

UNIVERSIDAD PANAMERICANA

ESCUELA DE INGENIERIA

CON ESTUDIOS INCORPORADOS A LA
UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

**“ANÁLISIS Y DISEÑO DE UN SISTEMA PARA LA
ADMINISTRACIÓN DE PORTAFOLIOS DE
INVERSIÓN APLICANDO MODELOS
FINANCIEROS NO LINEALES”**

T E S I S

**QUE PARA OBTENER EL TÍTULO DE
INGENIERO MECÁNICO ELECTRICISTA
ÁREA INGENIERÍA INDUSTRIAL**

P R E S E N T A

ALEJANDRO VILLASEÑOR ZERTUCHE

DIRECTOR DE TESIS:

ING. ANTONIO CASTRO D' FRANCHIS

N. Autorizo a la Dirección General de Bibliotecas de la UNAM a difundir en formato electrónico e impreso el contenido de mi trabajo recepcional.

NOMBRE: Alejandra Villaseñor Zertuche

FECHA: 4/ Septiembre 196

FIRMA: [Firma]

86414882-1

No autoriza la microfilmación y digitalización de mi obra.

[Firma]

54252107
alejandra-villaseñor@prodigy.net.mx

A mi esposa Luz Aurora y a mis hijos que son fuente de inspiración

A mis padres

A mi hermano Fernando

A mi hermano Jaime por todo su apoyo

ÍNDICE

Introducción	i
1. Antecedentes	1
1.1 El problema de optimización	1
1.1.1 Identificar el problema	1
1.1.2 Encontrando una solución	2
1.1.3 Modelos matemáticos.....	4
1.2 Breve historia de la Teoría de Portafolios de Inversión.....	5
1.3 Conceptos estadísticos	9
1.3.1 El valor esperado y la varianza de los rendimientos	10
1.3.1.1 Media y varianza de una muestra	11
1.3.2 La distribución de probabilidad conjunta.....	12
1.3.2.1 La covarianza de una muestra	13
1.3.2.2 Coeficiente de correlación.....	15
1.3.2.3 Coeficiente de determinación	15
1.3.3 Relación entre una acción y un portafolio.....	15
1.3.3.1 La línea característica	16
1.3.3.2 El factor Beta (β).....	17

1.3.3.3	Varianza residual.....	18
1.3.4	Portafolios.....	19
1.3.4.1	Estimación del rendimiento de un portafolio.....	19
1.3.4.2	La varianza de un portafolio.....	20
1.3.4.3	La línea de combinación.....	21
1.3.4.4	Correlación positiva y negativa perfecta.....	22
1.4	Métodos de solución de portafolios.....	23
1.4.1	Método gráfico.....	23
1.4.1.1	Variables de decisión.....	23
1.4.1.2	Función objetivo.....	24
1.4.1.3	Restricciones.....	24
1.4.2	Método de Programación Cuadrática.....	25
1.5	Modelos de Optimización de Portafolios de Inversión.....	29
1.5.1	Markowitz.....	29
1.5.2	Sharpe.....	32
1.5.3	Modelos multifactoriales.....	32
2.	Diseño de un Sistema de Administración de Portafolios.....	33
2.1	Carga e inicialización del sistema.....	33

2.1.1	Seguridad.....	36
2.1.2	Menú	37
2.1.3	Estructura	39
2.2	Base de datos.....	40
2.2.1	Crear base de datos.....	40
2.2.2	Seleccionar base de datos.....	42
2.2.3	Compactar base de datos.....	43
2.3	Datos	45
2.3.1	Cargar información a la base de datos	45
2.3.2	Importar datos a Excel.....	47
2.3.3	Borrar información de la base de datos.....	51
2.4	Hojas	52
2.4.1	Borrar hojas.....	53
2.5	Precios.....	54
2.5.1	Modificar precios	54
2.6	Rendimientos	59
2.7	Análisis	60
2.7.1	Promedio aritmético.....	65

2.7.2	Promedio ponderado	67
2.7.3	Normalidad.....	69
2.8	Gráficas	69
2.9	Matriz de varianzas-covarianzas y correlaciones.....	69
2.10	Portafolios	72
2.10.1	Portafolios de mínima varianza (PMV)	75
2.10.2	Portafolios con riesgo menor o igual a	76
2.10.3	Portafolios con rendimiento mayor o igual a	78
2.10.4	Portafolios de la frontera.....	79
2.10.5	Portafolios factibles	80
2.11	Idioma	81
2.12	Salida del sistema.....	83
3.	Caso práctico.....	84
	Conclusiones	95
	Bibliografía	97
	Apéndice “A” Código	99
	Índice de rutinas por orden alfabético.....	99
	Inicio	100

Seguridad	100
Menú	102
Verifica estructura.....	105
Base de datos.....	106
Hojas	120
Precios.....	129
Rendimientos	138
Análisis	140
Gráficas.....	149
Matriz de covarianzas y correlaciones.....	153
Portafolios	159
Idioma	175
Salida	178
Rutinas generales	178

Introducción

La información y la tecnología tienen un papel fundamental en la toma de decisiones para invertir los recursos de la mejor manera. El hacer cada vez más eficiente el manejo de la información y poder reaccionar rápidamente a los cambios en el entorno puede ser la diferencia entre tener un buen o mal desempeño en el rendimiento de un portafolio de inversiones. Este trabajo tiene como finalidad el presentar el diseño de un programa que facilite el manejo y la organización de la información así como la optimización de portafolios dependiendo del perfil de preferencias de riesgo del inversionista. Debido al concepto de diversificación, existen mejores activos financieros que otros no sólo por su rendimiento si no porque en su conjunto reducen el riesgo de la inversión. Empíricamente hemos aplicado el refrán popular “no hay que poner todos los huevos en la misma canasta”, aunque muy pocas veces nos hemos puesto a pensar porque es así y mucho menos hemos visto cual es la mejor combinación que nos de el resultado que esperamos.

La presente tesis está basada en la propuesta conceptual que Markowitz ideó a mediados del siglo pasado y que sigue teniendo vigencia en la industria. Las metodologías comenzaron a proliferar y esta investigación muestra que cualesquiera que sea el método elegido es necesario “sistematizar” este conocimiento mediante algún sistema informático, para así ponerlo en práctica.

En realidad las técnicas de inversión en México habían tenido algún retraso por la ausencia de “inversionistas institucionales”, históricamente habían sido sólo unos cuantos individuos (menos de 150 mil personas que hacían inversiones en lo personal). En el país, hace menos de diez años, comenzó a surgir la figura de las Sociedades de Inversión especializadas en Fondos para el Retiro (SIEFORES) y ahora estos vehículos de inversión tienen más de 55 mil millones de

dólares, que representan casi una tercera parte del Producto Interno Bruto de México. De ahí la importancia de que los métodos de inversión sean cada vez mas profesionales y menos intuitivos.

La tesis está organizada de la siguiente manera: en el primer capítulo veremos la explicación desde el punto de vista teórico, la manera en que la incorporación de activos financieros afecta la variabilidad del portafolio así como ecuaciones que demuestran este comportamiento.

En el segundo capítulo está el diseño de una aplicación, motivo de este trabajo, donde se pueden apreciar los procesos que facilitan la administración del portafolio y que sugieren la mejor manera de distribuir los recursos. También se muestran las pantallas del programa que hacen uso de las rutinas.

En el tercer capítulo se muestra un caso práctico con datos sobre índices mundiales que podrían ser susceptibles de inversión, mostrando el impacto que tendría el incorporar o no el índice mexicano "IPC" dentro de los posibles instrumentos de inversión. Por último se mide el desempeño de estos portafolios o "performance" para probar su eficacia asumiendo que las recomendaciones de inversión generadas son puestas en práctica y se presentan las conclusiones pertinentes.

Por último, en el Apéndice "A" Código se muestran las funciones y las rutinas generales que forman parte de la aplicación ordenados por tipo.

1. Antecedentes

1.1 El problema de optimización

Las principales herramientas que tienen los analistas para resolver los problemas relativos a la construcción de portafolios de inversión son los modelos matemático-financieros y los algoritmos computacionales que dan información cuantitativa; permitiendo obtener soluciones óptimas del portafolio de acuerdo con las características y expectativas de los inversionistas que están dispuestos a participar.

Para asignar eficientemente los recursos disponibles en diferentes instrumentos, el uso de estos modelos requiere de un proceso automatizado y eficiente dada la gran cantidad de datos y de cálculos aritméticos que se requiere procesar en este tipo de problemas. Asimismo, el contar con una base de datos confiable, eficiente y actualizada es fundamental para el análisis, lo cual es posible a través de un acceso continuo y oportuno a las fuentes de la información y por otro lado un sistema que proporcione el almacenamiento adecuado de la información.

1.1.1 Identificar el problema

El objetivo planteado es el diseñar portafolios de inversión en las dos formas como comúnmente se interpretan las inversiones financieras: por un lado, se seleccionan todas aquellas que tengan el mayor rendimiento promedio esperado dado un nivel de riesgo (i.e. varianza para nuestros propósitos); y otro que tenga el menor riesgo esperado (o varianza esperada) dado su rendimiento.

Los métodos cuantitativos permiten obtener o proponer uno o varios portafolios óptimos o eficientes, es decir, que no existan otros portafolios con un mejor rendimiento dado un mismo

riesgo, o bien, portafolios con un menor riesgo y que tenga el mismo rendimiento y que cumpla con el objetivo de un inversionista en cuanto al rendimiento que espera y el riesgo que está dispuesto a incurrir para obtenerlo. Es importante destacar que estos portafolios se establecen con la información histórica existente y como tales los portafolios propuestos asumen indirectamente una estabilidad o como dirían los economistas “caeteris-paribus”; es decir, manteniendo todo lo demás constante.

En la creación de portafolios existen variantes que se pueden considerar dentro del modelo como son: la opción de invertir en instrumentos con riesgo nulo (caso ideal ya que sabemos que no existen instrumentos sin riesgo pero que este es despreciable en comparación con otros); adquirir posiciones cortas sobre los activos; y pedir prestado recursos a una cierta tasa de interés. Sin embargo, para limitar nuestro escenario en este trabajo, se tomarán únicamente posiciones largas. Para el contexto mexicano actual, la realidad es que el “vender en corto” o “apalancar la inversión” es muy complicado y muy caro llevar a cabo. Por otro lado, sí hay instrumentos gubernamentales que técnicamente están libres de riesgo pero únicamente en el sentido crediticio; es decir, que el gobierno federal tendría dinero para pagar siempre estos instrumentos mientras estén denominados en pesos. El Banco Central tendría que estar de acuerdo en pagar con circulante nuevo y esto haría que el inversionista no debiera preocuparse por si le pagan o no. No obstante, estos instrumentos sí tiene riesgo denominado “de mercado” porque una fluctuación en las tasas de interés o en el tipo de cambio puede hacer que el valor de las inversiones fluctúe.

1.1.2 Encontrando una solución

Una vez identificado el problema de portafolio eficiente, se requiere de la formulación de un modelo matemático representativo para que la solución obtenida resulte atractiva para el

inversionista. Aunque sabemos que en la práctica las carteras tienen activos en forma discreta (la tenencia mínima es un título y no se pueden tener fracciones de éste) podremos hacer una aproximación continua como un porcentaje invertido en el activo y suponer que dichas diferencias son insignificantes para portafolios de tamaño considerable.

La validez de los resultados de un modelo depende de la cantidad e importancia de los supuestos que tome en cuenta así como de la forma en cómo se obtengan los datos de entrada. Una propuesta para la aplicación del modelo se presenta a continuación:

I. Probar el modelo y la solución obtenida.

Aún cuando en teoría hayamos conseguido un portafolio eficiente hay que ver que dicha solución cumpla con su objetivo, para ello se pueden hacer pruebas posteriores al análisis y comparar los resultados contra lo esperado (análisis ex-post). Dado que las condiciones del mercado son cambiantes en el tiempo no esperaríamos obtener los mismos resultados, pero sí un portafolio con un desempeño parecido. Si no tenemos un resultado aceptable es necesario revisar el planteamiento y las entradas de datos y repetir la prueba.

II. Establecer controles sobre la solución.

Como el resultado del modelo se presenta como el porcentaje invertido en cada activo y éstos van cambiando con el tiempo, es conveniente hacer una revisión periódica de la solución inicial (este procedimiento se conoce como rebalanceo), o bien, hacer nuevamente el cálculo del portafolio óptimo incorporando información reciente. Esto vuelve a conformar un portafolio con posiciones que pueden ser distintas a las originalmente encontradas.

III. Llevar la solución a la práctica.

La formación del portafolio inicial tiene ciertas restricciones de acuerdo a las condiciones diarias del mercado como pueden ser la oferta y demanda del instrumento, y que hacen que las operaciones no se puedan realizar o bien que el precio ya no sea atractivo. Por ello es conveniente establecer “portafolios objetivo” a donde se quieran colocar las carteras y monitorear constantemente el mercado para poder hacer las adecuaciones necesarias a nuestra cartera.

1.1.3 Modelos matemáticos

La ecuación que describe el problema de cartera (función objetivo) se puede ver como: 1) una minimización del riesgo dado un rendimiento; y 2) como una maximización del rendimiento dado un cierto nivel de riesgo. Dichos datos son obtenidos y aplicados de manera distinta dependiendo del modelo a seguir (Markowitz o CAPM, explicados posteriormente en este trabajo) y están en función del porcentaje del dinero invertido en cada instrumento (variables de decisión). El planteamiento del problema de tipo Markowitz es el siguiente:

$$1) \text{ Min } \sigma^2 (r_p)$$

$$\text{s.a.} \begin{cases} E(r_p) = r_* \\ \sum_{i=1}^n x_i = 1 \end{cases}$$

$$2) \text{ Max } E(r_p)$$

$$\text{s.a.} \begin{cases} \sigma(r_p) = \sigma_* \\ \sum_{i=1}^n x_i = 1 \end{cases}$$

Donde para Markowitz: $\sigma^2(r_p) = \sum_{i=1}^n \sum_{j=1}^n x_i x_j \sigma_{i,j}$

$$E(r_p) = \sum_{i=1}^n x_i r_i$$

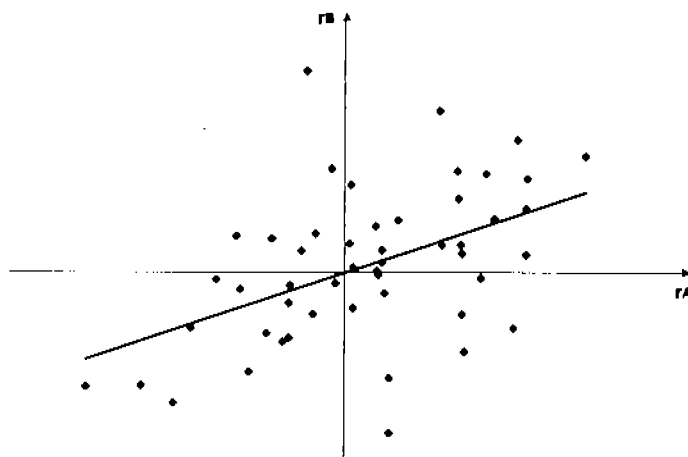
Estos conceptos estadísticos se explican posteriormente en el capítulo 1.3.

1.2 Breve historia de la Teoría de Portafollos de Inversión

La teoría de portafollos es utilizada por los inversionistas y analistas para reducir el riesgo o la incertidumbre respecto a los rendimientos de su inversión (en realidad el objetivo es obtener la mejor relación riesgo-rendimiento posible). Entre más alejado del presente esté nuestro horizonte de planeación mayor será su riesgo ya que las condiciones de los mercados financieros cambian con el tiempo. La preocupación del administrador del portafolio será llevar el dinero o los activos financieros en la mejor de las circunstancias posibles, esto es, con el mayor rendimiento. En general esperaríamos que un mejor rendimiento esté asociado a un mayor riesgo, por esto el problema se presenta como una optimización con restricciones de riesgo permisible o de rendimiento esperado, aunque no siempre sea factible obtenerlos por las características de los activos y del mercado. En la práctica esto podría llevar a modificar el sencillo planteamiento de la ecuación de optimización planteada e incorporar el que ninguna ponderación fuese negativa (i.e. no se podría hacer una “venta en corto”) u otras adecuaciones importantes.

La diversificación nos dice que si combinamos instrumentos cuyos rendimientos no se encuentran perfectamente correlacionados (ver Gráfica 1.1), encontramos que la desviación estándar conjunta del portafolio es menor que el promedio ponderado de las desviaciones estándar de dichos instrumentos individualmente debido a la relación que existe entre ellos. Es

por esto que logramos disminuir el riesgo cuando combinamos activos con diferente comportamiento dentro de un mismo portafolio.



Gráfica 1.1 Rendimientos no perfectamente correlacionados

El obtener la máxima utilidad dado un nivel de riesgo, o minimizar el riesgo del portafolio con cierto rendimiento es el principal objetivo que tienen los inversionistas bajo este concepto, por ello se requiere conocer la mezcla óptima de instrumentos, con rutinas que encuentren la mejor composición del portafolio.

La teoría de portafolios comienza de manera formal en 1952 cuando Harry Markowitz publica un artículo titulado "Portfolio Selection". En él explica cómo crear un conjunto de portafolios de inversión tal que cada uno de ellos tenga la máxima utilidad esperada dado un cierto nivel de riesgo. La gran cantidad de cálculos involucrados hacía que la propuesta de Markowitz fuera poco práctica en aquella época. Antes de ser difundida la teoría de portafolio masivamente, una versión simplificada de esta técnica, pero con más supuestos, conocida como el "Modelo de determinación de precios de activos financieros" o CAPM por las siglas en inglés de "Capital Asset Pricing Model" se le atribuye simultáneamente a Sharpe (1964), Lintner (1965) y Mossin

(1966) que independientemente estudiaron el impacto que tendría en la valuación de los instrumentos si todos los inversionistas tomaran una posición en la frontera usando la teoría de portafolios. Esta nueva versión hace que la teoría fuera más práctica en su cálculo, aún manejando una gran cantidad de instrumentos.

En 1965 se publicó la tesis de Eugene Fama en el Journal of Business en donde expone que dado que hay miles de inversionistas bien informados buscando instrumentos subvalorados, éstos afectan sus precios en forma colectiva por aquéllos que invierten en ellos. Ya que la información fluye rápidamente es imposible tener mayores rendimientos que el mercado por cualquier forma de análisis del instrumento, suponiendo que esto es cierto, el mercado sería eficiente. Por lo que hoy en día está en discusión la eficiencia del mercado. Este debate originó un gran número de estudios acerca de la cantidad y calidad de información reflejada en el precio de los instrumentos.

Suponiendo que el mercado es eficiente, los inversionistas podrán formar sus portafolios de acuerdo al nivel de riesgo que estén dispuestos a tomar ya que el rendimiento guarda una relación directa con su riesgo. En la actualidad, avances rápidos en computación han revolucionado el manejo profesional de la inversión y los administradores pueden sentarse en sus terminales y encontrar información detallada de un número ilimitado de compañías en todos los sectores del mercado. Esto tiene dos importantes implicaciones: la primera que la información es canalizada más rápidamente desde su fuente hacia los analistas e inversionistas que van a actuar sobre ellos, el probable efecto es el de disminuir el retraso entre el evento y su efecto sobre el precio de las acciones; la segunda es que el crecimiento en los sistemas computacionales hace posible usar sofisticados modelos financieros que toman en cuenta un mayor número de factores.

Por ser de importancia reciente, se hará una breve mención sobre lo que ahora se conoce como finanzas del comportamiento (“behavioral finance”). Una vez que Daniel Kahneman recibió

recientemente el premio de economía del Banco de Suecia, es prudente que en cualquier investigación respecto a la teoría de portafolios se incluyan cuando menos algunos de sus conceptos básicos.

Los tres temas más importantes en las finanzas del comportamiento son:

Heurísticas: Las personas comúnmente hacen decisiones basadas en “reglas de dedo” o simplificaciones prácticas y no en un análisis estrictamente racional.

Marco conceptual: La manera en que un problema o una decisión es presentada a algún tomador de decisiones afecta su decisión.

Ineficiencias de Mercado: Se intenta explicar los resultados de un mercado que son contrarios a lo que se considera una eficiencia en los mercados y expectativas racionales. En particular, Richard Thaler, ha escrito una larga serie de artículos describiendo ciertas anomalías del mercado desde un punto de vista del comportamiento.

Una referencia adicional sobre el comportamiento de los portafolios se presenta en el artículo publicado en el 2005, “A Behavioral Approach to Efficient Portfolio Formation” Yar Gulnur Muradoglu, City University, Aslihan Altay-Salih, Bilkent University, Muhammet Mercan, Yapi Kredi Yatirim.

“Este documento investiga el desempeño del portafolio de pronósticos subjetivos proporcionados en diferentes formas. En cuanto a la construcción de la frontera eficiente, nuestras expectativas se basan en el pronóstico subjetivo y en el comportamiento humano más que en los precios históricos. Estas estimaciones son proporcionadas por los administradores de portafolios dando un precio al que consideran que debe alcanzar. La primera contribución es el mostrar que el desempeño de un portafolio de pronósticos subjetivos es superior al modelaje de

series de tiempo. La segunda contribución se basa en el hecho de que usamos expertos en el manejo de estimaciones, administradores profesionales de fondos con suficiente experiencia. La tercera contribución es que los expertos usan pronósticos para puntos específicos de tiempo, intervalos y estimaciones probabilísticas que hacen que los resultados sean robustos.”

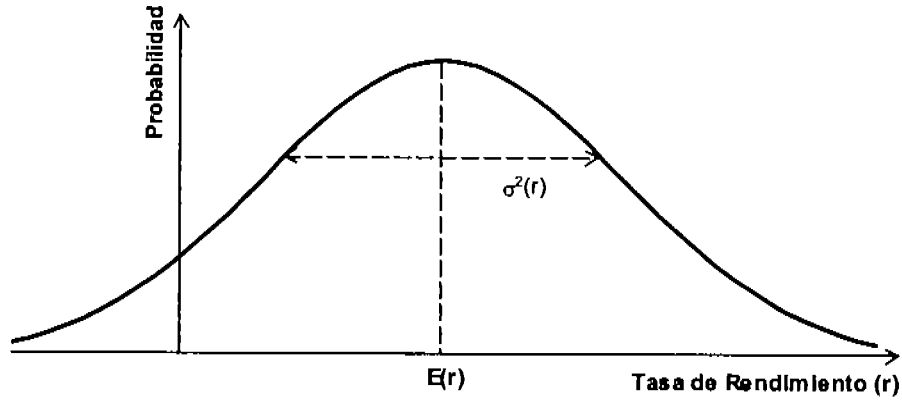
1.3 Conceptos estadísticos

Cuando queremos describir el comportamiento histórico o futuro de los rendimientos de los instrumentos podemos hacerlo con dos parámetros como son la media esperada $E(r)$ y la varianza $\sigma^2(r)$. Una manera de estimar el comportamiento futuro de los rendimientos es obtenerlos por el comportamiento histórico de los rendimientos observados (r), mientras que su comportamiento no cambie considerablemente en el tiempo.

En el caso de querer obtener el rendimiento de una acción, éste se mide con el ingreso asociado por tener una acción durante un período. Es igual al cambio entre el precio actual y el de adquisición más los dividendos pagados durante el periodo de medición.

$$r = \frac{P_1}{P_0} - 1 \dots\dots\dots \text{Ecuación 1.1}$$

Supongamos que la distribución simple de probabilidad representa la esperanza de obtener las tasas de rendimiento durante el transcurso del tiempo. La distribución de dichos rendimientos puede verse como la siguiente gráfica:



Gráfica 1.2 Distribución de probabilidad de la tasa de rendimiento

En esta gráfica se puede observar que el eje horizontal “x” representa la tasa de rendimiento, y en el eje vertical “y” la probabilidad de ocurrencia.

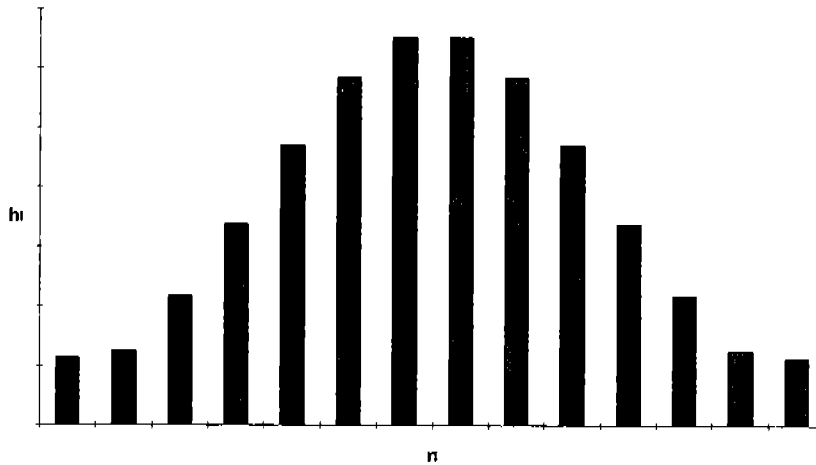
1.3.1 El valor esperado y la varianza de los rendimientos

Supongamos que estamos interesados en encontrar un punto donde están centralmente localizados los rendimientos. Los parámetros que describen esta localización son el rendimiento esperado y la varianza. El primero indica lo que esperaríamos recibir en el transcurso del siguiente período y se representa con la siguiente ecuación:

$$E(r) = \sum_{i=1}^n h_i r_i \dots\dots\dots \text{Ecuación 1.2}$$

Donde: h_i representa la probabilidad de ocurrencia de un rendimiento r_i .

n es el número de rendimientos posibles.



Gráfica 1.3 Distribución de rendimientos esperados

Para encontrar la tasa de rendimiento esperado hay que sumar todos los posibles rendimientos multiplicados por la probabilidad de que ocurran.

El segundo parámetro describe la varianza, que indica el grado de dispersión de los rendimientos con respecto a su rendimiento esperado. El riesgo de un portafolio generalmente es medido por la variabilidad de sus rendimientos. La forma de calcularlo es:

$$\sigma^2(r) = \sum_{i=1}^n h_i [r_i - E(r)]^2 \dots\dots\dots \text{Ecuación 1.3}$$

De la misma manera que se hizo en el rendimiento esperado, se suman todas las desviaciones y se multiplican por su probabilidad de ocurrencia.

1.3.1.1 Media y varianza de una muestra

Debido a que en la realidad no podemos saber la distribución que describe el comportamiento de las acciones, generalmente ésta se puede estimar por medio de una muestra. Se debe calcular el rendimiento esperado y la varianza para utilizarlos como estimados para usar las técnicas

empleadas en la teoría de portafolios (aun cuando es posible extender su alcance a pedir “juicios expertos” a personas o utilizar otros métodos para obtener la media esperada, es mas difícil en el caso de la varianza). Para obtener el estimado, necesitamos suponer que la distribución de la probabilidad es constante durante todo el tiempo. En general, la ecuación para calcular el valor esperado para el tiempo t de la muestra de n rendimientos tenemos:

$$\bar{r}_t = \frac{\sum_{i=t-n+1}^n r_i}{n} \dots\dots\dots \text{Ecuación 1.4}$$

La media de la muestra nos da un estimado del valor esperado sin mucha precisión. Esperaríamos que fuera más representativo entre más grande fuera la muestra, pero solamente cuando la distribución no cambia con el tiempo. Esta suposición es menos realista entre más tiempo pasa. En general, la muestra se tomaría por periodos tan largos que no hubieran cambios tan grandes en la forma de la distribución característica.

Lo mismo sucede para la varianza, no se puede saber exactamente cuál es su varianza por lo que se necesita sacar de una muestra. La varianza de una muestra la obtendremos de la siguiente ecuación:

$$\sigma_t^2(r) = \frac{\sum_{i=t-n+1}^n (r_i - \bar{r})^2}{n - 1} \dots\dots\dots \text{Ecuación 1.5}$$

Se divide entre n-1 porque se está manejando un estimado de la muestra.

1.3.2 La distribución de probabilidad conjunta

La tasa de rendimiento esperada y la varianza nos proporcionan la información de la distribución asociada a una sola acción, o para un portafolio de acciones. Pero esto no nos dice nada de la

forma como el rendimiento de un instrumento está relacionado con otro instrumento. La fórmula de esta distribución conjunta está dada por:

$$Cov(r_A, r_B) = \sum_{i=1}^n h_i [r_{A,i} - E(r_A)][r_{B,i} - E(r_B)] \dots\dots \text{Ecuación 1.6}$$

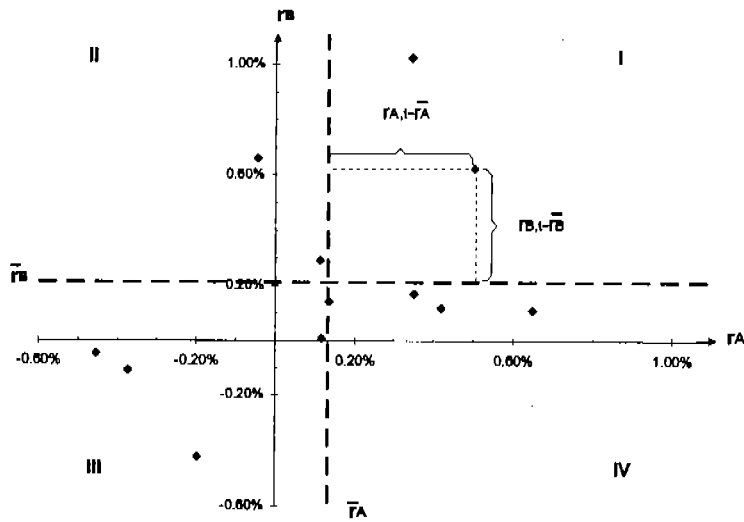
1.3.2.1 La covarianza de una muestra

Supongamos que tenemos los rendimientos históricos (r) de dos instrumentos A y B.

Tiempo (t)	r _A	r _B
1	0.42 %	0.12 %
2	0.14 %	0.14 %
3	0.65 %	0.11 %
4	0.34 %	1.02 %
5	0.12 %	0.01 %
6	-0.37 %	-0.11 %
7	-0.46 %	-0.05 %
8	0.35 %	0.17 %
9	-0.20 %	-0.42 %
10	0.11 %	0.29 %
11	0.50 %	0.62 %
12	-0.05 %	0.66 %

Tabla 1.1 Ejemplo de rendimientos observados de dos instrumentos

Representando los rendimientos en un plano observamos la relación que guardan entre sí.



Gráfica 1.4 Relación entre los rendimientos de dos acciones

La fórmula para encontrar la covarianza de una muestra de datos es:

$$Cov_{r_A r_B} = \frac{\sum_{t=1}^n [(r_{A,t} - \bar{r}_A)(r_{B,t} - \bar{r}_B)]}{n-1} \dots\dots\dots \text{Ecuación 1.7}$$

Aunque la covarianza no nos dice mucho acerca de la relación entre dos acciones sabemos que si ésta es positiva, cuando el rendimiento de una acción que esté por arriba del promedio, la otra con la que tiene covarianza positiva tenderá a hacer lo mismo y viceversa. En la Gráfica 1.4 podemos observar que mientras los puntos estén en los cuadrantes I y III, la diferencia con sus medias será con signos iguales, esto es, si uno da más o da menos que lo esperado, entonces el otro tenderá a hacer lo mismo. Contrariamente, en los cuadrantes II y IV mientras uno tenga mayor rendimiento el otro tenderá a ser menor a lo esperado.

Entre más positivo sea el valor de la covarianza diremos que tiene una relación directa y mientras más negativo tendremos una relación inversa. Para covarianzas cercanas a cero diremos que casi no tienen relación entre ellos.

1.3.2.2 Coeficiente de correlación

Teóricamente la covarianza puede ir desde menos infinito hasta más infinito, pero si lo dividimos entre el producto de las desviaciones estándar de ambos rendimientos éstos normalizan el estadístico y se obtiene el coeficiente de correlación, el cual puede tomar valores entre -1 y +1. Este coeficiente representa una descripción más intuitiva de la relación que la covarianza.

$$\rho_{A,B} = \frac{Cov(r_A, r_B)}{\sigma(r_A)\sigma(r_B)} \dots\dots\dots \text{Ecuación 1.8}$$

Si la correlación es +1 tendremos una correlación positiva perfecta y si es -1 tendremos una correlación negativa perfecta.

1.3.2.3 Coeficiente de determinación

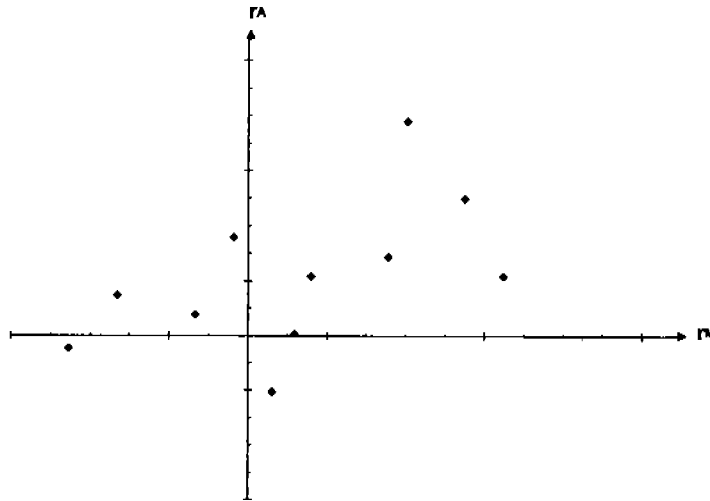
El coeficiente de determinación es el cuadrado del coeficiente de correlación y nos dice qué fracción de la variabilidad en el rendimiento de una inversión está asociada o explicada por la variabilidad en el rendimiento de otra.

$$\text{Coef_Det} = \rho_{A,B}^2 \dots\dots\dots \text{Ecuación 1.9}$$

1.3.3 Relación entre una acción y un portafolio

Consideraremos algunas estadísticas que describen la relación entre el rendimiento de una acción y lo que llamaremos portafolio de mercado. Un portafolio de mercado (M) tiene todas las

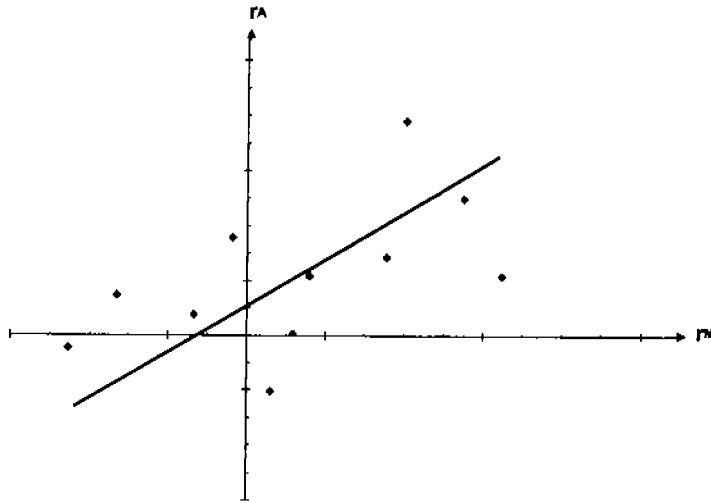
acciones riesgosas en el sistema financiero, y contiene cada activo financiero en proporción al total del mercado de ese bien relativo con el valor total de todos los activos financieros. Este tipo de portafolio es comúnmente llamado un portafolio ponderado de valores.



Gráfica 1.5 Relación entre el rendimiento una acción y el rendimiento del mercado

1.3.3.1 La línea característica

Cuando comparamos el rendimiento de una acción con el portafolio de mercado en la gráfica que utilizamos anteriormente y encontramos la línea que mejor se ajusta a ésta, la llamamos línea característica. La cual nos dice el rendimiento que esperamos de una acción dada una tasa de rendimiento correspondiente al mercado.



Gráfica 1.6 Línea característica de una acción

1.3.3.2 El factor Beta (β)

Como la línea característica es una línea recta, ésta puede ser completamente descrita por su pendiente y un punto por donde pasa sobre el eje vertical. La pendiente de la línea característica es normalmente referida como el factor beta (β) de una acción. La intersección la denotaremos con el símbolo A.

$$\hat{\beta} = \frac{Cov r_J r_M}{\sigma_{r_M}^2} \dots\dots\dots \text{Ecuación 1.10}$$

$$\hat{A}_J = \bar{r}_J - \hat{\beta}_J \bar{r}_M \dots\dots\dots \text{Ecuación 1.11}$$

El factor beta indica el grado en que la acción reacciona debido a cambios en el rendimiento provocados por el mercado. La ordenada únicamente sirve para fijar la posición de la línea.

1.3.3.3 Varianza residual

Otra dimensión de la relación entre la acción y el mercado es la tendencia de la acción a producir rendimientos que se desvían de la línea característica. La estadística que describe su tendencia se llama varianza residual. Mientras la varianza de la acción describe esa tendencia de producir rendimientos derivados de su valor esperado, la varianza residual describe la tendencia de la acción a producir rendimientos que se desvían de su línea característica.

La varianza residual es la línea vertical entre el par de rendimientos y su línea característica.

$$\varepsilon_{J,t} = r_{J,t} - (\hat{A}_J + \hat{\beta}_J r_{M,t}) \dots\dots\dots \text{Ecuación 1.12}$$

$$\sigma_{\varepsilon_J}^2 = \frac{\sum_{i=1}^n \varepsilon_{J,t}^2}{n-2} \dots\dots\dots \text{Ecuación 1.13}$$

Dividimos entre n-2 y no n-1 porque estamos empleando dos estimadores en vez de uno para hacer el cálculo.

Cuando calculamos la varianza de la muestra, empleamos un estimado del valor esperado, el promedio de la muestra. Aquí empleamos la ordenada y la pendiente de la curva característica para calcular residuales.

La varianza residual de una acción se aproxima a cero, mientras que el coeficiente de correlación se acerca a (+1) ó (-1), dependiendo si la línea característica tiene pendiente positiva o negativa. También es cierto para un número grande de muestras que si el coeficiente de correlación es cero, la varianza residual de la acción es aproximadamente a su varianza, porque con una correlación o covarianza de cero, el factor beta de la acción también es cero. Regresando a la

ordenada, vemos que para betas iguales a cero, la línea característica intercepta al eje vertical en el promedio de la muestra.

1.3.4 Portafolios

Un portafolio de inversión por definición consta de varios instrumentos. El problema radica en saber cuáles, de todos los diferentes productos financieros se debe escoger. Se podrían seleccionar aquellos instrumentos que minimicen el riesgo del portafolio, pero ello también supondría muy poco rendimiento esperado, de ahí el problema de elegir. En realidad se pretende que cada quién fije un límite de riesgo esperado tolerable y de ahí obtener el máximo rendimiento posible.

1.3.4.1 Estimación del rendimiento de un portafolio

Si conocemos los rendimientos esperados de los m instrumentos y la fracción que representa de un portafolio, podremos conocer el rendimiento esperado del portafolio por medio de la siguiente fórmula:

$$r_p = \sum_{j=1}^m x_j r_j \dots\dots\dots \text{Ecuación 1.14}$$

$$x_j = \frac{\text{Inversión total para el activo J}}{\text{Inversión total del portafolio}}$$

Es simplemente el promedio ponderado de los rendimientos esperados de cada una de las acciones que componen el portafolio. Una fracción positiva significa que se está tomando una

posición larga¹, mientras que en caso de ser negativo tenemos una posición corta. Todas estas fracciones también llamadas ponderaciones del portafolio deben de sumar uno.

$$\sum_{j=1}^m x_j = 1 \dots\dots\dots \text{Ecuación 1.15}$$

El rendimiento esperado de un portafolio será:

$$E(r_p) = \sum_{j=1}^m x_j E(r_j) \dots\dots\dots \text{Ecuación 1.16}$$

Por lo general, incrementando nuestro rendimiento esperado aumentaríamos también el riesgo del portafolio, aunque más adelante se verá que existen beneficios de diversificación que hacen que esto no sea siempre cierto.

1.3.4.2 La varianza de un portafolio

Para calcular la varianza del portafolio necesitamos la matriz de covarianzas de los instrumentos que estamos manejando en el portafolio. Las dimensiones de esta matriz cuadrada son de m renglones por m columnas donde m es el número de instrumentos que conforman el portafolio. En la diagonal simplemente tenemos la varianza².

¹ Una posición es larga cuando compramos el instrumento con la expectativa de que suba el precio. Caso contrario, la posición corta o venta en corto se observa cuando se piden prestadas las acciones y luego se venden estando obligados a regresar la misma cantidad de acciones después de un cierto tiempo y probablemente con una prima o premio al dueño original de los instrumentos. Esta estrategia se usa cuando esperamos que el precio del instrumento baje.

