo2) Then  
MsgBox "**Error. Valor de Salvamento inválido. Debe ser mayor que 0 y menor que el Costo."  
Exit Sub  
End If  
End If  
If Not IsNumeric(Me.Vutil1) Then 'Validación de la Vida Util  
MsgBox "**Error. Años de Vida Util no numérico."  
Exit Sub  
Else  
If Val(Me.Vutil1) < 1 Then  
MsgBox "**Error. Años de Vida Util inválido. Debe ser mayor que 0."  
Exit Sub  
End If  
End If  
Set dbs = CurrentDb 'Se indica que se utilizará esta base de datos: EmCalc  
Set rs = dbs.OpenRecordset("tblDepr1") 'Se abre la tabla tblDepr1 para guardar datos a mostrar  
Me.Deprrec1 = SLN(Me.Costo1, Me.Salv1, Me.Vutil1) 'SLN es una función preconstruída de Access.  
For i = 1 To Vutil 'Ciclo para todos los años de vida útil del bien.  
With rs 'Indica que se usa el registro de tblDepr1  
.AddNew 'Se agrega un registro (fue borrada previamente)  
!Per = i 'Se crea el período para cada renglón  
!Depr = Me.Deprrec1 'Se guarda la depreciación  
DeprAcum = DeprAcum + Me.Deprrec1 'Se crea la depreciación acumulada.  
!DeprAcum = DeprAcum 'Se guarda la depreciación acumulada  
!CostoRemanente = Me.Costo1 - DeprAcum 'Se guarda el costo remanente del bien.  
rs.Update 'Se actualiza el registro en el archivo.  
End With 'Termina cláusula With.  
Next 'Cierra ciclo For.  
rs.Close 'Se cierra el archivo "tblDepr1".  
dbs.Close 'Se cierra la base de datos (EmCalc).  
Set dbs = Nothing 'Se desasigna la variable de base de datos.  
Me.Refresh 'Se refresca la pantalla para desplegar la subforma.  
End Sub 'Termina el código.
```

Esta subrutina se corre automáticamente al abrir la forma. Borra las tablas que alimentan las subformas de información.

```
Private Sub Form_Open(Cancel As Integer)
```

```
    Dim stDocName As String
    stDocName = "qryDelDepr1"           'Query que borra la tabla de depreciación lineal
    DoCmd.OpenQuery stDocName, acNormal, acEdit
    stDocName = "qryDelDepr2"         'Query que borra la tabla de depreciación de doble
    DoCmd.OpenQuery stDocName, acNormal, acEdit 'declinación.
```

```
End Sub
```



***Borra la tabla de la depreciación lineal.**

```
Private Sub Command30_Click()
```

```
    Dim stDocName As String
    Me.Costo1 = Null           'variable auxiliar de texto.
    Me.Vuti1 = Null           'Costo del bien.
    Me.Salv1 = Null           'Vida útil
    Me.Deprac1 = Null         'Valor de salvamento
    stDocName = "qryDelDepr1" 'Depreciación
    DoCmd.OpenQuery stDocName, acNormal, acEdit 'Query que borra la tabla
    Me.Refresh                'Ejecuta el comando de borrar datos.
                                'Refresca la pantalla mostrando la subforma y
                                'datos borrados.
```

```
End Sub
```

***Borra la tabla de la depreciación de doble declinación.**

```
Private Sub Command31_Click()
```

```
    Dim stDocName As String
    Me.Costo2 = Null           'variable auxiliar de texto.
    Me.Vuti2 = Null           'Costo del bien.
    Me.Salv2 = Null           'Vida útil
    Me.Deprac2 = Null         'Valor de salvamento
    Me.fact2 = Null           'Depreciación
    stDocName = "qryDelDepr2" 'Factor de declinación que se emplea
    DoCmd.OpenQuery stDocName, acNormal, acEdit 'Query que borra la tabla
    Me.Refresh                'Ejecuta el comando de borrar datos.
                                'Refresca la pantalla mostrando la subforma y
                                'datos borrados.
```

```
End Sub
```

4.6.9 Tasa Interna de Retorno.

Tasa Interna de Retorno

Instrucciones/Ayuda

TECLAR los flujos de efectivo ó traer un archivo
 [m]T[1]
 La subforma de captura de flujos se formatea en esta área. Se puede agrandar ó achicar según se necesite.

La tabla tbITIR sólo tiene un campo llamado "Valor"

Form

Form Header

Tecler los Datos

Detail

Valor

Form Footer

SELECT (tbITIR...

forma en diseño.

Option Compare Database

Instrucciones/Ayuda

TIRgral

Calcular T.I.R.

TIR: Unbound

Comandos de la forma

Borrar Pantalla

TIRArch

Archivo: Unbound Traer Archivo

Guardar Datos en Archivo

Archivo: Unbound Borrar Archivo

"Pantallas de ayuda. Instrucción que se cambia en la subrutina de carga de pantalla.
 stLinkCriteria = "[Forma]=" & "" & Me![TIRgral] & ""
 stLinkCriteria = "[Forma]=" & "" & Me![TIRArch] & ""
 TIRgral, TIRArch, son las variables auxiliares definidas para cargar las formas de ayuda.

"Crea la Tasa Interna de retorno leyendo los datos de la tabla de captura. Máximo no. de flujos = 1000.

```
Private Sub Command25_Click()
    Dim Valor(1000) As Double, i As Integer, inversion, ingresos As Double 'Variables auxiliares.
    Dim dbs As DAO.Database 'Declara una base de datos (EmCalc)
    Dim rs As DAO.Recordset 'Declara un archivo (empleará una tabla)

    i = DCount("[Valor]", "tbITIR") 'Verifica que haya flujos a utilizar
    If i < 2 Then
        MsgBox "***Error. No existen datos o son insuficientes. Al menos debe haber 2: uno positivo y uno negativo"
        Exit Sub
    End If

    ContNulls = 0
    Set dbs = CurrentDb 'Se verifica que no haya datos omitidos (que aparezcan en blanco).
    Set rs = dbs.OpenRecordset("tbITIR") 'Se conecta a EmCalc
    'Abre la tabla de datos como archivo.

    rs.MoveFirst 'Lee el primer registro.
    Do While Not rs.EOF 'Ejecuta el ciclo mientras haya registros en el archivo.
        If IsNull(rs!Pago) Or IsNull(rs!Valor) Then ContNulls = ContNulls + 1 'Encuentra valores omitidos.
        rs.MoveNext 'Lee el siguiente registro del archivo.
    Loop
End Sub
```

```

Loop                                     *Cierra la estructura Do While.
rs.Close                                 *Cierra el archivo.

If ContNulls > 0 Then                    *Verifica si encontró valores omitidos.
MsgBox ""**ERROR**". Existen: " & ContNulls & " valores en blanco. Borre los renglones o reemplácelos por
ceros"
Exit Sub                                 *termina la subrutina por haber encontrado errores.
End If                                   *Cierra la estructura IF.

Set dbs = CurrentDb                      *Asigna variable de base de datos a EMCalc
Set rs = dbs.OpenRecordset("tblTIR")     *Asigna tabla tblTIR como archivo a usar
rs.MoveFirst                             *Lee el primer registro de la tabla de captura

With rs                                  *Indica que usará los campos del registro de tbl TIR
Do While Not rs.EOF                       *Ciclo que se repite para todos los registros
i = i + 1                                 *incrementa el contador
If i > 1000 Then                          *Verifica que no se exceda el límite de flujos declarado
MsgBox "El no. de flujos excede el máximo permitido de 1000."
Exit Sub
End If

If IsNull(!Valor) Then                   *Validación de cada uno de los flujos
MsgBox ("Error. Use ceros en lugar de 'Núlos'")
Exit Sub
End If
If !Valor = "" Then
MsgBox ("Error. Use ceros en lugar de 'Espacios'")
Exit Sub
End If
Valor(i) = !Valor                         *Guarda los flujos en un arreglo
If Valor(i) < 0 Then
inversion = inversion + Valor(i)         *Crea cálculos de chequeo: inversión
Else
ingresos = ingresos + Valor(i)          *Crea cálculos de chequeo: retornos
End If
.MoveNext
Loop                                     *Lee el siguiente registro de datos
End With                                 *Cierra ciclo de lectura de datos
                                           *Cierra cláusula With

If ingresos <= Abs(inversion) Then       *Verifica que haya suficientes retornos.
MsgBox ""**Error. Retornos de Capital no cubren la Inversión."
Exit Sub
End If

Me.TIR = IRR(Valor())                    *Calcula la tasa interna de retorno

rs.Close                                 *Cierra el archivo de flujos.
dbs.Close                                 *Cierra la base de datos
Set dbs = Nothing                         *Desasigna la variable de base de datos
End Sub                                   *Termina la subrutina.

```

Borra la tabla de captura de datos "limpiando" la subforma de captura.

```
Private Sub Command26_Click()  
    Me.TIR = Null                'Tasa Interna de Retorno  
    Me.NombArch = Null          'Nombre del archivo de datos a guardar (caja-combo)  
    Me.Datos = Null            'Nombre del archivo de datos a traer (caja-combo)  
    Dim stDocName As String  
  
    stDocName = "qry01DelTir"    'Query que borra la tabla de captura  
    DoCmd.OpenQuery stDocName, acNormal, acEdit 'Borra la tabla de captura  
    Me.Refresh                  'Muestra la pantalla ya "limpia".  
End Sub                        'termina el código
```

Se guardan los flujos que existen en el área de captura.