² La covarianza de una acción sobre si misma es la varianza.

$$Cov(r_A, r_A) = \sum_{i=1}^m h_i [r_{A,i} - E(r_A)] [r_{A,i} - E(r_A)] = \sum_{i=1}^m h_i [r_{A,i} - E(r_A)]^2 = \sigma^2(r_A)$$

Instrumento	A	B	C	...	m
A	$\sigma^2(r_A)$	$Cov(r_A, r_B)$	$Cov(r_A, r_C)$...	$Cov(r_A, r_m)$
B	$Cov(r_B, r_A)$	$\sigma^2(r_B)$	$Cov(r_B, r_C)$...	$Cov(r_B, r_m)$
C	$Cov(r_C, r_A)$	$Cov(r_C, r_B)$	$\sigma^2(r_C)$...	$Cov(r_C, r_m)$
...
m	$Cov(r_m, r_A)$	$Cov(r_m, r_B)$	$Cov(r_m, r_C)$...	$\sigma^2(r_m)$

Tabla 1.2 Matriz de varianzas-covarianzas de los instrumentos

También podemos observar que la covarianza de una acción sobre otra es igual a la covarianza de la segunda con respecto a la primera.

$$\begin{aligned}
 Cov(r_A, r_B) &= \sum_{i=1}^m h_i [r_{A,i} - E(r_A)] [r_{B,i} - E(r_B)] \\
 &= \sum_{i=1}^m h_i [r_{B,i} - E(r_B)] [r_{A,i} - E(r_A)] = Cov(r_B, r_A)
 \end{aligned}$$

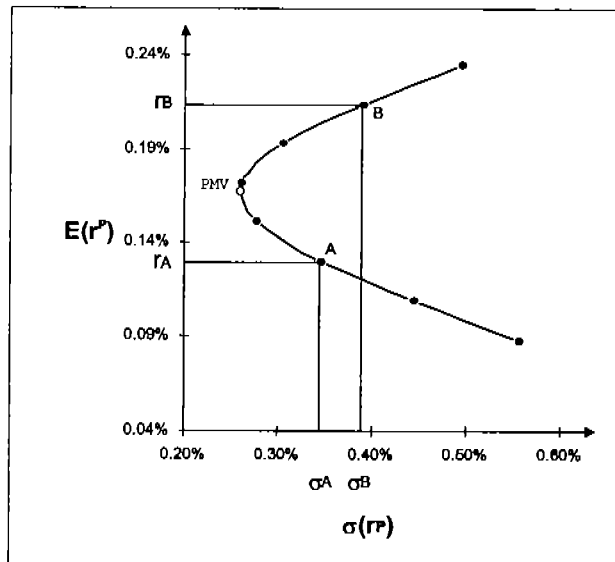
Por lo tanto, necesitaríamos calcular $(m^2-m)/2+m$ elementos. Por ejemplo el caso de 200 instrumentos sería $(200^2-200)/2+200$ (20,100 elementos).

Para obtener la varianza del portafolio necesitamos multiplicar la covarianza por cada uno de los pesos de las acciones y sumarlos. Obviamente, la desviación estándar es la raíz cuadrada de la varianza.

$$\sigma^2(r_P) = \sum_{j=1}^m x_j^2 \sigma^2(r_j) + 2 \sum_{j=1}^{m-1} \sum_{i=j+1}^m x_j x_i Cov(r_j, r_i) \dots\dots\dots \text{Ecuación 1.17}$$

1.1.1.1 La línea de combinación

Ésta se dibuja en una gráfica como la que se muestra a continuación, donde en el eje vertical tenemos $E(r)$ y en el eje horizontal $\sigma(r)$. Cada punto sobre la línea representa el rendimiento esperado y la desviación estándar de un portafolio dado el peso de cada acción.



Gráfica 1.7 Línea de combinación entre dos acciones

Para puntos sobre la curva entre los puntos A y B estamos tomando posiciones positivas en ambas acciones. Portafolios al sureste de A se están tomando posiciones cortas de B. Portafolio al noreste de B se están tomando posiciones cortas de A.

Los portafolios por debajo del punto de mínima varianza (PMV), no son atractivos para los inversionistas, ya que tienen un menor rendimiento que los de la parte superior y con un mismo nivel de riesgo.

1.3.4.4 Correlación positiva y negativa perfecta

En el caso de que los rendimientos tengan una correlación de uno, esto es, correlación positiva perfecta, todos los pares de rendimientos caerían sobre la misma línea y la pendiente sería la relación que hay entre las desviaciones estándar de ambos instrumentos. En dado caso la desviación estándar del portafolio es simplemente el promedio ponderado de ambos activos, por lo que podríamos obtener un portafolio con un riesgo de cero vendiendo en corto una acción e invirtiendo en otra.

$$\sigma(r_p) = \sqrt{x_A^2 \sigma^2(r_A) + 2x_A x_B \rho \sigma(r_A) \sigma(r_B) + x_B^2 \sigma^2(r_B)} \quad \dots\dots\dots \text{Ecuación 1.18}$$

$$\rho = 1, \quad x_A = 1 - x_B \quad \dots\dots\dots \sigma(r_p) = x_A \sigma(r_A) + (1 - x_A) \sigma(r_B)$$

$$0 = x_A (\sigma(r_A) - \sigma(r_B)) + \sigma(r_B) \quad \dots\dots\dots x_A = \frac{\sigma(r_B)}{\sigma(r_B) - \sigma(r_A)}$$

Se puede observar que el rendimiento de dicho portafolio es:

$$E(r_p) = x_A E(r_A) + (1 - x_A) E(r_B) \quad \dots\dots\dots \text{Ecuación 1.19}$$

Donde $E(r_B) = a_0 + a_1 E(r_A)$

1.4 Métodos de solución de portafolios

1.4.1 Método gráfico

Este método sirve para resolver problemas de hasta tres instrumentos ya que pueden ser representados en un plano de dos ejes. Al conjunto de combinaciones de instrumentos que conforman una inversión la llamaremos portafolios o puntos sobre la gráfica. Los dos ejes representan el porcentaje de inversión de dos de los activos y el tercero está determinado por la restricción de la suma de todos los pesos.

1.4.1.1 Variables de decisión

Las variables son los porcentajes que se van a invertir en cada uno de los instrumentos y éstos pueden tomar valores continuos. Al momento de crear el portafolio, se deberán adquirir títulos redondeando al número más cercano al resultado.

1.4.1.2 Función objetivo

Existen dos planteamientos para encontrar un portafolio óptimo. Las ecuaciones de la función objetivo son:

- 1) Minimizar el riesgo de un portafolio

$$\sigma^2(r_p) = \sum_{i=1}^n \sum_{j=1}^n x_i x_j \sigma_{i,j} \dots\dots\dots \text{Ecuación 1.20}$$

Gráficamente es posible obtener los puntos a la derecha de la bala cambiando el problema de minimización a uno de maximización con la ecuación anterior, los cuales serían portafolios no deseables excepto para los puntos extremos.

- 2) Maximizar el rendimiento de un portafolio

$$E(r_p) = \sum_{i=1}^n x_i r_i \dots\dots\dots \text{Ecuación 1.21}$$

Es común utilizar el primer planteamiento ya que se puede reducir a un problema de programación cuadrática. De esta manera se simplifica la metodología para encontrar la solución al problema.

1.4.1.3 Restricciones

Para los casos anteriores las restricciones a los planteamientos son:

- 1) $E(r_p) = \sum_{i=1}^n x_i r_i = r_p^* \dots\dots\dots \text{Ecuación 1.22}$

- 2) $\sigma^2(r_p) = \sum_{i=1}^n \sum_{j=1}^n x_i x_j \sigma_{i,j} = \sigma_p^* \dots\dots\dots \text{Ecuación 1.23}$

1) y 2) $\sum_{i=1}^n x_i = 1$ Ecuación 1.24

Para obtener posiciones largas usaremos la siguiente restricción para cada instrumento.

$$x_i \geq 0, \text{ para } i = 1 \dots n$$

1.4.2 Método de Programación Cuadrática

El problema de minimización del riesgo en una cartera de inversión es un caso de programación cuadrática. La programación lineal difiere de la cuadrática en que la función objetivo ($f(x)$) es de segundo grado, esto es, las variables de decisión se presentan como (x_j^2) y $(x_j x_k)$, las restricciones ($g(x)$) únicamente tienen términos de primer grado, es decir, con una sola variable (x_1, x_2, \dots, x_j) . El problema general de programación lineal quedaría planteado de la siguiente manera:

Función Objetivo	Max $f(x)$
Restricciones	$g_1(x) \leq b_1$
	...
	$g_m(x) \leq b_m$

Para poder plantear el problema de portafolio eficiente y resolverlo con el método simplex es necesario modificar las ecuaciones iniciales de tal manera que se presenten en la forma general. La formulación para encontrar el PMV (punto de mínima varianza) que se presentó anteriormente es el caso más sencillo.

$$\text{Min } \sigma^2 = \sum_{i=1}^n \sum_{j=1}^n x_i x_j \text{Cov}_{i,j}$$

$$\text{sa } \sum_{i=1}^n x_i = 1$$

Un conjunto de valores de las n variables x puede ser óptimo sólo si existen m números, u_1, u_2, \dots, u_m tales que cumplan las condiciones KKT³.

$$\frac{\partial f}{\partial x_j} - \sum_{i=1}^m u_i \frac{\partial g_i}{\partial x_j} \leq 0 \quad \text{para } j = 1, 2, \dots, n$$

$$x_j^* \left(\frac{\partial f}{\partial x_j} - \sum_{i=1}^m u_i \frac{\partial g_i}{\partial x_j} \right) = 0 \quad \text{para } j = 1, 2, \dots, n$$

$$g_i(x^*) - b_i \leq 0 \quad \text{para } i = 1, 2, \dots, m$$

$$u_i (g_i(x^*) - b_i) = 0 \quad \text{para } i = 1, 2, \dots, m$$

$$x_j^* \geq 0 \quad \text{para } j = 1, 2, \dots, n$$

$$u_i \geq 0 \quad \text{para } i = 1, 2, \dots, m$$

Para que la solución sea óptima, no sólo debe cumplir lo anterior sino que la función objetivo $f(x)$ sea cóncava y las m restricciones $g_i(x)$ convexas.

³ Karush-Kuhn-Tucker. Karush, W.: "Minima of Functions of Several Variables with Inequalities as Side Conditions", MS Thesis, Department of Mathematics, University of Chicago, 1939.. Kuhn, H.W. and A.W. Tucker: "Nonlinear Programming" in Jerzy Neyman (ed), Proceedings of the Second Berkeley Symposium, University of California Press, Berkeley, 1951, pp. 481-492.

Para el caso de dos variables x_1, x_2 (porcentaje de inversión en el activo 1 y 2 respectivamente)

las ecuaciones quedarían de la siguiente manera:

$$\text{Max } \sigma^2 = x_1^2 \text{Cov}_{1,1} + x_1 x_2 \text{Cov}_{1,2} + x_2 x_1 \text{Cov}_{2,1} + x_2^2 \text{Cov}_{2,2}$$

$$\text{sa } x_1 + x_2 = 1$$

Simplificando la ecuación y cambiando el problema a minimización quedaría:

$$\text{Min } -\sigma^2 = -\text{Cov}_{1,1}x_1^2 - \text{Cov}_{2,2}x_2^2 - 2\text{Cov}_{1,2}x_1x_2$$

$$\text{sa } x_1 + x_2 = 1$$

En forma matricial el problema se puede ver como:

$$\text{Maximizar: } f(x) = cx - \frac{1}{2}x^T Qx$$

$$\text{sa: } Ax \leq b, x \geq 0$$

Donde: c , vector renglón con los coeficientes de los términos de grado uno (c_1, c_2, \dots, c_n).

x , vector columna de las n variables (x_1, x_2, \dots, x_n).

b , vector columna de las m restricciones (b_1, b_2, \dots, b_m).

A , matriz de coeficientes de las restricciones ($m \times n$).

Q , matriz que se obtiene de los coeficientes de los términos de grado dos ($n \times n$).

Por lo tanto, para el problema de minimización del riesgo quedaría:

$c = 0$ ya que en la función objetivo nunca tiene variables únicas.

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$Q = \begin{bmatrix} 2Cov_{1,1} & 2Cov_{1,2} & \cdots & 2Cov_{1,n} \\ 2Cov_{2,1} & 2Cov_{2,2} & \cdots & 2Cov_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 2Cov_{n,1} & 2Cov_{n,2} & \cdots & 2Cov_{n,n} \end{bmatrix}$$

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix}$$

$$b = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$

Otra manera de verificar que esté definido positivamente es obteniendo el determinante. Para el caso de tres instrumentos.

$$Q = \begin{bmatrix} 2Cov_{1,1} & 2Cov_{1,2} & 2Cov_{1,3} \\ 2Cov_{2,1} & 2Cov_{2,2} & 2Cov_{2,3} \\ 2Cov_{3,1} & 2Cov_{3,2} & 2Cov_{3,3} \end{bmatrix}$$

Tomando una matriz de varianzas y covarianzas como ejemplo:

$$|Q| = 2 \begin{vmatrix} .120 & .085 & .095 \\ .085 & .100 & .090 \\ .095 & .090 & .110 \end{vmatrix} = 0.0002085 \geq 0$$

Una vez cumplidas dichas condiciones entonces podemos garantizar que sea un óptimo. Las condiciones de KKT en la forma matricial para resolver el problema son:

$$Qx + A^T u - y = c^T$$

$$Ax + v = b$$

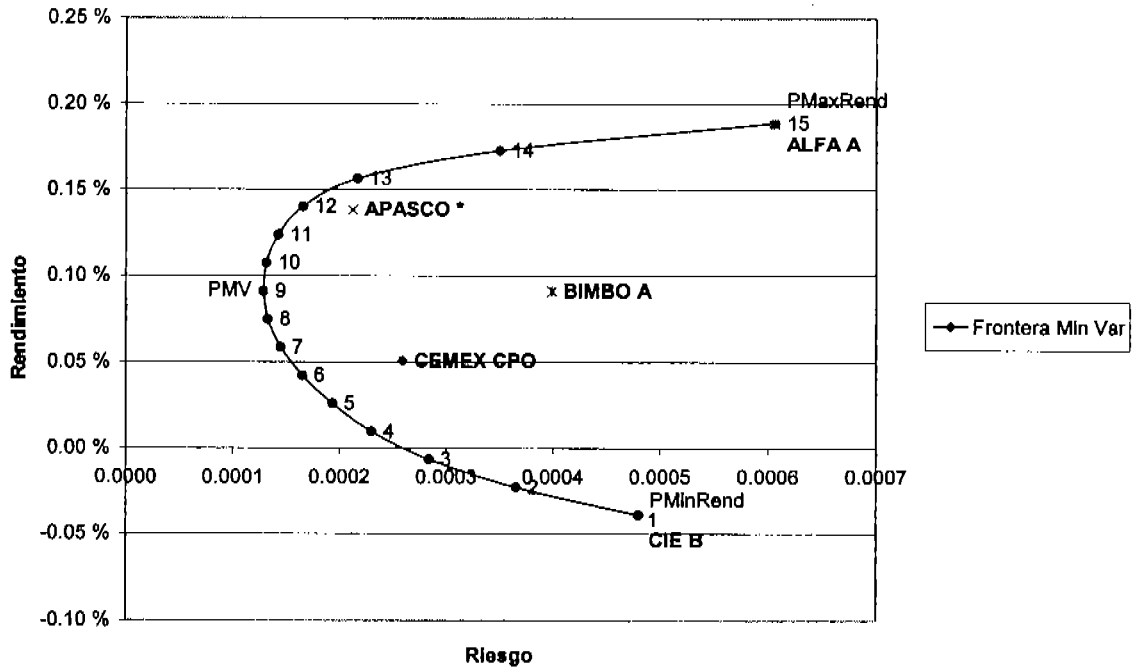
$$x^T y + u^T v = 0$$

1.5 Modelos de Optimización de Portafolios de Inversión

1.5.1 Markowitz

Podemos observar de la fórmula de desviación estándar de un portafolio que los activos con mayor correlación negativa lograrían mejores resultados en el riesgo total del portafolio para un rendimiento dado. La selección de los activos que lograrían encontrar una cartera situada en la frontera eficiente de Markowitz es por medio de programación cuadrática (no lineal) y se encontrarían cada uno de los puntos minimizando el riesgo para cada rendimiento entre los activos con menor y mayor rendimientos.

Markowitz



Gráfica 1.8 Gráfica de una frontera de Mínima Varianza

La parte de la frontera que nos interesaría analizar es aquella situada en la parte superior del punto de mínima varianza (PMV) donde a mayor riesgo incurrido, mayor rendimiento esperaríamos y viceversa. Existiría un portafolio único dependiendo de las características de los inversionistas, esto es el punto de tangencia entre la frontera eficiente y las curvas de indiferencia o la relación riesgo rendimiento para un determinado perfil de inversionista. A continuación se proporciona un ejemplo para ilustrar el procedimiento a seguir:

1. Escoger los mejores activos evaluados de acuerdo un análisis fundamental y que nos ayudaría a un primer acercamiento.
2. Catalogar de acuerdo al tipo de activo al que pertenece cada uno de los activos para poder determinar de manera agregada los límites de inversión (Asset Allocation).

3. Determinar la frecuencia de los datos, éstos pueden ser diarios, semanales, quincenales, mensuales, etc.. El ruido producido por fluctuaciones especulativas puede ser eliminado tomando datos más separados en el tiempo siempre y cuando los instrumentos tengan suficiente historia para ser tomados en cuenta.
4. Calcular los rendimientos como simple razón de cambio o cambios logarítmicos.
5. Determinar por cada uno de los activos si las posiciones pueden ser largas o también cortas.
6. Establecer límites conjuntos de inversión, esto es, determinar restricciones conjuntas para activos con las mismas características. Esto puede aplicar en México para las Afores.
7. Escoger el tipo de función que va a utilizar: con riesgo menor o igual a...; con rendimiento mayor o igual a...; con punto de mínima varianza.
8. Correr el algoritmo de optimización iterativamente para recalculando los nuevos portafolios. Esto puede incurrir en costos de transacción como pueden ser comisiones y/o penalizaciones por adquirir o deshacerse de activos.
9. Evaluar la cartera periódicamente para saber el desempeño en cuanto al riesgo y rendimiento reales (ex-post) para hacer las correcciones pertinentes a los parámetros del modelo.

1.5.2 Sharpe

Una simplificación del problema propuesto por Markowitz, es calcular los rendimientos a partir de una relación lineal entre un activo libre de riesgo y el portafolio del mercado. Las ecuaciones que describen el rendimiento son:

$$R_i = R_{TLR} + \beta_i * (R_M - R_{TLR}) \dots\dots\dots\text{Ecuación 1.25}$$

Donde:

$$\beta = \rho_{i,M} \frac{\sigma_i}{\sigma_M} = \frac{\sigma_{i,M}}{\sigma_M^2} \dots\dots\dots\text{Ecuación 1.26}$$

1.5.3 Modelos multifactoriales

Los modelos multifactoriales o APT (Arbitrage Pricing Theory) consideran la relación del activo con factores que afectan su comportamiento. Este es similar al CAPM, pero no solamente es comparado contra el mercado, ya que no todo el comportamiento de un activo está explicado por movimientos de éste.

Estos factores pueden ser tasas de interés, inflación, cambios en el Producto Interno Bruto y otros factores macroeconómicos que son difíciles de separar del portafolio por medio de la diversificación.

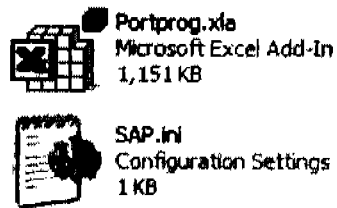
2. Diseño de un Sistema de Administración de Portafolios

El sistema trabaja sobre el paquete Microsoft Excel, que de ahora en adelante llamaremos Excel, como un complemento que puede ser instalado y retirado sin alterar ninguna de las funciones de dicho paquete. Las versiones anteriores a Excel 2000 no son compatibles con el programa ya que no ofrecen la flexibilidad que necesita el sistema. Ahora bien, en la optimización de portafolios es necesario tener instalado alguno de los "solvers", éste puede ser el que viene incluido en Excel, el Premium Solver o el Premium Solver Platform. Los dos últimos son paquetes que se instalan por separado y que hacen los cálculos de optimización de manera más eficiente ya que tienen mejores algoritmos para resolver los problemas en comparación con versiones anteriores.



2.1 Carga e instalación del sistema

La aplicación creada está compilada como "Excel Add-In" o "Complemento" y contiene la "maquinaria" para hacer el análisis la cual deberá cargarse desde Excel por primera vez. Utiliza un archivo de configuración llamado SAP.ini donde está el nombre de la base de datos que va a usar y el idioma deseado el cual podrá ser cambiado desde la aplicación sin necesidad hacerlo desde un editor de texto.



La aplicación debe ser instalada por primera vez desde el programa Excel seleccionando “Add-Ins...” del menú “Tools” como se ve en la figura 2.1.

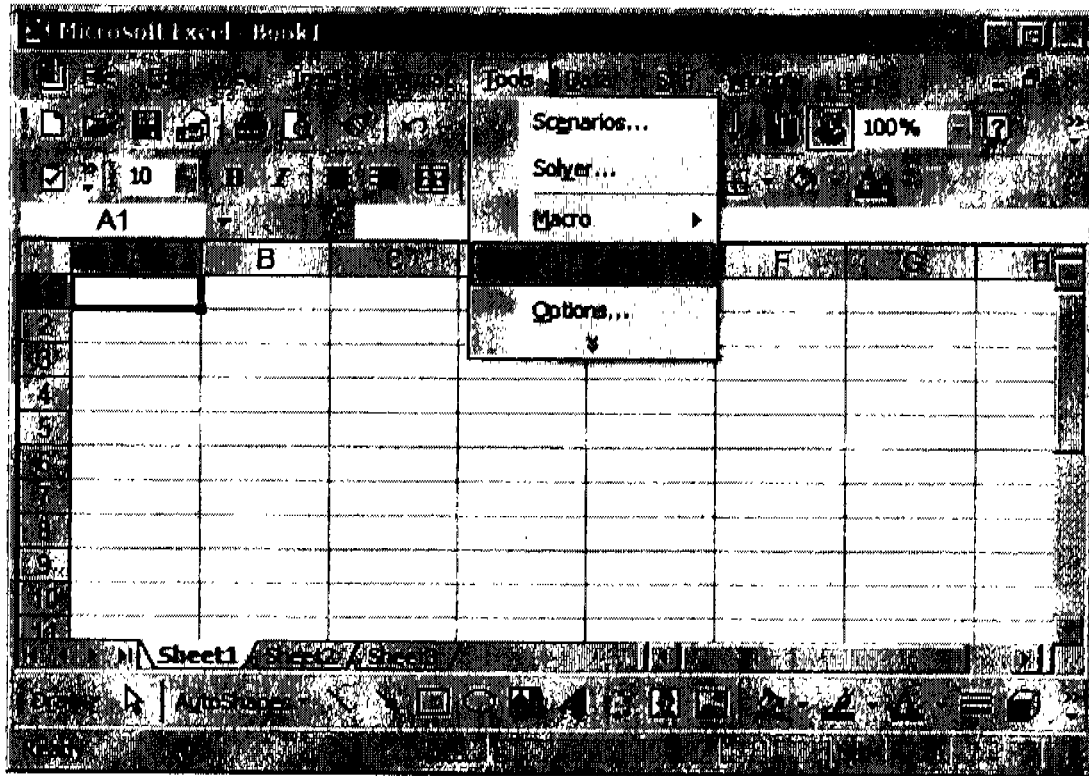


Figura 2.1 Carga del programa “Add-Ins”...

Luego aparecerá una lista de todas las aplicaciones instaladas en Excel. Antes de ver esta aplicación, necesitamos seleccionar su ubicación. Esto se hace presionando el botón “Browse...”.

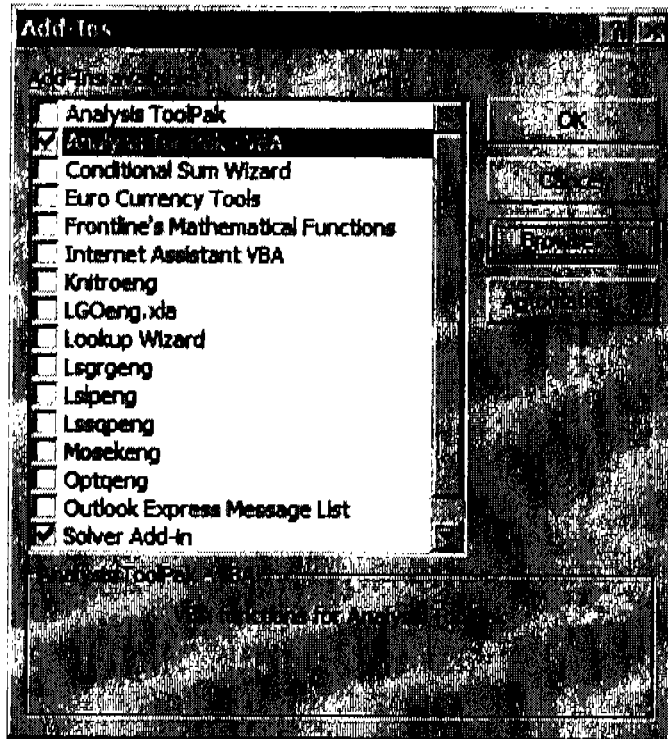


Figura 2.2 Add-Ins instalados

Seleccionaremos el archivo "Portprog.xls" como se muestra en la pantalla y presionaremos "OK".

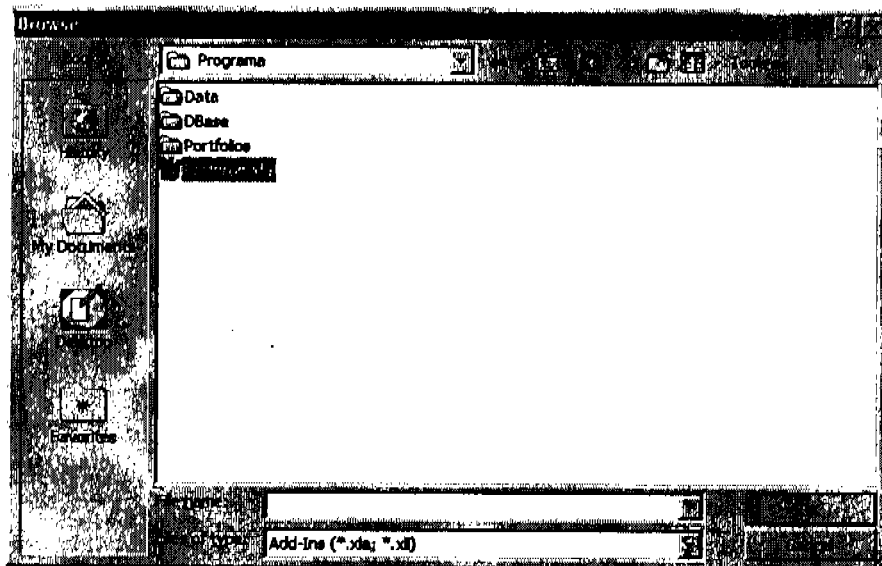


Figura 2.3 Selección del archivo Portprog.xls

Observe que aparecerá un nuevo elemento en la lista de aplicaciones instaladas llamado "Optimización de Portafolios". Asegúrese de que el recuadro de la izquierda esté seleccionado con una paloma, esto hará que se cargue cada vez que inicie Excel, por último presione "OK".

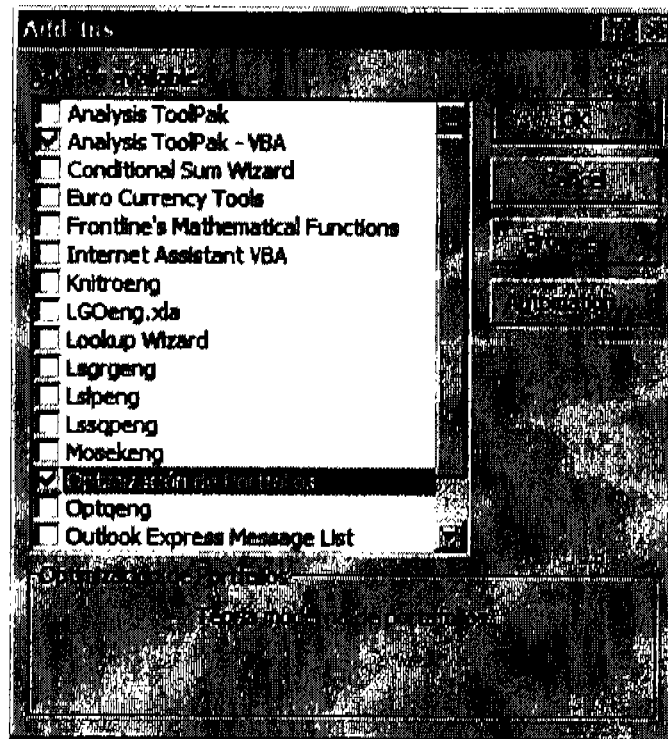


Figura 2.4 "Add-In" -Optimización de Portafolios-

Excel correrá automáticamente la rutina de inicio donde se mandan llamar las funciones de seguridad, creación del menú y validación de directorios.

Para facilidad en la lectura de este capítulo, el código de este trabajo se encuentra en el Apéndice "A" Código donde se podrán consultar las rutinas que usa el sistema.

2.1.1 Seguridad

A continuación se muestra la pantalla de seguridad llamada por el programa para recibir el usuario y contraseña o bien cancelar la validación y salir del sistema.

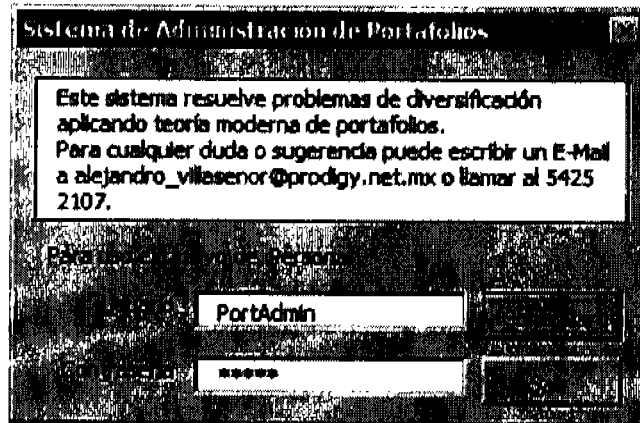


Figura 2.5 Seguridad

En la seguridad se podrá ingresar hasta un máximo de tres contraseñas antes de que se cierre la aplicación, pero en el caso de coincidir el usuario con la contraseña, la carga continuará. Si se trata de detener manualmente el procedimiento mientras se verifica su información se cerrará la aplicación.

2.1.2 Menú

Una vez validado el usuario, el sistema creará un nuevo menú antes de "Windows" llamado "SAP" como se muestra en la figura y de donde se podrán usar las funciones de la aplicación.

Dentro del menú SAP (ver Figura 2.6) podemos crear bases de datos, cargar datos de un archivo de texto a una base de datos, importar series desde la base de datos a hojas de cálculo, obtener rendimientos esperados, generar matrices de covarianzas, sacar los porcentajes de inversión de un portafolio óptimo, entre otras funciones.

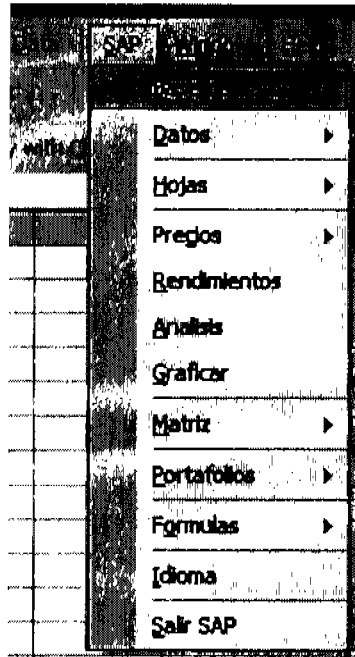


Figura 2.6 Menú del sistema

A continuación se muestra una tabla con cada uno de los elementos del menú, su rutina asociada que se puede consultar en el Apéndice A y una breve descripción de lo que hace cada una.

Menú	SubMenú	Rutina	Argumentos	Descripción
Base de Datos	Seleccionar	SeleccionaBD		Selecciona una BD
	Crear	CreaBD		Crea una Base de Datos (*.mdb)
	Compactar	CompactaBD		Compacta BD
Datos	Importar	ImportarDatosBD		Importa datos de una Base de Datos
	Cargar	CargarSeriesBD		Carga datos de un archivo a una BD
	Borrar	BorrarSeriesBD		Borrar datos de una Base de Datos
Hojas	Borrar	BorrarHoja		Borra hojas
Precios	Modificar	ModificarSeries		Completa y modifica las series
Rendimientos		ObtieneRends		Obtiene rendimientos de los precios
Análisis		CreaAnálisis		Obtiene análisis de los datos
Gráficas		CreaGrafica		Gráfica datos de las hojas
Matriz	Covarianzas	CreaMatCovCorr	"Covarianzas"	Matriz de covarianzas entre series
	Correlaciones	CreaMatCovCorr	"Correlaciones"	Matriz de correlaciones entre series
	Gráfica de Calor	GrafCalor		Identifica de los valores
	Excel Add-Inn	CambiaOpcionMenu	"Excel Add-In", "Matriz"	Crea la matriz usando Excel
Portafolios	Markowitz	CreaPortafolio	"Markowitz"	Creación de portafolios
	Límites	CambiaOpcionMenu	"Limits", "&Portfolios"	Establece límites para crear portafolios
Fórmulas	Análisis	CambiaOpcionMenu	"Análisis", "Fórmulas"	Conserva las fórmulas de análisis
	Matriz	CambiaOpcionMenu	"Matriz", "Fórmulas"	Conserva las fórmulas de matrices
	Portafolio	CambiaOpcionMenu	"Portafolio", "Fórmulas"	Conserva las fórmulas de portafolios
Idioma		CamblalIdioma		Cambiar Idioma
Salir		SalirSistema		Salir del Sistema

Tabla 2.1 Descripción de los elementos del menú

2.1.3 Estructura

Dado que el sistema usa datos de diferentes fuentes de información, vamos a necesitar un lugar donde almacenarla temporalmente para luego cargarlos a nuestra base de datos. El directorio "Data" contiene los archivos de texto que alimentarán la base de datos, en el directorio "DBase" residirán las bases de datos con la historia de los instrumentos y en el directorio de "Portfolios" estarán las hojas de cálculo con los portafolios.



El sistema revisa que los directorios antes mencionados existan dentro del directorio de la aplicación, si no es así, preguntará al usuario para crearlos. Si se encuentran todos, la aplicación seguirá, de lo contrario, se deshabilitará el menú.

2.2 Base de datos

Podemos almacenar y ordenar toda la información de una manera eficiente en una Base de Datos. En el siguiente menú “Base de Datos” (ver Figura 2.7) se pueden ejecutar las funciones de Crear, Seleccionar⁴ y Compactar una Base de Datos explicadas más adelante.

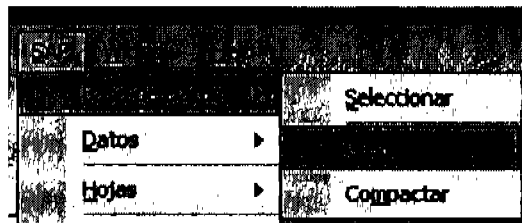


Figura 2.7 Menú Crear Base de Datos

2.2.1 Crear base de datos

Las bases de datos pueden ser creadas desde esta opción solamente indicando el nombre del archivo que deberá ser distinto a los que ya existen en el directorio “DBase”⁵.

⁴ Aún cuando en orden cronológico primero se debe crear una base de datos antes de seleccionarla, es más usada esta opción y por ello aparece primero dentro del menú.

⁵ Como se había visto anteriormente, en el directorio “DBase” están las Bases de Datos de MS Access.

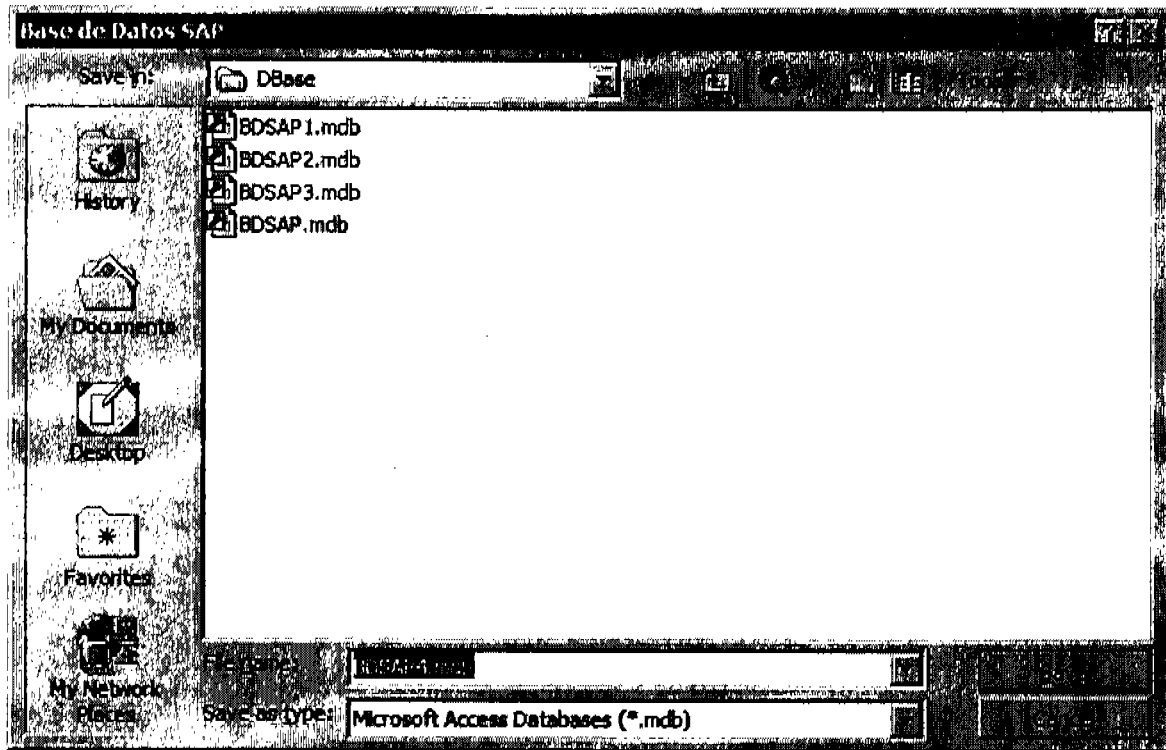


Figura 2.8 Nombre de la Base de Datos

Las Bases de Datos creadas tienen tres tablas: "Cat_TipoInstr" donde están los tipos de instrumentos; "Cat_Instr" donde están descritos todos los instrumentos que han sido cargados a la base de datos; y "Cotizacion" que tiene los valores de todas las series.

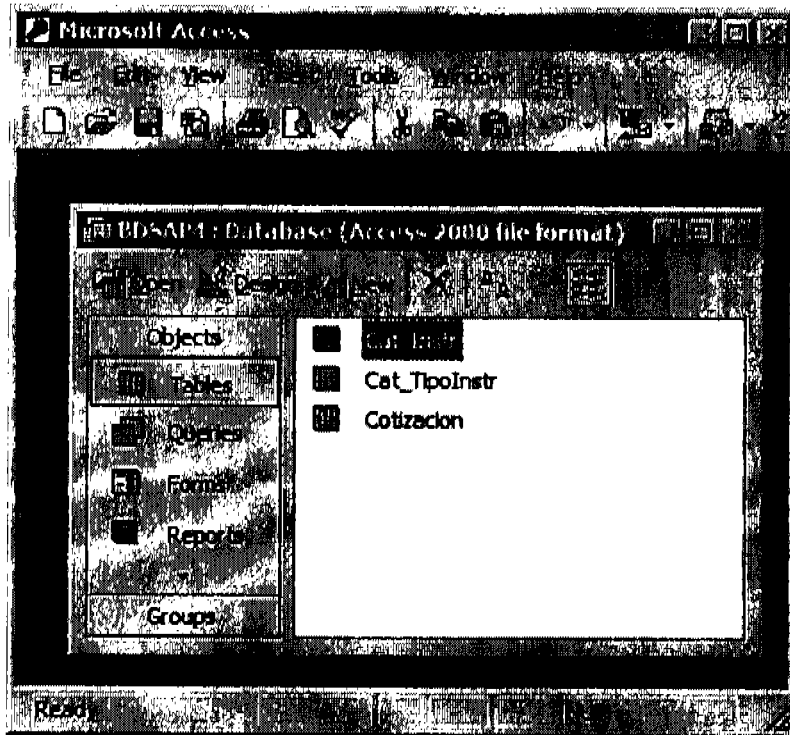


Figura 2.9 Tablas de la Base de Datos

Después de crear la Base de Datos entonces nos preguntará si desea que sea predeterminada, esto es, sea la base de donde se van a estar importando los datos a Excel.

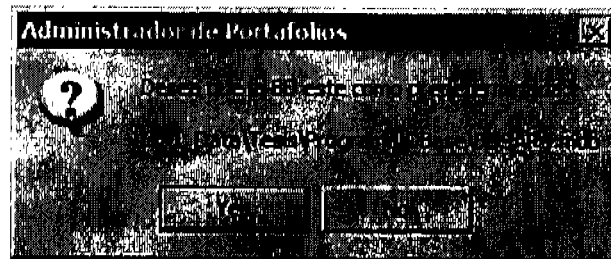


Figura 2.10 Establecer Base de Datos como predeterminada

2.2.2 Seleccionar base de datos

La aplicación nos da la facilidad de cambiar de Bases de Datos y para ello, únicamente tendremos entrar a la opción "Seleccionar" del menú y seleccionarla de la lista.

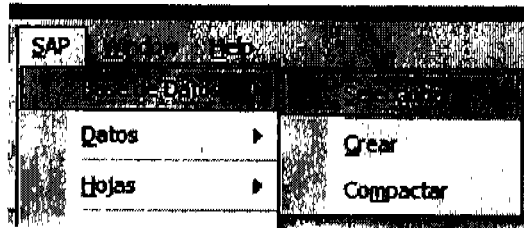


Figura 2.11 Menú Seleccionar Base de Datos

En la siguiente figura aparecerá la lista de todas las Bases de Datos existentes en el directorio "Dbase" de donde seleccionará alguna de ellas para importar las series.

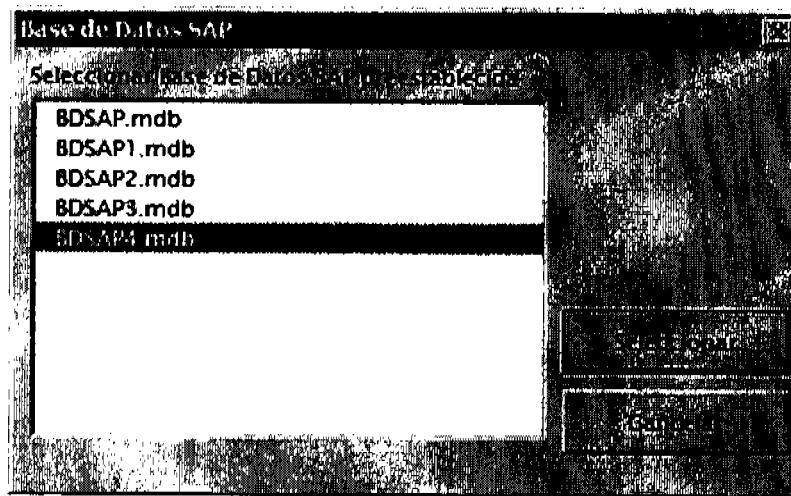


Figura 2.12 Lista seleccionar Base de Datos preestablecida

2.2.3 Compactar base de datos

La opción de compactar la Base de Datos se usa para mejorar su desempeño cuando ésta se encuentra fragmentada⁶. Después de hacer una carga de series desde un archivo de texto a la Base de Datos el sistema hace una compactación de la misma. Aún con esto, es posible que por

⁶ Cuando se borran datos u objetos de la Base de Datos quedan huecos y se dice que está fragmentada.

el uso, la Base de Datos se daña y se requiera compactar. Una manera de hacerlo es seleccionando “Compactar” de manera periódica para prevenir este problema⁷.

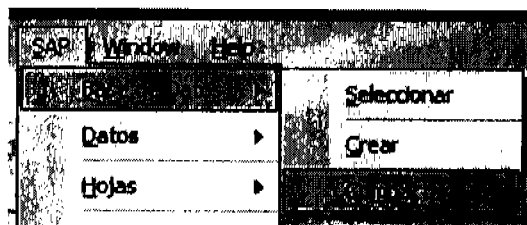


Figura 2.13 Menú Compactar Base de Datos

Posteriormente se debe seleccionar de la lista la base de datos y presionar compactar.

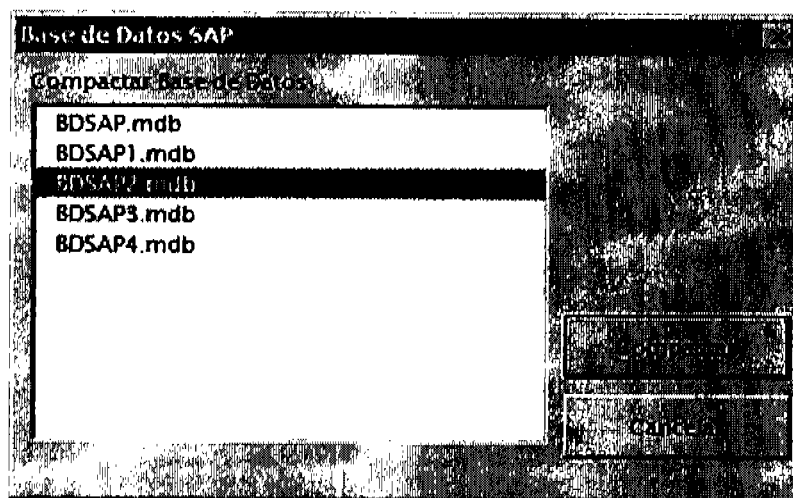


Figura 2.14 Lista de bases de datos a compactar

Este proceso puede durar varios segundos o minutos dependiendo del tamaño de la “Base de Datos”. Al final indicará que la Base de Datos ya fue compactada.

⁷ Para mayor información, consultar la documentación de MS Access.

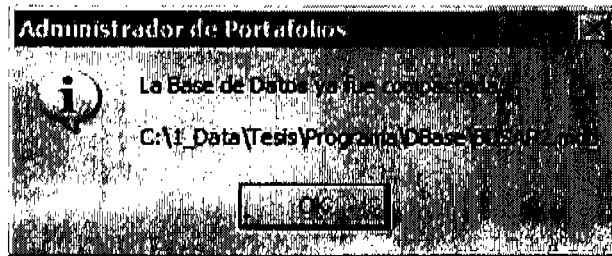


Figura 2.15 Mensaje de compactación de la Base de Datos

2.3 Datos

En el menú de Datos podemos Cargar a la Base de Datos así como Importar las series a Excel con la ventaja de no perder la fuente de información original. Otra función es de Borrar datos de la Base de Datos que ya no van a ser usados en nuestro análisis.

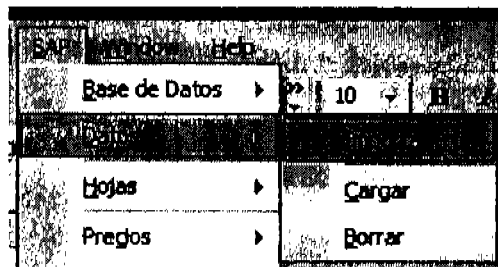


Figura 2.16 Menú Datos

2.3.1 Cargar información a la base de datos

Antes de hacer cualquier operación, necesitamos cargar la información ya que de lo contrario mandará un mensaje indicando que se necesita cargar previamente.

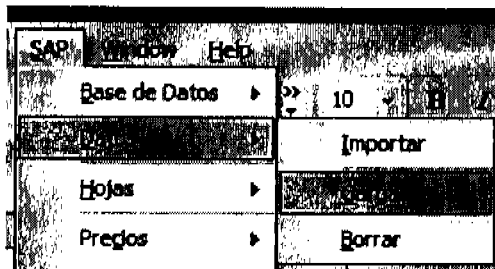


Figura 2.17 Menú Cargar Datos

La carga de información se podrá hacer a partir de archivos de texto con los datos del directorio de datos. Todos éstos aparecerán en una lista para ser seleccionados uno o varios archivos.

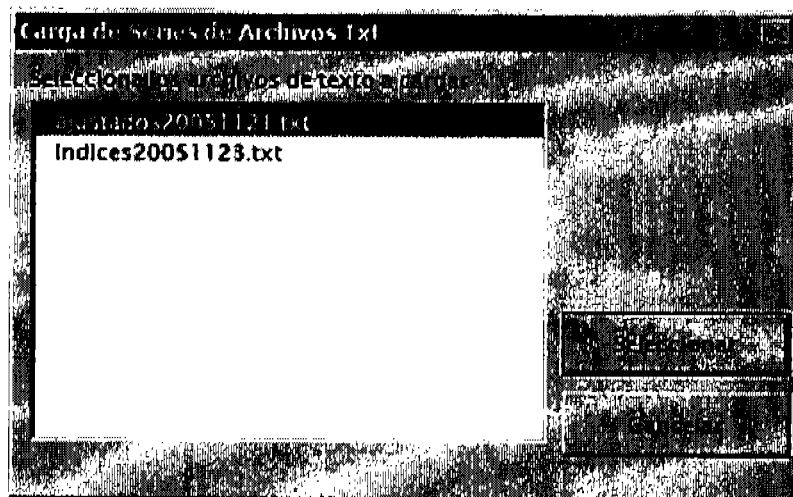


Figura 2.18 Lista de archivos de texto (*.txt)

Las series escogidas serán copiadas a la tabla "Cotizacion". La información desplegada es el Id del instrumento, nombre de la emisora y la serie, así como el tipo de instrumento.

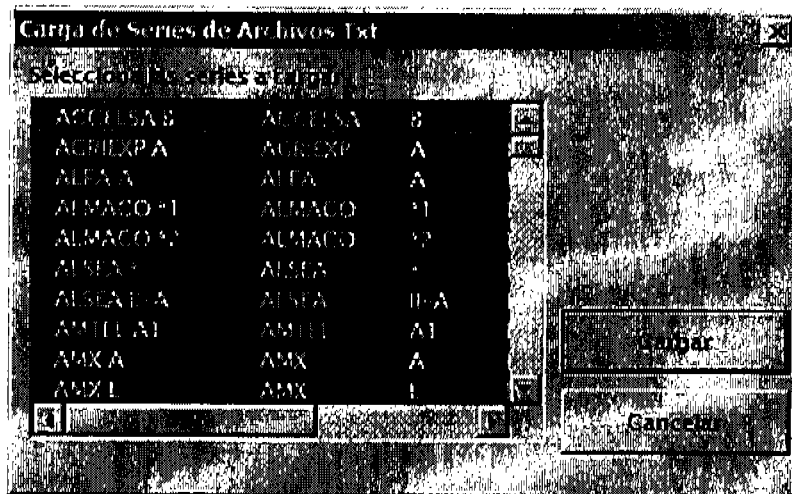


Figura 2.19 Series de Datos a cargar

El programa carga la información a la Base de Datos por medio de un vínculo (“link”) que se crea con el archivo de texto para hacer más rápido su lectura. También genera una tabla temporal donde se le asigna un nombre único a cada serie llamado “IdSerie” y que identifica al instrumento financiero que junto con la fecha, forman la llave primaria.

Es posible que tengamos que reemplazar los datos de algunas series ya existentes pero el sistema preguntará si desea reemplazarlas antes de hacer la carga para evitar que se pierda información. Una vez que hayamos incorporado nuevas series o después de reemplazarlas, el sistema calcula el número de datos para todas las series y compacta nuevamente la Base de Datos.

2.3.2 Importar datos a Excel

Una vez que cargamos la información a la Base de Datos podemos seleccionar e importar los instrumentos financieros a una hoja de cálculo Excel.

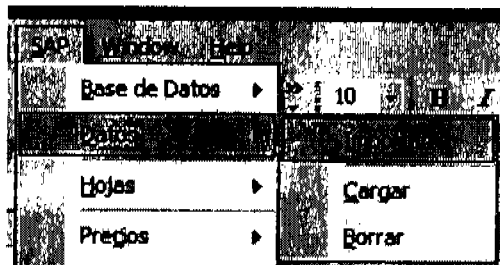


Figura 2.20 Menú Importar Datos a Excel

La opción de importar hace un primer filtro desplegando únicamente los instrumentos de los tipos de series seleccionados. Para seleccionar varios tipos, se necesita presionar **Ctrl** y con el mouse los tipos de instrumentos deseados o bien para secciones continuas es necesario seleccionar el primer tipo, presionar **Shift** y seleccionar el último tipo deseado. Automáticamente todos los tipos entre estos dos quedarán seleccionados.

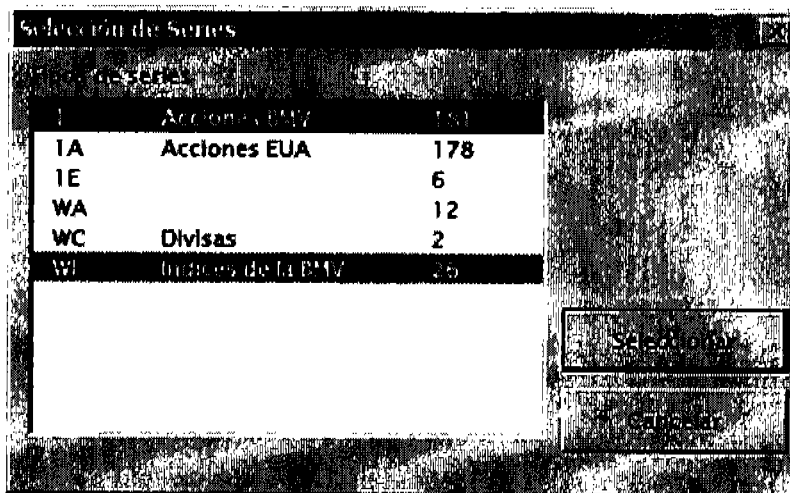


Figura 2.21 Tipos de series a importar

El tipo de serie de los instrumentos se puede ver a la derecha dentro de la misma pantalla como se ve en la Figura 2.22. En esta pantalla se podrán seleccionar las series individualmente.

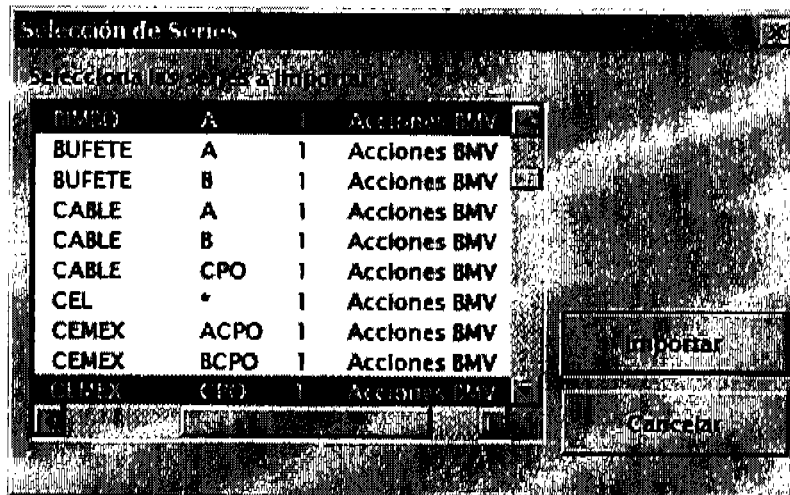


Figura 2.22 Series a importar

Una vez escogidas las series, tendremos que darle un nombre que identifique a nuestra nueva base en Excel y que formará parte del nombre de la hoja así como a cada una de las transformaciones o análisis posteriores⁸. Si queremos datos diferentes, una nueva selección se podrá importar. El nombre que llevará esta nueva hoja estará formada por el tipo de información (en este caso serán "Precios") que hay en la hoja, un número consecutivo y el nombre que le hayamos asignamos "Precios(# consecutivo)-< nombre de la base>", un ejemplo es "Precios(1)-BMV".

⁸ En el transcurso de este capítulo se verán los tipos de transformaciones y análisis que se podrán hacer a la base.



Figura 2.23 Ingreso del nombre de los datos a importar

El programa creará una nueva hoja de cálculo y vaciará la información seleccionada. De esta manera se evitará que reemplace información existente dentro del libro activo.

 A screenshot of Microsoft Excel showing a table of data. The table has 11 columns and 17 rows of data. The first column contains dates from 12/3/04 to 01/20/05. The second column contains numerical values ranging from 51.1 to 54.8. The remaining columns contain various numerical values. The table is titled "Precios(1)-BMW" in the bottom left corner.

Fecha	Valor 1	Valor 2	Valor 3	Valor 4	Valor 5	Valor 6	Valor 7	Valor 8	Valor 9	Valor 10
12/3/04	54.6	28.2	40.6	36.2	12.6	3.0	3.2	103.5	20.0	24.0
01/03/05	54.8	28.3	40.7	36.0	12.4	3.0	3.2	100.8	20.0	23.7
01/04/05	53.9	28.4	40.7	36.8	12.2	3.0	3.1	95.8	19.8	23.3
01/05/05	52.7	28.5	39.6	36.0	12.1	3.0	3.0	94.3	19.4	23.3
01/06/05	53.3	28.7	39.9	33.6	12.1	3.0	3.0	96.3	19.6	23.1
01/07/05	52.5	28.9	38.1	32.6	12.1	3.0	3.1	95.5	19.3	23.2
01/10/05	51.1	28.0	39.7	33.5	12.1	2.7	3.1	93.7	19.3	22.9
01/11/05	49.3	28.2	39.4	32.5	11.9	2.7	3.0	92.0	18.9	22.6
01/12/05	49.6	28.7	40.1	32.5	12.1	2.7	3.1	91.0	19.0	22.6
01/13/05	51.0	29.6	40.4	32.3	12.1	2.7	3.0	91.9	19.2	22.6
01/14/05	52.6	29.8	41.6	32.4	12.3	2.7	3.1	94.3	19.2	22.8
01/17/05	53.8	29.8	42.3	32.6	12.3	2.7	3.0	95.9	19.6	22.9
01/18/05	54.4	31.2	42.1	33.8	12.2	2.7	3.1	96.6	20.4	22.7
01/19/05	53.8	30.9	41.5	34.0	12.2	2.7	3.0	96.6	21.3	23.3
01/20/05	52.6	30.3	40.8	32.8	12.3	2.8	3.0	98.0	20.8	23.2

Figura 2.24 Datos importados a Excel

2.3.3 Borrar información de la base de datos

La opción Borrar Series de la Base de Datos tiene como finalidad depurar las series de acciones e índices que no van a ser analizadas o utilizadas para formar los portafolios. Solamente hay que seleccionar "Borrar" del menú "Datos".

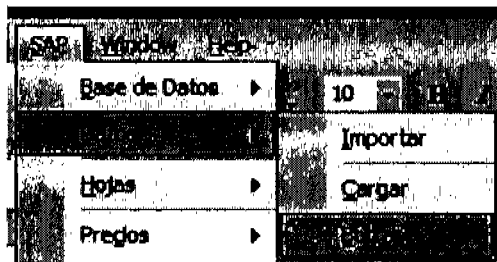


Figura 2.25 Menú Borrar Información de la Base de Datos

Luego hay que seleccionar los activos que ya no queremos. Siempre podremos restaurar la misma información a la base de datos si conservamos los archivos de texto y no hayan sido reemplazados.

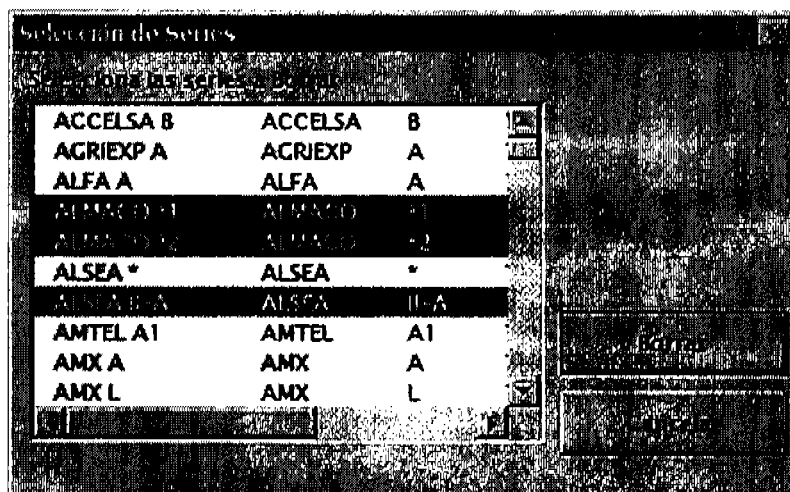


Figura 2.26 Series a borrar

2.4 Hojas

Como hemos visto, en cualquier libro de Excel se pueden almacenar las hojas de cálculo creadas por el sistema con la ventaja de que los resultados se pueden explotar como en una hoja de cálculo normal. Cada una de estas hojas tiene una identificación del tipo de información que el sistema pone automáticamente al momento de ser creadas aún cuando se haya modificado el nombre de la hoja. A continuación se muestra una lista de los tipos de hojas que el sistema usa y una breve descripción de ellos.

Tipo de Hoja	Descripción
Precios	Incluye índices, valores o precios de diferentes instrumentos a través del tiempo. Ésta se origina con la importación de la Base de Datos.
Rendimientos	Indica la razón de cambio de precios o índices de un periodo a otro.
Análisis	Contiene datos estadísticos acerca de las hojas de Precios o Rendimientos.
Gráfica	Son gráficas de la información de las hojas.
Matriz de Covarianzas	Contiene la matriz de Varianzas-Covarianzas.
Matriz de Correlaciones	Tiene la matriz de Correlaciones.
Gráfica de Calor	Es alguna de las matrices anteriores pero indicando mediante colores el grado en que un activo puede contribuir a bajar el riesgo del portafolio.
Portafolios	Contiene información de la composición y de los resultados de la optimización de portafolios.

Tabla 2.2 Tipos de hojas y su descripción

El sistema ofrece una herramienta que ayuda al manejo de la información y que hace que sea amigable con las siguientes características:

- Filtrar hojas por tipo. No solamente desplegará aquéllas con el tipo seleccionado si no que también se podrá desplegar todas aquellas hojas creadas por el sistema

- Seleccionar una o varias hojas. Previamente el sistema marcará aquellas que estén seleccionadas en Excel y el usuario podrá conservar o cambiar su selección. En algunas opciones, por su función, no se podrá seleccionar más de una hoja a la vez.
- Abrir y cerrar libros en el momento de la selección sin tener que salir de la herramienta que ofrece el sistema.
- Mostrar las hojas antes de ser usadas con la finalidad de que el usuario pueda ver su selección. En el momento que se cambie seleccione o deseccione una hoja el sistema la desplegará.

Como parte de las ventajas de usar Excel es que podemos crear, borrar u ordenar las series o bien modificar o rellenar los datos faltantes con procedimientos adicionales a lo que el sistema maneja. También podemos borrar información en el tiempo sin que se imposibilite la generación automática de portafolios.

2.4.1 Borrar hojas

El sistema permite eliminar las hojas que no son útiles para el análisis y evitar que el tamaño de la hoja de cálculo crezca y con ello que aumente el tiempo para hacer los cálculos. A continuación está la pantalla que muestra las hojas creadas por el sistema y su tipo antes de ser borradas.

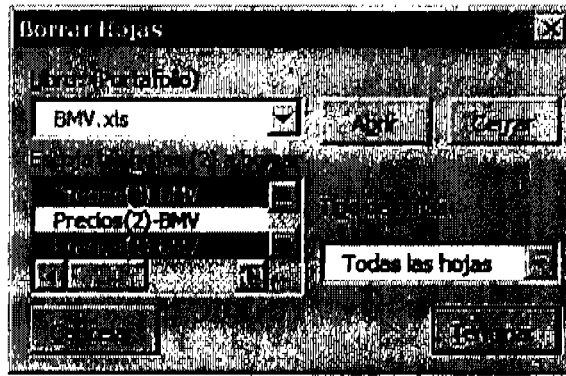


Figura 2.27 Borrar hojas

2.5 Precios

2.5.1 Modificar precios

Con frecuencia nos encontramos que no todas las series tienen información en las mismas fechas. Por ejemplo, algunos índices como la inflación se publican quincenalmente, o bien, las tasas de interés pueden presentarse semanal, mensual, etc.. Por ello es recomendable hacer una revisión de las series para poder hacer homogénea la información y tener una mejor entrada de datos. A continuación se muestra como el sistema ayuda a este propósito.

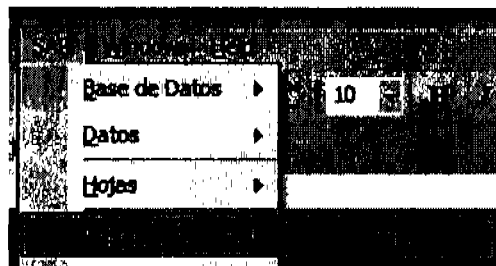


Figura 2.28 Modificar hojas

Dentro de las opciones que tenemos para modificar las series de precios están:

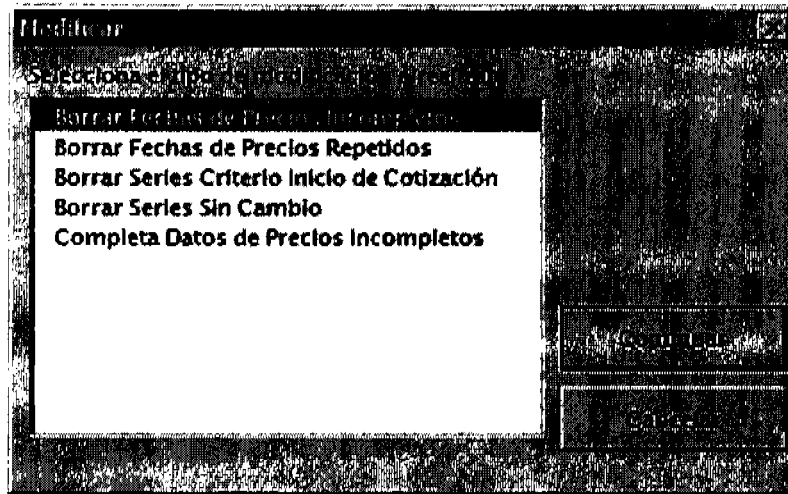


Figura 2.29 Tipo de modificación a precios

a) "Borrar Fechas de Precios Incompletos"

Esta opción borra el renglón completo de la o las fechas en que no exista al menos un valor en los datos seleccionados como se muestra a continuación.

01/01/03	35.9					3.8
01/02/03						3.8
01/03/03						3.8
01/06/03						3.8
01/07/03						3.8
01/08/03						3.8
01/09/03						3.8
01/10/03	35.8	35.6	13.8	8.4	7.4	3.8

Figura 2.30 Precios incompletos

Entonces el programa preguntará si desea borrar la fecha antes de ser eliminada. Si escoge "Si" entonces solamente borrará la fecha por la que esta preguntando, si escoge "Si Todo" borrará todas las fechas sin que vuelva a preguntar, si escoge "No", no borrará dicha fecha y seguirá validando la información y por último, "Cancelar" terminará con la búsqueda de siguientes datos faltantes.



Figura 2.31 Pregunta para borrar fechas sin datos completos

Un uso frecuente de esta opción es cuando tenemos series como Cetes u otros instrumentos cuyos datos no existan diariamente o al menos no con la misma frecuencia.

b) "Borrar Fechas de Precios Repetidos"

Con frecuencia la información se repite solamente para completar la información de algunas series en las fechas donde no cotiza. Si deseamos, esta opción quita los periodos en los que todos los precios seleccionados se repiten con la fecha anterior.

12/27/02	173.8				6,120.2
12/30/02					6,124.5
12/31/02					6,127.1
01/01/03					6,127.1
01/02/03					6,226.4
01/03/03					6,263.3
01/06/03					6,333.6
01/07/03	171.4	108.8	216.8	15.2	6,279.3
01/08/03	171.4	108.8	216.8	15.2	6,290.6

Figura 2.32 Precios repetidos

Como en todas las opciones, el programa preguntará si desea modificar los datos para este caso o también para los siguientes.

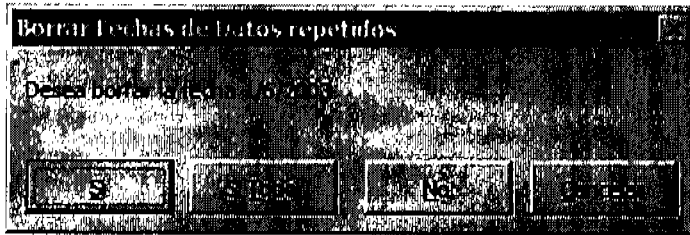


Figura 2.33 Pregunta para borrar fechas de datos repetidos

c) "Borrar Series Criterio Inicio de Cotización"

Con esta opción se busca que las series que no tienen datos en el rango seleccionado sean borradas. Esto se puede presentar cuando las series no empiezan a cotizar desde el inicio.

01/01/01	177.0				
01/02/01					
01/03/01					
01/04/01					
01/05/01					
01/08/01					
01/09/01	177.0	80.6	21.2	5,908.5	
01/10/01	177.0	80.6	222.2	21.2	5,915.0

Figura 2.34 Borra series criterio inicio de cotización

Si no existe al menos un dato en el rango seleccionado para alguna de las series, se pregunta si desea borrarla.

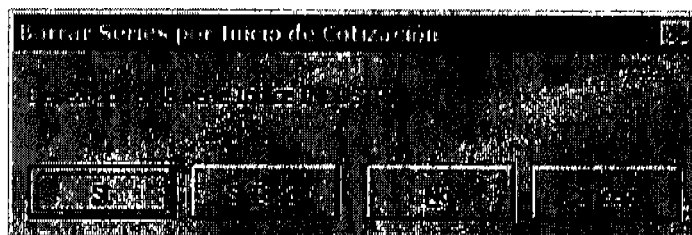
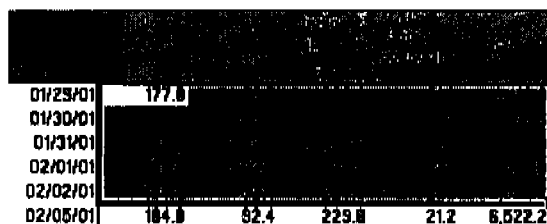


Figura 2.35 Pregunta para borrar las series criterio inicio de cotización

d) "Borrar Series Sin Cambio"

Usamos esta opción cuando existen algunas series que no presentan suficientes cambios y que no deseáramos tomarlas en cuenta para el estudio.



01/23/01	177.0				
01/30/01					
01/31/01					
02/01/01					
02/02/01					
02/05/01	184.0	52.4	229.0	21.2	6,522.2

Figura 2.36 Borra series sin cambio

El programa preguntará el porcentaje mínimo de cambio requerido en el rango de datos seleccionado, esto es, el porcentaje de veces que deseáramos que las series tuvieran al menos un cambio en su valor.

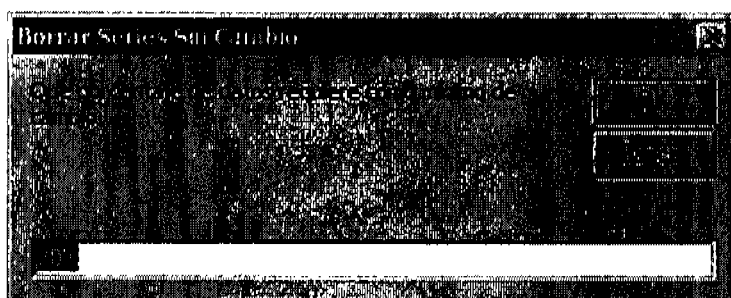


Figura 2.37 Pregunta porcentaje de cambio

Se preguntarán si desea borrar aquellas series que tienen un porcentaje menor o igual de cambio.

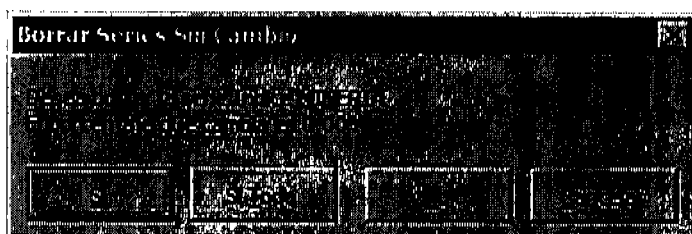
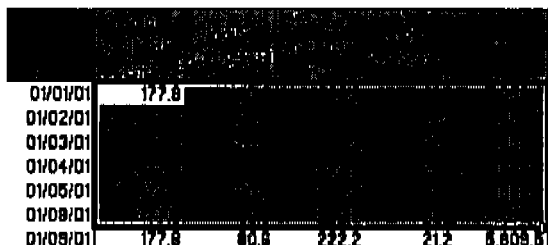


Figura 2.38 Pregunta para borrar las series sin cambio

e) "Completa Datos de Precios Incompletos"

Los datos en algunas series pueden faltar por algún error en el proceso de captura manual o automatizada, o bien simplemente no existen tales valores y que no representan una variación significativa para la estimación de los datos. Esta opción simplemente toma el dato anterior para llenar la serie.



01/01/01	177.8				
01/02/01					
01/03/01					
01/04/01					
01/05/01					
01/08/01					
01/09/01	177.8	80.8	222.2	21.2	8,809.5

Figura 2.39 Completa datos de precios incompletos

Para cada dato faltante se pregunta si desea completar la serie con el dato de la fecha anterior.



Figura 2.40 Pregunta para completar datos de precios incompletos

2.6 Rendimientos

Los precios son usados por el sistema para calcular los rendimientos históricos. El siguiente menú hace dicho cálculo.

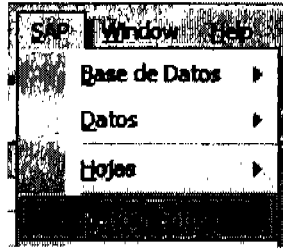


Figura 2.41 Menú calcular rendimientos

Se genera una nueva hoja de rendimientos definidos como el cambio porcentual entre el precio anterior y el actual. Una muestra de los datos generados se ve a continuación.

FECHA	PREVIO	PREVIO	PREVIO	PREVIO	PREVIO	PREVIO	PREVIO	PREVIO	PREVIO	PREVIO
01/03/05	0.68 %	0.50 %	0.20 %	2.48 %	-0.64 %	0.87 %	-2.17 %	-2.64 %	-0.13 %	-1.29 %
01/04/05	-1.71 %	0.32 %	-0.09 %	-0.81 %	-1.77 %	0.00 %	-3.17 %	-4.92 %	-0.77 %	-1.65 %
01/05/05	-2.26 %	0.36 %	-2.53 %	-2.23 %	-1.23 %	0.00 %	-1.97 %	-1.60 %	-2.19 %	0.00 %
01/06/05	1.16 %	0.89 %	0.63 %	-4.14 %	0.58 %	0.00 %	0.33 %	1.06 %	1.93 %	-0.94 %
01/07/05	-1.61 %	0.42 %	-1.97 %	-2.99 %	-0.58 %	0.00 %	2.67 %	0.24 %	-2.16 %	0.52 %
01/10/05	-2.59 %	-2.96 %	1.42 %	2.92 %	0.00 %	-8.87 %	0.32 %	-1.94 %	-0.02 %	-1.26 %
01/11/05	-3.59 %	0.64 %	-0.54 %	-2.87 %	-1.24 %	-1.46 %	-2.27 %	-1.77 %	-2.16 %	-1.79 %
01/12/05	0.70 %	1.74 %	1.72 %	-0.16 %	1.28 %	0.00 %	0.99 %	-1.09 %	0.46 %	0.44 %
01/13/05	2.87 %	3.94 %	0.72 %	-0.58 %	0.09 %	0.00 %	-0.33 %	1.03 %	0.93 %	0.00 %
01/14/05	3.15 %	-0.20 %	2.86 %	0.29 %	1.59 %	0.00 %	1.97 %	2.57 %	-0.02 %	0.99 %

Figura 2.42 Hoja de rendimientos

2.7 Análisis

Aún cuando lo que le interesa a este modelo (Markowitz) son el riesgo y el rendimiento esperados para usarlos como estimados para la creación de portafolios, dentro del sistema, podemos hacer una exploración inicial de las series para conocer su comportamiento de manera

individual y poder hacer una mejor selección de éstas. A continuación se presenta la lista de las funciones que podemos aplicar y una breve explicación.

Función	Descripción
Máximo	Valor máximo
Mínimo	Valor mínimo
Mediana	Mediana
Total	Número de datos (no se cuentan los vacíos)
Promedio	Promedio aritmético
Prom. Pond.	Promedio ponderado
Prom. Geom.	Promedio geométrico
Prom. Armon.	Promedio armónico
Varianza	Varianza definido como el cuadrado de la desviación de los datos de su promedio
Desv. Estand.	Raíz cuadrada de la varianza
Sesgo	Sesgo
Curtosis	Curtosis
Normalidad	Arroja el valor de una prueba de normalidad
Prueba	Prueba de normalidad (Jarque-Vera)
Beta	Pendiente (m)
Alfa	Ordenada al origen
Var. Residual	Varianza Residual
Cuartil 50	Cuartil 50
Cuartil 25	Cuartil 25
Cuartil 75	Cuartil 75

Tabla 2.3 Descripción de Funciones de análisis

El sistema ofrece la opción de pegar dichos resultados como valores o bien dejar las fórmulas que se recalculan cuando queramos hacer otra operación, haciendo el proceso muy lento en especial con una cantidad de datos considerable. La opción de fórmulas sirve para poder hacer un seguimiento de los cálculos o bien para hacer escenarios con los datos iniciales (que pasaría si...), caso contrario se recomiendan usar valores. Esta opción se puede seleccionar presionando "Análisis" en el menú "Formulas". Una marca ("✓") indica que en las siguientes hojas creadas, se conservarán las fórmulas.

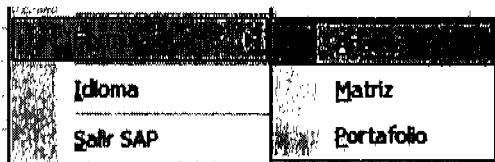


Figura 2.43 Preferencia de Fórmulas en las hojas de Análisis

El análisis se puede hacer con varias funciones a la vez desplegando todos los valores en la misma hoja.

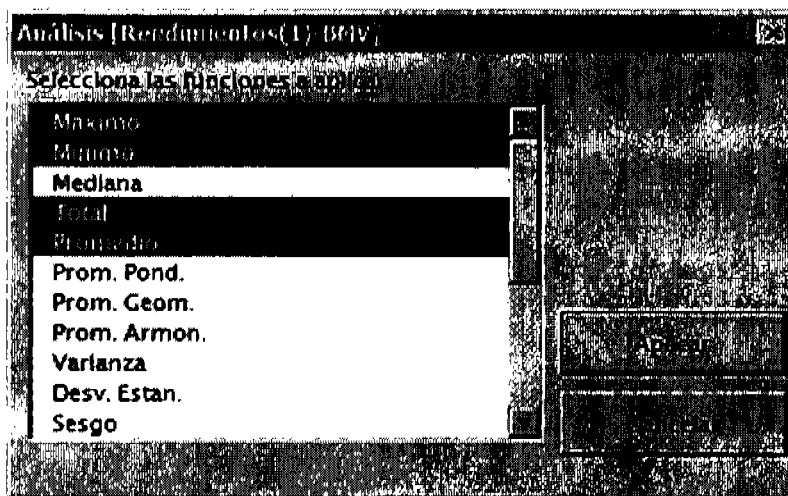


Figura 2.44 Selección múltiple de funciones de análisis

Se deberá escoger la fecha de análisis y especificar el tamaño de la muestra de datos como se muestra en la siguiente figura.

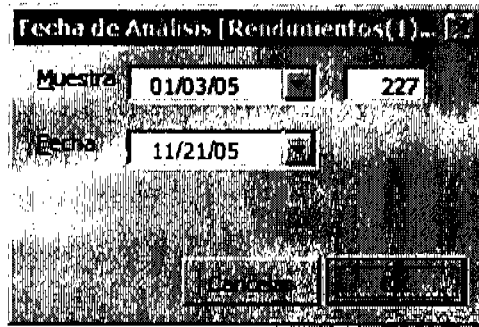


Figura 2.45 Fecha de análisis múltiple

En este caso, la hoja de resultados quedará con las fórmulas como se ve en la siguiente figura pero será necesario conservar los datos a los cuales hace referencia ya que de lo contrario arrojarían errores en los resultados.