```
Private Sub Command28_Click()  
    a = GuardaTbl(3)  
End Sub
```

***Borra permanentemente un archivo de flujos.**

```
Private Sub Command38_Click()  
    DoCmd.DeleteObject acTable, Me.NombArch 'NombArch tiene el nombre del archivo a borrar.  
End Sub
```

***Esta subrutina se corre automáticamente al abrir la forma. Borra la tabla que alimenta a la subforma de captura.**

```
Private Sub Form_Open(Cancel As Integer)  
    Dim stDocName As String  
  
    stDocName = "qry01DelTir"    'Query que borra la subforma de captura.  
    DoCmd.OpenQuery stDocName, acNormal, acEdit 'Se ejecuta el query  
End Sub
```

***Esta subrutina extrae una copia del archivo de datos deseado y lo pega en el área de captura de datos.**

```
Private Sub TraeDat_Click()  
    Dim a, stDocName As String  
    If IsNull(Me.Datos) Then Exit Sub 'Verifica que haya un nombre de archivo válido.  
    If (Me.Datos = "") Then Exit Sub  
  
    stDocName = "qry01DelTir"    'Antes de traer los nuevos datos, se cerciora que no  
    DoCmd.OpenQuery stDocName, acNormal, acEdit 'existan ningunos, porque de otra forma se agregarían  
    a = TraeTbl(Me.Datos, 3)     'al final.  
End Sub
```

4.7 Utilerías/Solución de Problemas.

Option Compare Database

Option Explicit

Private Sub Command48_Click()

If IsNull(Me.ArchivoABorrar) Then

 MsgBox "Acción de borrar archivo cancelada."

Exit Sub

End If

'Borra archivos de datos de regresión.

'Verifica que haya un nombre de archivo a borrar

'Pide verificación al usuario para borrar el archivo

If MsgBox("Borrar archivo " & Me.ArchivoABorrar & "?", vbYesNoCancel) <> vbYes Then

 MsgBox "Acción de borrar archivo cancelada."

'El usuario cancela la operación

Exit Sub

End If

DoCmd.DeleteObject acTable, Me.ArchivoABorrar

'Se ejecuta el comando de borrado de archivo

Me.Refresh

'El archivo borrado ya no aparecerá en la caja-combo

End Sub

***Proporciona un ejemplo de como se debe ver el archivo de Excel a importar para Análisis de datos**

Private Sub Command11_Click()

 Dim stDocName As String

 Dim stLinkCriteria As String

 stDocName = "EmCalcDAT"

'Nombre del archivo a importar

 DoCmd.OpenForm stDocName, acFormDS

End Sub

***Proporciona un ejemplo de como se debe ver el archivo de Excel a importar para Análisis de Regresión**

Private Sub Command12_Click()

 Dim stDocName As String

 Dim stLinkCriteria As String

 stDocName = "EmCalcREG"

 DoCmd.OpenForm stDocName, acFormDS

End Sub

4.7.1 Macros para Importar Archivos.

Importa archivos de Excel, para utilizar en Análisis de Datos de Funciones Estadísticas.

Nombre de la macro: Importar.ImportUSR

OpenQuery, qryVacialmportUSR

Se cerciora que esté vacía la tabla que recibe los datos

TransferSpreadsheet, Transfer Type: Import

Se emite comando para importar archivos de Excel

Table Name: ImportUSR ,

tabla que recibe los datos

File Name: C:\My Documents\EmCalcDAT,

Archivo original de Excel y su ubicación

CopyObject Destination Database:

New Name: USRImport

Se crea una copia de la tabla para que el usuario de

Source Object Type: Table

EmCalc no tenga acceso a la tabla receptora.

Source Object Name: ImportUSR

USRImport es ya un archivo de usuario

OpenQuery: qryVacialmportUSR

Vacía la tabla que recibió los datos.

MsgBox

Emite mensaje de terminación.

Message: Archivo importado como 'UXYImport'. Puede verse en Sección 'Funciones Estadística, Análisis de Regresión'

Title: EmCalc: Importación de Archivo para Análisis de Regresión.

EmCalc: Importación de Archivo para Análisis de Datos

X

Archivo importado como 'USRImport'. Puede verse en Sección 'Funciones Estadística, Análisis de Datos'

OK

Nombre de la macro: Importar.ImportUXY

Openquery: qryUXYVacialmport:

Se cerciora que esté vacía la tabla que recibe los datos

TransferSpreadsheet, Transfer Type: Import

Comando para importar una hoja de cálculo de Excel

Table Name: ImportUXY

Tabla que recibe los datos

File Name: C:\My Documents\EmCalcREG

Documento original en Excel

CopyObject Destination Database:

Crea copia de la tabla que recién recibio los datos

New Name: UXYImport

Nombre del archivo para que el usuario la pueda usar

Source Object Name: ImportUXY

Tabla que recibió los datos

OpenQuery Query Name: qryUXYVacialmport

Vacía la tabla que recibió los datos.

MsgBox

Emite mensaje de terminación.

Message: Archivo importado como 'UXYImport'. Puede verse en Sección 'Funciones Estadística, Análisis de Regresión'

Title: EmCalc: Importación de Archivo para Análisis de Regresión.

EmCalc: Importación de Archivo para Análisis de Regresión

X

Archivo importado como 'UXYImport'. Puede verse en Sección 'Funciones Estadística, Análisis de Regresión'

OK

5. Módulo General de EmCalc (Funciones Creadas).

Un módulo general contiene funciones y subrutinas que se pueden emplear desde cualquier objeto en que Acces permite referenciar funciones y/o subrutinas: formas, reportes, queries, funciones y subrutinas de módulos locales, macros y aún funciones y subrutinas de éste y de otros módulos generales.

Para EmCalc sólo se desarrolló un módulo general y contiene solamente funciones. Una característica de estas funciones es que el usuario no tiene acceso a ellas directamente, sino que al utilizar ciertos procesos, éstos hacen la llamada a las funciones.

El lenguaje utilizado para escribir las subrutinas y funciones es Visual Basic, el cuál viene integrado a Access.

Para su descripción, las funciones se agrupan según su utilización, tál agrupación es irrelevante para su funcionamiento.

Funciones generadas, clasificadas según su utilización, se muestran los argumentos que requieren.

5. Funciones del Módulo General.

5.1. Operaciones y Funciones Básicas.

- 5.1.1. Operacion(Op1, Opr, Op2) As Double
- 5.1.2. PreConst(Funcion, Arg) As Double

Double significa que el resultado se proporciona en punto flotante de 4 bytes.

5.2. Manejo de archivos

- 5.2.1. TraeTbl(tblName, Tipo)
- 5.2.2. GuardaTbl(Tipo)

Estas funciones no generan resultados, ejecutan acciones.

5.3. Valuación de Expresiones para Funciones

- 5.3.1. Equation(strMidEquation, dblValue, ArreglaErr, ErrCero, ErrOverFlow, ErrOtro) As Double
- 5.3.2. StrEcuacion(strEquation, dblValue) as String Esta función genera texto.

5.4. Creación de Puntos automáticamente para graficar

- 5.4.1. CreaSerie(Inicio, Fin, Puntos) As Double
- 5.4.2. CreaSerie2(Inicio, Fin, Incr) As Double

5.5. Funciones del módulo de Estadística

- 5.5.1. AnDatos()
- 5.5.2. Regresion(Grado, Desplegar)
- 5.5.3. NAlatorios(TotNos, LInf, LSup) As Double

Estas funciones generan acciones. Leen una tabla y acorde a parámetros que el usuario proporciona, llenan una tabla con información calculada, para desplegarla.

5.1 Operaciones y Funciones Básicas. Para este módulo se desarrollaron 2 funciones que generan la valuación de las funciones preconstruidas y el resultado de operaciones. Tales funciones son exclusivas de este módulo.

5.1.1 Función Operacion. Genera el resultado de la operación en la pantalla de Operaciones y funciones básicas. Utiliza 3 parámetros: Operando 1, Operador y Operando 2. Analiza el operador por medio de una estructura Select-Case, generando en ese análisis el resultado según la operación indicada.

Ejemplo de su empleo: Resultado = Operacion(3.25/,2.780). Operando 1 = 3.25; Operador = "/"; Operando 2 = 2.780.

Operaciones				
Operando 1 (op1)	Operador	Operando 2 (op2)	<Click>	Resultado <Click>
3.25	/	2.780	=	1.16906474820144
Pasar resultado a Operando 1	+	Suma		
	-	Resta		
	*	Multiplicación		
	/	División		
a * b%; 50 % 5 = (50 * 5%) = 2.5				

Function Operacion(Op1, Opr, Op2) As Double

On Error GoTo Err_Operacion_Click

Dim Operando1, Operando2, Result1 As Double

Dim TodoOK As Boolean

TodoOK = True

If IsNull(Op1) Or IsNull(Op2) Or IsNull(Opr) Then

MsgBox ("Error: especifique los operandos y el operador")

TodoOK = False

End If

If Not IsNumeric(Op1) Or Not IsNumeric(Op2) Then

MsgBox ("Los operandos deben ser numéricos. Verifique que tengan valores o 0 (cero)")

TodoOK = False

End If

If TodoOK Then

Operando1 = Op1

Operando2 = Op2

Select Case Opr

Case "+": Result1 = Val(Op1) + Val(Op2)

Case "-": Result1 = Op1 - Op2

Case "*": Result1 = Op1 * Op2

Case "/":

If Op2 = 0 Then

MsgBox ("Division por cero.")

Result1 = 0

TodoOK = False

Else

Result1 = Op1 / Op2

End If

Case "^": Result1 = Op1 ^ Op2

Case "%":

If Op2 = 0 Then

MsgBox ("Division por cero.")

Result1 = 0

TodoOK = False

Else

*Parámetros: Operando 1, Operador y Operando 2

*Se pide que el sistema Access atrape errores.

*Se declaran variables auxiliares.

*TodoOK indicará si hay errores o no.

*TodoOK Se inicializa a "No hay errores".

*Se validan los valores de los parámetros.

*Si hay error se emite el mensaje de error y se

prende el switch de error.

*Se validan que los parámetros sean numéricos, si hay

que tengan valores o 0 (cero)"

*error, se emite mensaje y se prende switch de error.

*Si no hubo error se procede a usar las variables

*auxiliares y examinar el valor del operador por medio

* de una estructura Select-Case:

*Suma: la función Val asegura usar valores numéricos.

*Sustracción.

*Multiplicación.

*En la división se verifica que el divisor sea diferente

*de cero, de ser cero se emite mensaje y se prende el

*switch de error.

*No hay error y se calcula la división.

*Exponenciación.

*El operador % genera una división y por eso se válida

*El valor del divisor (Operando 2).

*En caso de error se emite mensaje y se prende el

*switch de error.

```

Result1 = Op1 / Op2 * 100
End If
Case "@": Result1 = Op1 * Op2 / 100
Case "Lga":
    If Op1 = 0 Or Op2 = 0 Then
        MsgBox ("Error. Base y/o argumento 0.")
        Exit Function
    Else: Result1 = Log(Op2) / Log(Op1)
    End If
Case "Pi": Result1 = 3.14159265358979
Case "e": Result1 = 2.71828182845905
End Select
End If

Operacion = Result1

If TodoOK Then GoTo Fin_Operacion
Err_Operacion_Click:
    MsgBox Err.Description
Fin_Operacion:
End Function

```

*No hay error y se crea el porcentaje.
*Op2 se puede considerar como un porcentaje.
*Logaritmo base "a". No está definido logaritmo de 0, de modo que se válida que Op1 y Op2 sean > 0.
*Se crea el logaritmo de Op2 en base Op1.
*El valor de Pi se transfiere al resultado.
*El valor de e se transfiere al resultado.
*Cierra la estructura Select-Case.
*Cierra la estructura IF de "no hay error"
*Se asigna el valor a regresar
*Si no hubo error, termina la función.
*Si hubo error detectado por Access, se emite el mensaje de error correspondiente.
*Cierra el código de la función.

5.1.2 Función PreConst. El código genera el resultado de la función en la pantalla de Operaciones y funciones básicas. Utiliza 2 parámetros: la función y su argumento. Sólo se utilizan funciones con argumento numérico. Analiza la función por medio de una estructura Select-Case, generando en ese análisis el resultado según la función indicada. Ejemplo de su empleo: Resultado = PreConst(Log,3.75). Función= Log; Argumento = 3.75.

Funciones Preconstruidas		
Función	Argumento (arg)	Resultado
Log	3.75	1.32175583998232
Log	Log natural(arg)	
Exp	e^arg (arg <= 709.782712893)	
Sin	Sen(arg) {arg en radianes}	
Cos	Cos(arg)	
Abs	Abs(arg)	
Atn	Arco Tan(arg) {result en radianes}	
Fix	parte entera(arg). Ej Fix(-8.4)=-8; Fix(8.4)=9	
Int	parte entera(b). Ej Int(-8.4)=-9; Int(8.4)=8	

```

Function PreConst(Funcion, Arg) As Double
On Error GoTo Err_PreConst_Click
Dim Arg1, Result1 As Double
Dim TodoOK As Boolean
TodoOK = True
If IsNull(Funcion) Then
    MsgBox ("Error: especifique la Función.")
    TodoOK = False
End If
If IsNull(Arg) Then
    MsgBox ("Error: argumento no especificado.")
    TodoOK = False

```

*Sus parámetros son la función y el argumento.
*Se pide que el sistema Access atrape errores.
*Se declaran variables auxiliares.
*TodoOK indicará si hay errores o no.
*TodoOK Se inicializa a "No hay errores".
*Se valida el valor de la función.
*Si hay error se emite el mensaje de error y se prende el switch de error.
*Se valida el valor del argumento.
*Si hay error se emite el mensaje de error y se prende el switch de error.

```

End If
If Not IsNumeric(Arg) Then
    MsgBox ("El argumento debe ser numérico. Verifique que tenga valor o 0 (cero)")
    TodoOK = False
End If

If TodoOK Then
    Arg1 = Arg
    Select Case Funcion
        Case "1/X":
            If Arg1 = 0 Then
                MsgBox ("Error: argumento 0. No válido en esta función")
                GoTo Fin_PreConst
            Else: Result1 = 1 / Arg1
        End If
        Case "Log":
            If Arg1 = 0 Then
                MsgBox ("Error: argumento 0. No válido en esta función")
                GoTo Fin_PreConst
            Else: Result1 = Log(Arg1)
        End If
        Case "Exp": Result1 = Exp(Arg1)
        Case "Sin": Result1 = Sin(Arg1)
        Case "Cos": Result1 = Cos(Arg1)
        Case "Abs": Result1 = Abs(Arg1)
        Case "Atn": Result1 = Atn(Arg1)
        Case "Fix": Result1 = Fix(Arg1)
        Case "Int": Result1 = Int(Arg1)
        Case "Rnd":
            Randomize
            Result1 = Rnd(Arg1)
        Case "Sgn": Result1 = Sgn(Arg1)
        Case "Tan": Result1 = Tan(Arg1)
        Case "Sqr":
            If Arg1 < 0 Then
                MsgBox ("Error: El argumento debe ser mayor o igual a 0")
                Result1 = 0
                TodoOK = False
            Else:
                Result1 = Sqr(Arg1)
        End If
        Case "GaR": Result1 = Arg1 * 3.14159265358979 / 180
        Case "RaG": Result1 = Arg1 * 180 / 3.14159265358979
        Case "Pi": Result1 = 3.14159265358979
        Case "e": Result1 = 2.71828182845905
        Case Else:
            MsgBox "Función no disponible" & Funcion & "(" & Arg & ")"
    End Select
End If

PreConst = Result1

If TodoOK Then GoTo Fin_PreConst
Err_PreConst_Click:
    MsgBox Err.Description
Fin_PreConst:
End Function

```

*Se valida que el argumento sea numérico.

*Se valida que el argumento sea numérico.

*No se encontró error y se procede usar variable

*auxiliar y analizar la función con la estructura

*Select-Case.

*Inverso multiplicativo, se valida que $arg < 0$,

*si es cero se emite mensaje de error.

*El argumento tiene un valor válido.

*Para logaritmo se valida que el argumento sea

*diferente de cero.

*Error: argumento 0. No válido en esta función")

*Log es una función preconstruída de Access.

*Exp, Sin, Cos, Abs, Atn, Fix, Int, Rnd también son

*funciones preconstruídas de Access.

*Para crear un número aleatorio, se inicializa la semilla

*de números aleatorios con el reloj de la computadora y

*se crea el número aleatorio con el argumento.

*Función Signo, Tan, Sqr (raíz cuadrada) también son

*funciones preconstruídas.

*La raíz de un número menor que cero no está definida.

*Error: El argumento debe ser mayor o igual a 0")

Result1 = 0

TodoOK = False

*Se crea la raíz cuadrada.

*Grados a Radianes.

*Radianes a Grados.

*El valor de Pi se transfiere al resultado de la función.

*El valor de Pi se transfiere al resultado de la función.

*Evento imposible, la caja-combo no permite

* & Arg & ")" *seleccionar funciones no definidas aquí.

*Termina estructura Select-Case

*Se asigna el valor a regresar.

*Si no hubo error termina

* Si Access detectó algún error, emite mensaje y

*termina la función.

*Fin del código.

5.2 Manejo de archivos. EmCalc provee medios para almacenar, acceder y borrar información simulando archivos de datos, los cuáles son guardados localmente dentro de la base de datos EmCalc, es decir, los archivos creados no se almacenan directamente en un directorio de la computadora, sino en una tabla interna. (Véase Utilerías y Solución de problemas para interactuar con Excel). Se describen aquí el acceso y guardado de datos, el borrado de archivos se realiza en los módulos locales de las pantallas que utilizan tal función de borrado.

5.2.1 Función TraeTbl. Esta función permite el acceso de datos almacenados en un archivo. tblName es una variable con el nombre del archivo a traer y es proporcionado en una caja-combo ligada por medio de un query a la tabla del sistema MsysObjects, que contiene entre otros, los nombres de tablas. El query se crea ex-profeso para la sección en que se pide el archivo y filtra los archivos según su tipo:

Tipo 1: Funciones Estadísticas. Análisis de Datos; accesa archivos con el prefijo "USR" (1 sola columna).

Tipo 2: Funciones Estadísticas. Análisis de Regresión; accesa archivos con el prefijo "UXY" (6 columnas).

Tipo 3: Funciones Financieras. T.I.R.; accesa archivos con el prefijo "TIR" (1 columna).

Ejemplos de su uso (se hace una asignación dummy, la variable a no requiere ningún valor; el primer parámetro es una caja-combo ligada a la tabla MsysObjects que contiene el nombre del archivo a traer):

- a = TraeTbl(Mc.Datos, 1) ...Módulo de Análisis de Datos.
- a = TraeTbl(Mc.Arch, 2) ...Módulo de Regresión.
- a = TraeTbl(Mc.Datos, 3) ...Módulo de Tasa Interna de Retorno.

```

Function TraeTbl(tblName, Tipo)
If tblName = "" Then
    MsgBox ("Accion de traer datos cancelada")
Exit Function
End If

On Error Resume Next

Select Case Tipo

'En los 3 casos realiza las mismas funciones pero, con objetos ad-hoc a cada sección.

Case 1: DoCmd.DeleteObject acTable, "tblAnDatFija"
DoCmd.CopyObject , "tblAnDatFija", acTable, tblName

stDocName = "qry001_02AppendAnDat"
DoCmd.OpenQuery stDocName, acNormal, acEdit

Case 2: DoCmd.DeleteObject acTable, "tblAnRegFija"
DoCmd.CopyObject , "tblAnRegFija", acTable, tblName
stDocName = "qry01_7AppRegXY"
DoCmd.OpenQuery stDocName, acNormal, acEdit

Case 3: DoCmd.DeleteObject acTable, "tblTirFija"
DoCmd.CopyObject , "tblTirFija", acTable, tblName
stDocName = "qry01AppTirFija"
DoCmd.OpenQuery stDocName, acNormal, acEdit

End Select

If Err.Number = 2501 Then MsgBox ("Accion de traer datos cancelada")
fin_funcion:
End Function
    