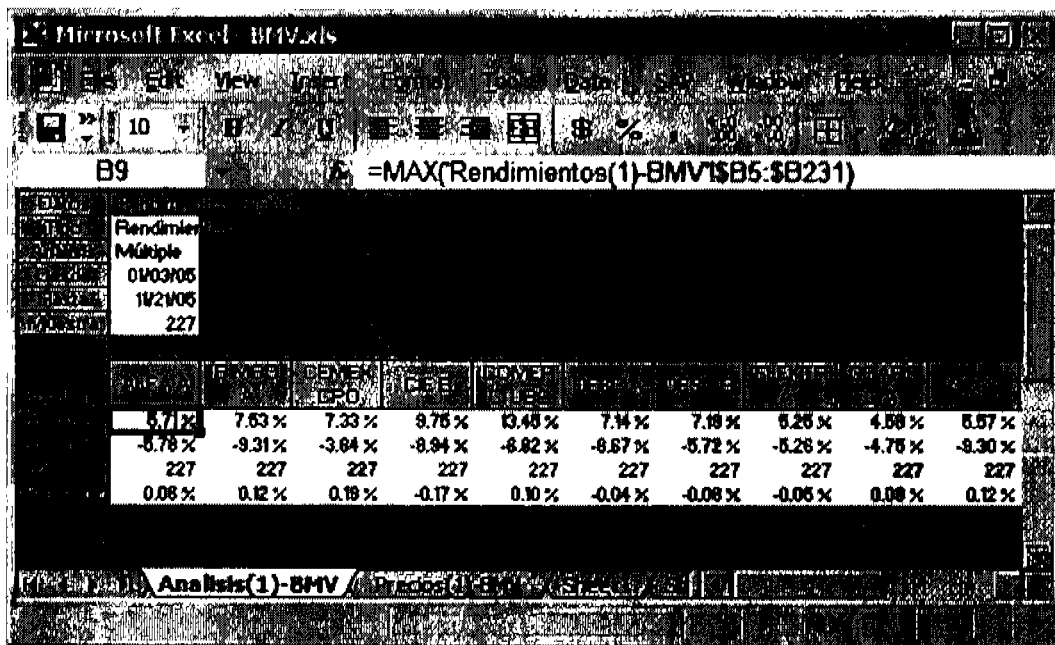


Figura 2.46 Resultados de análisis múltiple con fórmulas

Si solamente seleccionamos una fórmula como se muestra en la siguiente figura podemos obtener dicha función a través del tiempo.

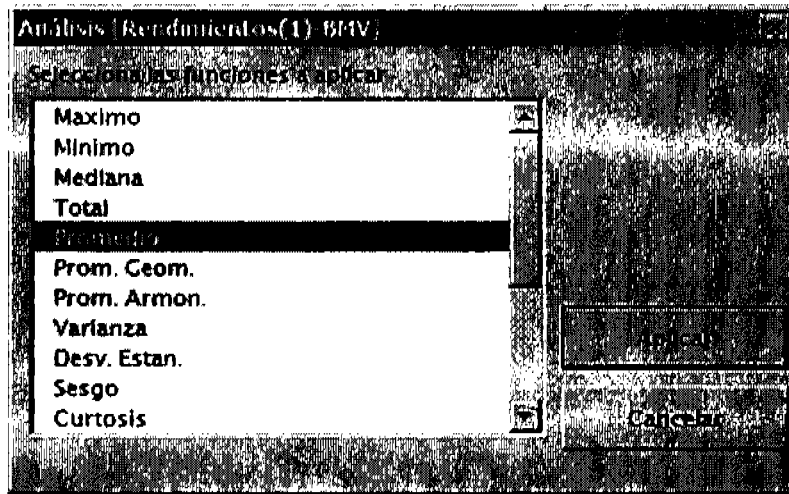


Figura 2.47 Selección de una función para análisis

Para especificar el periodo y el tamaño de los datos de muestra se despliega la siguiente pantalla donde aparecerá un dato más, hasta que tiempo queremos hacer el análisis.

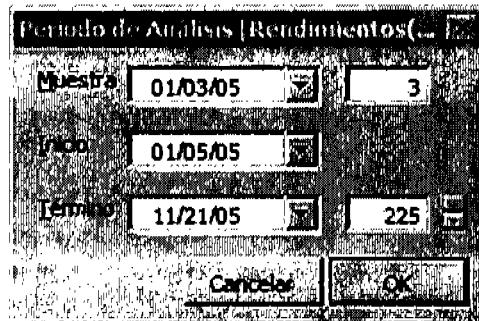


Figura 2.48 Periodo de análisis de una sola función

La ventana de datos a la cual se le aplica la función se va desplazando de tal manera que el tamaño de muestra permanezca constante hasta el término del análisis. El resultado mostrado a una cierta fecha (i) corresponde a los datos tomados desde $i-n+1$ hasta el i -ésimo dato.

Para el caso específico de usar las funciones "Máximo", "Mínimo", "Mediana", "Promedio" y "Prom. Geom." la hoja de resultados sería igual a la que la originó (Precios o Rendimientos) ya que dicha transformación generaría una del mismo tipo. Por ejemplo, escoger la función de

promedio sobre los rendimientos de un cierto periodo generaría otra hoja de rendimientos llamada “promedios móviles”.

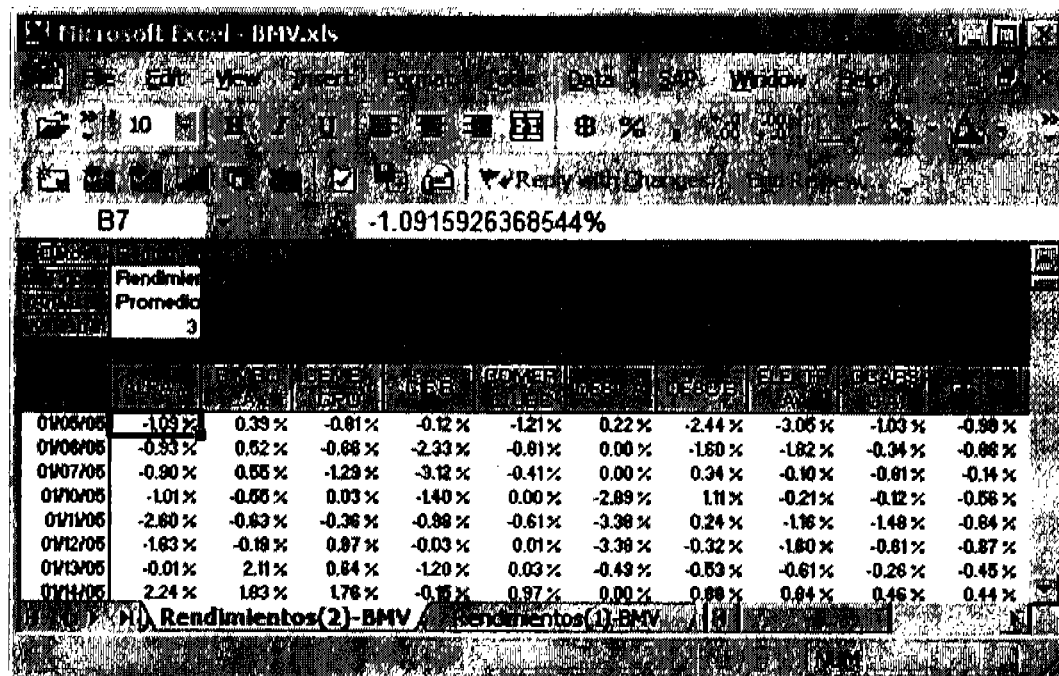


Figura 2.49 Promedios móviles

Algunas de las funciones se describen a continuación.

2.7.1 Promedio aritmético

La siguiente función es la que utiliza para el promedio aritmético:

$$d_{t,j}^* = \frac{1}{n} \left(\sum_{i=t-n+1}^t d_{i,j} \right)$$

Donde:

n = rezago

$t =$ tiempo t

$j =$ j-ésimo instrumento

$d^* =$ dato transformado

$d =$ dato original

El rezago indica el número de datos que toma para hacer el promedio incluyendo el dato actual.

La serie transformada tendrá $m-n+1$ datos, donde m es la cantidad de datos de la serie original.

Esto es, $n-1$ datos menos que la serie original.

Ejemplo:

Tiempo (t)	Serie 1	Serie 2
1	14.50	7.20
2	14.42	6.80
3	14.71	6.55
4	14.94	6.70
5	15.20	6.43
6	15.05	6.26
7	15.35	6.15

$n=4,$

$$d_{5,1}^* = \frac{1}{4} \left(\sum_{i=2}^5 d_{i,1} \right) = \frac{1}{4} (d_{2,1} + d_{3,1} + d_{4,1} + d_{5,1}) = 14.8175 \approx 14.82$$

Tiempo (t)	Serie 1	Serie 2
4	14.64	6.81
5	14.82	6.62
6	14.98	6.49
7	15.14	6.39

2.7.2 Promedio ponderado

Una generalización del promedio aritmético es el promedio ponderado donde se le asignan pesos a cada uno de los datos.

Fórmula:

$$d_{t,j}^* = \sum_{i=t-n+1}^t (d_{i,j} \times w_{\text{mod}(i-1,n)+1})$$

Donde:

$w_1 \cdots w_n$ = ponderaciones

n = rezago

t = tiempo t

j = j -ésimo instrumento

d^* = dato transformado

d = dato original

$\text{mod}(i-1,n)+1$ = remanente de i/n

El rezago es la cantidad de pesos (o ponderaciones) que toma para hacer el promedio ponderado.

El rango debe tener una columna y la suma de sus ponderaciones suma uno. Si solamente tomamos dos datos como muestra, lo llamaremos suavización exponencial.

$$\sum_{k=1}^n w_k = 1$$

Nota: La serie transformada tendrá n-1 periodos menos que la serie original.

Ejemplo:

Vector W:

# Ponderación	w
1	0.2
2	0.3
3	0.5

Tiempo (t)	Serie 1	Serie 2
1	14.50	7.20
2	14.42	6.80
3	14.71	6.55
4	14.94	6.70
5	15.20	6.43
6	15.05	6.26
7	15.35	6.15

La cantidad de datos que se toman es el rezago (n=3).

$$d_{4,2}^* = \sum_{k=1}^3 (d_{i,2} \times w_k) = d_{2,2} \times w_1 + d_{3,2} \times w_2 + d_{4,2} \times w_3 = 6.675 \approx 6.68$$

Tiempo (t)	Serie 1	Serie 2
3	14.58	6.76
4	14.77	6.68
5	15.02	6.54
6	15.07	6.40
7	15.23	6.24

2.7.3 Normalidad

Permite evaluar si la muestra se distribuye como una normal en la que solamente se necesitan dos momentos (media y varianza). Si los datos no se distribuyen normalmente no se puede hacer inferencia estadística (pruebas de t, de F y χ^2). Una serie distribuida normalmente significa que su valor esperado es igual a la media y que su varianza es igual a una constante pequeña.

2.8 Gráficas

El tipo de gráfica a elegir depende de los datos que se van a usar (precios, rendimientos, portafolio). Estos pueden ser:

1. De dispersión (riesgo vs. rendimiento)
2. Gráficas en el tiempo
3. Gráfica de barras o líneas

2.9 Matriz de varianzas-covarianzas y correlaciones

Así como el rendimiento esperado es necesario para la creación de portafolios, también necesitamos de una matriz de Varianzas-Covarianzas para calcular su riesgo esperado. Aún cuando en la creación de los portafolios las matrices son creadas automáticamente, podemos hacer un análisis previo a dicha corrida como se muestra en la siguiente figura.

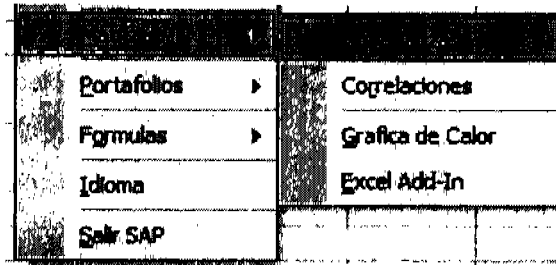


Figura 2.50 Menú para la creación de la matriz de Varianzas-Covarianzas

Los resultados se desplegarán en una hoja nueva con el nombre de MatCov(#)-<nombre> para matriz de Varianzas-Covarianzas.

Pendientes: 01/03/05 1/21/05 227 Muestral FALSE						
	0.00026	0.00008	0.00008	0.00012	0.00002	0.00011
	0.00008	0.00035	0.00010	0.00023	0.00004	0.00010
	0.00008	0.00010	0.00029	0.00011	0.00001	0.00008
	0.00012	0.00023	0.00011	0.00002	0.00003	0.00012
	0.00002	0.00004	0.00001	0.00002	0.00018	0.00002
	0.00011	0.00010	0.00008	0.00012	0.00002	0.00029
	0.00008	0.00007	0.00005	0.00011	0.00002	0.00006

Figura 2.51 Matriz de Varianzas-Covarianzas

Una hoja con el nombre MatCor(#)-<nombre> para matriz de Correlaciones será creada como se ve a continuación.

Plendimia	01003406	11/21/06	227	Total	FALSE		
	1.0000	0.2963	0.3406	0.3144	0.0603	0.4129	0.3717
	0.2963	1.0000	0.3277	0.5326	0.1576	0.3198	0.2666
	0.3406	0.3277	1.0000	0.2780	0.0404	0.2819	0.1963
	0.3144	0.5326	0.2780	1.0000	0.1091	0.3258	0.3361
	0.0603	0.1576	0.0404	0.1091	1.0000	0.0787	0.0611
	0.4129	0.3198	0.2819	0.3258	0.0787	1.0000	0.2336
	0.3717	0.2666	0.1963	0.3361	0.0611	0.2336	1.0000

MatCor(1)-BMV

Figura 2.52 Matriz de Correlaciones

Hay que recordar que se necesitarán cuando menos tres datos por serie y cuando menos dos series para crear las matrices.

El sistema genera las matrices de dos maneras, la primera es por medio de fórmulas que el sistema copia y la segunda es llamando la herramienta "Data Analysis" que viene en el paquete Excel. Ambas son rápidas, pero si queremos hacer escenarios debemos conservar las fórmulas que crea el sistema tal y como se muestra en la siguiente figura:

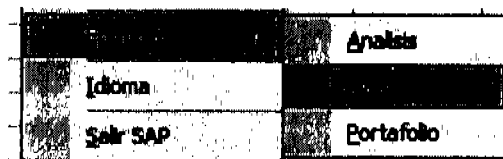


Figura 2.53 Opción de conservar fórmulas en la creación de matrices

Para usar la herramienta que ofrece Excel debemos marcar en el menú la opción "Excel Add-In".

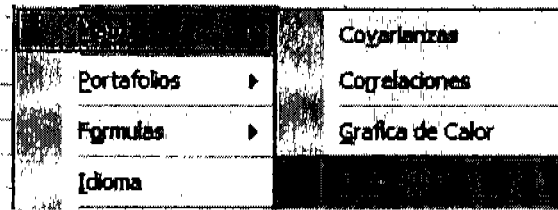


Figura 2.54 Menú para la creación de matrices usando "Data Análisis" de Excel

También podemos usar una "gráfica de calor" para identificar a partir de una matriz de Varianzas-Covarianzas o de una matriz de Correlaciones aquellos instrumentos que hacen que el portafolio incremente o reduzca significativamente su riesgo. Esta opción facilita la búsqueda con colores "cálidos" o rojos aquellos valores más negativos y con colores "fríos" o azules aquéllos más positivos.

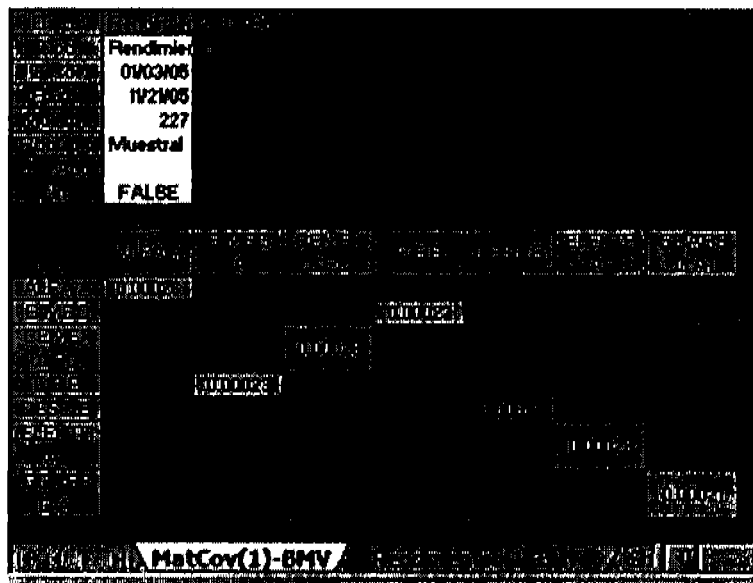


Figura 2.55 Gráfica de Calor

2.10 Portafolios

En la creación de portafolios requerimos de rendimientos que tomará el sistema para generar hojas temporales en el momento de ejecución. Si especificamos que queremos conservar las

fórmulas, como se ve en la siguiente figura, tendremos todas las hojas que se hayan usado, de lo contrario las hojas temporales serán borradas de manera automática.

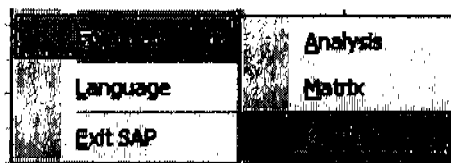


Figura 2.56 Opción de conservar fórmulas en la creación de portafolios

Los dos tipos de hojas “temporales” que se generan son la de rendimientos estimados y de matriz de varianza y covarianzas. La primera usa la función promedio aritmético vista en la sección de “Análisis” para formar una hoja de “promedios móviles”. El segundo tipo es el de Varianzas-Covarianzas y esta es creada tantas veces como puntos en el tiempo hayamos escogido.

El primer paso para empezar a formar los portafolios es seleccionando “Markowitz” del menú de Portafolios.



Figura 2.57 Creación de portafolios

A continuación escogemos la hoja de rendimientos observados para obtener las entradas al modelo.

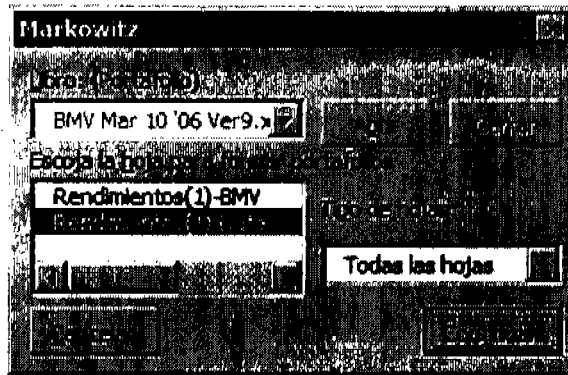


Figura 2.58 Selección de hoja de rendimientos para crear portafolios

Posteriormente vamos a seleccionar el tipo de portafolios que queremos obtener. Éstos pueden ser:

- de Mínima Varianza (PMV). Estos portafolios son los puntos que están en la punta bala en la gráfica de riesgo-rendimiento.
- con Riesgo \leq a Si conocemos el riesgo que un inversionista está dispuesto a incurrir, podemos ingresarle al modelo dicho dato para que forme los portafolios con el mejor rendimiento.
- con Rendimiento \geq a Si estamos tratando de alcanzar un rendimiento entonces generarán portafolios con el menor riesgo esperado.
- de la Frontera. Esta opción calculará portafolios a lo largo de la frontera óptima.
- puntos Factibles. Esta opción enseña no solamente la frontera óptima si no una muestra de portafolios dentro del área factible.

En la siguiente pantalla se podrá escoger alguna de las opciones antes mencionadas.

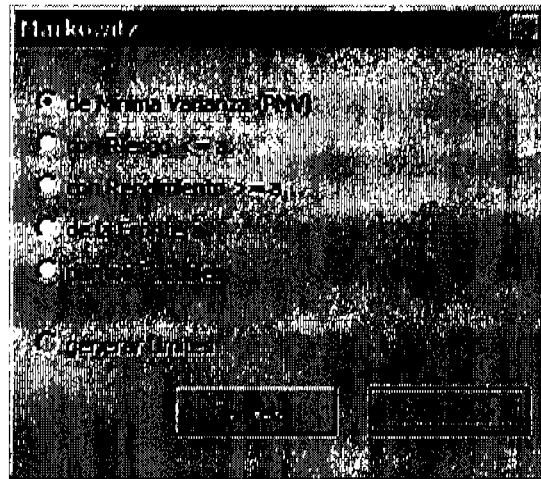


Figura 2.59 Selección de tipo de portafolio a calcular

Por último hay que seleccionar el periodo y la muestra en la cual queremos formar los portafolios. Cabe destacar que el tamaño de la muestra permanecerá constante en todo momento incorporando los datos mas recientes y eliminando los anteriores.

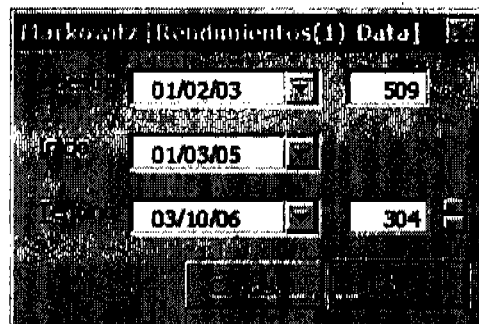


Figura 2.60 Selección de la muestra y el periodo para formar portafolios

2.10.1 Portafolios de mínima varianza (PMV)

Crearé una hoja de los portafolios formados con los datos de rendimiento y varianza esperadas, y el porcentaje invertido de cada uno de los activos. A continuación se muestra un ejemplo.

04/05/06										
PMV										
01/03/06										
03/10/06										
509										
12/30/05	6.46E-06	0.12 %	100.00 %	0.62 %	14.17 %	8.18 %	9.37 %	0.00 %	4.68 %	2.51 %
01/02/06	6.41E-06	0.11 %	100.00 %	0.70 %	14.08 %	8.20 %	9.36 %	0.00 %	4.71 %	2.47 %
01/03/06	6.39E-06	0.11 %	100.00 %	0.78 %	14.88 %	8.03 %	9.26 %	0.04 %	4.67 %	2.46 %
01/04/06	6.18E-06	0.11 %	100.00 %	0.64 %	15.09 %	8.52 %	9.58 %	0.27 %	4.59 %	2.38 %
01/05/06	6.17E-06	0.11 %	100.00 %	0.55 %	14.97 %	8.55 %	9.48 %	0.32 %	4.80 %	2.38 %
01/06/06	6.19E-06	0.11 %	100.00 %	0.68 %	15.05 %	8.48 %	9.48 %	0.33 %	4.81 %	2.38 %
01/08/06	6.17E-06	0.11 %	100.00 %	0.57 %	15.28 %	8.42 %	9.44 %	0.33 %	4.83 %	2.38 %
01/10/06	6.15E-06	0.11 %	100.00 %	0.46 %	15.47 %	8.34 %	9.38 %	0.28 %	4.91 %	2.38 %
01/11/06	6.15E-06	0.11 %	100.00 %	0.46 %	15.48 %	8.36 %	9.37 %	0.26 %	4.94 %	2.38 %
01/12/06	6.15E-06	0.11 %	100.00 %	0.46 %	15.53 %	8.37 %	9.38 %	0.28 %	4.96 %	2.38 %

Figura 2.61 Portafolios de Mínima Varianza con una muestra de dos años

Como se puede ver en el ejemplo anterior, los portafolios cambian conforme avanza el tiempo debido a que se incorpora nueva información. Si nuestro tamaño de muestra es muy grande, los cambios serán marginales, esto es, el cambio en el portafolio será inversamente proporcional al tamaño de la muestra.

2.10.2 Portafolios con riesgo menor o igual a ...

Para formar este tipo de portafolios necesitamos conocer el riesgo que estaríamos dispuestos a incurrir en términos de varianza. Previamente, el usuario puede generar los puntos sobre la frontera y así ubicar su riesgo ya que de lo contrario podría ingresar una varianza que no es posible alcanzar (menor al PMV) o bien ser demasiado holgada generando soluciones de “esquina” donde se invertirá todo en el activo con mayor rendimiento esperado.

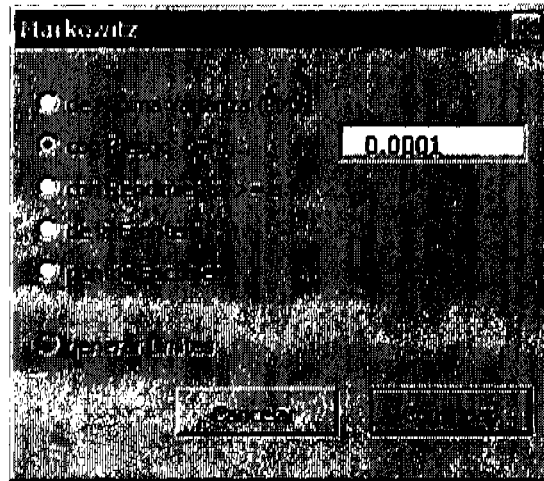


Figura 2.62 Selección portafolios con Riesgo menor o igual a...

En el ejemplo que se muestra a continuación se puede observar que aún para muestras de dos años de información los porcentajes cambian varios puntos porcentuales de un día a otro.

04/08/06										
Riesgo										
12/30/05										
03/10/06										
0.0001										
12/30/05	0.0001	0.28 %	100.00 %	10.26 %	3.96 %	5.14 %	2.22 %	0.90 %	6.48 %	0.13 %
01/02/06	0.0001	0.28 %	100.00 %	9.82 %	4.28 %	5.35 %	2.83 %	0.68 %	5.65 %	0.38 %
01/03/06	0.0001	0.28 %	100.00 %	2.38 %	3.68 %	8.84 %	4.09 %	0.37 %	6.09 %	0.78 %
01/04/06	0.0001	0.28 %	100.00 %	4.28 %	3.39 %	10.48 %	3.10 %	0.83 %	6.44 %	0.16 %
01/05/06	0.0001	0.27 %	100.00 %	1.45 %	3.87 %	11.81 %	3.34 %	0.64 %	6.55 %	10.10 %
01/06/06	0.0001	0.27 %	100.00 %	2.21 %	3.98 %	12.63 %	3.50 %	0.24 %	6.03 %	12.01 %
01/09/06	0.0001	0.27 %	100.00 %	0.00 %	3.37 %	16.08 %	5.05 %	10.77 %	3.22 %	12.09 %
01/10/06	0.0001	0.27 %	100.00 %	0.00 %	3.13 %	15.82 %	4.71 %	10.47 %	2.50 %	11.84 %
01/11/06	0.0001	0.27 %	100.00 %	0.06 %	3.16 %	15.77 %	6.05 %	10.51 %	1.75 %	11.10 %
01/12/06	0.0001	0.28 %	100.00 %	3.51 %	3.34 %	12.58 %	4.27 %	10.47 %	0.32 %	11.06 %

Figura 2.63 Portafolios con riesgo menor o igual a 0.0001 con una muestra de dos años

2.10.3 Portafolios con rendimiento mayor o igual a ...

De igual manera si conocemos la expectativa de rendimiento del inversionista, podremos ingresarla y así obtener los portafolios óptimos.

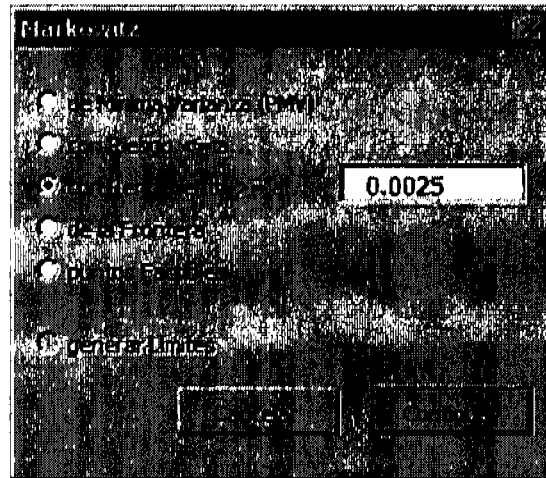


Figura 2.64 Selección portafolios con Rendimiento mayor o igual a...

A continuación está un ejemplo con portafolios cuyo rendimiento esperado es de al menos 0.25%.

04/08/06										
Rend										
12/30/05										
03/10/06										
509										
0.25%										
12/30/05	5.87E-05	0.26 %	100.00 %	10.11 %	3.07 %	3.02 %	3.42 %	0.24 %	6.72 %	5.84 %
01/02/06	5.69E-05	0.26 %	100.00 %	10.02 %	3.33 %	2.93 %	3.89 %	0.28 %	6.78 %	5.69 %
01/03/06	5.6E-05	0.26 %	100.00 %	5.24 %	2.82 %	2.88 %	5.57 %	0.13 %	6.58 %	5.55 %
01/04/06	5.81E-05	0.25 %	100.00 %	5.44 %	2.78 %	4.27 %	5.61 %	0.00 %	6.80 %	6.21 %
01/05/06	5.14E-05	0.25 %	100.00 %	4.23 %	3.12 %	5.56 %	4.77 %	0.00 %	7.50 %	6.64 %
01/06/06	6.1E-05	0.26 %	100.00 %	4.89 %	3.08 %	6.94 %	5.08 %	0.12 %	7.49 %	5.78 %
01/09/06	5.15E-05	0.26 %	100.00 %	2.79 %	2.76 %	10.23 %	5.53 %	0.08 %	8.20 %	3.88 %
01/10/06	6.12E-05	0.26 %	100.00 %	1.96 %	2.55 %	11.09 %	4.31 %	0.18 %	8.14 %	2.38 %
01/11/06	6E-05	0.26 %	100.00 %	3.54 %	2.52 %	11.16 %	5.33 %	0.00 %	7.93 %	1.95 %
01/12/06	5.8E-05	0.26 %	100.00 %	6.03 %	2.55 %	6.67 %	5.04 %	0.00 %	7.88 %	1.02 %

Figura 2.65 Portafolios con rendimiento mayor o igual a 0.25% con una muestra de dos años

2.10.4 Portafolios de la frontera

Para el caso de la frontera eficiente es necesario especificar el número de puntos que se calcularán.

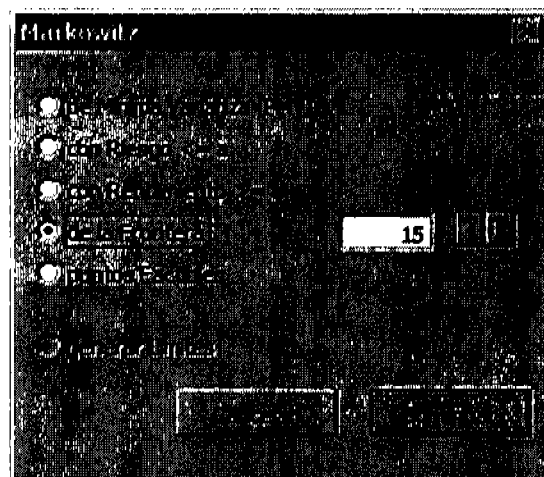


Figura 2.66 Selección portafolios de la Frontera Eficiente

Un ejemplo de 15 portafolios sobre la frontera se ve a continuación.

Fecha	04/08/06											
Fecha	04/08/06											
Tipo	Frontera											
Fecha	12/30/05											
Fecha	12/30/06											
Muestra	509											
Modelo	Eficiente											
Portafolios	15											

12/30/05	0.000649	0.37 %	16 Efic	100.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	100.00 %
12/30/05	0.000439	0.36 %	14 Efic	100.00 %	0.00 %	6.83 %	0.00 %	14.33 %	0.00 %	0.00 %	0.00 %	78.03 %
12/30/05	0.000315	0.34 %	13 Efic	100.00 %	0.00 %	11.42 %	0.00 %	15.18 %	0.00 %	0.00 %	0.00 %	62.89 %
12/30/05	0.000227	0.32 %	12 Efic	100.00 %	0.00 %	11.35 %	4.40 %	13.48 %	0.00 %	0.00 %	0.00 %	48.90 %
12/30/05	0.000161	0.30 %	11 Efic	100.00 %	7.32 %	10.55 %	7.19 %	11.35 %	0.00 %	0.00 %	0.00 %	37.71 %
12/30/05	0.000113	0.28 %	10 Efic	100.00 %	10.74 %	9.33 %	7.77 %	9.31 %	0.00 %	0.00 %	5.60 %	28.47 %
12/30/05	0.000077	0.26 %	9 Efic	100.00 %	10.48 %	7.83 %	6.70 %	7.46 %	0.00 %	0.00 %	16.33 %	21.48 %
12/30/05	0.000051	0.25 %	8 Efic	100.00 %	9.75 %	6.38 %	6.53 %	6.72 %	3.65 %	22.52 %	16.36 %	16.36 %
12/30/05	0.000033	0.23 %	7 Efic	100.00 %	8.11 %	5.09 %	4.04 %	4.13 %	11.00 %	22.93 %	10.48 %	10.48 %
12/30/05	0.000021	0.21 %	6 Efic	100.00 %	6.13 %	4.01 %	2.45 %	2.85 %	13.87 %	20.62 %	6.78 %	6.78 %
12/30/05	0.000014	0.19 %	5 Efic	100.00 %	4.25 %	2.88 %	0.80 %	1.84 %	15.25 %	17.91 %	3.58 %	3.58 %
12/30/05	0.000011	0.17 %	4 Efic	100.00 %	3.15 %	2.09 %	0.00 %	1.03 %	13.66 %	15.15 %	2.01 %	2.01 %
12/30/05	0.000009	0.16 %	3 Efic	100.00 %	2.44 %	1.52 %	0.00 %	0.63 %	11.77 %	12.70 %	0.83 %	0.83 %
12/30/05	0.000007	0.13 %	2 Efic	100.00 %	1.76 %	1.09 %	0.00 %	0.23 %	10.07 %	10.54 %	0.00 %	0.00 %
12/30/05	0.000007	0.12 %	1 PMV	100.00 %	0.02 %	0.86 %	0.00 %	0.00 %	5.95 %	6.58 %	0.00 %	0.00 %

Figura 2.67 Portafolios sobre la frontera con una muestra de dos años

2.10.5 Portafolios factibles

Los puntos factibles crearán el espacio donde se colocan todos los posibles portafolios.

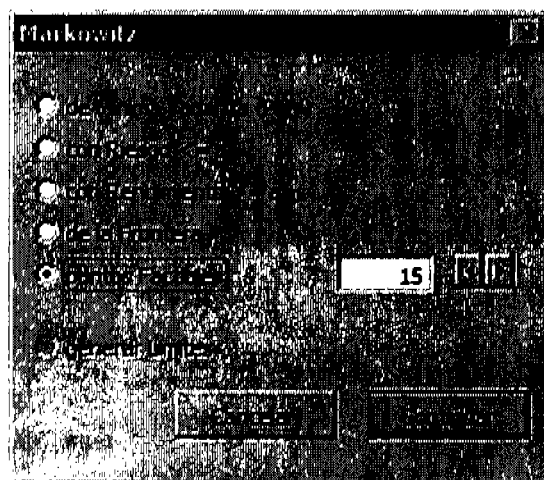


Figura 2.68 Selección portafolios factibles

El área delimitada por el perímetro representa todas las combinaciones de activos que forman los portafolios. La cantidad de puntos especificada son los que se van a calcular entre los activos.

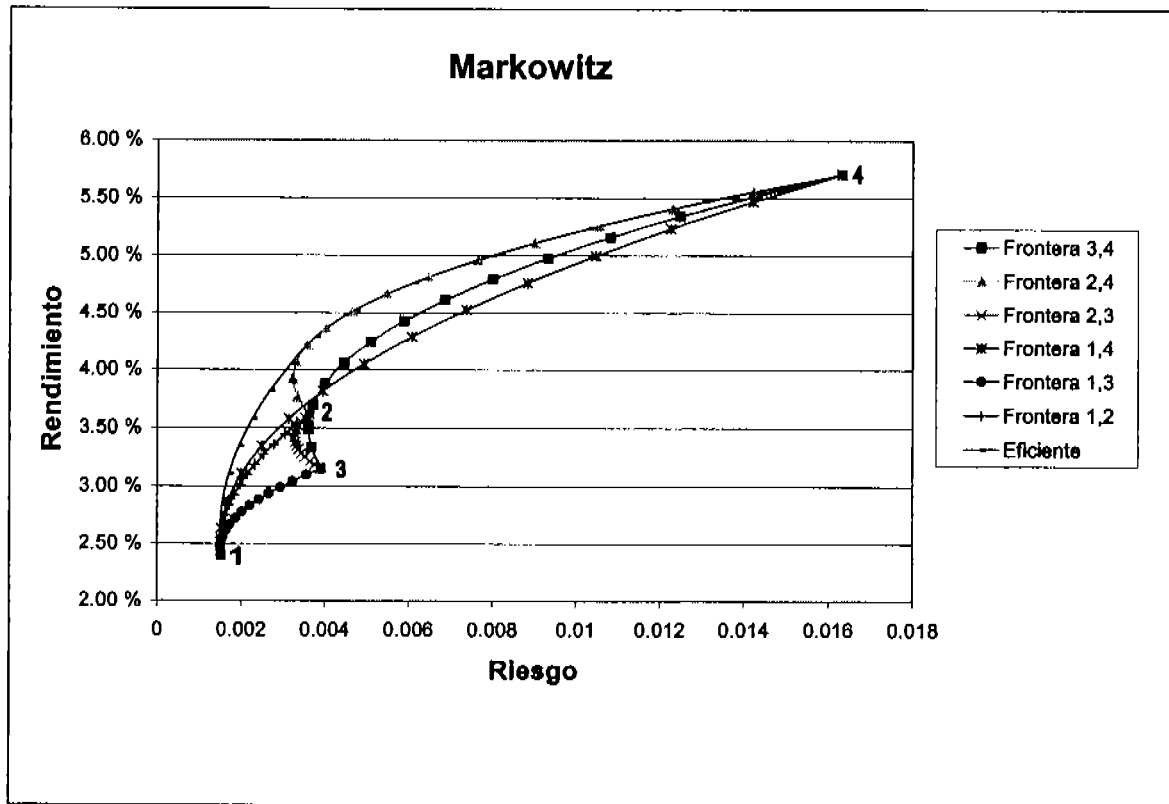


Figura 2.69 Gráfica del espacio factible

2.11 Idloma

Si bien el usuario tiene una preferencia de lenguaje y ésta por lo general no cambia, existe la posibilidad de que tengamos que usar los términos de la región con la que estamos tratando. Parte del éxito en la implementación local de algún sistema es proporcionarles a los usuarios las herramientas con el lenguaje en el que ellos interactúan diariamente aunque poco a poco estos términos sean más conocidos debido a la globalización.

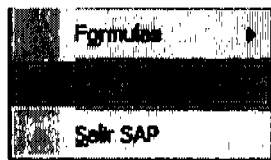


Figura 2.70 Menú cambio de Idioma

En este menú se puede seleccionar el idioma del sistema guardando dicha preferencia en el archivo de configuración inicial para ser usado cada vez que se abra el sistema. El sistema tiene la facilidad de agregar nuevos idiomas sin tener que hacer muchos cambios al código ya que toda esta información reside en una base de datos independiente.

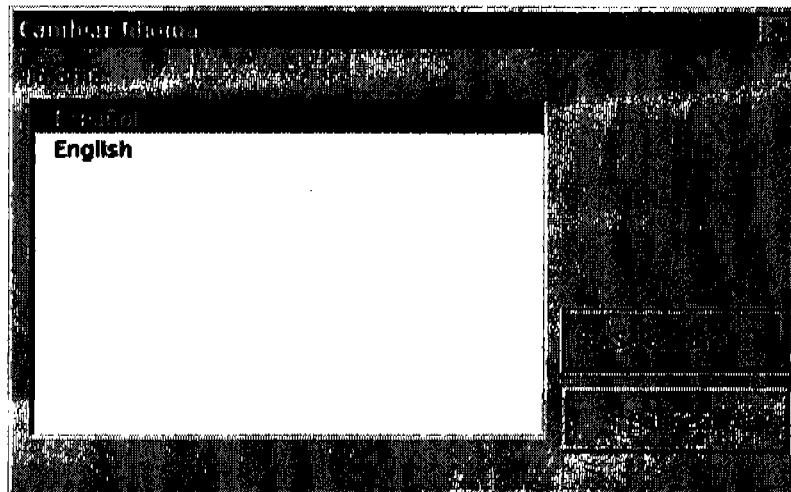


Figura 2.71 Lista de Idiomas

Los títulos del menú cambiarán al idioma seleccionado guardando el mismo orden como se muestra en la siguiente figura.