```

5.2.2 Función GuardaTbl. Esta función permite guardar datos en un archivo. El parámetro "tipo" indica que pantalla emite el comando:

Tipo 1: Funciones Estadísticas. Análisis de Datos. 1 sola columna.

Tipo 2: Funciones Estadísticas. Análisis de Regresión. Varias columnas (6)

Tipo 3: Funciones Financieras. T.I.R. 1 sola columna.

Ejemplos de su uso (se hace una asignación dummy, la variable **a** no requiere ningún valor):

- a = GuardaTbl (Me.Datos, 1) ...Módulo de Análisis de Datos.

- a = GuardaTbl (Me.Arch, 2) ...Módulo de Regresión

- a = GuardaTbl (Me.Datos, 3) ...Módulo de Tasa Interna de Retorno.

El nombre del archivo se solicita en línea, al hacer click en la caja-comando que guarda los datos.

Function GuardaTbl(Tipo)

Dim tblName As String

Select Case Tipo

Case 1: tblName = InputBox("Nombre de Archivo de Datos? -15 caracteres máx-. Se le agregará 'USR' al inicio del nombre.")

Case 2: tblName = InputBox("Nombre de Archivo de Datos? -15 caracteres máx-. Se le agregará 'UXY' al inicio del nombre.")

Case 3: tblName = InputBox("Nombre de Archivo de Datos? -15 caracteres máx-. Se le agregará 'TIR' al inicio del nombre.")

End Select

If tblName = "" Then 'El usuario puede cancelar la acción al usar un nombre en blanco.

MsgBox ("Acción de guardar datos cancelada")

Exit Function

End If

If Len(tblName) > 15 Then 'Se verifica que el nombre no exceda la longitud establecida.

MsgBox ("**Error. El nombre excede de 15 caracteres. Acción de guardar datos cancelada.")

Exit Function

End If

On Error Resume Next 'Si hay error, el programa reanuda después del Select-Case.

Select Case Tipo 'Se explora el tipo de archivo

'Los tres casos son idénticos, con los objetos ad-hoc a la sección.

Case 1: tblName = "USR" & tblName
DoCmd.CopyObject , tblName, acTable, "tblAnDat"

'Caso 1: Análisis de Datos.

'Se agrega el prefijo USR al nombre.

'Se crea una copia de la tabla de captura con el nombre solicitado.

Case 2: tblName = "UXY" & tblName
DoCmd.CopyObject , tblName, acTable, "tblAnReg01"

'Caso 2: Análisis de Regresión.

'Se agrega el prefijo UXY al nombre.

'Se crea una copia de la tabla de captura.

Case 3: tblName = "TIR" & tblName
DoCmd.CopyObject , tblName, acTable, "tblTIR"

'Caso 3: Tasa Interna de Retorno.

'Se agrega el prefijo TIR al nombre.

'Se crea una copia de la tabla de captura.

End Select

If Err.Number = 2501 Then MsgBox ("Acción de guardar datos cancelada")

End Function

5.3 Valuación de Expresiones para Funciones. Access provee la función Eval para valorar expresiones, la cuál utiliza valores, operadores y funciones preconstruidas de Access. Aprovechando esta facilidad, se desarrollaron una serie de pasos que permiten reemplazar el carácter "X" por su valor y así lograr la valuación de una expresión escrita en términos de X (Módulo de Valuación de Funciones F(X)). De un modo similar, se desarrolla la facilidad para reemplazar el carácter "Y" por su valor y lograr la valuación de funciones en términos de X y Y (Módulo de Valuación de Funciones F(X,Y)), dando la facilidad adicional de valorar funciones escritas en términos de Y solamente o de "X" solamente, en este último módulo. En cada módulo se proporciona la expresión resultante al sustituir la o las variables por su valor.

5.3.1 Función Equation. La función realiza las siguientes acciones:

- Debido a que una expresión para una función puede ser escrita en términos de X y/o Y, primeramente se verifica que la función esté escrita en términos de X, transformando todas las Y's que pueda haber, a X. En los módulos de funciones F(X) y F(X,Y) se complementan las subrutinas con precauciones para evitar errores.
- Se reemplaza el carácter "X" por el valor dado. (En el módulo de funciones F(X,Y), cuando la expresión tiene X's y Y's, se hacen dos llamadas, la primera reemplazando X's por su valor y al quedar la expresión en términos de Y, se hace otra llamada).
- Se reemplaza el resultado por el valor proporcionado. Se valúa la expresión de la función.
- Control de errores. En caso de haber división por cero, overflow ú otro error, se puede o no "arreglar el error" sustituyendo el resultado por el valor que el usuario proporciona para control de errores. En las secciones de valuación de funciones NO se arregla el error, por lo que Access emite un mensaje de error; en las secciones de graficación y regresión, SI se controlan los errores. Esta acción no es controlada por el usuario.
- Ejemplos:
 - Resultado = Equation(Expresión F(X), Valor de X, Arreglar errores: True/False, Resultado en caso de División por cero, Resultado en caso de Overflow, Resultado en caso de Otro error).
 - Ejemplo: Y = Equation("3X+X^2", 2.7, Yes, 0, 0, 0).
 - Expresión a valorar es: 3X+X^2, cuando X = 2.7; en caso de cualquier error (Div. por cero, Overflow, ú otro), el resultado se hace cero.
 - Ejemplo: Y = Equation("3X+X^2", 2.7, No, 0, 0, 0)
 - Misma función y valor, sólo que si hay algún error, Access emite un mensaje de error.

Function Equation(strMidEquation, dblValue, ArreglaErr, ErrCero, ErrOverFlow, ErrOtro) As Double	
Dim i As Long	'Variable auxiliar como índice.
Dim lngL As Long	'Auxiliar para contar caracteres de la expresión.
Dim strNewEquation, strEquation As String	'Auxiliares para almacenar la expresión.
On Error GoTo Mensajes_Error	'Errores no atrapados por EmCalc, sino por Access.
If IsNull(strMidEquation) Or IsNull(dblValue) Then	'Se verifica que haya una expresión de función.
Equation = 0	'En caso de no haber función, el resultado se hace cero
Exit Function	'y se sale de la función.
End If	
strMidEquation = UCase(strMidEquation)	'Todos los caracteres se hacen mayúsculas.
lngL = Len(strMidEquation)	'Se toma la longitud de la expresión.
strEquation = ""	'Se inicializa variable auxiliar.
For i = 1 To lngL 'Convierte Y's a X's	'Se recorre la expresión, carácter por carácter para
If Mid(strMidEquation, i, 1) = "Y" Then	'reemplazar Y's por X's.
strEquation = strEquation + "X"	'Se encuentra Y y se sustituye por X.
Else	
strEquation = strEquation + Mid(strMidEquation, i, 1)	'Mid extrae caracteres del texto, en este caso 1, el
End If	'i-ésimo del recorrido.
Next i	'Cierra ciclo For.
lngL = Len(strEquation)	'Se toma la longitud de la nueva expresión.
strNewEquation = ""	'Se inicializa nueva variable auxiliar.
For i = 1 To lngL 'Se substituye X por su valor	'Se recorre la expresión, carácter por carácter para
If Mid(strEquation, i, 1) = "X" Then	'reemplazar X's por el valor proporcionado.
If i >= 2 Then	'Se hace una salvedad para no reemplazar X cuando se

```

If Mid(strEquation, i - 1, 3) <> "EXP".Then      'trata de la función exponencial.
strNewEquation = strNewEquation + "(" + Trim(Str(dblValue)) + ")"      'El valor de X se pone entre ().
Else
  strNewEquation = strNewEquation + Mid(strEquation, i, 1)      'X es de función EXP
End If
Else
  strNewEquation = strNewEquation + "(" + Trim(Str(dblValue)) + ")"      'El valor de X se pone entre ().
End If
Else
  strNewEquation = strNewEquation + Mid(strEquation, i, 1)      'No se encontró X.
End If
Next i      'Termina análisis del carácter i-ésimo.
'Cierra ciclo For.

```

*Se verifica que la expresión resultante (X reemplazado por su valor), no exceda 255 caracteres, para que pueda ser evaluado con la función Eval de Access.

```

lngL = Len(strNewEquation)      'Longitud máxima para poder valuar es 255.

```

```

If lngL <= 255 Then
  Equation = Eval(strNewEquation)      'Se valúa la expresión.

```

```

Else:
  MsgBox ("Expresión muy larga al substituir valor de X. Long. Máx=255. Longitud: " & Str(lngL))
  Equation = 0
End If

```

```

GoTo Salida_Funcion

```

*En esta sección de la función se atrapan ciertos errores al valuar la función, tomándolos de la variable de Access que

*proporciona el código de error: **Err.Number**

*Nota: excepto en las pantallas de valuación de funciones F(X) y F(X,Y), en cualquier otra pantalla donde se usan funciones de usuario, al ocurrir un error al resultado se le asigna el valor dado en control de errores..

Mensajes_Error:

```

If ArreglaErr Then      'Esto es para secciones de graficación y regresión.
  Select Case Err.Number      'Valúa el número de error y dá valor preasignado a la función

```

```

  Case 7: Equation = ErrOverflow

```

```

  Case 11: Equation = ErrCero

```

```

  Case Else: Equation = ErrOtro

```

```

End Select

```

```

Else      'Sólo se atrapan los errores más comunes.

```

```

  Select Case Err.Number      'Evalua el error y envia el mensaje correspondiente

```

```

  Case 5: MsgBox ("Error de Sintáxis: La expresión tiene errores.")

```

```

  Case 7: MsgBox ("Overflow: El resultado es un número muy grande que no puede ser representado.")

```

```

  Case 11: MsgBox ("Error: División por Cero.")

```

```

  Case 13: MsgBox ("Número inválido. Número teclado tiene caracteres no numéricos.")

```

```

  Case 2420: MsgBox ("Error de Sintáxis: Un número está escrito erróneamente, P.Ej. 9.524.5")

```

operadores necesarios, p. ej. 3X debe escribirse 3*X. Posible falta de Paréntesis")

```

  Case 2425: MsgBox ("Error de Sintáxis: La expresión tiene una función preconstruida mal escrita.")

```

operadores necesarios, p. ej. 3X debe escribirse 3*X. Posible falta de Paréntesis")

```

  Case 2435: MsgBox ("Error de Sintáxis: Falta un paréntesis que abre una expresion, o está cerrando paréntesis de más.")

```

```

  Case 2436: MsgBox ("Error de Sintáxis: Falta un paréntesis que cierre la expresión.")

```

```

  Case Else: MsgBox ("Error: " & Err.Number & ": " & Err.Description & ". Posible error de sintáxis.")

```

```

End Select

```

```

End If

```

```

Salida_Funcion:

```

```

End Funcion

```


5.3.2 Función StrEcuación. Esta función proporciona la expresión de la función F(X) con X reemplazado por sus valores. En el caso de F(X,Y), también se realiza la misma acción, comandada desde el módulo local, haciendo dos pasadas a esta función, la primera para reemplazar las X's por su valor y la segunda para reemplazar las Y's por su valor.

- Ejemplo del módulo de Valuación de funciones F(X):
 - TraducciónExpresion = StrEcuacion("Sin(X)+3*X^2+7", 7.75); crea:
 - TraducciónExpresion = Sin((7.75)) + 3*(7.75)^2+7
- Desde el módulo de valuación de funciones F(X,Y) se produce remplazo de la expresión en dos pasadas, por ejemplo,
 - TraducciónExpresion = Sin(X) * X + Cos(Y) * Y, donde X = 1.85, Y = 2.77; queda:
 - 1er. pasada, reemplaza X's: TraducciónExpresion = Sin(1.85) * 1.85 + Cos(Y) * Y
 - 2a. pasada, reemplaza Y's" TraducciónExpresion = Sin(1.85) * 1.85 + Cos(2.77) * 2.77

Function StrEcuacion(strEquation, dblValue) As String	
On Error GoTo Reporta_Error	'Se pide a Acces que atrape errores.
Dim i As Long	'Variable auxiliar para indices.
Dim lngL As Long	'Variable para medir longitud de expresiones.
Dim strNewEquation As String	'Variable auxiliar para la expresión de la función.
If Not IsNull(strEquation) Then	'Se verifica que exista una expresión.
If Not IsNull(dblValue) Then	'Se verifica que haya un valor para X.
strEquation = UCase(strEquation)	'Los caracteres se hacen mayúsculas.
lngL = Len(strEquation)	'Se mide la longitud de la expresión de F(X).
strNewEquation = ""	'Se inicializa variable auxiliar.
For i = 1 To lngL	'Se recorre la expresión carácter por carácter para
If Mid(strEquation, i, 1) = "X" Then	'reemplazar X por su valor.
If i >= 2 Then	'Se hace una excepción cuando se encuentra X en el
If Mid(strEquation, i - 1, 3) <> "EXP" Then	'nombre de la función EXP, exponencial.
strNewEquation = strNewEquation + "(" + Trim(Str(dblValue)) + ")"	'El valor de X se pone entre
Else	'paréntesis.
strNewEquation = strNewEquation + Mid(strEquation, i, 1)	'Aquí X está en EXP.
End If	'Cierra IF que checa si X está en EXP.
Else	'Aquí X es el 1er. Carácter, se reemplaza por su valor.
strNewEquation = strNewEquation + "(" + Trim(Str(dblValue)) + ")"	
End If	'Cierra IF que checa desde el 2o. carácter.
Else	'Aquí no se encontró X. se usa el carácter i-ésimo de la
strNewEquation = strNewEquation + Mid(strEquation, i, 1)	'de la expresión.
End If	'Cierra IF que analiza carácter por carácter.
Next i	'Cierra ciclo For.
StrEcuacion = strNewEquation	'Se asigna expresión a regresar a la pantalla.
Else	'parte de If Not IsNull(dblValue) Then...
StrEcuacion = ""	'El argumento es nulo y regresa expresión en blanco.
End If	'Cierra If Not IsNull(dblValue) Then...
End If	'Cierra If Not IsNull(strEquation) Then
GoTo Fin_StrEcuacion	'No hubo errores y termina.
Reporta_Error:	'Hubo errores y se emite el mensaje de error por
MsgBox Err.Description	'parte del sistema.
Fin_StrEcuacion:	'Fin normal.
End Function	'Cierra el código de la función.

5.4 Creación de Puntos automáticamente para graficar. Estas funciones se utilizan en las pantallas de graficación con generación de puntos automáticamente: para una función y para múltiples funciones.

¿ Rango de X a utilizar para la Serie Gráfica:

Inicio:
Fin:

La figura muestra las diferentes opciones que existen en EmCalc para la generación automática de puntos X, así como los parámetros que se requieren. Se especifican los parámetros (Inicio, Fin, No de puntos) o (Inicio, Fin, Incremento). Cuando se oprime una de las cuatro opciones, se considera No. de puntos o Incremento y se descarta el otro parámetro.

¿ Seleccione una de las opciones:

No de puntos: Incremento:

Crear Serie Solamente

Crear Serie Solamente

Crear Serie y Graficar

Crear Serie y Graficar

Las 4 opciones generarán la serie de puntos X.

5.4.1 Función CreaSerie. La siguiente función se utiliza en las secciones de graficación con generación de puntos automáticamente.

- En caso de número inválido de puntos, se generan 10 puntos.
- Se limita la cantidad de puntos a crear, a 1000. Una cantidad mayor de puntos generalmente no se puede apreciar en la pantalla.
- Usa tres parámetros punto inicial, punto final y número de puntos a crear en ese rango.
- Ejemplo: CreaSerie(-10,10,150); aquí X se encuentra en el rango[-10,10] y se generarán 150 puntos intermedios.

Function CreaSerie(Inicio, Fin, Puntos) As Double

Dim dbs As DAO.Database

Dim rs As DAO.Recordset

Dim Incremento As Double

Dim i As Integer

*Se declara una base de datos.

*rs se declara como un archivo

*incremento es una variable auxiliar.

*i se usa como indice.

If Puntos = 0 Then

Puntos = 10

End If

*Si no se especifican puntos, se hacen 10 por default.

Set dbs = CurrentDb

Set rs = dbs.OpenRecordset("tbl1Serie")

If Puntos > 1000 Then Puntos = 1000

Incremento = (Fin - Inicio) / Puntos

*Se establece la conexión a EmCalc

*Se usa la tabla tbl1Serie como archivo para guardar

*los puntos X que se crearán.

*Se calcula el incremento entre puntos.

For i = 0 To Puntos 'Esto es 1 punto mas de los solicitados para evaluar el inicio y el fin.

With rs

rs.AddNew

rs!Serie1 = Inicio + i * Incremento

rs.Update

End With

Next

*Se indica que se usarán los campos del archivo.

*Se agrega un registro en blanco al archivo.

*Se guarda punto por punto generado, en el archivo.

*Se guarda el registro con los valores dados.

*Termina cláusula With.

*Cierra ciclo For.

rs.Close

dbs.Close

Set dbs = Nothing

End Function

*Cierra el archivo.

*Cierra la base de datos.

*Se desconecta de EmCalc.

*termina código de la función.

5.4.2 Funcionamiento CreaSerie2. La función se utiliza en las secciones de graficación con generación de puntos automáticamente: para una función y para múltiples funciones.

- En caso de un incremento inválido, se genera de tal forma que proporcione 11 puntos.
- Se limita la cantidad de puntos a crear a 1000. Una cantidad mayor de puntos generalmente no se puede apreciar en la pantalla.
- Usa tres parámetros punto inicial, punto final y número de puntos a crear en ese rango.
- Ejemplo: CreaSerie2(-10,10,0.01); X se encuentra en el rango[-10,10] y se pide que los puntos disten .01, uno del otro, ésto implica que se generarán 2000 puntos, como se rebasa el límite impuesto de 1000; la cantidad de puntos se hace igual a 1000, el incremento también se recalcula para desplegarse apropiadamente.

Function CreaSerie2(Inicio, Fin, Incr) As Double

```

Dim dbs As DAO.Database
Dim rs As DAO.Recordset
Dim i, NPuntos As Integer, Incremento, Puntos, ValX, Rango As Double
'Se declara una base de datos.
'rs se declara como un archivo
'Declaración de variables auxiliares.

Set dbs = CurrentDb
Set rs = dbs.OpenRecordset("tbl1Serie")
'Se establece la conexión a EmCalc
'Se usa la tabla tbl1Serie como archivo para guardar
'los puntos X que se crearán.

Incremento = Incr
If Inicio >= Fin Then
    MsgBox ("Error: Inicio y Fin estan reversados o son iguales.")
    'Puntos del rango reversados.
    GoTo finCreaSerie2
Else: Rango = Fin - Inicio
    'Se calcula el rango de X a graficar.
End If

If Incremento > 0 And Incremento < Rango Then
    'Válida Incremento con respecto al rango.
    Puntos = Int(Rango) / Incremento
    'Calcula un número entero de puntos a crear.
    If Puntos > 1000 Then
        'Limita a 1000 el número de puntos a crear.
        MsgBox "la función se valorará y graficará en 1000 puntos solamente."
        Puntos = 1000
        Incremento = Rango / 1000
    End If
    NPuntos = Puntos
    'Variable auxiliar para número de puntos.
    ValX = Inicio
    'Variable auxiliar para punto de inicio.
    For i = 0 To NPuntos
        'Generar punto por punto.
        With rs
            'Usará campos del archivo.
            rs.AddNew
            'Agrega un nuevo registro al archivo.
            rs!Serie1 = ValX
            'Asigna punto creado al campo del registro
            rs.Update
            'Guarda el registro con el valor recién dado.
        End With
        'Cierra cláusula With
        ValX = Val(ValX) + Val(Incremento)
        'Genera el punto para el siguiente registro.
    Next
    'Cierra ciclo For.
Else: MsgBox ("Error: Incremento es <=0 o excede el rango de graficacion. " & Incr & Str(NPuntos))
End If

rs.Close
dbs.Close
'Cierra el archivo.
'Cierra la base de datos.
Set dbs = Nothing
'Se desconecta de EmCalc.