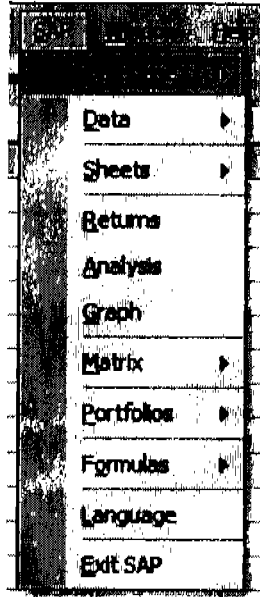


Figura 2.72 Menú del sistema en inglés

2.12 Salida del sistema

El sistema puede descargarse usando el menú o bien por medio de la lista de complementos de Excel explicado en el capítulo “Carga e inicialización del sistema”.



Figura 2.73 Menú salir del sistema

3. Caso práctico

Para el caso práctico, todas las herramientas teóricas y prácticas descritas anteriormente se usarán con información verdadera para construir portafolios de la manera propuesta por la teoría. Se consiguió información de varios índices de las bolsas mundiales y se obtuvieron los portafolios sobre la frontera eficiente. Esta forma de invertir es muy buena en términos de diversificación y se asemeja cada vez más a la manera como se estructuran los fondos de inversión a través de índices y no de acciones individuales.

Existen muchas decisiones importantes que deben ser tomadas al momento de usar estas metodologías. En un momento dado estamos creando un portafolio de acciones (en este caso índices) que no conviene ser modificado constantemente porque los costos de transacción absorberían una parte importante de la utilidad de la inversión. Los costos de transacción son los costos que incurre un individuo o inversionista institucional para solicitar a un intermediario que compre una acción o en este caso un portafolio que replique un índice como los aquí mostrados.

La información que se obtuvo estaba de manera diaria y así se establecieron los pronósticos de rendimiento y de varianza (o desviación estándar). Esto quiere decir implícitamente que los resultados tienen un horizonte de un día. Es posible, sin ningún problema con el sistema mostrado, solicitar que el horizonte de cálculo se extienda a una semana, un mes o seis meses inclusive. Esto es algo relevante y que debe ser estudiado continuamente por los tomadores de decisiones para poder hacer la mejor conformación del portafolio. Como se mencionó en el primer capítulo, se podría llegar al extremo de no usar la información histórica para las estimaciones sino utilizar "opiniones expertas" de analistas para obtener una media y varianza esperada y sobre éstas calcular las fronteras eficientes. Esto último no invalida los conceptos

importantes de Markowitz porque el más relevante propuesto por él se refiere a la diversificación y la revisión frecuente de ésta basada en estimaciones de riesgo-rendimiento. El cómo se obtengan éstas, si con información histórica o con juicios expertos, no afecta su aplicación práctica.

Los índices aquí mostrados representan a todos los continentes privilegiando un poco a los europeos (en este primer ejemplo eliminamos al Índice de Precios y Cotizaciones de México para concentrar los esfuerzos en la evidencia empírica metodológica. Se podrán ver los cambios de incorporar el IPC más adelante). Se tomó información de 2001 a 2005 y la gráfica muestra la frontera eficiente calculada con los dos últimos años de información (522 días hábiles para ser exactos). Existen varias complicaciones prácticas de hacer este tipo de inversiones. Como por ejemplo, que los días hábiles y feriados no coinciden entre países y esto complica establecer las relaciones de riesgo y rendimiento puesto que la información de varios días de la muestra no representa información válida en realidad. Existen esfuerzos a nivel internacional, como el del Grupo de los 30, para tratar de armonizar los estándares de operación a nivel internacional, inclusive sincronizar los horarios de operación, lo cual generalmente implica extender los horarios operativos. De hecho, este año en México vimos una iniciativa que se aprobó para hacer coincidir los días feriados con los lunes, muy al estilo de países europeos y también los EE.UU. Así mismo desde hace años se tiene trabajando con el horario de verano para aprovechar mejor la luz solar, lo que nos ha homologado a la mayor parte de los horarios mundiales. En fin, lo relevante es que la todavía existente asincronía operativa a nivel mundial y que complica un poco poner en práctica los conceptos teóricos planteados. No obstante vemos que aún en países donde no es fácil conseguir acuerdos, existe una tendencia mundial a la sincronización operativa de los mercados, lo que facilitará tener portafolios “globales”.

La gráfica obtenida (ver gráfica 3.1) es totalmente coincidente con la propuesta por la teoría, en donde todos los índices individuales se encuentran dentro de la frontera eficiente y ésta genera combinaciones de portafolios que tienen una relación riesgo-rendimiento esperado mucho mejor que cualquier índice en lo individual. Con excepción del IBOV (índice Bovespa de Brasil) que al tener un alto rendimiento para el periodo de cálculo pero también un alto riesgo se conserva como una inversión viable en la frontera para el portafolio más rentable pero también el más riesgoso.

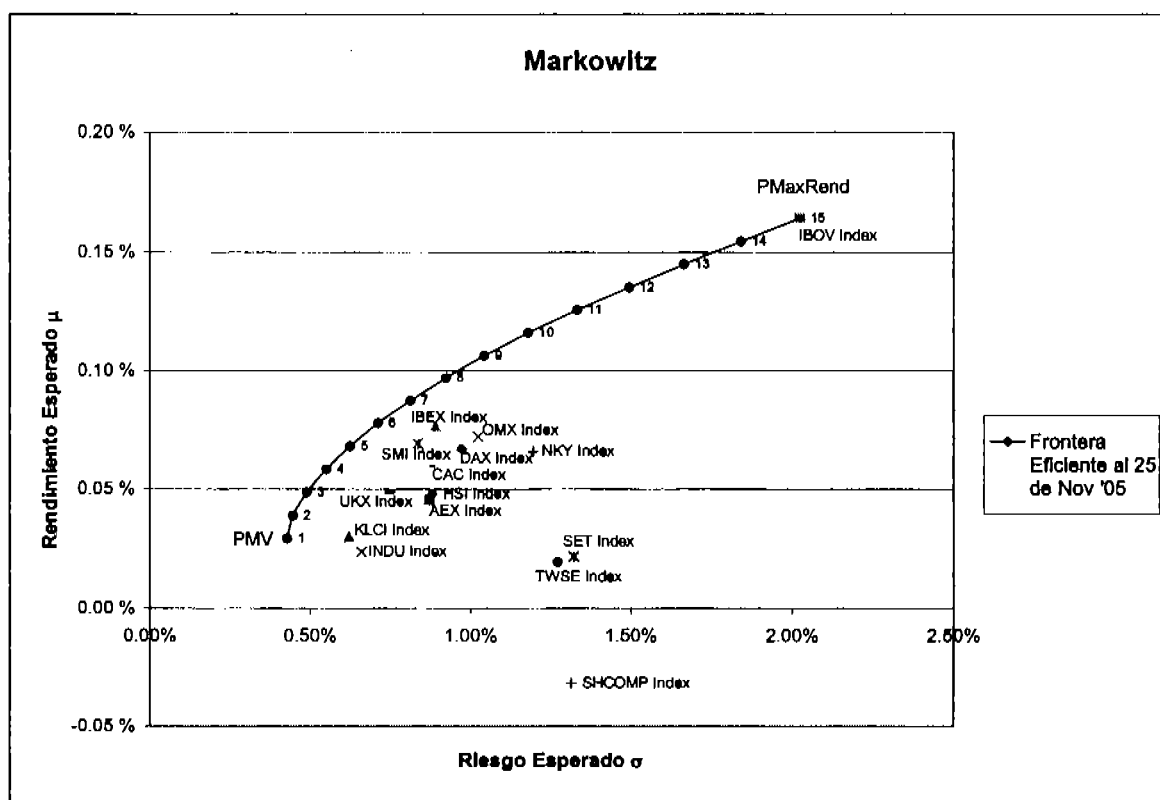


Figura 3.1 Frontera óptima al 25 de Nov. de 2005 sin IPC

Una vez que obtenemos los resultados de la frontera usando el sistema, el inversionista debe proceder a escoger un portafolio dentro de ésta, acorde con sus preferencias de riesgo-rendimiento. Por ejemplo, el Portafolio de Mínima Varianza le indicaría invertir un 36 % del

portafolio en el índice de Malasia KLCI y el Dow Jones Industrial Average (INDU) en un 34%, etc. descritos en la tabla 3.1.

Bolsa	Indice	Porcentaje de Inversión
SINGAPUR	KLCI Index	36.0%
EEUU	INDU Index	33.8%
SUIZA	SMI Index	10.0%
CHINA	SHCOMP Index	7.5%
LONDRES	UKX Index	6.2%
HONG KONG	HSI Index	5.6%
TAILANDIA	SET Index	0.9%

Tabla 3.1 Porcentaje de inversión de PMV al 25 de Noviembre de 2005

Una persona cuyo perfil trate de minimizar el riesgo del portafolio de inversión debe tomar sus recursos de inversión y asignarlos en los porcentajes indicados. En el verdadero mundo de las inversiones, es poco práctico tener porcentajes de inversión menores al 1.0% en un activo, primordialmente porque los costos de operación y de dar seguimiento a un activo pueden ser mucho más altos que los beneficios. Por ello, es mejor tener una diversificación sin llegar a una pulverización aunque esto va a estar determinado por el “estilo” del administrador de portafolios.

Es importante ver que la forma de la frontera cambia dependiendo del momento en el que se hacen las mediciones. En la siguiente gráfica se muestran los cálculos de la frontera eficiente que se obtuvieron con el sistema para diferentes puntos en el tiempo. Por ejemplo, la frontera calculada en 2003 tiene una expectativa de riesgo y rendimiento mucho peor que las de los siguientes periodos. La curva calculada en 2004 es una que tiene menor riesgo y un buen rendimiento esperado.

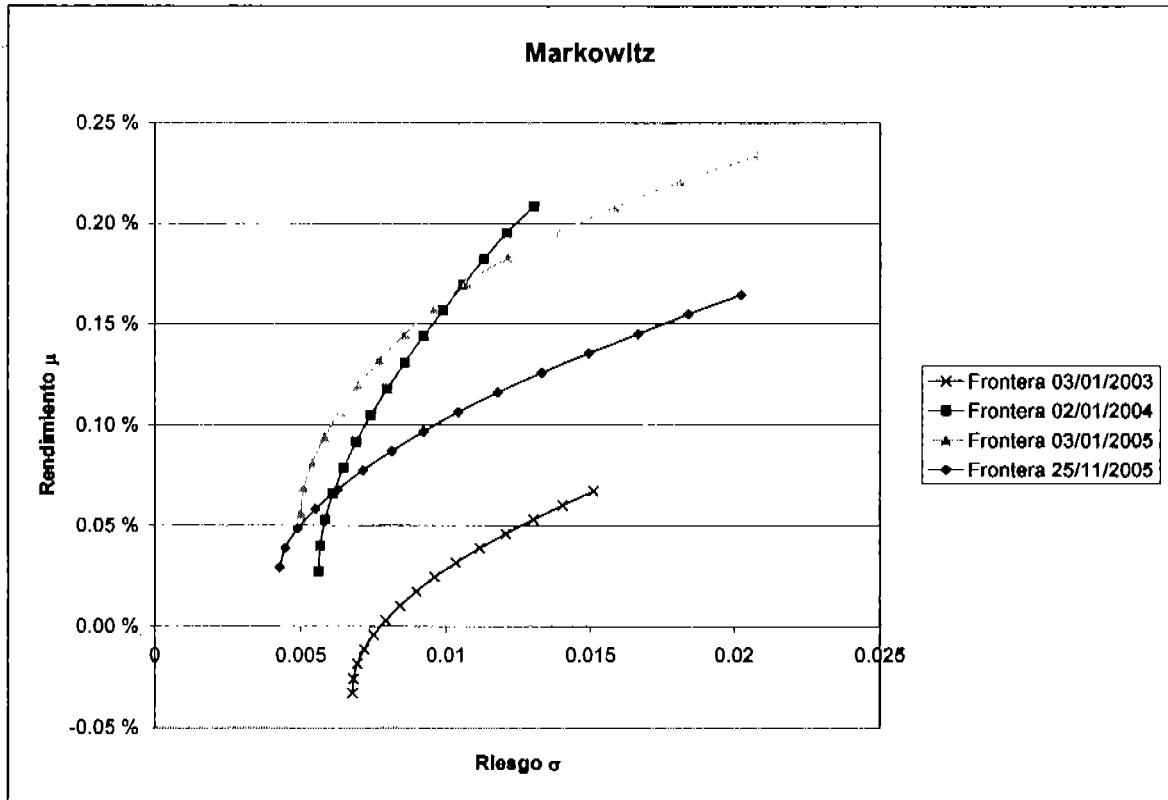


Figura 3.2 Cambios en el tiempo de la frontera sin IPC

Se puede ver en la siguiente tabla que cada punto obtenido en diferentes momentos, tiene una composición muy distinta entre sí, por ello el administrador de un portafolio de inversiones tiene que “rebalancear” su portafolio reflejando las ponderaciones actualizadas prescritas por el modelo de Markowitz.

Bolsa	Ene 2003	Ene 2004	Ene 2005	Nov 2005
BRASIL	1.1 %	0.2 %	0.0 %	0.0 %
CHINA	21.9 %	17.2 %	14.0 %	7.5 %
EEUU	14.5 %	15.9 %	26.8 %	33.8 %
ESTOCOLMO	0.0 %	0.0 %	0.0 %	0.0 %
HONG KONG	0.0 %	7.8 %	5.4 %	5.6 %
LONDRES	7.6 %	3.1 %	6.5 %	6.2 %
MADRID	0.0 %	1.4 %	0.9 %	0.0 %
SINGAPUR	36.4 %	48.9 %	41.4 %	36.0 %
SUIZA	5.8 %	1.3 %	4.2 %	10.0 %
TAILANDIA	4.1 %	3.7 %	0.5 %	0.9 %
TAIWAN	5.8 %	0.5 %	0.1 %	0.0 %
TOKIO	2.8 %	0.0 %	0.0 %	0.0 %

Tabla 3.2 Cambios en el porcentaje de inversión del PMV

El que las curvas cambien tiene que ver con que cada índice va teniendo trayectorias distintas que lo posicionan de manera distinta cada vez que se vuelve a calcular la frontera eficiente. Es importante entender que la dinámica observada es normal para cualquier metodología que se emplee para estimar el riesgo y el rendimiento. Si se maneja la metodología que usa la información histórica, se incorporará nueva información de rendimientos y volatilidad que va a ser usada por el algoritmo de optimización. Por otro lado, en el caso de usar “juicios expertos”, también las expectativas de rendimiento y volatilidad cambian conforme los analistas tienen nuevos elementos de valoración. Se reitera que la forma cóncava de la curva y el comportamiento de las fronteras eficientes “tipo Markowitz” son siempre las mismas independientemente de cómo se hagan las estimaciones.

A continuación se puede ver qué es lo que sucede con la curva que se mostró en la frontera óptima de la figura 3.1 si incorporamos un instrumento adicional, en este caso el Índice de Precios y Cotizaciones de la Bolsa Mexicana de Valores (IPC). Comparando las dos curvas es fácil observar que el IPC es un activo financiero atractivo en el sentido de que desplaza la curva hacia la izquierda y hacia arriba. Es decir, que con excepción del portafolio número 15 donde se invierte todo en el Índice Brasileño – BOVESPA, todos los demás puntos están desplazados

hacia la esquina superior izquierda; ¡mayor rendimiento con menos riesgo! Esta es una situación ideal de diversificar incorporando un activo que tiene atributos que se comportan bien con el resto de los activos financieros. Por supuesto que todas las ponderaciones de nuevos portafolios deben ser ahora distintas, en general deben incorporar el IPC disminuyendo la ponderación de otros índices. Inclusive ahora el portafolio 12 se compone casi exclusivamente del IPC, como se aprecia en la siguiente figura.

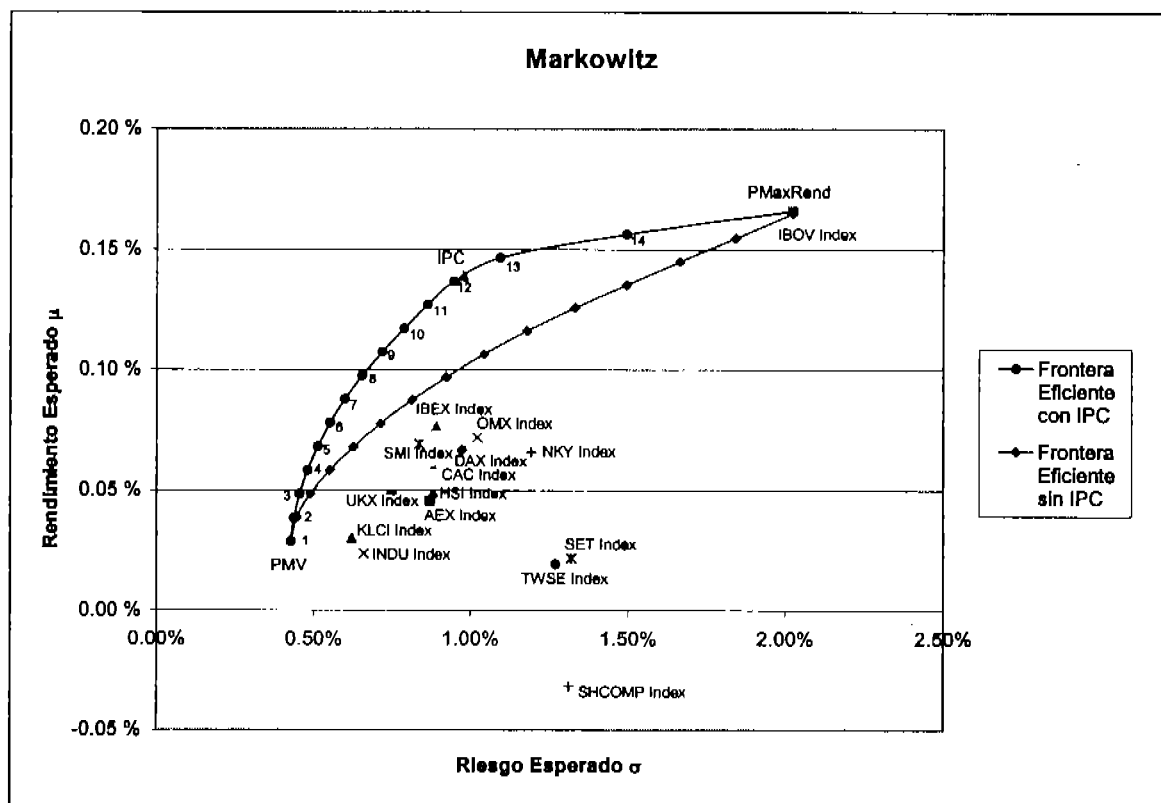


Figura 3.3 Fronteras óptimas al 25 de Nov. de 2005 con y sin IPC

Ahora terminamos esta parte presentando las fronteras en diferentes momentos de tiempo pero ahora incorporando el IPC. En la frontera para Enero de 2004 encontramos una mejoría en la curva marginal. Es decir, con la información empleada para estimar la curva de 2004, el IPC no tenía características de riesgo-rendimiento que mejoraran los portafolios originalmente establecidos. Lo anterior tiene que ver primordialmente con la correlación que tiene el IPC con

el resto de los activos. Con seguridad la correlación debe ser cercana a uno; es decir, el comportamiento estimado del IPC es muy similar a la de algunos activos que ya se encuentran en el portafolio y por ende no contribuyen mayormente a obtener portafolios muy diferentes a los originales. En el fondo esta es la característica más importante de los activos de un portafolio de inversión. Por supuesto, el riesgo y rendimiento esperado es relevante pero termina siendo la correlación de cada instrumento con los demás mucho más importante que sus características individuales. He ahí la contribución más importante de Markowitz a la teoría de las finanzas.

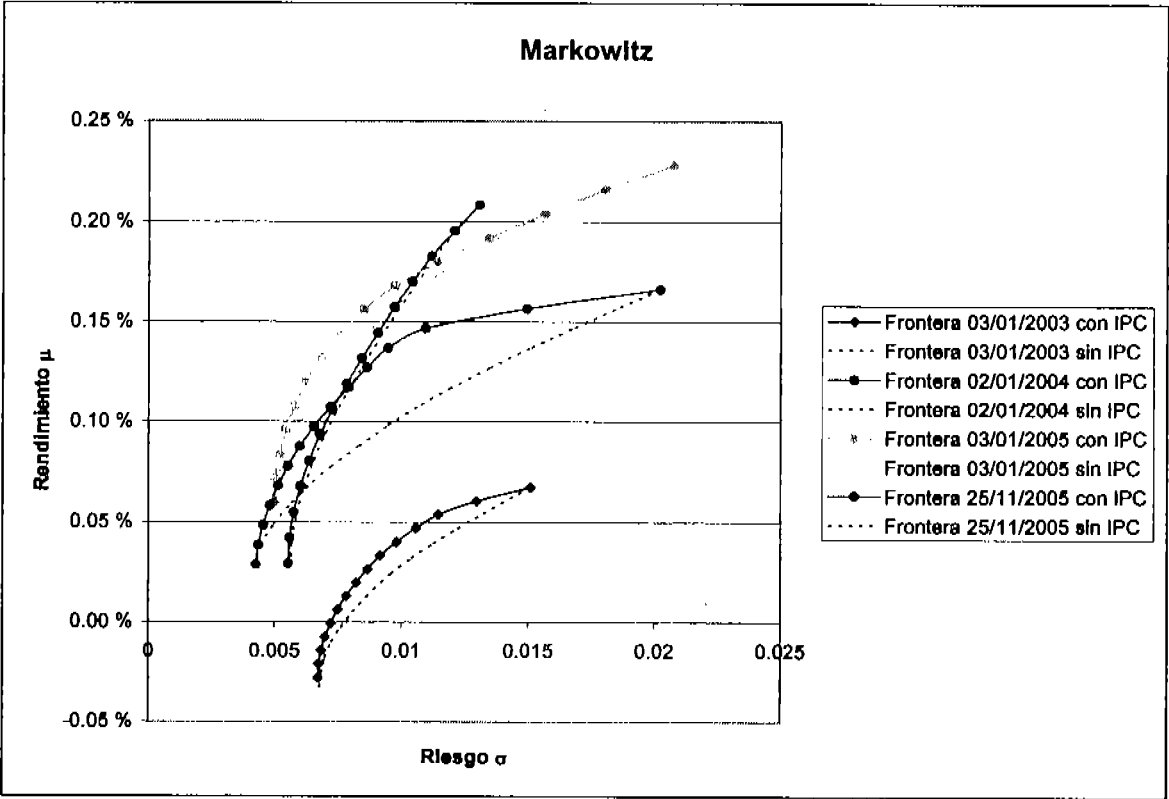


Figura 3.4 Comparativo de fronteras en el tiempo con IPC vs. sin IPC

Como dijimos al inicio de la tesis, vamos a ver lo que pasa con los portafolios arrojados por el sistema en un análisis ex-post para diferentes perfiles de los inversionistas. Se hizo una simulación durante casi 3 años de los rendimientos un portafolio formado con los porcentajes arrojados por el sistema y haciendo un rebalanceo cada año. Podemos observar que para

periodos grandes de tiempo, aquellos portafolios cuyo riesgo esperado es bajo, su rendimiento ex-post también resultó bajo y para aquellos que tienen un alto riesgo estimado, el rendimiento observado fue alto.

Riesgo esperado	Rendimiento esperado	Rendimiento ex-post
PMV	0.02 %	0.05 %
Bajo	0.06 %	0.08 %
Mediano	0.10 %	0.10 %
Alto	0.17 %	0.16 %

Tabla 3.3 Rendimiento ex-post de portafolios de varios perfiles de riesgo durante casi 3 años (Ene 2003 – Nov 2005)

Esto no necesariamente va ocurrir así siempre y mucho menos en periodos de tiempo cortos, por ejemplo, durante el 2004 los resultados que se muestran en la siguiente tabla dan evidencia de que un año donde las bolsas mundiales presentaron en promedio un crecimiento menor que el anterior, el rendimiento no guardó la misma relación directa entre su riesgo y su rendimiento inclusive hubo pérdida en el portafolio más riesgoso. Esto sigue siendo consistente con la teoría presentada ya que la variación en el rendimiento puede ser hacia arriba o hacia abajo.

Riesgo esperado	Rendimiento esperado	Rendimiento ex-post
PMV	0.03 %	0.04 %
Bajo	0.08 %	0.04 %
Mediano	0.13 %	0.03 %
Alto	0.21 %	-0.04 %

Tabla 3.4 Rendimiento ex-post de portafolios de varios perfiles de riesgo en el 2004

Otro tema relevante para la discusión de este caso práctico es la elección de índices de bolsas en el mundo en lugar de acciones individuales. El tema amerita análisis porque existe la creencia de que no es factible invertir en éstos. En realidad, hoy día, de los activos financieros más operados están relacionados con índices accionarios. Inclusive existe regulación que obliga a algunos fondos a invertir predominantemente en índices; y esto no es un asunto ajeno a México.

A continuación se muestra un pequeño extracto de una circular emitida por la Comisión Nacional del Sistema de Ahorro para el Retiro que regula la manera en cómo se invierten los fondos de pensiones en México. Es el llamado SAR, que cada trabajador afiliado al IMSS (ahora ya inclusive se puede tener siendo trabajador independiente) tiene con una contribución que se hace de manera bimestral y cuyo dinero es canalizado a una cuenta individual que administran las AFORES (Administradoras de Fondos de Ahorro para el Retiro). Estos fondos valen hoy día casi 60 mil millones de dólares y constituyen la fuente de ahorro más importante del país en este momento.

Sin profundizar más en antecedentes analicemos el texto de la circular mencionada:

“REGLAS GENERALES PARA VALORES ADQUIRIDOS POR LAS SOCIEDADES DE INVERSIÓN ESPECIALIZADAS DE FONDOS PARA EL RETIRO

SEGUNDA.- Para los efectos de estas reglas generales, se entenderá por:

...

III. Canasta de Índices, al conjunto de índices accionarios, o subíndices que de ellos se deriven,...

VIGESIMA SEPTIMA.- Las Sociedades de Inversión podrán adquirir Instrumentos de Renta Variable y Valores Extranjeros de Renta Variable ... únicamente con el fin de replicar los índices o Canasta de Índices previstos en las reglas de carácter general expedidas por la Comisión en materia de régimen de inversión”

Es decir, que la CONSAR sólo permite que se invierta en instrumentos accionarios locales y extranjeros (llamados de renta variable), sólo si replican los índices o canastas de índices que

permite la misma Comisión. Un poco extremo, pero en el fondo sólo permite que se invierta en índices o instrumentos que repliquen índices accionarios. Es decir, en México es obligatorio invertir en índices, como los empleados en el caso práctico analizado. Para mayor información el lector puede remitirse al sitio en internet de la CONSAR⁹ para profundizar en las reglas generales de inversión que son muy interesantes en donde se detalla los índices en los cuales se permite la inversión, porque tampoco todos los índices son permitidos.

⁹ CONSAR: www.consar.gob.mx

Conclusiones

El sistema aprovecha las ventajas de un “software” ampliamente usado por los analistas de portafolios (MS Excel) y reduce significativamente el tiempo para hacer un profundo estudio de riesgo-rendimiento. Los reportes del sistema pueden ser modificados o bien se pueden crear nuevos sin tener que recurrir a personal especializado en otro tipo de herramientas.

En el caso práctico podemos observar que hay una relación directa entre el riesgo y rendimiento esperado sobre la frontera eficiente y el observado aunque no siempre va a ser así, sobre todo en periodos cortos. Los porcentajes sugeridos indican al gestor del portafolio la manera en que se debe estructurar el portafolio de inversión y que permiten dar seguimiento continuo a éste. También se cuenta con la posibilidad de incorporar de manera sencilla y continua nueva información y no perder oportunidad en el uso de variables relevantes en todo momento.

Con seguridad la evolución conceptual de las propuestas para invertir seguirán avanzando y su uso podrá ser relevante en la medida en que se utilicen las tecnologías de información de manera adecuada. Cuanto antes se realicen los cambios, mayor provecho podrá obtenerse. Al propagarse el conocimiento de manera paulatina, la explotación de éste será menos rentable hasta que nuevos paradigmas surjan y unos cuantos los incorporen para su beneficio, sin duda en una actividad similar a la planteada en este trabajo, en donde todas las herramientas que requiere son integradas para facilitar su uso. Esta propuesta da indicios de la urgencia de ofrecer el soporte informático para hacer factible el uso de nuevas ideas.

Las propuestas para mejorar el sistema son:

i) Obtener información en línea (por ejemplo Bloomberg o Infotel) con el fin de tener actualizada en todo tiempo nuestra base de datos. La posibilidad de conectarse con las fuentes

de información mediante una “liga” evitando así la captura manual. El éxito en el proceso de integración de nuevos sistemas requiere de conocer las necesidades de información, así como los recursos tecnológicos que disponen los usuarios. Es necesario hacer una buena planeación para llegar al objetivo con las limitantes existentes para maximizar el impacto positivo en el menor tiempo posible.

ii) Incorporar, no solamente información histórica si no cualitativa de analistas y de investigaciones especializadas que alimenten el sistema haciendo una especie de modelo híbrido que tome, por un lado información que pronostique los rendimientos y por otro lado utilice las ventajas de la diversificación. Parte fundamental del éxito de este modelo es que el comportamiento de los activos financieros no difiere mucho de cómo se ha venido comportando anteriormente, pero esto no necesariamente va a ser así todo el tiempo. Adicionalmente, para cuando los rendimientos históricos de los activos financieros nos muestren su tendencia, el invertir en ellos en ese momento puede ya no ser la mejor opción (“Beat the market”).

iii) Una aportación adicional al sistema es la posibilidad de formar índices de los precios o rendimientos a partir de las series existentes para tomarlos como referencias (“benchmarks”) y hacer comparaciones sobre su desempeño. Aún cuando exista en el mercado información sobre índices, éstos no necesariamente son los que queremos para medir nuestros portafolios.

Por último, basta decir que una revisión a las revistas especializadas en el tema de inversiones para dar cuenta de que existe una gran vigencia en el uso de la tecnología para resolver problemas propios del sector financiero. Grandes cantidades de recursos económicos y humanos son invertidos anualmente para esta actividad, y sin duda, los servicios financieros están basados para su proveeduría en el apoyo de las tecnologías de información. El sustento conceptual que la ingeniería da para llevar a cabo estas actividades, permite armonizar las complejidades de modelación que se puedan presentar por el lado financiero con el correcto uso de la informática.

Bibliografía

Robert A. Haugen, Modern Investment Theory
Prentice Hall
Engle Wood Clifs and New Jersey, 1986.

Frank K Reilly, Keith C. Brown, Investment Analysis and Portfolio Management
Dryden
United States of America, Fifth Edition , 1997.

Eppen, G. D., Investigación de Operaciones en la Ciencia Administrativa
Prentice Hall
México, 1987.

Hillier, Lieberman, Introducción a la Investigación de Operaciones
Mac Graw Hill
México, 1990.

Taha, Operations Research
Macmillan Publishing Co., Inc.
United States of America, 1976.

Wells, Developing Microsoft Excel 95 Solutions with Visual Basic for Application
Microsoft Press
United States of America, 1995.

Bracken, McCormik, Selected Applications of Nonlinear Programming
John Wiley & Sons, Inc.
New York, 1968

Harry M. Markowitz, Portfolio Selection
John Wiley & Sons, Inc.
New York, 1959

Harry M. Markowitz, Mean-Variance Analysis in Portfolio Choice and Capital Markets
Basil Blackwell Ltd
Oxford, 1987

G. Hadley, Nonlinear and Dynamic Programming
Addison-Wesley, Reading
Mass., 1964

Lawrence, Guerard, Reeves, Advances in Mathematical Programming and Financial Planning, Vol 1, Research Annual
Jai Press Inc.
London, 1987

Lawrence, Guerard, Reeves, Advances in Mathematical Programming and Financial Planning, Vol 2, Research Annual

Jai Press Inc.
London, 1990

Apéndice "A" Código

Como se mencionó en el Capítulo 2 "Diseño de un Sistema de Administración de Portafolios", el código es presentado en este apéndice como referencia. El índice de las funciones y los procedimientos de los que consta este sistema se muestra a continuación en orden alfabético.

Índice de rutinas por orden alfabético

AbreBD, 108
AgregaDatos, 185
AplicaFormato, 128
Auto_Close, 178
Auto_Open, 100
BorraFechasDeDatosIncompletos, 130
BorraFechasDeDatosRepetidos, 132
BorrarHoja, 128
BorrarSeriesBD, 114
BorraSeriesCriterioInicioCotizacion, 133
BorraSeriesSinCambio, 135
CalculaNoSeriesPorTipo, 119
CambiaIdioma, 175
CambiaOpcionMenu, 105
CambiaPref, 180
CargarSeriesBD, 108
ChiC, 148
ChiCuad, 146
CompactaBD, 119
CompletaPreciosIncompletos, 137
CopiaValoresFrontera, 175
CorreSeguridad, 100
CreaAnálisis, 140
CreaArchivoIni, 179
CreaBD, 106
CreaFrontera, 169
CreaGrafica, 149
CreaHistograma, 148
CreaMatCovCorr, 153
CreaMenu, 102
CreaPortafolio, 159
EmpiezaTerminaCotizar, 174
FormatoHojas, 125
GenDim, 147
GeneraStrDeArreglo, 184
GeneraStrDeRecordset, 118
GenFrecEsp, 147
GenZ, 147
GrafCalor, 158
GraficaDePortafolio, 150
HabilitarMenu, 105
ImportarDatosBD, 112
ListaArchivos, 182
ModificarSeries, 129
ObtenerHojas, 122
ObtienePosiciones, 173
ObtieneRends, 138
RegresaComentEnIdioma, 176
RegresaConsecutivo, 121
RegresaFechas, 184
RegresaPath, 179
RegresaPreferencia, 180
RegresaValorDePref, 181
RegresaVectorDeRango, 185
SalirSistema, 178
SeleccionaBD, 178
SigHoja, 120
TomaBD, 183
TomaBD_Default, 183
TomaIdioma, 177
TomaLibro, 124
TomaPrefIni, 179
TomaSeries, 115
TomaTipo, 152
TomaTxt, 181
Verifica_Estructura, 105

Inicio

```
Sub Auto_Open()

Dim Comentario As String

Idioma = TomaIdioma
On Error GoTo MnjErr
Application.EnableCancelKey = xlErrorHandler
If Not CorreSeguridad Then Exit Sub
On Error GoTo 0
CreaMenu
VerificaEstructura

Exit Sub

MnjErr:
Select Case Err
Case 18 "Se ha producido una interrupción por parte del usuario"
If Not RegresaComentEnIdioma(Comentario, _
"No se puede abrir por una interrupción del usuario") Then Exit Sub
MsgBox Prompt:=Comentario, Buttons:=vbCritical, _
Title:=TituloSistema & " Seguridad"
Case Else
MsgBox Prompt:=Error(), Title:="Error " & Err
End Select

End Sub
```

Seguridad

```
Private Function CorreSeguridad
(Optional CorrerCerrar) As Boolean

Dim Usuario As String
Dim Password As String
Dim ContIntentos As Integer

On Error GoTo SeguridadCerrar

Application.EnableCancelKey = xlErrorHandler
With frm_Inicio
If Idioma = IdiomaEspanol Then
.TxtMensaje.Text = "Este sistema resuelve problemas de " _
& "diversificación aplicando teoría moderna de portafolios." & Chr(13) _
& "Para cualquier duda o sugerencia puede escribir un " _
& "E-Mail a alejandro_villasenor@prodigy.net.mx o llamar al 5425 2107."
.LblOrganización.Caption = "Para uso exclusivo de " _
& Application.OrganizationName
.LblUsuario.Caption = "Usuario"
.LblContraseña.Caption = "Contraseña"
ElseIf Idioma = IdiomaIngles Then
.TxtMensaje.Text = "This system solves portfolio problems using " _
& "portfolio modern investment theory ." & Chr(13) _
& "For any issue please send an e-mail to " _
& "alejandro_villasenor@prodigy.net.mx or call 5425 2107."
.LblOrganización.Caption = "For exclusive use of " _
& Application.OrganizationName
```

```

        .LblUsuario.Caption = "User"
        .LblContraseña.Caption = "Password"
    End If
End With
RepiteInicio:
With frm_Inicio
    Usuario = "PortAdmin"
    .TxtUsuario.Text = Usuario
    .TxtContraseña.SelStart = 0
    .TxtContraseña.SelLength = Len(.TxtContraseña.Text)
    .TxtContraseña.SetFocus
    .UserCancel = False
    .Show
    If .UserCancel Then Error 8
    ContIntentos = ContIntentos + 1
    Select Case .TxtUsuario.Text
        Case ""
            MsgBox Prompt:="Ingrese un nombre de usuario", Buttons:=vbCritical, _
                Title:=SiglaSistema & "-Error en seguridad"
            ContIntentos = ContIntentos - 1
            GoTo RepiteInicio
        Case "PortAdmin"
            Password = "optim"
        Case Else
            MsgBox Prompt:="El nombre del usuario no esta registrado.", _
                Buttons:=vbCritical, Title:=SiglaSistema & "-Error en seguridad"
            Error 4
    End Select
    If .TxtContraseña.Text <> Password Then
        MsgBox Prompt:="No coincide la contraseña de " & .TxtUsuario.Text _
            & Chr(13) & "Verifique las mayusculas" & Chr(13) & "Intento " _
            & ContIntentos & " de 3.", _
            Buttons:=vbCritical, Title:=SiglaSistema & "-Error en seguridad"
        If ContIntentos > 2 Then Error 4
        GoTo RepiteInicio
    Else
        CorreSeguridad = True
    End If
End With

Exit Function

SeguridadCerrar:

Select Case Err
    Case 4
        MsgBox Prompt:="No se pudo abrir por error en seguridad", _
            Title:=SiglaSistema & "-Error en seguridad"
    Case 8
        Application.Caption = Empty
    Case 18 "'Se ha producido una interrupción por parte del usuario"
        MsgBox Prompt:="No se puede abrir por interrupcion del usuario", _
            Title:=SiglaSistema & "-Error en seguridad"
    Case Else
        MsgBox Prompt:=Error(), Title:="Error en Seguridad " & Err
    End Select
    If IsMissing(CorrerCerrar) Then CorrerCerrar = False
    If CorrerCerrar Then ThisWorkbook.RunAutoMacros xlAutoClose
    ThisWorkbook.Close SaveChanges:=False

End Function