finCreaSerie2:
End Function
'termina código de la función.

```

5.5 Funciones del módulo de Estadística. Se desarrollaron dos funciones para crear las tablas que proporcionan la información pertinente a datos de entrada de las secciones de Análisis de Datos y de Análisis de Regresión; una tercera función genera una tabla de números aleatorios de acuerdo a parámetros proporcionados por el usuario.

5.5.1 Función AnDatos. La función se utiliza en la pantalla de Funciones Estadísticas: Análisis de Datos. Genera la información acerca del grupo de datos que se proporciona. Se graban la información a desplegar en una tabla y se muestran por medio de una subforma ligada a dicha tabla.

Nota: la subrutina que llama a esta función, realiza validación adicional sobre la información que proporciona el usuario, para evitar llamar la función en casos innecesarios.

Function AnDatos()

Dim dbs As DAO.Database	'Se declara una base de datos.
Dim rs, rsReg As DAO.Recordset	'Se declaran dos archivos.
Dim Cont, Cont1, Cont2, Puntos, ContNulls As Integer	'Contadores, variables auxiliares, enteras.
Dim RCont, RCont1, RCont2, RMediana, RMediana1, RMediana2, RIntQ, Media, VarM, VarP, DstM, DstP, Suma, Mínimo, Máximo, Rango As Double	'Variables de punto flotante.
Dim Mdn, Mdn1, Mdn2 As Variant	'Variables del tipo mas general
Dim Par, Par1, Par2, IntQuart As Boolean	'Variables booleanas.
ContNulls = 0	'Contador de valores omitidos (nulos o en blanco)
Set dbs = CurrentDb	'Abre la base de datos actual
Set rs = dbs.OpenRecordset("SELECT Valor FROM tblAnDat ORDER BY Valor, Valor")	
rs.MoveFirst	'Accesa el primer registro de datos.
Do While Not rs.EOF	'Realiza el ciclo para todos los registros
If IsNull(rs!Valor) Then ContNulls = ContNulls + 1	'Cuenta los valores inválidos u omitidos.
rs.MoveNext	'Lee el siguiente registro de datos.
Loop	'Cierra la estructura Do While.
rs.Close	'Cierra el archivo de datos.
If ContNulls > 0 Then	'Checa si hubo valores inválidos.
MsgBox "****ERROR****. Existen: " & ContNulls & " valores en blanco. Borre los renglones o reemplácelos por ceros"	
Exit Function	'Termina la función por haber errores.
End If	'Cierra estructura IF.
'La siguiente información se crea con funciones preconstruídas de Access:	
Puntos = DCount("[Valor]", "tblAnDat")	'Número de observaciones.
Suma = DSum("[Valor]", "tblAnDat")	'Suma de las observaciones.
Media = DAvg("[Valor]", "tblAnDat")	'Promedio de las observaciones.
VarP = DVarP("[Valor]", "tblAnDat")	'Varianza Poblacional.
VarM = DVar("[Valor]", "tblAnDat")	'Varianza Muestral.
DstP = Sqr(VarP)	'Desviación Stándard Poblacional.
DstM = Sqr(VarM)	'Desviación Stándard Muestral.
Máximo = DMax("[Valor]", "tblAnDat")	'Máximo de las observaciones
If Puntos < 2 Then Exit Function	'Si no hay puntos suficientes, se termina la función
Set dbs = CurrentDb	'Se conecta a EmCalc como base de datos.
'Se selecciona el campo Valor (datos de captura) de la tabla tblAnDat, ordenado ascendentemente.	
Set rs = dbs.OpenRecordset("SELECT Valor FROM tblAnDat ORDER BY Valor, Valor")	
'Se abre la tabla tblMdn para guardar la información que se calcula.	
Set rsReg = dbs.OpenRecordset("tblMdn")	

*Se generan los datos para el rango intercuartil

RCont = (Puntos + 1) / 2

RCont1 = (Puntos + 3) / 4

RCont2 = (Puntos + 1) - RCont1

Cont = Int(RCont)

Cont1 = Int(RCont1)

Cont2 = Int(RCont2)

If Puntos < 5 Then IntQuart = False

rs.MoveFirst

With rs

Minimo = !Valor

Rango = Maximo - Minimo

Cont1 = Cont1 - 1

Cont2 = Cont2 - 1

Cont = Cont - 1

For lmed = 1 To Cont1

rs.MoveNext

Next lmed

Mdn1 = !Valor

If (Cont1 + 1) <> RCont1 Then

rs.MoveNext

Mdn1 = (Mdn1 + !Valor) / 2

Cont1 = Cont1 + 1

End If

For lmed = (Cont1 + 1) To Cont

rs.MoveNext

Next lmed

Mdn = !Valor

If (Cont + 1) <> RCont Then

rs.MoveNext

Mdn = (Mdn + !Valor) / 2

Cont = Cont + 1

End If

For lmed = (Cont + 1) To Cont2

rs.MoveNext

Next lmed

Mdn2 = !Valor

If (Cont2 + 1) <> RCont2 Then

rs.MoveNext

Mdn2 = (Mdn2 + !Valor) / 2

End If

*Se procede a guardar la información recién calculada en una tabla que se liga a una subforma para desplegarlos.

rsReg.AddNew

rsReg!Nombre = "Número de Observaciones"

rsReg!dato = Puntos

rsReg.Update

*Rcont, Rcont1, Rcont2 se utilizan para conocer el número de puntos en cada cuartil.

*Se encuentra el entero.

*Menos de 5 puntos? No se calcula el rango intercuartil.

*Se usarán campos de este registro:

*El primer dato que se lee es el mínimo, se asigna a su variable.

*Ahora ya se puede calcular el rango.

*Ya se leyó el primer registro, por lo tanto se resta 1 a los contadores.

*Busca Mediana1, la mediana de la primera mitad de observaciones.

*Avanza al siguiente registro.

*Cierra For.

*Encuentra la posible mediana,

*Si Cont1 (entero) no coincide con Rcont1 (Real), implica que debe leer otro registro y obtener el promedio de ambos.

*El cuartil tiene un número par de observaciones.

*Se avanza el contador porque se leyó un registro extra.

*Busca Mediana, la mediana de los datos en general.

*Avanza al siguiente registro.

*Encuentra la posible mediana,

*Si no coinciden, implica que el número de observaciones es par y hay que leer otro registro para obtener un promedio de ambas observaciones.

*Se calcula la mediana de los datos.

*Se avanza el contador de registros.

*Busca Mediana3, la mediana de la segunda mitad de observaciones.

*Avanza al siguiente registro

*Encuentra la posible mediana,

*Si no coinciden, hay que obtener el siguiente punto y promediarlo con el ya obtenido para encontrar la mediana.

*Esta instrucción agrega un registro al archivo.

*campo Nombre de la tabla tblMdn.

*campo Dato de la tabla tblMdn.

*Se guarda el registro.

rsReg.AddNew
rsReg!Nombre = "Suma"
rsReg!dato = Suma
rsReg.Update

*Suma de la observaciones.

rsReg.AddNew
rsReg!Nombre = "Media"
rsReg!dato = Media
rsReg.Update

*Media de las observaciones.

rsReg.AddNew
rsReg!Nombre = "Varianza Poblacional (N)"
rsReg!dato = VarP
rsReg.Update

*Varianza Poblacional

rsReg.AddNew
rsReg!Nombre = "Varianza Muestral (n)"
rsReg!dato = VarM
rsReg.Update

*Varianza Muestral

rsReg.AddNew
rsReg!Nombre = "Desv. Standard Pob. (N)"
rsReg!dato = DstP
rsReg.Update

*Desviación Standard Poblacional

rsReg.AddNew
rsReg!Nombre = "Desv. Standard muestra (n)"
rsReg!dato = DstM
rsReg.Update

*Desviación Standard Poblacional

rsReg.AddNew
rsReg!Nombre = "Mínimo"
rsReg!dato = Minimo
rsReg.Update

*Mínimo

rsReg.AddNew
rsReg!Nombre = "Máximo"
rsReg!dato = Maximo
rsReg.Update

*Máximo

rsReg.AddNew
rsReg!Nombre = "Rango"
rsReg!dato = Rango
rsReg.Update

*Rango

rsReg.AddNew
rsReg!Nombre = "Mediana"
rsReg!dato = Mdn
rsReg.Update

*Mediana

rsReg.AddNew 'Guarda Mediana 1
rsReg!Nombre = "Mediana 1"
rsReg!dato = Mdn1
rsReg.Update

*Mediana 1.

rsReg.AddNew 'Guarda Mediana 3

*Mediana 3.

```
rsReg.Nombre = "Mediana 3"  
rsReg.Idato = Mdn2  
rsReg.Update
```

```
RIntQ = Mdn2 - Mdn1  
rsReg.AddNew
```

```
rsReg.Nombre = "Rango Intercuartil"  
rsReg.Idato = RIntQ  
rsReg.Update
```

```
End With
```

```
Medianax = Mdn
```

```
rs.Close
```

```
rsReg.Close
```

```
dbms.Close
```

```
Set dbs = Nothing
```

```
End Function
```

*Rango Intercuartil

*Asignación Dummy

*Cierra archivos empleados.

*Cierra la base de datos.

*Desconecta la base de datos

5.5.2 Función Regresión. La función genera la tabla de información del módulo de regresión. Utiliza una subforma de captura de datos y otra subforma para presentar información calculada, ambas tablas están ligadas a sendas tablas. Los parámetros que utiliza son el grado de la ecuación de ajuste (grado 1: recta $y = a + bX + o$, grado 2, curva: $a + bX + cX^2$). La curva que se genera es por el método de mínimos cuadrados. Se generan también los puntos de ajuste y los residuales. Se ofrece la capacidad de desplegar datos sumariados o detallados, usando una variable-chequeo.

Function Regresion(Grado, Desplegar)

```
Dim dbs As DAO.Database
```

```
Dim rs, rsReg, RsNvo As DAO.Recordset
```

```
Dim n As Integer
```

```
Dim SUMX, SUMY, SUMX2, SUMXY, SUMX2Y, SUMX3, SUMX4, a, b, Xavg, Yavg As Double
```

```
Dim a0, a1, a2, a3, b0, b1, b2, b3, c0, c1, c2, c3, d0, d1, d2, d3 As Double
```

```
Dim SumResid2, Aux, StdErr, VarErr, r, CovXY, VarX, VarY, DstdX, DstdY, XtransfX, XtransfY As Double
```

```
Dim Modelo As String, SwNulls As Boolean
```

*Almacenará la expresión del

módulo de ajuste.

*Declaración de una base de datos.

*Declaración de archivos a usar.

*Declaración de variables auxiliares.

*Verifica que haya datos suficientes para realizar la regresión.

```
n = DCount("[X]", "tblAnReg01")
```

*Se cuenta el número de observaciones.

```
If n <= 2 Then
```

*Si no hay observaciones suficientes, se termina la

```
MsgBox "**Error. Datos insuficientes o no hay datos a analizar."
```

*función.

```
Exit Function
```

```
End If
```

*Desplegar los cálculos detallados para las regresiones $\Sigma(X_i)$, $\Sigma(X-X_i)^2$, etc ?

```
If IsNull(Desplegar) Then Desplegar = False
```

*Se chequea el valor de variable-chequeo.

```
Set dbs = CurrentDb
```

*Asigna base de datos EmCAIc.

*Se trabaja con las transformadas de X y Y, en caso que no haya transformaciones $\text{transfX} = X$ y $\text{transfY} = Y$.

```
Set rs = dbs.OpenRecordset("SELECT,TransfX,TransfY,Yajuste,Residual FROM tblAnReg01")
```