```

Menú

```
Global Const NombreMenu = "&SAP"
Global Const SiglaSistema = "SAP"

Sub CreaMenu()

Dim i As Integer
Dim SubMenu 'Variable que va a contener MenuItem

Dim Comentario As String

With MenuBars(xlWorksheet)
  For i = 1 To .Menus.Count
    If NombreMenu = .Menus(i).Caption Then
      If Not RegresaComentEnIdioma(Comentario, _
        "El menú SAP ya existe. Desea reemplazarlo") Then Exit Sub
      If MsgBox(Prompt:=Comentario, Buttons:=vbYesNo + vbQuestion, _
        Title:=TituloSistema) = vbYes Then
        .Menus(i).Delete
        Exit For
      Else
        Exit Sub
      End If
    End If
  Next i
  .Menus.Add Caption:=NombreMenu, Before:="Window"

  With .Menus(NombreMenu).MenuItem
    Select Case Idioma
      Case IdiomaEspanol
        Set SubMenu = .AddMenu(Caption:="&Base de Datos")
        SubMenu.MenuItem.Add Caption:="&Seleccionar", _
          OnAction:="SeleccionaBD", _
          StatusBar:="Selecciona una BD"
        SubMenu.MenuItem.Add Caption:="-"
        SubMenu.MenuItem.Add Caption:="&Crear", _
          OnAction:="CreaBD", _
          StatusBar:="Crea una Base de Datos (*.mdb)"
        SubMenu.MenuItem.Add Caption:="Co&mpactar", _
          OnAction:="CompactaBD", _
          StatusBar:="Compacta BD"
        Set SubMenu = .AddMenu(Caption:="&Datos")
        SubMenu.MenuItem.Add Caption:="&Importar", _
          OnAction:="ImportarDatosBD", _
          StatusBar:="Importa datos de una Base de Datos"
        SubMenu.MenuItem.Add Caption:="-"
        SubMenu.MenuItem.Add Caption:="&Cargar", _
          OnAction:="CargarSeriesBD", _
          StatusBar:="Carga datos de un archivo CSV a una Base de Datos"
        SubMenu.MenuItem.Add Caption:="&Borrar", _
          OnAction:="BorrarSeriesBD", _
          StatusBar:="Borrar datos de una Base de Datos"
        .Add Caption:="-"
        Set SubMenu = .AddMenu(Caption:="&Hojas")
        SubMenu.MenuItem.Add Caption:="&Borrar", _
          OnAction:="BorrarHoja", _
          StatusBar:="Borra hojas"
        .Add Caption:="-"
        Set SubMenu = .AddMenu(Caption:="Pre&cios")
        SubMenu.MenuItem.Add Caption:="&Modificar", _
          OnAction:="ModificarSeries", _
          StatusBar:="Modificar series"
        .Add Caption:="&Rendimientos", _
```

```

    OnAction:="ObtieneRends", _
    StatusBar:="Obtiene rendimientos de los precios"
.Add Caption:="&Análisis", _
    OnAction:="CreaAnálisis", _
    StatusBar:="Obtiene análisis de los datos"
.Add Caption:="&Graficar", _
    OnAction:="CreaGrafica", _
    StatusBar:="Gráfica datos de las hojas"
.Add Caption:="-"
Set SubMenu = .AddMenu(Caption:="&Matriz")
    SubMenu.MenuItems.Add Caption:="Co&varianzas", _
        OnAction:="'CreaMatCovCorr "Covarianzas"', _
        StatusBar:="Matriz de covarianzas entre series"
    SubMenu.MenuItems.Add Caption:="Co&rrelaciones", _
        OnAction:="'CreaMatCovCorr "Correlaciones"', _
        StatusBar:="Matriz de correlaciones entre series"
    SubMenu.MenuItems.Add Caption:="-"
    SubMenu.MenuItems.Add Caption:="&Grafica de Calor", _
        OnAction:="GrafCalor", _
        StatusBar:="Identifica con colores los valores de la matriz"
    SubMenu.MenuItems.Add Caption:="&Excel Add-In", _
        OnAction:="'CambiaOpcionMenu "&Excel Add-In"', _
        StatusBar:="Crea la matriz de la herramienta de Excel"
.Add Caption:="-"
Set SubMenu = .AddMenu(Caption:="&Portafolios")
    SubMenu.MenuItems.Add Caption:="&Markowitz", _
        OnAction:="'CreaPortafolio "Markowitz"', _
        StatusBar:="Portafolio por matriz de Varianzas y Covarianzas"
    SubMenu.MenuItems.Add Caption:="-"
    SubMenu.MenuItems.Add Caption:="&Límites", _
        OnAction:="'CambiaOpcionMenu "&Límites"', _
        StatusBar:="Establece límites para la generación de portafolios"
.Add Caption:="-"
Set SubMenu = .AddMenu(Caption:="F&ormulas")
    SubMenu.MenuItems.Add Caption:="&Análisis", _
        OnAction:="'CambiaOpcionMenu "&Análisis"', _
        StatusBar:="Conserva las formulas para generar el análisis"
    SubMenu.MenuItems.Add Caption:="&Matriz", _
        OnAction:="'CambiaOpcionMenu "&Matriz"', _
        StatusBar:="Conserva las formulas para generar la matriz"
    SubMenu.MenuItems.Add Caption:="&Portafolio", _
        OnAction:="'CambiaOpcionMenu "&Portafolio"', _
        StatusBar:="Conserva las formulas para generar el portafolio"
.Add Caption:="-"
.Add Caption:="&Idioma", _
    OnAction:="CambiaIdioma", _
    StatusBar:="Cambiar Idioma"
.Add Caption:="-"
.Add Caption:="&Salir " & SiglaSistema, _
    OnAction:="SalirSistema", _
    StatusBar:="Salir del Sistema"
Case IdiomaIngles
    Set SubMenu = .AddMenu(Caption:="&Database")
        SubMenu.MenuItems.Add Caption:="&Select", _
            OnAction:="SeleccionaBD", _
            StatusBar:="Selects a DB"
        SubMenu.MenuItems.Add Caption:="-"
        SubMenu.MenuItems.Add Caption:="&Create", _
            OnAction:="CreaBD", _
            StatusBar:="Creates a DB (*.mdb)"
        SubMenu.MenuItems.Add Caption:="Co&mpact", _
            OnAction:="CompactaBD", _
            StatusBar:="Compacts a DB"
    Set SubMenu = .AddMenu(Caption:="&Data")
        SubMenu.MenuItems.Add Caption:="&Import", _
            OnAction:="ImportarDatosBD", _

```



```

        StatusBar:="Imports data from a Database"
    SubMenu.MenuItems.Add Caption:="-"
    SubMenu.MenuItems.Add Caption:="&Load", _
        OnAction:="CargarSeriesBD", _
        StatusBar:="Loads data from a CSV file into a Database"
    SubMenu.MenuItems.Add Caption:="&Delete", _
        OnAction:="BorrarSeriesBD", _
        StatusBar:="Deletes data from a Database"
    .Add Caption:="-"
    Set SubMenu = .AddMenu(Caption:="&Sheets")
    SubMenu.MenuItems.Add Caption:="&Delete", _
        OnAction:="BorrarHoja", _
        StatusBar:="Delete sheets"
    .Add Caption:="-"
    Set SubMenu = .AddMenu(Caption:="Pri&ces")
    SubMenu.MenuItems.Add Caption:="&Modify", _
        OnAction:="ModificarSeries", _
        StatusBar:="Modify series"
    .Add Caption:="&Returns", _
        OnAction:="ObtieneRends", _
        StatusBar:="Get returns from prices"
    .Add Caption:="&Analysis", _
        OnAction:="CreaAnalisis", _
        StatusBar:="Get analysis from data"
    .Add Caption:="&Graph", _
        OnAction:="CreaGrafica", _
        StatusBar:="Graphs the results"
    .Add Caption:="-"
    Set SubMenu = .AddMenu(Caption:="&Matrix")
    SubMenu.MenuItems.Add Caption:="Co&variance", _
        OnAction:="'CreaMatCovCorr "Covarianzas"','', _
        StatusBar:="Covariance Matrix from series"
    SubMenu.MenuItems.Add Caption:="Co&rrelation", _
        OnAction:="'CreaMatCovCorr "Correlaciones"','', _
        StatusBar:="Correlation Matrix from series"
    SubMenu.MenuItems.Add Caption:="-"
    SubMenu.MenuItems.Add Caption:="&Heat graph", _
        OnAction:="GrafColor", _
        StatusBar:="Identifies with colors the values of the matrix"
    SubMenu.MenuItems.Add Caption:="&Excel Add-In", _
        OnAction:="'CambiaOpcionMenu "&Excel Add-In"', "&Matrix"','', _
        StatusBar:="Creates the matrix from Excel Add-In"
    .Add Caption:="-"
    Set SubMenu = .AddMenu(Caption:="&Portfolios")
    SubMenu.MenuItems.Add Caption:="&Markowitz", _
        OnAction:="'CreaPortafolio "Markowitz"','', _
        StatusBar:="Keeps formulas for the Variance-Covariance matrix"
    SubMenu.MenuItems.Add Caption:="-"
    SubMenu.MenuItems.Add Caption:="&Limits", _
        OnAction:="'CambiaOpcionMenu "&Limits"', "&Portfolios"','', _
        StatusBar:="Stablish Limits in the Portfolio generation"
    .Add Caption:="-"
    Set SubMenu = .AddMenu(Caption:="F&ormulas")
    SubMenu.MenuItems.Add Caption:="&Analysis", _
        OnAction:="'CambiaOpcionMenu "&Analysis"', "F&ormulas"','', _
        StatusBar:="Keeps formulas for the analysis"
    SubMenu.MenuItems.Add Caption:="&Matrix", _
        OnAction:="'CambiaOpcionMenu "&Matrix"', "F&ormulas"','', _
        StatusBar:="Keeps formulas for the matrix"
    SubMenu.MenuItems.Add Caption:="&Portfolio", _
        OnAction:="'CambiaOpcionMenu "&Portfolio"', "F&ormulas"','', _
        StatusBar:="Keeps formulas for the portfolio"
    .Add Caption:="-"
    .Add Caption:="&Language", _
        OnAction:="CambiaIdioma", _
        StatusBar:="Change Language"

```

```

        .Add Caption:="-"
        .Add Caption:="&Exit " & SiglaSistema, _
            OnAction:="SalirSistema", _
            StatusBar:="Exit System"
    Case Else
        MsgBox "Language not defined " & Idioma, vbExclamation, TituloSistema
    End Select
End With
End With
End Sub

```

```

Sub HabilitarMenu(Habilitar As Boolean)

Dim SubMenu

For Each SubMenu In MenuBars(xlWorksheet).Menus(NombreMenu).MenuItems
    Select Case SubMenu.Caption
        Case "-", "&Exit SAP"
        Case Else
            SubMenu.Enabled = Habilitar
    End Select
Next SubMenu

End Sub

```

```

Sub CambiaOpcionMenu(Menu1 As String, Optional Menu2)

With MenuBars(xlWorksheet).Menus(NombreMenu)
    If IsMissing(Menu2) Then
        .MenuItems(Menu1).Checked = _
            Not .MenuItems(Menu1).Checked
    Else
        .MenuItems(Menu2).MenuItems(Menu1).Checked = _
            Not .MenuItems(Menu2).MenuItems(Menu1).Checked
    End If
End With

End Sub

```

Verifica estructura

```

Global Const TituloSistema = "Administrador de Portafolios"
Global Const dir_Data = "Data"
Global Const dir_DB = "DBase"
Global Const dir_Port = "Portfolios"

Function Verifica_Estructura() As Boolean

```

```

Dim ArrDirs
Dim i As Integer
Dim PathDirectorio As String
Dim NombreDir As String

Verifica_Estructura = True
ArrDirs = Array(dir_Data, dir_DB, dir_Port)

For i = LBound(ArrDirs) To UBound(ArrDirs)
    PathDirectorio = ThisWorkbook.Path & "\" & ArrDirs(i)
    NombreDir = Dir(PathDirectorio, vbDirectory)
    If NombreDir = "" Then
        If MsgBox(Prompt:="No se encuentra el directorio: " & String(2, vbCr) _
            & PathDirectorio & String(2, vbCr) & "Desea crearlo", _
            Buttons:=vbQuestion + vbYesNo, Title:=TituloSistema) = vbYes Then
            MkDir PathDirectorio
        Else
            Verifica_Estructura = False
        End If
    End If
Next i

'El siguiente código deshabilita el menu
'del sistema si falta algun directorio
If Not Verifica_Estructura Then HabilitarMenu Habilitar:=False

End Function

```

Base de datos

```

Sub CreaBD ()
'Requiere tener la referencia con "Microsoft ActiveX Data Objects Library"
'
Dim Path_BDSAP As String
Dim i_NoBD As Integer

Const NombreBDInicial = "BDSAP.mdb"

Dim NombreDBPropuesto As String
Dim NombreBDSAP As String

Dim cat As New ADOX.catalog
Dim DB As New ADODB.Connection
Dim cmd As New ADODB.Command
Dim appAccess As New Access.Application

Dim Mensaje As String

If Not RegresaPath(Directorio:=vbSAPBD, DirPath:=Path_BDSAP) Then Exit Sub
ChDir Path:=Path_BDSAP

i_NoBDSAP = 0
NombreDBPropuesto = NombreBDInicial
Do Until Dir(NombreDBPropuesto) = ""
    i_NoBDSAP = i_NoBDSAP + 1
    NombreDBPropuesto = Left(NombreBDInicial, Len(NombreBDInicial) - 4) _
        & i_NoBDSAP & ".mdb"

```

```

Loop
Do
  If Not RegresaComentEnIdioma("Base de Datos SAP", Mensaje) Then Exit Sub
  NombreBDSAP = Path_BDSAP & "\" & NombreDBPropuesto
  NombreBDSAP = Application.GetSaveAsFilename(InitialFilename:=NombreBDSAP, _
    FileFilter:="Microsoft Access Databases (*.mdb),*.mdb", Title:=Mensaje)
  If NombreBDSAP = "False" Then Exit Sub
  If Dir(NombreBDSAP) <> "" Then
    If Not RegresaComentEnIdioma("La Base de Datos ya existe", Mensaje) Then
      Exit Sub
    End If
    MsgBox Prompt:=Mensaje & ": " & NombreBDSAP, Buttons:=vbExclamation, _
      Title:=TituloSistema
  End If
Loop Until Dir(NombreBDSAP) = ""

cat.Create "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & NombreBDSAP
If AbreBD(DB, BaseDatos:=Dir(NombreBDSAP)) Then
  With DB
    .Execute "CREATE TABLE " & NombreTablaCat & " (IdSerie TEXT (20) " _
      & "CONSTRAINT IdSerie PRIMARY KEY, Nombre TEXT (20), Serie TEXT (5), " _
      & "Tipo TEXT (2), PrimeraFecha DATETIME, UltimaFecha DATETIME, " _
      & "NoDatos LONG);"
    .Execute "CREATE TABLE " & NombreTablaCot & " (Fecha DATETIME, " _
      & "IdSerie TEXT (20), Valor DOUBLE, " _
      & "CONSTRAINT MyTableConstrain PRIMARY KEY (Fecha, IdSerie));"
    .Close
  End With
  cmd.CommandText = "TRANSFORM Avg(Valor) AS [The Value] SELECT Fecha " _
    & "FROM " & NombreTablaCot & " GROUP BY Fecha PIVOT IdSerie;"
  cat.Views.Append "Cotizacion_Crosstab", cmd

  Set appAccess = CreateObject("Access.Application")
  appAccess.OpenCurrentDatabase NombreBDSAP
  appAccess.DoCmd.TransferDatabase acImport, "Microsoft Access", _
    ThisWorkbook.Path & "\Translate.mdb", acTable, "MSysIMEXSpecs", _
    "MSysIMEXSpecs", StructureOnly:=False
  appAccess.DoCmd.TransferDatabase acImport, "Microsoft Access", _
    ThisWorkbook.Path & "\Translate.mdb", acTable, "MSysIMEXColumns", _
    "MSysIMEXColumns", StructureOnly:=False
  appAccess.DoCmd.TransferDatabase acImport, "Microsoft Access", _
    ThisWorkbook.Path & "\Translate.mdb", acTable, NombreTablaCatTipo, _
    NombreTablaCatTipo, StructureOnly:=False
  appAccess.CloseCurrentDatabase
  Set appAccess = Nothing

  If Not RegresaComentEnIdioma("La Base de Datos fue creada exitosamente", _
    Mensaje) Then Exit Sub
  MsgBox Mensaje & ": " & String(2, Chr(13)) & NombreBDSAP, _
    vbExclamation, TituloSistema

  If Not RegresaComentEnIdioma("Desea que la BD este como predeterminada", _
    Mensaje) Then Exit Sub
  If MsgBox(Mensaje & ": " & String(2, Chr(13)) & NombreBDSAP, _
    vbYesNo + vbQuestion, TituloSistema) = vbYes Then
    CambiaPref "DB_Name", Dir(NombreBDSAP)
  End If
Else
  If Not RegresaComentEnIdioma("La Base de Datos NO fue creada", _
    Mensaje) Then Exit Sub
  MsgBox Mensaje, vbCritical, TituloSistema
  Kill NombreBDSAP

```

```

End If
Set cat = Nothing

End Sub

```

```

Function AbreBD _
    (DB As ADODB.Connection, Optional BaseDatos) As Boolean

Dim Valor As String
Dim cnn As New ADODB.Connection
Dim Comentario As String

If IsMissing(BaseDatos) Then
    If Not RegresaPreferencia(NombrePref:="DB_Name", ValorPref:=Valor) Then
        If Not RegresaComentEnIdioma("Base de Datos no definida", _
            Comentario) Then Exit Function
        MsgBox Comentario & " " & BaseDatos, vbExclamation, TituloSistema
        Exit Function
    End If
Else
    Valor = BaseDatos
End If

If Valor = "" Then Exit Function

cnn.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & _
    ThisWorkbook.Path & "\DBase\" & Valor & ";"
cnn.Open
Set DB = cnn
AbreBD = True

End Function

```

```

Sub CargarSeriesBD ()

Dim Nombre_BD As String
Dim BD As ADODB.Connection
Dim BD_path As String
Dim TablaDeBusqueda As String
Dim appAccess As New Access.Application

Dim DirDatos As String
Dim ArrTxt() As String
Dim i As Integer
Dim NombreArchivoLinkTxt As String
Dim NombreArchivoLink As String

Dim SQLstr As String
Dim rst As New ADODB.Recordset
Dim rst1 As New ADODB.Recordset

Dim Ejemplo As String

Dim ArrSeries() As String
Dim StrEmisoras As String
Dim StrIdSeries As String

```

```

Dim StrIdSeriesRepetidas As String
Dim Importado As Boolean

Dim Comentario As String

Dim DescripcionTipo As String

If Not TomaTxt(ArrTxt:=ArrTxt) Then Exit Sub
If Not RegresaPath(Directorio:=vbSAPData, DirPath:=DirDatos) Then Exit Sub

For i = LBound(ArrTxt) To UBound(ArrTxt)
    Application.Cursor = xlWait
    NombreArchivoLinkTxt = ArrTxt(i)

    NombreArchivoLink = Left(NombreArchivoLinkTxt, Len(NombreArchivoLinkTxt) - 4)

    If Not RegresaComentEnIdioma("Cargando", Comentario) Then Exit Sub
    Application.StatusBar = Comentario & " " & NombreArchivoLinkTxt & " ..."

    If Not AbreBD(BD) Then Exit Sub

    NombreArchivoLink = Left(NombreArchivoLinkTxt, Len(NombreArchivoLinkTxt) - 4)

    'Valida que no exista una tabla con el nombre seleccionado NombreArchivoLink
    On Error GoTo MnjErrores
    TablaDeBusqueda = NombreArchivoLink
    SQLstr = "SELECT * FROM " & TablaDeBusqueda
    rst.Open SQLstr, BD, adOpenStatic
    rst.Close
    SQLstr = "DROP TABLE " & NombreArchivoLink
    BD.Execute SQLstr

TablaLinkNoExiste:
    If Not TomaBD_Default(Nombre_BD:=Nombre_BD) Then Exit Sub
    RegresaPath Directorio:=vbSAPBD, DirPath:=BD_path
    BD_path = BD_path & "\" & Nombre_BD

    Set appAccess = Nothing
    Set appAccess = CreateObject("Access.Application")
    appAccess.OpenCurrentDatabase BD_path

    On Error GoTo 0
    appAccess.DoCmd.TransferText TransferType:=acLinkDelim, _
        SpecificationName:="ImportData", TableName:=NombreArchivoLink, _
        FileName:=DirDatos & "\" & NombreArchivoLinkTxt, HasFieldNames:=False
    appAccess.CloseCurrentDatabase
    Set appAccess = Nothing
    Importado = True

    Application.Cursor = xlDefault

    If TomaSeries(BD:=BD, NombreTabla:=NombreArchivoLink, _
        ArrSeries:=ArrSeries, TipoAccion:="Cargar") Then

        StrIdSeries = GeneraStrDeArreglo(ArrSeries, 0)
        StrEmisoras = GeneraStrDeArreglo(ArrSeries, 1)
        SQLstr = "SELECT DISTINCT IdSerie FROM " & NombreTablaCat & _
            " WHERE IdSerie IN (" & StrIdSeries & ")"
        rst.Open SQLstr, BD, adOpenStatic
        If rst.RecordCount > 0 Then
            StrIdSeriesRepetidas = GeneraStrDeRecordset(rst)
            If MsgBox(Prompt:="Ya existen las siguientes series (" & _
                rst.RecordCount & "): " & StrIdSeriesRepetidas & String(2, vbCr) & _
                "Desea reemplazarlas", Buttons:=vbYesNo + vbQuestion, _
                Title:=TituloSistema) = vbNo Then
                Exit Sub
            End If
        End If
    End If

```

```

End If

Application.StatusBar = "Borrando las series del catalogo. " & _
    rst.RecordCount & " series ..."
SQLstr = "DELETE * FROM " & NombreTablaCat & " WHERE IdSerie IN (" & _
    StrIdSeriesRepetidas & ")"
BD.Execute SQLstr

Application.StatusBar = "Borrando las series de la tabla de " & _
    "Cotizaciones. " & rst.RecordCount & " series ..."
SQLstr = "DELETE * FROM " & NombreTablaCot & " WHERE IdSerie IN (" & _
    StrIdSeriesRepetidas & ")"
BD.Execute SQLstr

Application.StatusBar = False
End If
rst.Close

On Error GoTo MnjErrores
TablaDeBusqueda = NombreTablaCotTemp
SQLstr = "SELECT * FROM " & TablaDeBusqueda
rst.Open SQLstr, BD, adOpenStatic
rst.Close
SQLstr = "DROP TABLE " & NombreTablaCotTemp
BD.Execute SQLstr

TablaCotTempNoExiste:
On Error GoTo 0
SQLstr = "CREATE TABLE " & NombreTablaCotTemp & " (Fecha DATETIME, " & _
    "IdSerie TEXT (20), Nombre TEXT (20), Serie TEXT (5), Tipo TEXT (2), " & _
    "Valor DOUBLE, CONSTRAINT MyTableConstrain PRIMARY KEY (Fecha, IdSerie));"
BD.Execute SQLstr

Application.Cursor = xlWait
Application.StatusBar = "Cargando series (" & _
    UBound(ArrSeries, 2) + 1 & ")..."
SQLstr = "INSERT INTO " & NombreTablaCotTemp & " (Fecha, IdSerie, " & _
    "Nombre, Serie, Tipo, Valor) SELECT Fecha, Emisora & "" "" & " & _
    "Serie AS IdSerie, Emisora, Serie, Tipo, Valor FROM " & _
    NombreArchivoLink & " WHERE Emisora IN (" & StrEmisoras & ")"
BD.Execute SQLstr

SQLstr = "INSERT INTO " & NombreTablaCot & " (Fecha, IdSerie, Valor) " & _
    "SELECT Fecha, IdSerie, Valor FROM " & NombreTablaCotTemp & _
    " WHERE IdSerie IN (" & StrIdSeries & ")"
BD.Execute SQLstr

Application.StatusBar = "Actualizando información " & _
    "en el catálogo de Series..."
SQLstr = "INSERT INTO " & NombreTablaCat & " (IdSerie, Nombre, Serie, " & _
    "Tipo, PrimeraFecha, UltimaFecha, NoDatos) SELECT IdSerie, Nombre, " & _
    "Serie, Tipo, MIN(Fecha), MAX(Fecha), COUNT(Fecha) FROM " & _
    NombreTablaCotTemp & " WHERE IdSerie IN (" & StrIdSeries & ") " & _
    "GROUP BY IdSerie, Nombre, Serie, Tipo"
BD.Execute SQLstr

SQLstr = "DROP TABLE " & NombreTablaCotTemp
BD.Execute SQLstr

SQLstr = "SELECT DISTINCT Tipo FROM " & NombreTablaCat & " WHERE Tipo " & _
    "NOT IN (SELECT Tipo FROM " & NombreTablaCatTipo & ")"
rst.Open SQLstr, BD, adOpenStatic

Do Until rst.EOF
    SQLstr = "SELECT IdSerie FROM " & NombreTablaCat & " WHERE Tipo = '" & _
        rst("Tipo") & "'"
    rst1.Open SQLstr, BD, adOpenStatic

```

```

Do Until rst1.EOF
    Ejemplo = Ejemplo & rst1("IdSerie") & ", "
    rst1.MoveNext
Loop
rst1.Close
Ejemplo = Left(Ejemplo, Len(Ejemplo) - 2)

DescripcionTipo = InputBox("Dame la descripcion del tipo de Serie "" & _
    rst("Tipo") & ""."" & Chr(13) & "Ejemplo: " & Ejemplo)

If DescripcionTipo <> "" Then
    SQLstr = "INSERT INTO " & NombreTablaCatTipo & " SELECT '" & _
        rst("Tipo") & "' AS Tipo, '" & DescripcionTipo & "' AS Descripcion"
    BD.Execute SQLstr
End If

    rst.MoveNext
Loop
rst.Close

CalculaNoSeriesPorTipo BD

MsgBox "Las series ya fueron cargadas " & UBound(ArrSeries, 2) + 1, _
    vbInformation, TituloSistema

End If

SQLstr = "DROP TABLE " & NombreArchivoLink
BD.Execute SQLstr

BD.Close

Application.StatusBar = False
Application.Cursor = xlDefault

Next i

If Importado Then
    CompactaBD Nombre_BD, EsBatch:=True
End If

Exit Sub

MnjErrores:

Select Case Err()
    Case -2147217865
        'The Microsoft Jet database engine cannot find
        'the input table or query 'CotizacionImportTemp'.
        'Make sure it exists and that its name is spelled correctly.
        Select Case TablaDeBusqueda
            Case NombreArchivoLink
                Resume TablaLinkNoExiste
            Case NombreTablaCotTemp
                Resume TablaCotTempNoExiste
            Case Else
                Stop
        End Select
    Case Else
        Stop
End Select

End Sub

```



```

Sub ImportarDatosBD ()

Dim Nombre_BD As String
Dim BD_path As String
Dim BD As ADODB.Connection
Dim ArrSeries() As String
Dim j As Integer
Dim Agrup As String

Dim SQLstr As String
Dim rst As New ADODB.Recordset
Dim m As Integer
Dim TipoPrimCol As String

Dim Libro As Workbook

Dim NombreDatos As String
Dim HojaNueva As Worksheet

Dim ArrLetrasReservadas

Dim ArrDeValInfo
Dim PrimRen
Dim Comentario As String

If Not RegresaComentEnIdioma("importar", Comentario) Then Exit Sub
If Not TomaLibro(Libro, Comentario) Then Exit Sub

If Not TomaBD_Default(Nombre_BD:=Nombre_BD) Then Exit Sub
If Not RegresaPath(Directorio:=vbSAPBD, DirPath:=BD_path) Then Exit Sub

BD_path = BD_path & "\" & Nombre_BD
If Dir(BD_path) = "" Then
    If RegresaComentEnIdioma("No existe la Base de Datos", Comentario) Then
        MsgBox Comentario & ":" & String(2, vbCrLf) & BD_path, _
            Buttons:=vbCritical, Title:=SiglaSistema
    End If
    Exit Sub
End If
If Not AbreBD(BD) Then Exit Sub

If TomaSeries(BD:=BD, NombreTabla:=NombreTablaCot, _
    ArrSeries:=ArrSeries, TipoAccion:="Importar") Then
    If Not RegresaComentEnIdioma("Consultando los precios de la Base " & _
        "de Datos", Comentario) Then Exit Sub
    Application.StatusBar = Comentario & "..."
    Application.Cursor = xlWait
    For j = LBound(ArrSeries) To UBound(ArrSeries)
        Agrup = Agrup & "'" & ArrSeries(j) & "',"
    Next j
    Agrup = Left(Agrup, Len(Agrup) - 1)
    SQLstr = "TRANSFORM Avg(Valor) AS [The Value] SELECT Fecha FROM " & _
        NombreTablaCot & " WHERE IdSerie IN (" & Agrup & ") " & _
        "GROUP BY Fecha PIVOT IdSerie"
    rst.Open SQLstr, BD, adOpenStatic

    If rst.EOF Then
        If Not RegresaComentEnIdioma("No hay informacion sobre las series " & _
            "seleccionadas", Comentario) Then Exit Sub
        MsgBox Comentario & ":" & String(2, Chr(13)) & Agrup, _
            vbCritical, TituloSistema
    Else
        rst.MoveLast
        m = rst.RecordCount
        TipoPrimCol = "Fechas"
    End If
End If

```

```
Application.StatusBar = False
Application.Cursor = xlDefault
```

```
PideNombre:
```

```
NombreDatos = "Data"
If Not RegresaComentEnIdioma("Ingrese el nombre de los datos", _
Comentario) Then Exit Sub
NombreDatos = Application.InputBox(Comentario, _
Title:=TituloSistema, Default:=NombreDatos)
If Len(NombreDatos) = 0 Then
If Not RegresaComentEnIdioma("No puede estar en blanco", _
Comentario) Then Exit Sub
MsgBox "No puede estar en blanco", Buttons:=vbExclamation, _
Title:="Error-Data": GoTo PideNombre
End If
If Len(NombreDatos) > 31 Then
If Not RegresaComentEnIdioma("No puede ser mayor de 31 caracteres", _
Comentario) Then Exit Sub
MsgBox "No puede ser mayor de 31 caracteres", Buttons:=vbExclamation, _
Title:="Error-Data": GoTo PideNombre
End If

If Not RegresaComentEnIdioma("No puede tener", Comentario) Then Exit Sub
ArrLetrasReservadas = Array(":", "\", "/", "?", "*", "[", "]")
For j = LBound(ArrLetrasReservadas) To UBound(ArrLetrasReservadas)
If InStr(1, NombreDatos, ArrLetrasReservadas(j)) > 0 Then
MsgBox Comentario & " "" & ArrLetrasReservadas(j) & """, _
Buttons:=vbExclamation, Title:="Error-Data"
GoTo PideNombre
End If
Next j

If NombreDatos <> "False" Then
Application.Cursor = xlWait
Libro.Activate

ReDim ArrDeInfo(0) As String
ArrDeInfo(0) = "Datos"
ReDim ArrDeValInfo(0) As String
ArrDeValInfo(0) = "Precios"

Application.ScreenUpdating = False
If Not SigHoja(Libro:=Libro, TipoHoja:="Precios", _
NombreDatos:=NombreDatos, NuevaHoja:=HojaNueva, _
SaltaBorrar:=False) Then
Exit Sub
End If
rst.MoveFirst
With HojaNueva
.Cells(4, 1).CopyFromRecordset rst
For j = 0 To rst.Fields.Count - 1
.Cells(3, j + 1).Value = rst.Fields(j).Name
Next
End With

FormatoHojas Hoja:=HojaNueva, ArrInfor:=ArrDeInfo, _
ArrValInf:=ArrDeValInfo, m:=m, n:=rst.Fields.Count - 1, _
TipoPrimCol:=TipoPrimCol, SigRen:=PrimRen
If Not RegresaPreferencia(NombrePref:="Idioma", ValorPref:=Idioma) Then
ActiveWorkbook.Names.Add Name:="Idioma", RefersTo:=Idioma
End If
If Not RegresaPreferencia(NombrePref:="DB_Name", _
ValorPref:=Nombre_BD) Then
ActiveWorkbook.Names.Add Name:="DB_Name", RefersTo:=Nombre_BD
End If
```

```

        Application.ScreenUpdating = True

    End If

End If

Application.Cursor = xlDefault
rst.Close
End If

BD.Close

End Sub

```

```

Sub BorrarSeriesBD ()

Dim Nombre_BD As String
Dim BD_path As String
Dim BD As ADODB.Connection
Dim ArrSeries() As String
Dim j As Integer

Dim SQLstr As String

If Not TomaBD_Default(Nombre_BD:=Nombre_BD) Then Exit Sub
If Not RegresaPath(Directorio:=vbSAPBD, DirPath:=BD_path) Then Exit Sub
BD_path = BD_path & "\" & Nombre_BD
If Dir(BD_path) = "" Then
    Exit Sub
End If

If Not AbreBD(BD) Then Exit Sub

If TomaSeries(BD:=BD, NombreTabla:=NombreTablaCot, _
    ArrSeries:=ArrSeries, TipoAccion:="Borrar") Then

    For j = LBound(ArrSeries) To UBound(ArrSeries)
        SQLstr = "DELETE * FROM " & NombreTablaCot & " WHERE IdSerie = '" & _
            ArrSeries(j) & "'"
        BD.Execute SQLstr
        SQLstr = "DELETE * FROM " & NombreTablaCat & " WHERE IdSerie = '" & _
            ArrSeries(j) & "'"
        BD.Execute SQLstr
    Next j
    CalculaNoSeriesPorTipo BD
    BD.Close
    If UBound(ArrSeries) Then
        CompactaBD Nombre_BD:=Nombre_BD, EsBatch:=True
    End If

Else

    BD.Close

End If

End Sub

```

```

Function TomaSeries(BD As ADODB.Connection, _
    NombreTabla As String, ArrSeries() As String, TipoAccion As String) As Boolean

Dim SQLstr As String
Dim rst As New ADODB.Recordset
Dim Idioma As String
Dim j As Integer
Dim cont_sel As Integer
Dim Titulo As String
Dim Comentario1 As String
Dim Comentario2 As String
Dim ArrTipoSeries() As String
Dim StrTipoSeries As String

Dim i As Long
Dim ContCiclos As Integer

Application.Cursor = xlWait
Select Case TipoAccion
    Case "Importar", "Borrar"
        If Not RegresaComentEnIdioma("Selección de Series", Titulo) Then _
            Exit Function

        'Filtro tipo de instrumento
        SQLstr = "SELECT * FROM " & NombreTablaCatTipo & " WHERE NoSeries > 0;"
        rst.Open SQLstr, BD, adOpenStatic
        If rst.EOF Then
            Application.Cursor = xlDefault
            If Not RegresaComentEnIdioma("No hay ninguna serie en la Base de Datos", _
                Comentario1) Then Exit Function
            MsgBox Comentario1 & String(2, Chr(13)) & BD.Properties(7), _
                vbExclamation, Titulo
            Exit Function
        End If
        rst.MoveLast
        ReDim ArrTipoSeries(rst.RecordCount - 1, 2) As String
        rst.MoveFirst
        j = 0
        Do Until rst.EOF
            ArrTipoSeries(j, 0) = rst("Tipo").Value
            If Not IsNull(rst("Descripcion").Value) Then ArrTipoSeries(j, 1) = _
                rst("Descripcion").Value
            ArrTipoSeries(j, 2) = rst("NoSeries").Value
            j = j + 1
            rst.MoveNext
        Loop
        With frm_Select
            .Caption = Titulo
            If Not RegresaComentEnIdioma("Tipos de series", Comentario1) Then _
                Exit Function
            .LblSelect.Caption = Comentario1
            If Not RegresaComentEnIdioma("Seleccionar", Comentario1) Then _
                Exit Function
            .BtnOk.Caption = Comentario1
            .BtnOk.Enabled = True
            If Not RegresaComentEnIdioma("Cancelar", Comentario1) Then Exit Function
            .BtnCancel.Caption = Comentario1
            .BtnCancel.Enabled = True

            .LstLists.Clear
            .LstLists.MultiSelect = fmMultiSelectExtended
            .LstLists.ColumnCount = 3
            .LstLists.ColumnWidths = "40 pt;100 pt;20 pt"
            .LstLists.List() = ArrTipoSeries
            .UserCancel = False
            Application.Cursor = xlDefault
        End With
    End Select
End Function

```

```

.Show
If .UserCancel Then
    Exit Function
Else
    cont_sel = -1
    StrTipoSeries = ""
    ReDim ArrTipoSeries(.LstLists.ListCount - 1) As String
    For j = 0 To .LstLists.ListCount - 1
        If .LstLists.Selected(j) Then
            cont_sel = cont_sel + 1
            ArrTipoSeries(cont_sel) = .LstLists.List(j)
            StrTipoSeries = StrTipoSeries & "'" & .LstLists.List(j) & "', "
        End If
    Next j
    If cont_sel < 0 Then Exit Function
    ReDim Preserve ArrTipoSeries(cont_sel)
    StrTipoSeries = Left(StrTipoSeries, Len(StrTipoSeries) - 2)
End If
End With
rst.Close

SQLstr = "SELECT " & NombreTablaCat & ".*, " & NombreTablaCatTipo & _
".Descripcion FROM " & NombreTablaCat
SQLstr = SQLstr & " LEFT JOIN " & NombreTablaCatTipo & " ON " & _
NombreTablaCat & ".Tipo = " & NombreTablaCatTipo & ".Tipo "
SQLstr = SQLstr & "WHERE " & NombreTablaCat & ".Tipo IN (" & _
StrTipoSeries & ") "
SQLstr = SQLstr & "ORDER BY " & NombreTablaCat & ".Tipo, IdSerie;"

rst.Open SQLstr, BD, adOpenStatic
If rst.EOF Then
    If Not RegresaComentEnIdioma("No hay ninguna serie en la Base de Datos", _
Comentariol) Then Exit Function
    MsgBox Comentariol & String(2, Chr(13)) & BD.Properties(7), _
vbExclamation, Titulo
    Exit Function
End If
rst.MoveLast
ReDim ArrSeries(rst.RecordCount - 1, 4) As String
rst.MoveFirst
j = 0
Do Until rst.EOF
    ArrSeries(j, 0) = rst("IdSerie").Value
    If Not IsNull(rst("Nombre").Value) Then _
        ArrSeries(j, 1) = rst("Nombre").Value
    ArrSeries(j, 2) = rst("Serie").Value
    ArrSeries(j, 3) = rst("Tipo").Value
    If Not IsNull(rst("Descripcion").Value) Then _
        ArrSeries(j, 4) = rst("Descripcion").Value
    j = j + 1
    rst.MoveNext
Loop

Case "Cargar"

    If Not RegresaComentEnIdioma("Carga de Series de Archivos Txt", Titulo) Then
        Exit Function
    End If

    'El link tarda en actualizarse en la base de datos por lo que puede
    'generar un error esta consulta. Manda a esperar unos segundos mas.
    On Error GoTo MngErrores
    SQLstr = "SELECT * FROM " & NombreTabla & ";"
    rst.Open SQLstr, BD, adOpenStatic
    On Error GoTo 0
    If rst.Fields(0).Name <> "Tipo" Or rst.Fields(1).Name <> "Emisora" Or _
rst.Fields(2).Name <> "Serie" Or rst.Fields(3).Name <> "Fecha" Or _

```

```

rst.Fields(4).Name <> "Valor" Then
  Application.Cursor = xlDefault
  If Not RegresaComentEnIdioma("Error en el formato del archivo Txt. " & _
    "El archivo debe tener los siguientes campos en dicho orden:", _
    Comentario1) Then Exit Function
  MsgBox Prompt:=Comentario1 & Chr(13) & "Tipo" & Chr(13) & "Emisora" & _
    Chr(13) & "Serie" & Chr(13) & "Fecha" & Chr(13) & "Valor", _
    Buttons:=vbExclamation, Title:=TituloSistema
  Exit Function
End If
rst.Close

SQLstr = "SELECT DISTINCT Emisora, Serie, " & NombreTabla & ".Tipo, "
SQLstr = SQLstr & "Descripcion FROM " & NombreTabla & " LEFT JOIN "
SQLstr = SQLstr & "Cat_TipoInstr ON " & NombreTabla & _
  ".Tipo = Cat_TipoInstr"
SQLstr = SQLstr & ".Tipo ORDER BY " & NombreTabla & ".Tipo, Emisora, Serie;"

rst.Open SQLstr, BD, adOpenStatic

rst.MoveLast
ReDim ArrSeries(rst.RecordCount - 1, 4) As String
rst.MoveFirst
j = 0
Do Until rst.EOF
  ArrSeries(j, 0) = rst("Emisora") & " " & rst("Serie")
  ArrSeries(j, 1) = rst("Emisora")
  ArrSeries(j, 2) = rst("Serie")
  ArrSeries(j, 3) = rst("Tipo")
  If Not IsNull(rst("Descripcion")) Then
    ArrSeries(j, 4) = rst("Descripcion")
  End If
  j = j + 1
  rst.MoveNext
Loop

End Select
rst.Close

With frm_Select
  .Caption = Titulo
  If Not RegresaComentEnIdioma("Selecciona las series a", _
    Comentario1) Then Exit Function
  If Not RegresaComentEnIdioma(TipoAccion, Comentario2) Then Exit Function
  .LblSelect.Caption = Comentario1 & " " & Comentario2
  .BtnOk.Caption = Comentario2
  .BtnOk.Enabled = True
  If Not RegresaComentEnIdioma("Cancelar", Comentario1) Then Exit Function
  .BtnCancel.Caption = Comentario1
  .BtnCancel.Enabled = True

  .LstLists.Clear
  .LstLists.MultiSelect = fmMultiSelectExtended
  .LstLists.ColumnCount = 5
  .LstLists.ColumnWidths = "77 pt;55 pt;35 pt;20 pt;100 pt"
  .LstLists.List() = ArrSeries

  'Selecciona todas las series
  For j = 0 To .LstLists.ListCount - 1
    .LstLists.Selected(j) = True
  Next j

  Application.Cursor = xlDefault
  .UserCancel = False
  RepiteSeleccion:
  .Show
  If Not .UserCancel Then

```

```

cont_sel = -1
If NombreTabla = NombreTablaCot Then
  ReDim ArrSeries(.LstLists.ListCount - 1) As String
  For j = 0 To .LstLists.ListCount - 1
    If .LstLists.Selected(j) Then
      TomaSeries = True
      cont_sel = cont_sel + 1
      ArrSeries(cont_sel) = .LstLists.List(j, 0)
    End If
  Next j
  ReDim Preserve ArrSeries(cont_sel)
Else
  'ArrSeries son 3 datos por serie para importar.
  '1 SerieId, 2 Emisora, 3 Serie
  ReDim ArrSeries(2, .LstLists.ListCount - 1) As String
  For j = 0 To .LstLists.ListCount - 1
    If .LstLists.Selected(j) Then
      TomaSeries = True
      cont_sel = cont_sel + 1
      ArrSeries(0, cont_sel) = .LstLists.List(j, 0)
      ArrSeries(1, cont_sel) = .LstLists.List(j, 1)
      ArrSeries(2, cont_sel) = .LstLists.List(j, 2)
    End If
  Next j
  If cont_sel > 255 Then
    MsgBox "No se puede tener mas de 255 series", _
      vbExclamation, TituloSistema
    GoTo RepiteSeleccion
  End If
  ReDim Preserve ArrSeries(2, cont_sel)
End If
End If
End With

Exit Function

MngErrores:
Select Case Err
  Case -2147217865
    'The Microsoft Jet database engine cannot find the input table or query 'x'.
    'Make sure it exists and that its name is spelled correctly.
    ContCiclos = ContCiclos + 1
    If ContCiclos >= 15 Then Stop
    For i = 1 To 100000000
      Next i
    Resume
  Case Else
    Stop
End Select

End Function

```

```

Function GeneraStrDeRecordset(rst As ADODB.Recordset) As String

Dim str As String

Do Until rst.EOF
  str = str & "'" & rst(0).Value & "',"
  rst.MoveNext
Loop
GeneraStrDeRecordset = Left(str, Len(str) - 1)

```

```
End Function
```

```
Sub CalculaNoSeriesPorTipo _  
    (BD As ADODB.Connection) _  
  
    Dim SQLstr As String  
    Dim rst As New ADODB.Recordset  
  
    'Inicializa a cero Numero de Series por Tipo  
    SQLstr = "UPDATE " & NombreTablaCatTipo & " SET NoSeries = 0"  
    BD.Execute SQLstr  
  
    'Calcula Numero de Series por Tipo  
    SQLstr = "SELECT Tipo, COUNT(IdSerie) AS NoSeries FROM " & NombreTablaCat & _  
        " GROUP BY Tipo"  
    rst.Open SQLstr, BD, adOpenStatic  
    Do Until rst.EOF  
        SQLstr = "UPDATE " & NombreTablaCatTipo & " SET NoSeries = " & _  
            rst("NoSeries") & " WHERE Tipo = '" & rst("Tipo") & "'"  
        BD.Execute SQLstr  
        rst.MoveNext  
    Loop  
    rst.Close  
  
End Sub
```

```
Sub CompactaBD (Optional Nombre_BD, Optional EsBatch)  
  
    Dim Comentario As String  
    Dim Comentario1 As String  
    Dim BD_path As String  
    Dim Nombre_TempBD As String  
  
    Dim Mensaje As String  
  
    If IsMissing(Nombre_BD) Then  
        If Not RegresaComentEnIdioma("Seleccionar Base de Datos a compactar", _  
            Comentario) Then Exit Sub  
        If Not RegresaComentEnIdioma("Compactar", Comentario1) Then Exit Sub  
        If Not TomaBD(Nombre_BD:=BD_path, Mensaje:=Comentario, _  
            NombreBoton:=Comentario1) Then Exit Sub  
        Nombre_BD = BD_path  
    End If  
  
    If Not RegresaPath(Directorio:=vbsAPBD, DirPath:=BD_path) Then Exit Sub  
    Nombre_BD = BD_path & "\" & Nombre_BD  
  
    If Not RegresaComentEnIdioma("Compactando Base de Datos", Mensaje) Then _  
        Exit Sub  
    Application.StatusBar = Mensaje & ": " & Nombre_BD & " ..."  
  
    Application.Cursor = xlWait  
  
    If IsMissing(EsBatch) Then EsBatch = False  
    Nombre_TempBD = Left(Nombre_BD, Len(Nombre_BD) - Len(Dir(Nombre_BD))) & _  
        "tmp.mdb"
```



```

On Error GoTo MnjErr
    DBEngine.CompactDatabase Nombre_BD, Nombre_TempBD
On Error GoTo 0
Kill Nombre_BD
Name Nombre_TempBD As Nombre_BD

Application.StatusBar = False
If Not EsBatch Then
    If Not RegresaComentEnIdioma("La Base de Datos ya fué compactada", _
        Mensaje) Then Exit Sub
    MsgBox Mensaje & ": " & String(2, Chr(13)) & Nombre_BD, vbInformation, _
        TituloSistema
End If
Application.Cursor = xlDefault

Exit Sub

MnjErr:

Select Case Err
Case 3356
    'You attempted to open a database that is already opened exclusively by
    'user 'Admin' on machine 'DS9'. Try again when the database is available.
    If Not RegresaComentEnIdioma("La base de datos que esta compactando " & _
        "esta abierta. Para continuar necesita cerrarla.", Comentario) Then _
        Exit Sub
    MsgBox Prompt:=Comentario & String(2, Chr(13)) & Nombre_BD, _
        Title:=TituloSistema
Case Else
    MsgBox Prompt:=Error, Title:=Err
Resume
End Select
Application.StatusBar = False
Application.Cursor = xlDefault

End Sub

```

Hojas

```

Function SigHoja(Libro As Workbook, TipoHoja As String, _
    NombreDatos As String, NuevaHoja As Worksheet, SaltaBorrar As Boolean, _
    Optional HojaPreferencias, Optional SubTipoHoja) As Boolean

Dim Consecutivo As Integer
Dim Siguiente As Integer
Dim Hoja As Worksheet
Dim Tipo As String
Dim Nombre As String
Dim Consec As Integer
Dim NombreSiguiente As String

Consecutivo = RegresaConsecutivo(Libro:=Libro, Nombre:=NombreDatos, _
    TipoHoja:=TipoHoja, NombreSiguiente:=NombreSiguiente)

If IsMissing(SaltaBorrar) Then SaltaBorrar = False
If SaltaBorrar Then
    Siguiente = Consecutivo
Else
    If NombreSiguiente <> "" Then

```

```

If Not IsMissing(HojaPreferencias) Then
  If HojaPreferencias.Name = NombreSiguiente Then
    Siguiente = Consecutivo
    GoTo NoBorrar
  End If
End If
Select Case MsgBox("Desea conservar " & NombreSiguiente & " de " & _
ActiveWorkbook.Name, Buttons:=vbYesNoCancel + vbQuestion, _
Title:=TituloSistema)
  Case vbYes
    Siguiente = Consecutivo
  Case vbNo
    BorrarHoja Sheets(NombreSiguiente), False
    Siguiente = Consecutivo - 1
  Case vbCancel
    Exit Function
End Select
NoBorrar:
  Else
    Siguiente = Consecutivo
  End If
End If
NombreSiguiente = TipoHoja & "(" & Siguiente & ")" & NombreDatos

If TipoHoja = "Grafica" Then
  Set NuevaHoja = Charts.Add
Else
  Set NuevaHoja = Worksheets.Add
End If

NuevaHoja.Name = NombreSiguiente
NuevaHoja.Names.Add NameLocal:="" & NombreSiguiente & "!Nombre", _
  RefersToR1C1Local:=NombreDatos
NuevaHoja.Names.Add NameLocal:="" & NombreSiguiente & "!Tipo", _
  RefersToR1C1Local:=TipoHoja
If Not IsMissing(SubTipoHoja) Then NuevaHoja.Names.Add _
  NameLocal:="" & NombreSiguiente & "!SubTipo", RefersToR1C1Local:=SubTipoHoja
NuevaHoja.Names.Add NameLocal:="" & NombreSiguiente & "!Consecutivo", _
  RefersToR1C1Local:=Siguiente

SigHoja = True
If TipoHoja <> "Grafica" Then ActiveWindow.Zoom = 70

End Function

```

```

Function RegresaConsecutivo(Libro As Workbook, _
  Nombre As String, TipoHoja As String, NombreSiguiente As String) As Integer

Dim Hoja As Worksheet
Dim Valor As String
Dim Consecutivo As Integer

Consecutivo = 1
For Each Hoja In Libro.Worksheets
  If RegresaPreferencia(NombrePref:="Nombre", ValorPref:=Valor, Hoja:=Hoja, _
  DespliegaComentario:=False) Then
    If Valor = Nombre Then
      If RegresaPreferencia(Hoja:=Hoja, NombrePref:="Tipo", _
      ValorPref:=Valor) Then
        If TipoHoja = Valor Then
          If RegresaPreferencia(Hoja:=Hoja, NombrePref:="Consecutivo", _

```

```

        ValorPref:=Valor) Then
        If Valor >= Consecutivo Then
            NombreSiguiente = Hoja.Name
            Consecutivo = Valor + 1
        End If
    End If
End If
End If
End If
End If
Next Hoja
RegresaConsecutivo = Consecutivo

End Function

```

```

Function ObtenerHojas(Optional Titulo, _
    Optional ArrTipo, Optional ArrSubTipo, Optional Comentario, _
    Optional MultipleHoja, Optional ArrHojas) As Boolean

Dim ArrDeTipoHoja
Dim Hoja As Worksheet
Dim TipoDeHoja As String
Dim SubTipoDeHoja As String
Dim i As Integer
Dim j As Integer
Dim l As Integer
Dim LibroSel As Integer
Dim EsSubTipo As Boolean
Dim k As Integer

Set HojaInicial = ActiveSheet
If IsMissing(Titulo) Then
    Titulo = "Obtener Datos [" & HojaInicial.Parent.Name & "]"
If IsMissing(Comentario) Then Comentario = "que contiene los datos:"
If IsMissing(ArrTipo) Then
    ArrDeTipoHoja = Array("Precios", "Rendimientos", "Análisis", _
        "MatCov", "MatCor", "Port")
    ArrSubTipo = Array("Precios", "Rendimientos", "Múltiple")
Else
    ArrDeTipoHoja = ArrTipo
End If
If IsMissing(MultipleHoja) Then MultipleHoja = False
With frm_Hoja
    .Caption = Titulo
    .LstHojas.Clear
    If MultipleHoja Then
        .LblHoja.Caption = "Escoja las hojas (0) " & Comentario
        .LstHojas.MultiSelect = fmMultiSelectMulti
    Else
        .LblHoja.Caption = "Escoja la hoja " & Comentario
        .LstHojas.MultiSelect = fmMultiSelectSingle
    End If
    .CmbTipo.List() = ArrDeTipoHoja
    .CmbTipo.AddItem pvargItem:="Todas las hojas", pvargindex:=0
    .CmbTipo.ListIndex = 0
    ReDim ArrDeLibros(0) As String
    l = 0
    For i = 1 To Workbooks.Count
        For Each Hoja In Workbooks(i).Worksheets
            If RegresaPreferencia("Tipo", TipoDeHoja, Hoja, _
                DespliegaComentario:=False) Then

```

```

For j = 0 To .CmbTipo.ListCount - 1
  If TipoDeHoja = .CmbTipo.List(j) Then
    EsSubTipo = False
    If TipoDeHoja = "Analisis" Then
      If RegresaPreferencia("SubTipo", SubTipoDeHoja, Hoja) Then
        For k = 0 To UBound(ArrSubTipo)
          If ArrSubTipo(k) = SubTipoDeHoja Then
            EsSubTipo = True
            Exit For
          End If
        Next k
      Else
        MsgBox "La hoja de Analisis " & Hoja.Name & _
          " no tiene subtipo", vbExclamation, TituloSistema
      End If
    Else
      EsSubTipo = True
    End If
    If EsSubTipo Then
      ReDim Preserve ArrDeLibros(1) As String
      ArrDeLibros(1) = Workbooks(1).Name
      If Workbooks(1).Name = ActiveWorkbook.Name Then LibroSel = 1
      l = l + 1
      GoTo OtroLibro
    End If
  End If
Next j
End If
Next Hoja
OtroLibro:
Next i
.CmbLibro.Clear
If l < 2 Then
  .CmbLibro.Enabled = False
Else
  .CmbLibro.Enabled = True
End If
.ArrSubTipoHoja = ArrSubTipo
If l = 0 Then
  .CmbLibro.Text = "No hay libros con datos"
  .CmbTipo.Enabled = False
  .BtnContinuar.Enabled = False
  .BtnCerrar.Enabled = False
Else
  .CmbLibro.List() = ArrDeLibros
  .CmbLibro.ListRows = Application.Min(8, UBound(ArrDeLibros) + 1)
  .CmbLibro.ListIndex = LibroSel
  .CmbTipo.Enabled = True
  If .BtnContinuar.Enabled Then .BtnContinuar.SetFocus
End If
.UserCancel = False
.Show
If Not .UserCancel Then
  ReDim ArrHojas(0)
  l = 0
  For i = 0 To .LstHojas.ListCount - 1
    If .LstHojas.Selected(i) Then
      ReDim Preserve ArrHojas(1)
      Set ArrHojas(1) = Sheets(.LstHojas.List(i))
      l = l + 1
    End If
  Next i
  If l > 0 Then ObtenerHojas = True
End If
End With
If Not HojaInicial Is Nothing Then
  HojaInicial.Activate

```

```
End If  
End Function
```

```
Function TomaLibro(Libro As Workbook, _  
    ComentarioDeSeleccion As String) As Boolean  
  
Dim ArrLibros() As String  
Dim i As Integer  
Dim Comentario As String  
  
If ThisWorkbook.IsAddin And Workbooks.Count = 0 Or  
    (Not ThisWorkbook.IsAddin) And Workbooks.Count < 2 Then  
    If Not RegresaComentEnIdioma("No hay ningun libro abierto para", _  
        Comentario) Then Exit Function  
    MsgBox Prompt:=Comentario & " " & ComentarioDeSeleccion, _  
        Buttons:=vbExclamation, Title:=TituloSistema  
Else  
    If ActiveWorkbook.Name = ThisWorkbook.Name Then  
        ReDim ArrLibros(Workbooks.Count - 2) As String  
        For Each Libro In Workbooks  
            If Libro.Name <> ThisWorkbook.Name Then  
                ArrLibros(i) = Libro.Name  
                i = i + 1  
            End If  
        Next Libro  
        With frm_Select  
            If Not RegresaComentEnIdioma("Libro", Comentario) Then _  
                Exit Function  
            .Caption = Comentario  
            If Not RegresaComentEnIdioma("Seleccione el libro", _  
                Comentario) Then Exit Function  
            .LblSelect.Caption = Comentario  
            If Not RegresaComentEnIdioma("Seleccionar", Comentario) Then _  
                Exit Function  
            .BtnOk.Caption = Comentario  
            If Not RegresaComentEnIdioma("Cancelar", Comentario) Then _  
                Exit Function  
            .BtnCancel.Caption = Comentario  
  
            .LstLists.MultiSelect = fmMultiSelectSingle  
            .LstLists.List = ArrLibros  
            .LstLists.ListIndex = 0  
            .UserCancel = False  
            .Show  
            If .UserCancel Then Exit Function  
            Set Libro = Workbooks(.LstLists.Value)  
            Libro.Activate  
        End With  
        TomaLibro = True  
    Else  
        Set Libro = ActiveWorkbook  
        TomaLibro = True  
    End If  
End If  
End Function
```

```

Sub FormatoHojas(Hoja, Optional ArrInfor, Optional ArrValInf, _
Optional ArrOtrosDatos, Optional ArrTitulos, Optional ArrPrimCol, _
Optional TipoPrimCol, Optional ArrDatos, Optional m, Optional n, _
Optional SigRen, Optional SigCol, Optional HojaPreferencias)

Dim NombreHoja As String
Dim ArrDeSecciones
Dim ArrBoolSecc
Dim k As Integer 'Contador de partes a cambiar
Dim ArrColModif
Dim RangoPrimCol As Range
'Indica si ya se le dió formato a la primera columna
Dim PrimColFormato As Boolean
Dim i As Integer
Dim j As Integer
Dim TieneOtrosDatos As Boolean
Dim TieneTiempo As Boolean
Dim TieneCaract As Boolean
Dim TieneOtrosTits As Boolean
Dim TieneDatos As Boolean
Dim Tipo As String

ArrDeSecciones = Array("Información", "OtrosDatos", "Tiempo", _
"Funciones", "OtrosTitulos", "Datos", "Titulos", "Series")
ReDim ArrBoolSecc(UBound(ArrDeSecciones)) As Boolean
If IsMissing(m) Then m = UBound(ArrPrimCol)
If IsMissing(n) Then n = UBound(ArrTitulos) + 1
If IsMissing(TipoPrimCol) Then TipoPrimCol = ""
Application.ScreenUpdating = False
With Hoja
NombreHoja = .Name
ActiveWindow.DisplayGridlines = False
ActiveWindow.DisplayHeadings = False
Cells.Interior.ColorIndex = 48 'Gris
SigRen = 1
For k = LBound(ArrDeSecciones) To UBound(ArrDeSecciones)
Select Case ArrDeSecciones(k)
Case "Información"
If IsArray(ArrInfor) Then
For i = LBound(ArrInfor) To UBound(ArrInfor)
.Cells(SigRen + i, 1) = ArrInfor(i)
If IsArray(ArrValInf(i)) Then
For j = 1 To UBound(ArrValInf(i))
.Cells(SigRen + i, j + 1) = ArrValInf(i)(j)
Next j
Else
.Cells(SigRen + i, 2) = ArrValInf(i)
End If
AplicaFormato CStr(ArrInfor(i)), .Cells(SigRen + i, 2)
Next i
ActiveWorkbook.Names.Add NameLocal:="" & NombreHoja & _
"!Información", RefersToR1C1Local:=.Cells(SigRen, 1) _
.Resize(UBound(ArrInfor) + 1, 2)
AplicaFormato "Titulos", _
.Cells(SigRen, 1).Resize(UBound(ArrInfor) + 1, 1)
SigRen = SigRen + UBound(ArrInfor) + 2
End If
If TipoPrimCol <> "" Then
SigCol = 1
Else
SigCol = 0
End If
Case "OtrosDatos"
If Not IsMissing(ArrOtrosDatos) Then
If IsMissing(ArrDatos) Then
For i = 1 To UBound(ArrOtrosDatos)

```

```

.Cells(SigRen + i - 1, SigCol).Value = ArrOtrosDatos(i)
Next i
ActiveWorkbook.Names.Add NameLocal:=.Name & "!OtrosDatos", _
RefersToR1C1Local:=.Cells(SigRen, SigCol + 1) _
.Resize(UBound(ArrOtrosDatos), n)
Else
For i = LBound(ArrOtrosDatos) To UBound(ArrOtrosDatos)
.Cells(SigRen + i, SigCol + UBound(ArrDatos) + 1) = _
ArrOtrosDatos(i)
Next i
ActiveWorkbook.Names.Add NameLocal:="" & .Name & "!OtrosDatos", _
RefersToR1C1Local:=.Cells(SigRen, SigCol + UBound(ArrDatos) + 2) _
.Resize(UBound(ArrOtrosDatos) + 1, n)
End If
.Cells(SigRen, 1).Resize(UBound(ArrOtrosDatos)).Rows.Group
SigRen = SigRen + UBound(ArrOtrosDatos) + 2
TieneOtrosDatos = True
End If
If TieneOtrosDatos And ArrBoolSecc(Application.Match("OtrosDatos", _
ArrDeSecciones, 0)) Then
If IsMissing(n) Then n = Range("Titulos").Columns.Count
ReDim Preserve ArrColModif(Range("OtrosDatos").Column + n - 1)
For i = 1 To Range("OtrosDatos").Rows.Count
ArrColModif(Range("OtrosDatos").Column + i - 1) = "Mayor"
Next i
ArrColModif(Range("OtrosDatos").Column - 1) = "Mayor"
End If
Case "Tiempo", "Funciones", "OtrosTitulos"
If Not PrimColFormato Then
If TipoPrimCol <> "" Then
.Cells(SigRen, 1) = TipoPrimCol
ActiveWorkbook.Names.Add NameLocal:="" & NombreHoja & _
"!TipoPrimCol", RefersToR1C1Local:=.Cells(SigRen, 1)
Set RangoPrimCol = .Cells(SigRen + 1, 1).Resize(m, 1)
Select Case TipoPrimCol
Case "Fechas", "Periodo"
ActiveWorkbook.Names.Add NameLocal:="" & NombreHoja & _
"!Tiempo", RefersToR1C1Local:=RangoPrimCol
TieneTiempo = True
If Left(NombreHoja, 6) = "Inform" Then
ActiveWorkbook.Names.Add NameLocal:=NombreHoja & _
"!Funciones", RefersToR1C1Local:=Range("Información") _
(Application.Match("Información", ArrInfor, 1), 2)
TieneCaract = True
End If
AplicaFormato Tipo:=CStr(TipoPrimCol), Rango:=.Range("Tiempo")
Case "Inform", "Función"
ActiveSheet.Names.Add NameLocal:="" & NombreHoja & _
"!Funciones", RefersToR1C1Local:=RangoPrimCol
TieneCaract = True
Case "Titulos"
If Not IsMissing(ArrTitulos) Then ArrPrimCol = ArrTitulos
ActiveWorkbook.Names.Add NameLocal:="" & NombreHoja & _
"!OtrosTitulos", RefersToR1C1Local:=RangoPrimCol
AplicaFormato Tipo:="Titulos", Rango:=.Range("OtrosTitulos")
TieneOtrosTits = True
End Select
If Not IsMissing(ArrPrimCol) Then RangoPrimCol.Value = _
Application.Transpose(ArrPrimCol)
If Not IsEmpty(ArrColModif) Then
For i = 1 To UBound(ArrColModif)
If ArrColModif(i) = "" Then ArrColModif(i) = "Exacto"
Next i
End If
End If
PrimColFormato = True
End If

```

```

Case "Datos"
  If Not IsMissing(ArrDatos) Then
    For i = 0 To UBound(ArrDatos)
      .Cells(SigRen, SigCol + i + 1) = ArrDatos(i)
    Next i
    ActiveWorkbook.Names.Add NameLocal:="" & NombreHoja & "!Datos", _
      RefersToR1C1Local:=.Cells(SigRen + 1, SigCol + 1) _
      .Resize(m, UBound(ArrDatos) + 1)
    SigCol = SigCol + UBound(ArrDatos) + 1
    TieneDatos = True
  End If
  If TieneDatos Then
    For i = 1 To Range("Datos").Columns.Count
      AplicaFormato Tipo:=Range("Datos").Cells(0, i), _
        Rango:=Range("Datos").Cells(1, i) _
        .Resize(Range("Datos").Rows.Count, 1)
    Next i
  End If
Case "Titulos"
  ActiveWorkbook.Names.Add NameLocal:="" & NombreHoja & "!Titulos", _
    RefersToR1C1Local:=.Cells(SigRen, SigCol + 1).Resize(1, n)
  ActiveWorkbook.Names.Add NameLocal:="" & NombreHoja & "!Series", _
    RefersToR1C1Local:=.Cells(SigRen + 1, SigCol + 1).Resize(m, n)
  Range("Titulos").Value = ArrTitulos
  AplicaFormato Tipo:="Titulos", Rango:=.Range("Titulos")
Case "Series"
  If Not RegresaPreferencia(NombrePref:="Tipo", ValorPref:=Tipo, _
    Hoja:=ActiveSheet) Then Stop
  AplicaFormato Tipo:=Tipo, Rango:=.Range("Series")
  Range("Series")(1, 1).Select
  ActiveWindow.FreezPanels = True
End Select
Next k
End With

Dim Interv As Integer
Dim MinObs As Integer
Dim MercadoNombre As String
Dim Alfa As Double
Dim ChiEst As Double

If IsMissing(HojaPreferencias) Then
  Interv = 8
  MinObs = 5
  MercadoNombre = ArrTitulos(0)
  Alfa = 5
  ChiEst = 38.41455338
Else
  If Not RegresaPreferencia(NombrePref:="Intervalos", _
    ValorPref:=Interv, Hoja:=HojaPreferencias) Then Stop
  If Not RegresaPreferencia(NombrePref:="Observaciones", _
    ValorPref:=MinObs, Hoja:=HojaPreferencias) Then Stop
  If Not RegresaPreferencia(NombrePref:="MercadoNombre", _
    ValorPref:=MercadoNombre, Hoja:=HojaPreferencias) Then Stop
  If Not RegresaPreferencia(NombrePref:="Alfa", _
    ValorPref:=Alfa, Hoja:=HojaPreferencias) Then Stop
  If Not RegresaPreferencia(NombrePref:="ChiEst", _
    ValorPref:=ChiEst, Hoja:=HojaPreferencias) Then Stop
End If
Hoja.Names.Add NameLocal:="" & Hoja.Name & _
  "!Intervalos", RefersToR1C1Local:=Interv
Hoja.Names.Add NameLocal:="" & Hoja.Name & _
  "!Observaciones", RefersToR1C1Local:=MinObs
Hoja.Names.Add NameLocal:="" & Hoja.Name & _
  "!MercadoNombre", RefersToR1C1Local:=MercadoNombre
Hoja.Names.Add NameLocal:="" & Hoja.Name & _

```



```

    "!Alfa", ReferstoR1C1Local:=Alfa
Hoja.Names.Add NameLocal:="" & Hoja.Name & _
    "!ChiEst", ReferstoR1C1Local:=ChiEst

Application.ScreenUpdating = True

End Sub

```

```

Sub AplicaFormato(Tipo As String, Rango As Range)

Select Case Tipo
  Case "Títulos"
    Rango.Interior.ColorIndex = 15 'Gris Claro
    Rango.HorizontalAlignment = xlCenter
    Rango.VerticalAlignment = xlCenter
    Rango.WrapText = True
    Rango.Borders.LineStyle = xlContinuous
  Case "Info", "Método", "Posiciones", "Portafolios"
    Rango.Interior.ColorIndex = 15 'Gris Claro
  Case "Datos"
    Rango.Interior.ColorIndex = 39 'Morado
  Case "Precios"
    Rango.Interior.ColorIndex = xlColorIndexNone
    Rango.NumberFormat = "#,##0.0"
  Case "Rendimientos", "Ren Esp", "Port", "Suma", "Rend. Min", "%Alfa"
    Rango.Interior.ColorIndex = xlColorIndexNone
    Rango.NumberFormat = "0.00 %"
  Case "Fechas", "Desde", "Hasta", "Creado"
    Rango.Interior.ColorIndex = xlColorIndexNone
    Rango.NumberFormat = "mm/dd/yy"
  Case "Muestra", "Tipo", "Análisis", "Análisis", "Población", "Usa Add-In", _
    "Frontera", "Puntos", "Punto", "Parte", "Índice", "Intervalos", "MinObs"
    Rango.Interior.ColorIndex = xlColorIndexNone
  Case "MatCov", "MatCor", "Var", "Riesgo Max"
    Rango.Interior.ColorIndex = xlColorIndexNone
    Rango.NumberFormat = "#,##0.0000"
  Case Else
    If Left(Tipo, 6) = "ChiEst" Then
      Rango.Interior.ColorIndex = xlColorIndexNone
      Rango.NumberFormat = "#,##0.0000"
    End If
End Select

End Sub

```

```

Sub BorrarHoja(Optional Hoja, Optional Pregunta)

Dim ArrHojas() As Worksheet
Dim i As Integer
Dim TipoDeHoja As String

If IsMissing(Hoja) Then
  If Not ObtenerHojas(Titulo:="Borrar Hojas", Comentario:="a borrar", _
    MultipleHoja:=True, ArrHojas:=ArrHojas) Then Exit Sub
Else
  ReDim ArrHojas(0) As Worksheet

```

```

Set ArrHojas(0) = Hoja
End If
If IsMissing(Pregunta) Then Pregunta = True
For i = LBound(ArrHojas) To UBound(ArrHojas)
    If Pregunta Then
        RegresaPreferencia "Tipo", TipoDeHoja, ArrHojas(i)
        Select Case MsgBox(Prompt:="Esta seguro de borrar la hoja de " & _
            TipoDeHoja & String(2, vbCr) & ArrHojas(i).Name, _
            Buttons:=vbYesNoCancel + vbQuestion, Title:="TituloSistema)
            Case vbYes
            Case vbNo
                GoTo OtraHoja
            Case vbCancel
                Exit Sub
        End Select
    End If
    If ActiveWorkbook.Sheets.Count = 1 Then Worksheets.Add
    Application.Calculation = xlManual
    Application.DisplayAlerts = False
    ArrHojas(i).Delete
    Application.DisplayAlerts = True
OtraHoja:
Next i

End Sub

```

Precios

```

Sub ModificarSeries()

Dim ArrModif
Const BorraFechPrecIncom = "Borrar Fechas de Precios Incompletos"
Const BorraFechPrecRepet = "Borrar Fechas de Precios Repetidos"
Const BorraSeriesIniCot = "Borrar Series Criterio Inicio de Cotización"
Const BorraSeriesSinCam = "Borrar Series Sin Cambio"
Const CompletaPrecIncom = "Completa Datos de Precios Incompletos"

ArrModif = Array(BorraFechPrecIncom, BorraFechPrecRepet,
    BorraSeriesIniCot, BorraSeriesSinCam, CompletaPrecIncom)
With frm_Select
    .Caption = "Modificar"
    .BtnOk.Caption = "Continuar"
    .LblSelect.Caption = "Selecciona el tipo de modificación a realizar"
    .LstLists.List = ArrModif
    .LstLists.MultiSelect = fmMultiSelectSingle
    .LstLists.ListIndex = 0
    .UserCancel = False
    .Show
    If .UserCancel Then Exit Sub
    Select Case .LstLists.Value
        Case BorraFechPrecIncom
            BorraFechasDeDatosIncompletos
        Case BorraFechPrecRepet
            BorraFechasDeDatosRepetidos
        Case BorraSeriesIniCot
            BorraSeriesCriterioInicioCotizacion
        Case BorraSeriesSinCam
            BorraSeriesSinCambio
        Case CompletaPrecIncom

```

```

        CompletaPreciosIncompletos
    Case Else
        Stop
    End Select
End With

End Sub

```

```

Sub BorraFechasDeDatosIncompletos()

Dim Titulo As String
Dim ArrHojas() As Worksheet

Dim RangoARevisar As Range
Dim RangoSel As Range
Dim NoSecARevisar As Integer
Dim k As Integer
Dim i As Integer
Dim j As Integer

Titulo = "Borrar Fechas sin Datos completos"
If Not ObtenerHojas(Titulo:=Titulo, _
    ArrTipo:=Array("Precios"), Comentario:="de Precios", _
    MultipleHoja:=False, ArrHojas:=ArrHojas) Then Exit Sub

ArrHojas(0).Activate
Set RangoARevisar = Application.Intersect(Range("Series"), Selection)
On Error GoTo MnjErr
    RangoARevisar.Select
On Error GoTo 0

On Error GoTo MnjErr
    Set RangoSel = Application.InputBox("Desea revisar el siguiente " & _
        "rango de datos", Titulo, RangoARevisar.Address, Type:=8)
On Error GoTo 0

Set RangoSel = Application.Intersect(Range("Series"), RangoSel)
On Error GoTo MnjErr
    RangoSel.Select
On Error GoTo 0

IniciaOtraVez:
NoSecARevisar = RangoSel.Areas.Count
For k = 1 To RangoSel.Areas.Count
    With RangoSel.Areas(k)
        .Select
        For i = .Rows.Count To 1 Step -1
            For j = 1 To .Columns.Count
                .Cells(i, j).Activate
                If .Cells(i, j) = "" Then
                    If frm_Pregunta.BtnSelec = "Si Todo" Then
                        RangoSel.Cells(i, 1).EntireRow.Delete Shift:=xlUp
                        On Error GoTo MnjErr
                        If RangoSel.Areas.Count < NoSecARevisar Then
                            GoTo IniciaOtraVez
                        End If
                        On Error GoTo 0
                    End If
                Else
                    .Cells(i, j).EntireRow.Select
                    .Cells(i, j).Activate
                End If
            Next j
        Next i
    End With
Next k

```

```

        If RangoSel.Areas.Count < NoSecARevisar Then
            GoTo IniciaOtraVez
        End If
        On Error GoTo 0
        Exit For
    Else
        .Cells(i, j).EntireRow.Select
        .Cells(i, j).Activate
        With frm_Pregunta
            .Caption = Titulo
            .UserCancel = False
            .LblPregunta = "Desea borrar la fecha " & _
                RangoSel.Offset(i - 1, 1 - RangoSel.Column)(1, 1)
            .BtnSelec = ""
            .Show
            If .UserCancel Then
                RangoSel.Select
                Exit Sub
            End If
            Select Case .BtnSelec
                Case "Si", "Si Todo"
                    'RangoSel.Cells(i, 1).EntireRow.Delete Shift:=xlUp
                    RangoSel.Cells(i, 1).EntireRow.Delete Shift:=xlUp
                    On Error GoTo MnjErr
                    If RangoSel.Areas.Count < NoSecARevisar Then
                        GoTo IniciaOtraVez
                    End If
                    On Error GoTo 0
                    Exit For
                Case "No"
                Case Else
                    Stop
            End Select
        End With
    End If
End If
Next j
Next i
End With
Next k

On Error Resume Next
RangoARevisar.Select
On Error GoTo 0

MsgBox "Revision terminada", vbInformation, Titulo

Exit Sub

MnjErr:
Select Case Err()
    Case 91 'Object variable or With block variable not set
        MsgBox Err() & " " & Error(), vbCritical
    Case 424 'Object required
        'Ya no hay celdas que revisar
    Case Else
        MsgBox Err() & " " & Error(), vbCritical
End Select

End Sub

```

```

Sub BorraFechasDeDatosRepetidos()

Dim Titulo As String
Dim ArrHojas() As Worksheet

Dim RangoARevisar As Range
Dim RangoSel As Range
Dim NoSecARevisar As Integer
Dim RenglonIgual As Boolean
Dim i As Integer
Dim j As Integer

Titulo = "Borrar Fechas de Datos repetidos"
If Not ObtenerHojas(Titulo:=Titulo, _
    ArrTipo:=Array("Precios"), Comentario:="de Precios", _
    MultipleHoja:=False, ArrHojas:=ArrHojas) Then Exit Sub

ArrHojas(0).Activate
Set RangoARevisar = Application.Intersect(Range("Series"), Selection)
On Error GoTo MnjErr
    RangoARevisar.Select
On Error GoTo 0

IniciaOtraVez:

On Error GoTo MnjErr
    Set RangoSel = Application.InputBox("Desea revisar el siguiente " & _
        "rango de datos", Titulo, RangoARevisar.Address, Type:=8)
On Error GoTo 0

Set RangoSel = Application.Intersect(Range("Series"), RangoSel)
On Error GoTo MnjErr
    RangoSel.Select
On Error GoTo 0

NoSecARevisar = RangoSel.Areas.Count
If NoSecARevisar > 1 Then
    MsgBox "El rango de datos debe ser continuo", vbExclamation, Titulo
    GoTo IniciaOtraVez
Else
    With RangoSel
        .Select
        frm_Pregunta.BtnSelec = ""
        For i = .Rows.Count To 2 Step -1
            RenglonIgual = True
            For j = 1 To .Columns.Count
                If frm_Pregunta.BtnSelec <> "Si Todo" Then .Cells(i, j).Activate
                If .Cells(i, j) <> .Cells(i - 1, j) Then
                    RenglonIgual = False
                    Exit For
                End If
            Next j
            If RenglonIgual Then
                If frm_Pregunta.BtnSelec = "Si Todo" Then
                    .Cells(i, 1).EntireRow.Delete Shift:=xlUp
                Else
                    .Cells(i, j).EntireRow.Select
                    .Cells(i, j).Activate
                    With frm_Pregunta
                        .Caption = Titulo
                        .UserCancel = False
                        .LblPregunta = "Desea borrar la fecha " & _
                            RangoSel.Offset(i - 1, 1 - RangoSel.Column)(1, 1)
                        .BtnSelec = ""
                        .Show
                    End With
                    If .UserCancel Then

```

```

        RangoARevisar.Select
        Exit Sub
    End If
    Select Case .BtnSelec
        Case "Si", "Si Todo"
            RangoSel.Cells(i, 1).EntireRow.Delete Shift:=xlUp
        Case "No"
        Case Else
            Stop
    End Select
End With
End If
End If
Next i
End With
End If

On Error Resume Next
RangoSel.Select
On Error GoTo 0

MsgBox "Revision terminada", vbInformation, Titulo

Exit Sub

MnjErr:
    Select Case Err()
        Case 91 'Object variable or With block variable not set
            MsgBox Err() & " " & Error(), vbCritical
        Case 424 'Object required
            'Ya no hay celdas que revisar
        Case Else
            MsgBox Err() & " " & Error(), vbCritical
    End Select

End Sub

```

```

Sub BorraSeriesCriterioInicioCotizacion()

Dim Titulo As String
Dim ArrHojas() As Worksheet

Dim RangoARevisar As Range
Dim RangoSel As Range
Dim NoSecARevisar As Integer
Dim EmpiezaACotizar As Boolean
Dim i As Integer
Dim j As Integer

Titulo = "Borrar Series por Inicio de Cotización"
If Not ObtenerHojas(Titulo:=Titulo, _
    ArrTipo:=Array("Precios"), Comentario:="de Precios", _
    MultipleHoja:=False, ArrHojas:=ArrHojas) Then Exit Sub

ArrHojas(0).Activate
Set RangoARevisar = Application.Intersect(Range("Series"), Selection)
Set RangoARevisar = Range("Series").Resize(RangoARevisar.Row + _
    RangoARevisar.Rows.Count - Range("Series").Row, Range("Series").Columns.Count)
On Error GoTo MnjErr
RangoARevisar.Select
On Error GoTo 0

```

```

IniciaOtraVez:
On Error GoTo MnjErr
Set RangoSel = Application.InputBox("Desea revisar el siguiente " & _
    "rango de datos", Titulo, RangoARevisar.Address, Type:=8)
On Error GoTo 0

Set RangoSel = Range("Series").Resize(RangoSel.Row + RangoSel.Rows.Count - _
    Range("Series").Row, Range("Series").Columns.Count)
On Error GoTo MnjErr
RangoSel.Select
On Error GoTo 0

NoSecARevisar = RangoSel.Areas.Count
If NoSecARevisar > 1 Then
    MsgBox "El rango de datos debe ser continuo", vbExclamation, Titulo
    GoTo IniciaOtraVez
Else
    With RangoSel
        .Select
        frm_Pregunta.BtnSelec = ""
        For j = .Columns.Count To 1 Step -1
            EmpiezaACotizar = False
            For i = 1 To .Rows.Count
                If frm_Pregunta.BtnSelec <> "Si Todo" Then .Cells(i, j).Activate
                If Range("Series").Cells(i, j) <> "" Then
                    EmpiezaACotizar = True
                    Exit For
                End If
            Next i
            If Not EmpiezaACotizar Then
                If frm_Pregunta.BtnSelec = "Si Todo" Then
                    .Cells(0, j).Resize(Range("Series").Rows.Count + 1, 1).Delete _
                        Shift:=xlShiftToLeft
                Else
                    .Cells(1, j).Resize(.Rows.Count, 1).Select
                    With frm_Pregunta
                        .Caption = Titulo
                        .UserCancel = False
                        .LblPregunta = "Desea borrar la serie " & _
                            Range("Titulos")(j)
                        .BtnSelec = ""
                        .Show
                    End With
                    If .UserCancel Then
                        RangoARevisar.Select
                        Exit Sub
                    End If
                    Select Case .BtnSelec
                        Case "Si", "Si Todo"
                            RangoSel.Cells(0, j).Resize(Range("Series").Rows.Count _
                                + 1, 1).Delete Shift:=xlShiftToLeft
                        Case "No"
                        Case Else
                            Stop
                    End Select
                End With
            End If
        Next j
    End With
End If

On Error Resume Next
RangoSel.Select
On Error GoTo 0

```

```

MsgBox "Revision terminada", vbInformation, Titulo

Exit Sub

MnjErr:
Select Case Err()
Case 91 'Object variable or With block variable not set
MsgBox Err() & " " & Error(), vbCritical
Case 424 'Object required
'Ya no hay celdas que revisar
Case Else
MsgBox Err() & " " & Error(), vbCritical
End Select

End Sub