```
Set rsReg = dbs.OpenRecordset("tblAnReg02")
```

*Tabla de presentación de información.

SWNulls = false

rs.MoveFirst

With rs

Do While Not rs.EOF

SUMX = SUMX + Nz(!TransfX,0)

SUMY = SUMY + Nz(!TransfY,0)

If IsNull(!transfx) Or IsNull(!transfy) Then SWNulls = True

SUMXY = SUMXY + Nz(!TransfX,0) * Nz(!TransfY,0)

SUMX2 = SUMX2 + Nz(!TransfX,0) ^ 2

If Grado = 2 Then

SUMX2Y = SUMX2Y + Nz(!TransfX,0) ^ 2, * Nz(!TransfY,0)

SUMX3 = SUMX3 + Nz(!TransfX,0) ^ 3

SUMX4 = SUMX4 + Nz(!TransfX,0) ^ 4

End If

rs.MoveNext

Loop

End With

rs.Close

***Coeficientes de la regresión**

Xavg = SUMX / n

Yavg = SUMY / n

***Verifica grado solicitado para el ajuste.**

If Grado = 1 Then

b = (n * SUMXY - SUMX * SUMY) / (n * SUMX2 - SUMX ^ 2)

a = Yavg - b * Xavg

a = Format(a, "##0.00000")

b = Format(b, "##0.00000")

***Si es un ajuste cuadrático se hacen cálculos adicionales.**

If Grado = 2 Then

a1 = n 'a1 se hace 1 al final

b1 = SUMX / n

c1 = SUMX2 / n

d1 = SUMY / n

a2 = SUMX

b2 = SUMX2 / SUMX - b1 'b2 se hace 1 al final

c2 = (SUMX3 / SUMX - c1) / b2

d2 = (SUMXY / SUMX - d1) / b2

a3 = SUMX2

b3 = SUMX3 / SUMX2 - b1 'b3 se hace 1 al final

c3 = (SUMX4 / SUMX2 - c1) / b3

d3 = (SUMX2Y / SUMX2 - d1) / b3

c = (d3 - d2) / (c3 - c2)

b = d2 - c2 * c

a = d1 - b1 * b - c1 * c

a = Format(a, "##0.00000")

b = Format(b, "##0.00000")

c = Format(c, "##0.00000")

End If '*** grado 2

*Inicializa detector de valores omitidos.

*Lee primer registro de datos dados.

*Usará el registro de datos.

*Ciclo que se repite hasta que se acaban los datos.

*Suma(X) (ó su transformada).

*Nz es una función de Access para detectar nulos,

*Identifica valores invalidos

*así, si TransfX es nulo, entonces se usa cero.

*Suma(X^2),

*Cálculos adicionales para ajuste con grado 2,

*Suma(X^2 * Y),

*Suma(X^3),

*Suma(X^4).

*Terminan cálculos para grado 2.

*Accesa el siguiente registro.

*Cierra ciclo Do While.

*Cierra cláusula With.

*Cierra archivo de datos de captura.

*Media de X

*Media de Y

*Cálculo para recta de ajuste:

*Pendiente de la recta.

*Término independiente.

*Se formatean a y b a 5 decimales.

*Cálculos para ajuste de grado 2.

*Suma(X)/n

*Suma(X^2)/n

*Suma(Y)/n

*Se desarrollaron las fórmulas resolviendo el sistema

*de ecuaciones:

*a1 b1 c1 d1

*a2 b2 c2 d2

*a3 b3 c3 d3

*a, b y c son los coeficientes del modelo

*Se formatean los coeficientes a 5 decimales.

***Si detectó valores omitidos, lo reporta en la subforma de presentación.**

```
If SWNulls Then
  rsReg.AddNew
  rsReg!a = " * * A V I S O * * * "
  rsReg!b = "El modelo y las gráficas pueden estar erróncos."
  rsReg.Update
  rsReg.AddNew
  rsReg!a = "Verifique datos:"
  rsReg!b = "Hay valores X y Y en blanco."
  rsReg.Update
  rsReg.AddNew
  rsReg!a = "Borre renglones"
  rsReg!b = "o teclée ceros en donde aparecen blancos."
  rsReg.Update
End If
```

*Variable detectora de valores omitidos.

***Guarda los datos parciales del modelo en variables auxiliares.**

```
rsReg.AddNew
rsReg!a = "Y ="
If b > 0 Then
  Modelo = Str(a) & " + " & Str(b) & " * X"
Else
  Modelo = Str(a) & " " & Str(b) & " * X"
End If
End If
'grado 1
```

*Genera la expresión de la recta ($Y = a + b \cdot X$)
*Pendiente positiva: se agrega el signo "+", la función
*Str hace que un valor sea considerado como texto y se
*pueda concatenar.
*Aqui "b" ya incluye el signo "-"

```
If Grado = 2 Then
  If c > 0 Then
    Modelo = Modelo & " + " & Str(c) & " * X^2"
  Else
    Modelo = Modelo & " " & Str(c) & " * X^2"
  End If
End If
```

*Si el ajuste es de grado 2, se complementa el modelo,
*adicionándole la parte cuadrática:
* $Y = a + b \cdot X + c \cdot X^2$

*Aqui el signo ya esta incluido.

***Se guardan los datos del modelo en una tabla que se liga a una subforma para desplegar la información.**

```
rsReg!b = Modelo
rsReg.Update
```

*Se guarda la expresión del modelo

```
rsReg.AddNew
rsReg!a = "a ="
rsReg!b = a
rsReg.Update
```

*Se crea un nuevo registro y se guardan los datos para
*el término independiente, (a).

```
rsReg.AddNew
rsReg!a = "b ="
rsReg!b = b
rsReg.Update
```

*Se crea un nuevo registro y se guardan los datos para
*el coeficiente de X, (b).

```
If Grado = 2 Then
  rsReg.AddNew
  rsReg!a = "c ="
  rsReg!b = c
  rsReg.Update
End If
```

*Se crea un nuevo registro y se guardan los datos para
*el coeficiente de X^2 , (c).

***Genera Residuales, Coeficiente de Correlación y Estimación de la Varianza (Errores).**

*Usa la tabla de captura de datos para guardar ahí Ajuste(Y) y los residuales.

*Nótese que se abre esta tabla por segunda vez, la primera fue para leer TransfX y TransfY para generar los datos de la curva de ajuste.

Set RsNvo = dbs.OpenRecordset("SELECT TransfX,TransfY,Yajuste,Residual FROM tblAnReg01")

RsNvo.MoveFirst

With RsNvo

Do While Not RsNvo.EOF

.Edit

XtransfX = Nz(!transfx, 0)

XtransfY = Nz(!transfy, 0)

!Yajuste = Equation(Modelo, XTransfX, True, 0, 0, 0)

Aux = XTransfY - !Yajuste

!residual = Aux

SumResid2 = SumResid2 + Aux ^ 2

Aux = XTransfX - Xavg

VarX = VarX + Aux ^ 2

Aux = XTransfY - Yavg

VarY = VarY + Aux ^ 2

.Update

.MoveNext

Loop

End With

RsNvo.Close

VarX = VarX / n

VarY = VarY / n

DstdX = Sqr(VarX)

DstdY = Sqr(VarY)

CovXY = SUMXY / n - Xavg * Yavg

r = CovXY / (DstdX * DstdY)

VarErr = SumResid2 / (n - 2)

StdErr = Sqr(VarErr)

***Todas las siguientes instrucciones son datos que se despliegan en la subforma de presentación:**

rsReg.AddNew

rsReg!a = "Est.Coef.Corr. ="

rsReg!b = r

rsReg.Update

rsReg.AddNew

rsReg!a = "Est.Var(e) ="

rsReg!b = VarErr

rsReg.Update

rsReg.AddNew

rsReg!a = "Est.dSt(e) ="

rsReg!b = StdErr

rsReg.Update

*Los siguientes datos se guardan en la tabla, sólo si se "checa" (palomea) la caja-chequeo "Desplegar Cálculos del Ajuste"

If Desplegar Then

*Accesa el primer registro de la tabla.

*Indica que se usará el registro RsNvo

*Se repite este ciclo hasta que se termine el archivo.

*Sólo edita porque (X,Y,TransfX,TransfY) ya existen.

*Usa variables auxiliares para poner ceros en caso que

*TransfX o TransfY tengan valores nulos.

*Genera ajuste(Y) con Equation del módulo general.

*Se calcula cada residual.

*Se genera Suma(residual^2).

*Cálculos para obtener la Covarianza y la correlación:

*Suma(diferencias^2 (X)).

*Suma(diferencias^2 (Y)).

*Guarda los cálculos de ajuste(Y) y residual(Y).

*Accesa el siguiente registro.

*Cierra ciclo Do While.

*Cierra cláusula With.

*Cierra archivo de datos.

*Calcula Var(X).

*Calcula Var(Y).

*Calcula Desviación Standard (X).

*Calcula Desviación Standard (Y).

*Calcula Covarianza(X,Y).

*Calcula Estimación del Coeficiente de Correlación.

*Calcula Estimación de Varianza(Error)

*Estimación de Desv.Std(Error).

*Guarda Estimación del Coeficiente de Correlación.

*Guarda Estimación de la Varianza(Error).

*Guarda Estimación de Desv.Std(Error).

rsReg.AddNew rsReg!a = " Cálculos:" rsReg!b = " * * * * *" rsReg.Update	'Encabezado de la sección.
rsReg.AddNew rsReg!a = "N =" rsReg!b = n rsReg.Update	'Guarda el número de observaciones.
rsReg.AddNew rsReg!a = "Sum(Xi) =" rsReg!b = SUMX rsReg.Update	'Guarda la Suma(X).
rsReg.AddNew rsReg!a = "Media(X) =" rsReg!b = Xavg rsReg.Update	'Guarda la Media(X).
rsReg.AddNew rsReg!a = "Sum(Yi) =" rsReg!b = SUMY rsReg.Update	'Guarda la Suma(Y).
rsReg.AddNew rsReg!a = "Media(Y) =" rsReg!b = Yavg rsReg.Update	'Guarda la Media(Y).
rsReg.AddNew rsReg!a = "Sum(XiYi) =" rsReg!b = SUMXY rsReg.Update	'Guarda la Suma(X * Y).
rsReg.AddNew rsReg!a = "Sum(Xi^2) =" rsReg!b = SUMX2 rsReg.Update	'Guarda la Suma(X ²).
If Grado = 2 Then rsReg.AddNew rsReg!a = "Sum(Xi^2Yi) =" rsReg!b = SUMX2Y rsReg.Update	'Si se solicita grado 2: 'Guarda la Suma(X ² * Y).
rsReg.AddNew rsReg!a = "Sum(Xi^3) =" rsReg!b = SUMX3 rsReg.Update	'Guarda la Suma(X ³).
rsReg.AddNew rsReg!a = "Sum(Xi^4) =" rsReg!b = SUMX4 rsReg.Update	'Guarda la Suma(X ⁴).
End If	

```
rsReg.AddNew          *Guarda la Suma(Residual?).
rsReg1a = "Sum(Residual?) ="
rsReg1b = SumResid2
rsReg.Update
End If      * * * if desplegar

rsReg.Close          *Cierra archivo de datos a desplegar.
dbs.Close           *Cierra la base de datos.
Set dbs = Nothing   *Desconecta la base de datos.
Regresion = Modelo  *Asigna la expresión a la función
End Function        *Fin del código.
```

5.5.3 **Función Naleatorios.** La función genera una tabla de números aleatorios cuyos valores oscilan en un rango dado por el usuario; la cantidad de números a generar también se especifica por el usuario. En caso de valores inválidos se hacen los siguientes default: **Rango(0,100);** cantidad de números: 10.

Function NAleatorios(TotNos, LInf, LSup) As Double	
Dim dbs As DAO.Database	*Se declara una base de datos.
Dim rs, rsNAleat As DAO.Recordset	*Se declaran dos archivos.
Dim NosByCol, i, Extra As Integer, TotNum, Menor, Mayor As Long	*Variables auxiliares.
stDocName = "qryDelNosAleat"	*Query para borrar la tabla de números aleatorios.
DoCmd.OpenQuery stDocName, acNormal, acEdit	*Se ejecuta el query.
Set dbs = CurrentDb	*La base de datos conectada es EmCalc.
Set rsNAleat = dbs.OpenRecordset("tblDatNAleat")	*Tabla de parámetros de los números aleatorios.
rsNAleat.AddNew	*Después de borrar, se crea un registro (el único).
With rsNAleat	*Se usa el registro de la tabla de parámetros.
!TotalNos = TotNos	*Se guardan los nuevos parámetros especificados por el
!LimInf = LInf	*usuario. Campos con "!" son del archivo. TotNos, LInf
!LimSup = Lsup	*y Lsup son dados por el usuario.
.Update	*Se guarda el registro
End With	*Cierra cláusula With.
rsNAleat.Close	*Cierra archivo de parámetros.
TotNum = TotNos	*Usa variables auxiliares.
Menor = LInf	
Mayor = LSup	
Set rs = dbs.OpenRecordset("tblNAleat")	*Asigna tabla de números al archivo rs.
Extra = TotNum Mod 10	*Verifica que la cantidad sea múltiplo de 10, si no, crea
NosByCol = TotNum / 10	*una columna extra, que contiene la cantidad solicitada.
If Extra > 0 And Extra < 6 Then	
NosByCol = NosByCol + 1	
End If	
Randomize	*Inicializa semilla de números aleatorios con reloj.
With rs	*Usa el registro del archivo de números.
For i = 1 To NosByCol	*Ciclo para todas las columnas a crear.
rs.AddNew	*Agrega un renglón a la tabla de números.
rs.Renglon = 1	*Asigna el número de renglón que esta creando.
rs!Col01 = Int((Mayor - Menor + 1) * Rnd + Menor)	*Crea el número aleatorio en el rango especificado,
rs!Col02 = Int((Mayor - Menor + 1) * Rnd + Menor)	*para cada columna (columnas 1 a 10).
rs!Col03 = Int((Mayor - Menor + 1) * Rnd + Menor)	
rs!Col04 = Int((Mayor - Menor + 1) * Rnd + Menor)	
rs!Col05 = Int((Mayor - Menor + 1) * Rnd + Menor)	
rs!Col06 = Int((Mayor - Menor + 1) * Rnd + Menor)	
rs!Col07 = Int((Mayor - Menor + 1) * Rnd + Menor)	
rs!Col08 = Int((Mayor - Menor + 1) * Rnd + Menor)	
rs!Col09 = Int((Mayor - Menor + 1) * Rnd + Menor)	
rs!Col10 = Int((Mayor - Menor + 1) * Rnd + Menor)	
rs.Update	*Guarda el registro.
Next	*Cierra ciclo For.
End With	*Cierra cláusula With.
rs.Close	*Cierra archivo de números.
dbs.Close	*Cierra la base de datos.
Set dbs = Nothing	*Desasigna la base de datos.
End Function	

6. Implementación de palabra clave para acceso a EmCalc.

La implementación de palabra clave se diseña en el módulo local de la forma de entrada o principal de EmCalc. Es una subrutina que se corre al momento de abrir la forma y simplemente permite el acceso en caso de dar la clave "Emcalc" adecuadamente.

Permite 10 intentos para dar la clave adecuada, en caso de no hacerlo, cierra Access. Igual, cierra Access cuándo el usuario le instruye "Salir"

Subrutina para permitir el acceso a EmCalc.

Private Sub Form_Open(Cancel As Integer)	'Se realiza al momento de abrir la forma.
Dim strMsg As String, strInput As String, i As Integer	'Variables auxiliares
i = 0	'Contador de intentos de acceso a EmCalc
strInput = ""	'Variable para recibir la palabra clave.
Do While strInput <> "EmCalc" And i < 10	'Realiza ciclo hasta por 10 ocasiones
strMsg = "Por favor proporcione la Palabra Clave, o Salir:"	'Mensaje para el usuario.
strInput = InputBox(Prompt:=strMsg, Title:"User Info", XPos:=2000, YPos:=2000)	'Envía mensaje
If strInput <> "EmCalc" Then	'La clave es errónea
If strInput <> "Salir" Then	'La clave tampoco es "Salir"
MsgBox "Palabra clave errónea, por favor intente nuevamente."	'Mensaje de intento fallido.
Else	
MsgBox "El software Access será cerrado."	'La clave fue "Salir"
Quit	'Cierra Access
End If	'Cierra If para checar si la clave es Salir.
End If	'Cierra If para checar si la clave es EmCalc.
i = i + 1	'Aumenta contador de intentos de acceso.
Loop	'Cierra estructura Do While.
If i >= 10 Or strInput <> "EmCalc" Then	'Verifica si se cumple una de estas condiciones.
MsgBox "Palabra clave errónea. El software Access será cerrado."	'De cumplirse, emite
Quit	'mensaje y cierra Access.
End If	'Cierra estructura IF.
End Sub	'Termina la subrutina.

7. Conclusiones.

El trabajo presentado cumple su cometido, proporcionando al usuario de la aplicación facilidades similares a las otorgadas por una calculadora. Como se mencionó al inicio del texto, al igual que una calculadora, requiere que el usuario tenga conocimientos específicos del área de influencia en que realice operaciones en EmCalc.

Las características que EmCalc hereda de una base de datos, crean facilidades adicionales de almacenamiento de información, de modo que el trabajo que el usuario crea puede ser guardado permanentemente, esta característica ayuda al usuario a hacer su trabajo más sencillo al utilizar eficientemente la facilidad de guardar datos y expresiones de funciones. Evidentemente la utilización del tamaño de desplegado de un monitor de computadora, otorga la gran ventaja de poner instrucciones en la pantalla, además de proveer las instrucciones y ayuda en la pantalla con un clic a un comando de ayuda.

EmCalc proporciona una idea concreta para crear aplicaciones de propósito general en las diversas áreas de la ciencia, en áreas administrativas o de cualquier índole, constituyendo entonces todas estas áreas un mercado potencial para el desarrollo de aplicaciones de este tipo. Para este propósito, el profesional debe contar con experiencia en la utilización de Access y saber programar en Visual Basic, muy importante también será que esté involucrado con el área en que se desarrolla la aplicación para crear una aplicación realmente útil, ad hoc al área en que esta encaminada la aplicación. Esto, cabe aclarar, es una idea interesante, pero no se puede garantizar que existan empresas que inviertan en desarrollar software de este tipo, debido a que la mayoría de las empresas prefiere adquirir software ya existente en el mercado; como en todas las cosas, persistencia y constancia abren las puertas.

En el caso específico de este trabajo, un paso futuro es complementarlo con la valuación de funciones de densidad para el módulo de funciones estadísticas; refinar la sección de utilerías permitiendo la importación y exportación de datos a Excel en forma más general, proporcionando un filtro para evitar la importación de valores faltantes (datos con valores no inicializados a cero); en la sección de utilerías también, se hará el cambio de variables $X \Leftrightarrow Y$ para la sección de Análisis de Regresión. Se tiene previsto también su publicación en inglés, para lo cual será necesario traducir las pantallas, mensajes e instrucciones, no así el código de programación que es idéntico en ambos idiomas.

También una futura generación de EmCalc se proporcionará de modo que corra sin la necesidad de utilizar a Access como plataforma de lanzamiento, es decir, se realizará un programa que corra en la computadora con sus propios recursos, tal programa, al igual que muchas otras emulaciones creadas en otros lenguajes de programación, se hará disponible por medio de Internet. En el caso del presente trabajo se utilizará en un portal de Internet, con distribución gratis ó con un pago mínimo, tal vez el pago que el usuario deba aceptar, sea el envío de correo electrónico de las empresas patrocinadoras.

Por ultimo, se desea aclarar que el desarrollo de EmCalc no estuvo vinculado a proyecto alguno de ninguna empresa ó grupo de personas, sino que se realizó como un proyecto personal, alterno a cualquier trabajo remunerado. La idea de distribuirlo públicamente surgió conforme la aplicación fue aumentando su capacidad. Con la idea que este párrafo proporciona, se invita a los profesionistas a que presenten sus desarrollos originales en algún sitio de Internet ó de acceso masivo para promocionar sus trabajos.

Apéndice I. Requerimientos para correr EmCalc.

- 1) EmCalc se puede correr en MS-Access 97 o en MS-Access 2000, sin embargo, se debe obtener la copia respectiva de EmCalc para la versión de MS-Access en que se corre, ambas versiones corren en forma idéntica y proveen las mismas funciones y facilidades:
 - EmCalc97.mde para MS-Access 97
 - EmCalc2000.mde para MS-Access 2000.

- 2) El espacio requerido en disco, al tener MS-Access ya instalado, es de 3.5MB. Este tamaño es el inicial ya que al guardar información su tamaño crecerá. Su tamaño máximo depende de la versión que se esté utilizando:
 - EmCalc, para MS-Access 97, puede crecer hasta 500 MB, dependiendo de la información que se almacene y los procesos que se corran.
 - EmCalc, para MS-Access 2000, puede crecer hasta 2 GB, dependiendo de la información que se almacene y los procesos que se corran.

 - EmCalc, debe ser "compactado y reparado" después de un uso extensivo para reducir su tamaño y aumentar su velocidad y capacidad de proceso. Un tamaño adecuado debe ser de unos 10MB como máximo, a menos que se generen tablas de cientos de miles de registros. Véase la sección de Utilerías y Solución de problemas de EmCalc para mayor información.

- 3) Se sugiere utilizar un procesador Pentium I ó superior, para mayor rapidez de procesamiento.

- 4) La copia de EmCalc se puede guardar en el directorio de preferencia. No requiere instalación alguna, sino simplemente un copiado. De preferencia, se sugiere crear un directorio llamado EmCalc para almacenar la aplicación.

Apéndice II. Bibliografía.

En la realización de esta tesis se utilizaron parcialmente los siguientes libros y textos en línea.

The Statistical Analysis of Experimental Data

Autor John Mandel

Dover Publications, Inc., New York, U.S.A. 1984.

Compiladores. Principios, técnicas y herramientas.

Alfred V. Aho

Ravi Sethi

Jeffrey D. Ullman

Editorial Addison-Wesley Iberoamericana. 1990.

Las ayudas en línea de los siguientes softwares:

Microsoft Access

Microsoft Excel

Visual Basic

Los siguientes sitios de Internet:

www.calculator.org

www.calculator.com