```

```

Sub BorraSeriesSinCambio()

Dim Titulo As String
Dim ArrHojas() As Worksheet

Dim RangoARevisar As Range
Dim RangoSel As Range
Dim NoSecARevisar As Integer
Dim i As Integer
Dim j As Integer
Dim ContestaPorcDif As String
Dim DividirEntre As Integer
Dim PorcDif As Double
Dim CasosDif As Long

Titulo = "Borrar Series Sin Cambio"
If Not ObtenerHojas(Titulo:=Titulo, _
ArrTipo:=Array("Precios"), Comentario:="de Precios", _
MultipleHojas:=False, ArrHojas:=ArrHojas) Then Exit Sub

ArrHojas(0).Activate
Set RangoARevisar = Application.Intersect(Range("Series"), Selection)
Set RangoARevisar = Range("Series")(RangoARevisar.Row - _
Range("Series").Row + 1, 1).Resize(RangoARevisar.Rows.Count, _
Range("Series").Columns.Count)
On Error GoTo MnjErr
RangoARevisar.Select
On Error GoTo 0

IniciaOtraVez:

On Error GoTo MnjErr
Set RangoSel = Application.InputBox("Desea revisar el siguiente " & _
"rango de datos", Titulo, RangoARevisar.Address, Type:=8)
On Error GoTo 0

Set RangoSel = Range("Series")(RangoSel.Row - Range("Series").Row + 1, 1) _
.Resize(RangoSel.Rows.Count, Range("Series").Columns.Count)
On Error GoTo MnjErr
RangoSel.Select
On Error GoTo 0

PreguntaPorcent:
ContestaPorcDif = Trim(InputBox("Que porcentaje de datos requiere " & _

```



```

"como mínimo de cambio", Titulo, Default:="20%"))
If ContestaPorcDif = "" Then Exit Sub
If Right(ContestaPorcDif, 1) = "%" Then
    DividirEntre = 100
    ContestaPorcDif = Trim(Left(ContestaPorcDif, Len(ContestaPorcDif) - 1))
Else
    DividirEntre = 1
End If
If IsNumeric(ContestaPorcDif) Then
    PorcDif = ContestaPorcDif / DividirEntre
    If PorcDif <= 0 Or PorcDif > 1 Then
        MsgBox "El porcentaje de diferencia tiene que ser mayor a 0% y " & _
            "como maximo 100% (1)", vbExclamation, Titulo
        GoTo PreguntaPorcent
    End If
Else
    MsgBox "El porcentaje tiene que ser un valor numerico (Ej. 0.2 o 20%)", _
        vbExclamation, Titulo
    GoTo PreguntaPorcent
End If

NoSecAREvisar = RangoSel.Areas.Count
If NoSecAREvisar > 1 Then
    MsgBox "El rango de datos debe ser continuo", vbExclamation, Titulo
    GoTo IniciaOtraVez
Else
    With RangoSel
        .Select
        frm_Pregunta.BtnSelec = ""
        For j = .Columns.Count To 1 Step -1
            CasosDif = 0
            For i = 1 To .Rows.Count
                If .Cells(i, j) <> .Cells(i - 1, j) Then
                    CasosDif = CasosDif + 1
                    If (CasosDif / .Rows.Count) > PorcDif Then Exit For
                End If
            Next i
            If (CasosDif / .Rows.Count) < PorcDif Then
                If frm_Pregunta.BtnSelec = "Si Todo" Then
                    Range("Series").Cells(0, j).Resize(
                        Range("Series").Rows.Count + 1, 1).Delete Shift:=xlShiftToLeft
                Else
                    .Cells(1, j).Resize(.Rows.Count, 1).Select
                    With frm_Pregunta
                        .Caption = Titulo
                        .UserCancel = False
                        .LblPregunta = "Desea borrar la serie " & Range("Titulos")(j) &
                            Chr(13) & "El porcentaje que cambia es " &
                            Format(CasosDif / RangoSel.Rows.Count, "0.00%")
                        .BtnSelec = ""
                    End With
                    .Show
                    If .UserCancel Then
                        RangoAREvisar.Select
                        Exit Sub
                    End If
                    Select Case .BtnSelec
                        Case "Si", "Si Todo"
                            Range("Series").Cells(0, j).Resize(Range("Series")
                                .Rows.Count + 1, 1).Delete Shift:=xlShiftToLeft
                        Case "No"
                        Case Else
                            Stop
                    End Select
                End With
            End If
        End If
    Next j

```

```

    End With
    End If

    On Error Resume Next
    RangoSel.Select
    On Error GoTo 0

    MsgBox "Revision terminada", vbInformation, Titulo

    Exit Sub

MnjErr:
    Select Case Err()
        Case 91 'Object variable or With block variable not set
            MsgBox Err() & " " & Error(), vbCritical
        Case 424 'Object required
            'Ya no hay celdas que revisar
        Case Else
            MsgBox Err() & " " & Error(), vbCritical
    End Select

End Sub

```

```

Sub CompletaPreciosIncompletos()

    Dim Titulo As String
    Dim ArrHojas() As Worksheet

    Dim RangoARevisar As Range
    Dim RangoSel As Range
    Dim NoSecARevisar As Integer
    Dim k As Integer
    Dim i As Integer
    Dim j As Integer

    Titulo = "Completa Precios Incompletos"
    If Not ObtenerHojas(Titulo:=Titulo, _
        ArrTipo:=Array("Precios"), Comentario:="de Precios", _
        MultipleHoja:=False, ArrHojas:=ArrHojas) Then Exit Sub

    ArrHojas(0).Activate
    Set RangoARevisar = Application.Intersect(Range("Series"), Selection)
    On Error GoTo MnjErr
    RangoARevisar.Select
    On Error GoTo 0

    On Error GoTo MnjErr
    Set RangoSel = Application.InputBox("Desea revisar el siguiente " & _
        "rango de datos", Titulo, RangoARevisar.Address, Type:=8)
    On Error GoTo 0

    Set RangoSel = Application.Intersect(Range("Series"), RangoSel)
    On Error GoTo MnjErr
    RangoSel.Select
    On Error GoTo 0

    IniciaOtraVez:
    NoSecARevisar = RangoSel.Areas.Count
    For k = 1 To NoSecARevisar
        With RangoSel.Areas(k)

```

```

.Select
For i = 1 To .Rows.Count
For j = 1 To .Columns.Count
If .Cells(i, j) = "" Then
If frm_Pregunta.BtnSelec = "Si Todo" Then
RangoSel.Cells(i, 1) = RangoSel.Cells(i - 1, 1)
Else
With frm_Pregunta
.Caption = Titulo
.UserCancel = False
.LblPregunta = "Desea completar el precio de " & _
RangoSel.Offset(i - 1, 1 - RangoSel.Column)(1, 1) & _
" con la fecha anterior " & _
RangoSel.Offset(i - 2, 1 - RangoSel.Column)(1, 1)
.BtnSelec = ""
.Show
If .UserCancel Then
RangoSel.Select
Exit Sub
End If
Select Case .BtnSelec
Case "Si", "Si Todo"
RangoSel.Cells(i, 1) = RangoSel.Cells(i - 1, 1)
Case "No"
Case Else
Stop
End Select
End With
End If
End If
Next j
Next i
End With
Next k

On Error Resume Next
RangoSel.Select
On Error GoTo 0

MsgBox "Revision terminada", vbInformation, Titulo

Exit Sub

MnjErr:
Select Case Err()
Case 91 'Object variable or With block variable not set
MsgBox Err() & " " & Error(), vbCritical
Case 424 'Object required
'Ya no hay celdas que revisar
Case Else
MsgBox Err() & " " & Error(), vbCritical
End Select

End Sub

```

Rendimientos

```
Sub ObtieneRends()
```

```

Dim ArrHojas() As Worksheet
Dim NombreDatos As String
Dim Libro As Workbook
Dim HojaNueva As Worksheet
Dim ArrDeInfo() As String
Dim ArrDeValInfo() As String
Dim ArrTitulos() As String
Dim i As Integer
Dim TipoPrimCol As String
Dim Mensaje As String

EscogeHojaDePrecios:
If Not ObtenerHojas(Titulo:="Rendimiento de Precios", _
  ArrTipo:=Array("Precios", "Análisis"), ArrSubTipo:=Array("Precios"), _
  Comentario:="de Precios", ArrHojas:=ArrHojas) Then Exit Sub
If ArrHojas(0).Range("Tiempo").Count < 2 Then
  If Not RegresaComentEnIdioma("No puede sacar rendimientos de " & _
    "menos de dos datos", Mensaje) Then Exit Sub
  MsgBox Prompt:=Mensaje, Buttons:=vbExclamation, Title:=TituloSistema
  GoTo EscogeHojaDePrecios
End If
If Not RegresaPreferencia(NombrePref:="Nombre", ValorPref:=NombreDatos, _
  Hoja:=ArrHojas(0)) Then Stop
Set Libro = ArrHojas(0).Parent

If SigHoja(Libro:=Libro, TipoHoja:="Rendimientos", NombreDatos:=NombreDatos, _
  NuevaHoja:=HojaNueva, SaltaBorrar:=False) Then
  ReDim ArrDeInfo(1) As String
  ReDim ArrDeValInfo(1) As String
  ArrDeInfo(0) = "Datos"
  ArrDeValInfo(0) = ArrHojas(0).Name
  ArrDeInfo(1) = "Tipo"
  ArrDeValInfo(1) = "Rendimientos"
  ReDim ArrTitulos(ArrHojas(0).Range("Titulos").Count - 1) As String
  For i = 1 To ArrHojas(0).Range("Titulos").Count
    ArrTitulos(i - 1) = ArrHojas(0).Range("Titulos")(i)
  Next i
  TipoPrimCol = ArrHojas(0).Range("TipoPrimCol")
  ReDim ArrPrimCol(ArrHojas(0).Range("Tiempo").Count - 1) As String
  For i = 2 To ArrHojas(0).Range("Tiempo").Count
    ArrPrimCol(i - 2) = ArrHojas(0).Range("Tiempo")(i)
  Next i
  FormatoHojas Hoja:=HojaNueva, ArrInfor:=ArrDeInfo, _
    ArrValInf:=ArrDeValInfo, ArrTitulos:=ArrTitulos, _
    ArrPrimCol:=ArrPrimCol, TipoPrimCol:=TipoPrimCol, SigRen:=PrimRen
  Application.ScreenUpdating = False
  Application.Calculation = xlCalculationManual
  HojaNueva.Range("Series").Cells(1, 1).Formula = "=" & _
    ArrHojas(0).Range("Series").Cells(2, 1).Address(RowAbsolute:=False, _
    ColumnAbsolute:=False, external:=True) & "/" & _
    ArrHojas(0).Range("Series").Cells(1, 1).Address(RowAbsolute:=False, _
    ColumnAbsolute:=False, external:=True) & "-1"
  HojaNueva.Range("Series").Cells(1, 1).Copy _
    Destination:=HojaNueva.Range("Series")
  Application.Calculate
  'En caso de no tener formulas con errores seguir
  On Error Resume Next
  HojaNueva.Range("Series").SpecialCells(xlCellTypeFormulas, xlErrors).Clear
  On Error GoTo 0
  HojaNueva.Range("Series").Copy
  HojaNueva.Range("Series").PasteSpecial xlPasteValues
  Application.CutCopyMode = False
  Application.ScreenUpdating = False
  Application.Calculation = xlCalculationAutomatic
End If

```

```
End Sub
```

Análisis

```
Function CreaAnalisis(Optional ArrDeFunc, _  
    Optional Tipo, Optional Multiple, Optional FechMuestra, _  
    Optional Muestra, Optional Periodo, Optional RangoDeDatos, _  
    Optional RangoDeTitulos, Optional RangoPrimCol, Optional TipoPrimCol)  
  
    Dim ArrTipo  
    Dim ArrHojas() As Worksheet  
    Dim NombreHoja As String  
    Dim FechInicio As Integer  
    Dim FechTermino As Integer  
    Dim HojaAnalisis As Worksheet  
    Dim NombreDatos As String  
    Dim TipoAnalisis As String  
    Dim RangoDeFechas As Range  
    Dim SaltaBorrar As Boolean  
    Dim n As Integer  
    Dim ArrInform  
    Dim ArrValInform  
    Dim ArrTitulos  
    'Dim SigRen As Integer  
    Dim ConFormulas As Boolean  
    'El primer elemento del arreglo es el Default de escoge sencillo  
    Dim ArrFuncDefault  
    Dim CambiaCeros As Boolean  
    Dim Serie  
    Dim Interv As Integer  
    Dim MinObs As Integer  
    Dim MercadoNombre As String  
    Dim Mercado As Integer 'Para obtener las betas  
    Dim Alfa As Double  
    Dim ChiEst As Double  
    Dim Comentario As String  
    Dim i As Integer  
    Dim j As Integer  
    Dim k As Integer  
    'Dim l As Integer  
    Dim Funcion As String  
    Dim ResChiCuad As Single  
    Dim ConMercado As Boolean  
    Dim ConNorm As Boolean  
    Dim ConPrueba As Boolean  
    Dim Cuartil As Integer  
    Dim RangoDeCambio As Range  
    Dim Menu As String  
    Dim TipoResultado As String  
  
    If IsMissing(RangoDeDatos) Then  
        RangoDeTitulos = ""  
        ArrTipo = Array("Precios", "Rendimientos", "Informacion", _  
            "Portafolio", "Mat Covarianzas")  
        If Not ObtenerHojas(Titulo:="Análisis", ArrTipo:=ArrTipo, _  
            Comentario:="para hacer el Análisis", MultipleHoja:=False, _  
            ArrHojas:=ArrHojas) Then Exit Function  
        TipoPrimCol = ArrHojas(0).Range("TipoPrimCol")
```

```

Set RangoPrimCol = ArrHojas(0).Range("Tiempo")
Set RangoDeDatos = ArrHojas(0).Range("Series")
If Not RegresaPreferencia(NombrePref:="Nombre", ValorPref:=NombreDatos, _
Hoja:=ArrHojas(0)) Then Exit Function
If Not RegresaPreferencia(NombrePref:="Tipo", ValorPref:=Tipo, _
Hoja:=ArrHojas(0)) Then Exit Function
Else
ReDim ArrHojas(0)
Set ArrHojas(0) = RangoDeDatos.Parent
If Not RegresaPreferencia(NombrePref:="Nombre", ValorPref:=NombreDatos, _
Hoja:=ArrHojas(0)) Then Exit Function
End If
NombreHoja = RangoDeDatos.Parent.Name
Select Case TipoPrimCol
Case "Fechas", "Periodo"
Set RangoDeFechas = RangoPrimCol
Case Else
End Select

If IsMissing(ArrDeFunc) Then
With frm_Select
.Caption = "Análisis [" & NombreHoja & "]"
.LblSelect.Caption = "Selecciona las funciones a aplicar"
.BtnOk.Caption = "Aplicar"
Select Case Tipo
Case "Precios", "Rendimientos"
ArrDeFunc = Array("Maximo", "Minimo", "Mediana", "Total", _
"Promedio", "Prom. Geom.", "Prom. Armon.", "Varianza", _
"Desv. Estan.", "Sesgo", "Curtosis", "Normalidad", "Prueba", _
"Beta", "Alfa", "Var. Residual", _
"Cuartil 50", "Cuartil 25", "Cuartil 75")
ArrFuncDefault = Array("Promedio", "Maximo", "Minimo", _
"Total", "Promedio")
Case "Información", "Portafolio"
ArrDeFunc = Array("Maximo", "Minimo", "Mediana", "Total", _
"Promedio", "Prom. Geom.", "Prom. Armon.", "Varianza", _
"Desv. Estan.", "Sesgo", "Curtosis", _
"Cuartil 50", "Cuartil 25", "Cuartil 75")
ArrFuncDefault = Array("Promedio", "Maximo", "Minimo", "Total")
Case "Mat Covarianzas"
ArrDeFunc = Array("Varianza", "Desv. Estan.")
ArrFuncDefault = Array("Desv. Estan.", "Varianza")
End Select
.LstLists.MultiSelect = fmMultiSelectExtended
.LstLists.List = ArrDeFunc
For i = 1 To UBound(ArrFuncDefault)
For j = 0 To .LstLists.ListCount - 1
If ArrFuncDefault(i) = .LstLists.List(j) Then
.LstLists.Selected(j) = True
Exit For
End If
Next j
Next i
.BtnOk.Enabled = True
.UserCancel = False
RepiteAnálisis:
.Show
If .UserCancel Then Exit Function
Set RangoDeFechas = ArrHojas(0).Range("Tiempo")

j = 0
For i = 0 To .LstLists.ListCount - 1
If .LstLists.Selected(i) Then
ReDim Preserve ArrDeFunc(j)
ArrDeFunc(j) = .LstLists.List(i)
j = j + 1

```

```

    End If
Next i
If j = 0 Then Exit Function

If UBound(ArrDeFunc) > 0 Then 'Multiple
    If Not RegresaFechas("Fecha de Análisis", ArrHojas(0), _
        RangoDeFechas, 3, Muestra, FechMuestra, FechInicio, _
        FechTermino, Periodo, False) Then GoTo RepiteAnálisis
    Else 'Sencillo
        If Not RegresaFechas("Periodo de Análisis", ArrHojas(0), _
            RangoDeFechas, 3, Muestra, FechMuestra, FechInicio, _
            FechTermino, Periodo, True) Then GoTo RepiteAnálisis
    End If

    Set RangoDeTitulos = ArrHojas(0).Range("Titulos")
End With
Else
    SaltaBorrar = True
End If
ArrTitulos = RegresaVectorDeRango(RangoDeTitulos)
Set Serie = RangoDeDatos.Cells(FechMuestra, 1) _
    .Resize(Muestra, RangoDeDatos.Columns.Count) _
If UBound(ArrDeFunc) > 0 Then
    Multiple = True
    TipoAnálisis = "Múltiple"
    TipoResultado = "Análisis"
Else
    Multiple = False
    TipoAnálisis = ArrDeFunc(0)
    Select Case TipoAnálisis
        Case "Maximo", "Minimo", "Mediana", "Promedio", "Prom. Geom."
            TipoResultado = Tipo
        Case "Total", "Prom. Armon.", "Varianza", "Desv. Estan.", _
            "Sego", "Curtosis", "Normalidad", "Prueba", "Beta", "Alfa", _
            "Var. Residual", "Cuartil 50", "Cuartil 25", "Cuartil 75"
            TipoResultado = "Análisis"
        Case Else
            MsgBox "Funcion no definida:" & Chr(13) & TipoAnálisis, vbExclamation,
                Titulo
    End Select
    Exit Function
End Select
End If
If Multiple Then
    ArrInform = Array("Datos", "Tipo", "Análisis", "Desde", "Hasta", "Muestra")
    ArrValInform = Array(NombreHoja, Tipo, "Múltiple", _
        RangoDeFechas(FechMuestra), RangoDeFechas(FechMuestra + Muestra - 1), _
        Muestra)
Else
    ArrInform = Array("Datos", "Tipo", "Análisis", "Muestra")
    ArrValInform = Array(NombreHoja, Tipo, ArrDeFunc(0), Muestra)
End If
For i = LBound(ArrDeFunc) To UBound(ArrDeFunc)
    Select Case ArrDeFunc(i)
        Case "Normalidad"
            ConNorm = True
        Case "Prueba"
            ConNorm = True
            ConPrueba = True
        Case "Beta", "Alfa", "Var. Residual"
            ConMercado = True
    End Select
Next i
If ConMercado Then
    If Not RegresaPreferencia(NombrePref:="MercadoNombre", _
        ValorPref:=MercadoNombre, Hoja:=ArrHojas(0)) Then Stop
    AgregaDatos Array("Indice"), ArrInform

```

```

AgregaDatos Array(MercadoNombre), ArrValInform
Mercado = Application.Match(MercadoNombre, ArrHojas(0).Range("Titulos"), 0)
End If
If Not RegresaPreferencia(NombrePref:="Intervalos", _
ValorPref:=Interv, Hoja:=ArrHojas(0)) Then Exit Function
If ConNorm Then
AgregaDatos Array("Intervalos", "MinObs"), ArrInform
If Not RegresaPreferencia(NombrePref:="Observaciones", _
ValorPref:=MinObs, Hoja:=ArrHojas(0)) Then Exit Function
AgregaDatos Array(Interv, MinObs), ArrValInform
End If
If ConPrueba Then
AgregaDatos Array("%Alfa", "ChiEst (" & Interv - Estim & ")"), ArrInform
If Not RegresaPreferencia(NombrePref:="Alfa", _
ValorPref:=Alfa, Hoja:=ArrHojas(0)) Then Stop
If Not RegresaPreferencia(NombrePref:="ChiEst", _
ValorPref:=ChiEst, Hoja:=ArrHojas(0)) Then Stop
AgregaDatos Array(Alfa / 1000, ChiEst / 10), ArrValInform
End If
If Not SigHoja(Libro:=ArrHojas(0).Parent, TipoHoja:=TipoResultado, _
NombreDatos:=NombreDatos, NuevaHoja:=HojaAnalisis, _
SaltaBorrar:=SaltaBorrar, HojaPreferencias:=ArrHojas(0), _
SubTipoHoja:=TipoAnalisis) Then
Exit Function
End If
If Multiple Then
TipoPrimCol = "Función"
Periodo = 1
Else
Stop
If TipoPrimCol = "Titulos" Then TipoPrimCol = "Fechas"
'Periodo = ?
End If

n = UBound(ArrTitulos) + 1

With HojaAnalisis
FormatoHojas Hoja:=HojaAnalisis, ArrInfor:=ArrInform, _
ArrValInf:=ArrValInform, ArrTitulos:=ArrTitulos, _
TipoPrimCol:=TipoPrimCol, m:=(UBound(ArrDeFunc) + 1) * Periodo, _
n:=n, HojaPreferencias:=ArrHojas(0) ', SigRen:=SigRen
For i = LBound(ArrDeFunc) + 1 To UBound(ArrDeFunc) + 1
If Multiple Then
.Range("Funciones")(i) = ArrDeFunc(i - 1)
Else
Stop
For j = 1 To Range("Tiempo").Count
.Range("Tiempo")(j).Value = RangoDeFechas.Cells(FechMuestra + _
k + Muestra + j - 2).Value
Next j
End If
Select Case ArrDeFunc(i - 1)
Case "Maximo", "Minimo", "Mediana", "Promedio", "Prom. Geom.", _
"Prom. Armon.", "Cuartil 50", "Cuartil 25", "Cuartil 75", "Alfa"
Select Case Tipo
Case "Precios", "Rendimientos"
.Range("Series")(i, 1).Resize(Periodo, n).NumberFormat = _
RangoDeDatos.Cells(1, 1).NumberFormat
End Select
Case "Varianza"
.Range("Series")(i, 1).Resize(Periodo, n).NumberFormat = "0.0000"
Case "Desv. Estan.", "Sesgo", "Curtosis", "Beta", _
"Var. Residual", "Normalidad"
.Range("Series")(i, 1).Resize(Periodo, n).NumberFormat = "0.000"
Case "Prueba", "Total"
Case Else

```



```

    MsgBox "No existe esa funcion", vbCritical
End Select
Select Case ArrDeFunc(i - 1)
Case "Maximo", "Minimo", "Mediana", "Total", "Promedio", _
    "Prom. Geom.", "Prom. Armon.", "Varianza", "Desv. Estan.", _
    "Sesgo", "Curtosis"
Select Case ArrDeFunc(i - 1)
    Case "Maximo"
        Funcion = "Max"
    Case "Minimo"
        Funcion = "Min"
    Case "Mediana"
        Funcion = "Median"
    Case "Total"
        Funcion = "Count"
    Case "Promedio"
        Funcion = "Average"
    Case "Prom. Geom."
        Funcion = "GeoMean"
    Case "Prom. Armon."
        Funcion = "HarMean"
    Case "Varianza"
        Funcion = "Var"
    Case "Desv. Estan."
        Funcion = "StDev"
    Case "Sesgo"
        Funcion = "Skew"
    Case "Curtosis"
        Funcion = "Kurt"
    Case Else
        Stop
End Select
If Tipo = "Mat Covarianzas" Then
    Select Case ArrDeFunc(i - 1)
    Case "Varianza"
        For j = 1 To n
            .Range("Series")(i, j) = "=" & _
                Serie(j, j).Address(External:=True)
        Next j
    Case "Desv. Estan."
        For j = 1 To n
            .Range("Series")(i, j) = "=SQRT" & _
                Serie(j, j).Address(External:=True)
        Next j
    Case Else
        Stop
    End Select
Else
    .Range("Series")(i, 1).Resize(Periodo, n) = "=" & Funcion & _
        "(" & Serie(1, 1).Resize(Muestra, 1)
        .Address(False, False, External:=True) & ")"
End If

Case "Normalidad"
    If Periodo > 1 Then
        Stop
    End If
    For j = 1 To n
        .Range("Series")(i, j) = ChiCud(Serie:= _
            Serie(1, j).Resize(Muestra, 1), Hoja:=HojaAnálisis, _
            Interv:=Interv, MinObs:=MinObs, Comentario:=Comentario)
    Next j
    If .Range("Series")(i, j) = 1 Then
        .Range("Series")(i, j).NoteText Text:=Comentario
    End If
End Case "Prueba"

```

```

If Not RegresaPreferencia(NombrePref:="ChiEst", ValorPref:=ChiEst, _
Hoja:=HojaAnalisis) Then
    MsgBox "Falta el valor predeterminado de ChiEst para " & _
        "hacer la prueba", vbExclamation, TituloSistema
    Exit Function
End If
For j = 1 To n
    If .Range("Funciones")(i - 1) = "Normalidad" Then
        If IsNumeric(.Range("Series")(i - 1, j)) Then
            If .Range("Series")(i - 1, j) > CInt(ChiEst) / 10 Then
                .Range("Series")(i, j) = False
            Else
                .Range("Series")(i, j) = True
            End If
        Else
            .Range("Series")(i, j) = "N.D."
        End If
    Else
        Stop
        ResChiCuad = ChiCuad(Serie:=Serie(1, j).Resize(Muestra, 1), _
            Hoja:=NombreHoja, Interv:=Interv, MinObs:=MinObs)
        .Range("Series")(i, j).NoteText Text:="ChiCuad: " & ResChiCuad
        If ResChiCuad > CInt(ChiEst) / 10 Then
            .Range("Series")(i, j) = False
        Else
            .Range("Series")(i, j) = True
        End If
    End If
Next j
Case "Beta", "Alfa"
    Select Case ArrDeFunc(i - 1)
        Case "Beta"
            Funcion = "Slope"
        Case "Alfa"
            Funcion = "Intercept"
        Case Else
            Stop
    End Select
    .Range("Series")(i, 1).Resize(Periodo, n) = "=" & Funcion & "(" & _
        Serie(1, 1).Resize(Muestra, 1).Address(False, False, External:=True) _
        & "," & Serie(1, Mercado).Resize(Muestra, 1) _
        .Address(False, External:=True) & ")"
Case "Var. Residual"
    .Range("Series")(i, 1).Resize(Periodo, n) = "=Intercept(" & _
        Serie(1, 1).Resize(Muestra, 1).Address(False, False, External:=True) _
        & "," & Serie(1, Mercado).Resize(Muestra, 1) _
        .Address(False, External:=True) & ")^2"
Case "Cuartil 50", "Cuartil 25", "Cuartil 75"
    Select Case ArrDeFunc(i - 1)
        Case "Cuartil 50"
            Cuartil = 2
        Case "Cuartil 25"
            Cuartil = 1
        Case "Cuartil 75"
            Cuartil = 3
        Case Else
            Stop
    End Select
    .Range("Series")(i, 1).Resize(Periodo, n) = "=Quartile(" & Serie(1, 1) _
        .Resize(Muestra, 1).Address(False, False, External:=True) & "," & _
        & Cuartil & ")"
End Select
Next i
On Error GoTo SinErrores
Set RangoDeCambio = .Range("Series") _
.SpecialCells(Type:=xlFormulas, Value:=xlErrors)

```

```

If CambiaCeros Then
    RangoDeCambio.Value = 0
Else
    RangoDeCambio.Value = NoDisp
End If
On Error GoTo 0
SinErrores:
If Not RegresaComentEnIdioma("&Análisis", Menu) Then Exit Function
ConFormulas = MenuBars(xlWorksheet).Menus(NombreMenu) _
    .MenuItems("F&ormulas").MenuItems(Menu).Checked
If Not ConFormulas Then
    Range("Series").Copy
    Range("Series").PasteSpecial Paste:=xlValues
    Application.CutCopyMode = False
End If
Set CreaAnálisis = .Range("Series")
End With

Exit Function

MnjErr:
Select Case Error()
    Case "Type mismatch" '13
        HojaAnálisis.Range("Series")(i, j) = "#NUM!"
        Resume Next
    Case Else
        MsgBox ("Error: " & Error() & " " & "Err: " & Err)
End Select

End Function

```

```

Function ChiCud(Serie As Range, Hoja, Interv, _
    MinObs, Optional EscrPrueba, Optional NomSerie, Optional Comentario)

Dim MaxInter As Integer
Dim n As Integer
Dim Seccion
Dim Prom As Double
Dim DesvStnd As Double
Dim ArrZ
Dim ArrDim
Dim ArrFrecEsp
Dim ArrFrecObs

n = Serie.Rows.Count
If Interv = 0 Then If Not RegresaPreferencia(NombrePref:="Intervalos", _
    ValorPref:=Interv, Hoja:=Hoja) Then ChiCud = "#NUM!": Exit Function
If MinObs = 0 Then If Not RegresaPreferencia(NombrePref:="Observaciones", _
    ValorPref:=MinObs, Hoja:=Hoja) Then ChiCud = "#NUM!": Exit Function
If n >= MinObs * Interv Then
    If CInt(n / MinObs) <= Interv Then Interv = CInt(n / MinObs)
    For Each Seccion In Serie.Areas
        ArrZ = GenZ(n, Interv)
        Prom = Application.Average(Serie)
        On Error GoTo MnjErr
        DesvStnd = Application.StDev(Serie)
        On Error GoTo 0
        ArrDim = GenDim(ArrZ, Prom, DesvStnd)
        ArrFrecEsp = GenFrecEsp(ArrZ, n)
        ArrFrecObs = CreaHistograma(Serie, ArrDim)
        ChiCud = ChiC(ArrFrecEsp, ArrFrecObs)
    Next Seccion
End If

```

```

Next Seccion
Else
Comentario = "El mínimo de observaciones necesarias de muestra " & _
"para hacer la prueba es de " & MinObs * Interv
End If
Exit Function
MnjErr:
Select Case Error()
Case "Type mismatch" '13
ChiCuad = "#NUM!"
Exit Function
Case Else
MsgBox ("Error: " & Error() & " " & "Err: " & Err)
Resume Next
End Select
End Function

```

```

Function GenZ(n, Interv)

Dim Tabla
Dim Porc, Z As Double
Dim i As Integer

ReDim Tabla(1 To Interv)
Porc = MinimoObs / n
Z = Application.NormSInv(Porc)
For i = 1 To Interv - 1
Tabla(i) = Z + (i - 1) * (-Z * 2 / (Interv - 2))
Next i
GenZ = Tabla

End Function

```

```

Function GenDim(ArrZ, Promedio, DesvStd)

Dim Tabla
Dim i As Integer

ReDim Tabla(1 To UBound(ArrZ))
For i = 1 To UBound(ArrZ) - 1
Tabla(i) = Promedio + ArrZ(i) * DesvStd
Next i
GenDim = Tabla

End Function

```

```

Function GenFrecEsp(ArrZ, n)

Dim Tabla
Dim i As Integer

```

```

ReDim Tabla(1 To UBound(ArrZ))
Tabla(1) = Application.NormSDist(ArrZ(1)) * n
For i = 2 To UBound(ArrZ) - 1
    Tabla(i) = (Application.NormSDist(ArrZ(i)) - _
        Application.NormSDist(ArrZ(i - 1))) * n
Next i
Tabla(UBound(ArrZ)) = (1 - _
    Application.NormSDist(ArrZ(UBound(ArrZ, 1) - 1))) * n
GenFrecEsp = Tabla

End Function

```

```

Function CreaHistograma
    (Valores, Intervalos, Optional Titulo)

Dim Result
Dim n As Integer
Dim i As Integer
Dim j As Integer
Dim Acum As Integer

If IsMissing(Titulo) Then
    Application.StatusBar = "Obteniendo histograma"
Else
    Application.StatusBar = "Obteniendo histograma de " & Titulo
End If
n = Valores.Rows.Count
ReDim Result(1 To UBound(Intervalos))
For i = 1 To n
    For j = 1 To UBound(Result) - 1
        If Valores.Cells(i) <= Intervalos(j) Then
            Result(j) = Result(j) + 1
            Acum = Acum + 1
        Exit For
    End If
    Next j
Next i
Result(UBound(Result)) = n - Acum
CreaHistograma = Result
Application.StatusBar = False

End Function

```

```

Function ChiC(Estimado, Real)

Dim i As Integer

For i = 1 To UBound(Estimado)
    ChiC = ChiC + (Estimado(i) - Real(i)) ^ 2 / Estimado(i)
Next i

End Function

```

Gráficas

```
Sub CreaGrafica()

Dim ArrHojas() As Worksheet
Dim TipoDatos As String
Dim RangoDeFechas As Range
Dim RangoDeDatos As Range
Dim RangoDeVars As Range
Dim VarIndep As String
Dim TipoGraf As String
Dim Variable
Dim Tipo As String
Dim i As Integer
Dim ArrTipoGraf
Dim ArrVarInd
Dim ArrVarDep

If Not ObtenerHojas(Titulo:="Graficar", _
    ArrTipo:=Array("Precios", "Rendimientos", "Volumen", _
    "Información", "Port"), Comentario:="a graficar", _
    MultipleHoja:=False, ArrHojas:=ArrHojas) Then Exit Sub
If Not RegresaPreferencia(NombrePref:="Tipo", _
    ValorPref:=TipoDatos, Hoja:=ArrHojas(0)) Then Stop

Select Case TipoDatos
    Case "Precio", "Rendimientos"
        Set RangoDeFechas = ArrHojas(0).Range("Tiempo")
        Set RangoDeDatos = ArrHojas(0).Range("Series")
        Set RangoDeVars = ArrHojas(0).Range("Titulos")
        VarIndep = ArrHojas(0).Range("TipoPrimCol")
        TipoGraf = "Dispersión"
    Case "Análisis"
        Set RangoDeDatos = Cells(1, 1).End(xlDown).End(xlDown).CurrentRegion
        Set RangoDeVars = RangoDeDatos.Resize(, 1)
        VarIndep = RangoDeDatos(1, 1)
        ArrVarDep = Array("Maximo", "Minimo", "Promedio")
        TipoGraf = "Dispersión"
    Case "Port"
        GraficaDePortafolio ArrHojas(0)
        Exit Sub
    Case ""
        Exit Sub
End Select

For Each Variable In RangoDeVars
    Tipo = TomaTipo(Variable)
    If VarType(ArrTipos) = 0 Then
        ReDim ArrTipos(2, 1)
        ArrTipos(1, 1) = Tipo
        ArrTipos(2, 1) = Variable.Address
        GoTo OtroTipo
    End If
    For i = 1 To UBound(ArrTipos, 2)
        If ArrTipos(1, i) = Tipo Then
            ArrTipos(2, i) = Union(Range(ArrTipos(2, i)), Variable).Address
            GoTo OtroTipo
        End If
    Next i
    ReDim Preserve ArrTipos(2, UBound(ArrTipos, 2) + 1)
    ArrTipos(1, UBound(ArrTipos, 2)) = Tipo
    ArrTipos(2, UBound(ArrTipos, 2)) = Variable.Address
OtroTipo:
Next Variable
```

```

ArrTipoGraf = Array("Area", "Columna", "Linea", _
    "Circular", "Anillos", "Dispersión", "Superficie")

With frm_Graf
    .CmbVarInd.List() = ArrVarInd
    '.LstVarsY1.List() = ArrVarDep
    '.CmbVarInd.ListIndex = Application.Match(VarIndep, RangoDeVars, 0) - 1
    .CmbTipoGraf1.List = ArrTipoGraf
    .CmbTipoGraf1.ListIndex = Application.Match(TipoGraf, ArrTipoGraf, 0) - 1
    .CmbTipoGraf2.List = ArrTipoGraf
    .CmbTipoGraf2.ListIndex = 0
    .CmbTipoY1.Clear
    .CmbTipoY2.Clear
    For i = 1 To UBound(ArrTipos, 2)
        .CmbTipoY1.AddItem ArrTipos(1, i)
        .CmbTipoY2.AddItem ArrTipos(1, i)
    Next i
    .CmbTipoY2.ListIndex = 0
    ' CambVars "CmbTipoY1"
    ' CambVars "CmbTipoY2"
    '.LstVarsY1.Selected = ArrSelecY1
    .ChkSegEje.Value = False
    SegundoEje
    .UserCancel = False
    .Show
    If .UserCancel Then Exit Sub
    Stop
End With

End Sub

```

```

Sub GraficaDePortafolio(Hoja As Worksheet)

Dim Grafica As Chart
Dim RangoDeDatos As Range
Dim RangoDeDatos1Renglon As Range
Dim RangoDeDatosTitulos As Range
Dim RangoDeDatosX As Range
Dim RangoDeDatosY As Range
Dim j As Integer
Dim i As Integer
Dim Figura As Shape
Dim IndicaPuntos As Boolean
Dim EsFronteraEfic As Boolean

Set RangoDeDatos = Hoja.Cells(1, 1).End(xlDown).End(xlDown).CurrentRegion
Set RangoDeDatos1Renglon = RangoDeDatos.Resize(1)

Set RangoDeDatos = RangoDeDatos.Offset(1, 0) _
    .Resize(RangoDeDatos.Rows.Count - 1, RangoDeDatos.Columns.Count)
Set RangoDeDatosX = RangoDeDatos.Offset(0, Application.Match("Var", _
    RangoDeDatos1Renglon, 0) - 1).Resize(RangoDeDatos.Rows.Count, 1)
Set RangoDeDatosY = RangoDeDatos.Offset(0, Application.Match("Ren Esp", _
    RangoDeDatos1Renglon, 0) - 1).Resize(RangoDeDatos.Rows.Count, 1)

Set Grafica = Charts.Add

With Grafica
    .ChartType = xlXYScatterSmooth
    .SetSourceData Source:=Union(RangoDeDatosX, RangoDeDatosY), PlotBy:=xlColumns

```

```

.HasTitle = True
.ChartTitle.Characters.Text = "Markowitz"
.PlotArea.Interior.ColorIndex = xlNone
.Axes(xlCategory, xlPrimary).HasTitle = True
.Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "Riesgo"
.HasAxis(xlCategory, xlPrimary) = True
.Axes(xlValue, xlPrimary).HasTitle = True
.Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Rendimiento"
.HasAxis(xlValue, xlPrimary) = True

j = 1
Do Until RangoDeDatos(1, 1) = ""
    Set RangoDeDatosX = RangoDeDatos.Offset(0, Application.Match("Var", _
        RangoDeDatos1Renglon, 0) - 1).Resize(, 1)
    Set RangoDeDatosY = RangoDeDatos.Offset(0, Application.Match("Ren Esp", _
        RangoDeDatos1Renglon, 0) - 1).Resize(, 1)
    Set RangoDeDatosTitulos = RangoDeDatos.Offset(0, _
        Application.Match("Punto", RangoDeDatos1Renglon, 0) - 1).Resize(, 1)
    If IsNumeric(RangoDeDatosTitulos(1)) Then
        If j > 1 Then
            .SeriesCollection.NewSeries
            j = j + 1
            .SeriesCollection(j).XValues = RangoDeDatosX
            .SeriesCollection(j).Values = RangoDeDatosY
        End If
        .SeriesCollection(j).Name = "Frontera"
        If RangoDeDatosTitulos(1, 2) = "Efic" Then
            IndicaPuntos = True
            EsFronteraEfic = True
        Else
            IndicaPuntos = False
            EsFronteraEfic = False
        End If
    Else
        IndicaPuntos = True
        EsFronteraEfic = False
    End If

    If IndicaPuntos Then
        For i = 1 To RangoDeDatosX.Count
            .SeriesCollection.NewSeries
            j = j + 1
            With .SeriesCollection(j)
                .Name = RangoDeDatosTitulos(i)
                .XValues = RangoDeDatosX(i)
                .Values = RangoDeDatosY(i)
                .HasDataLabels = False
                .ApplyDataLabels Type:=xlDataLabelsShowLabel, ShowSeriesName:=True
                .ApplyDataLabels Type:=xlDataLabelsShowLabel, ShowCategoryName:=False
                If EsFronteraEfic Then
                    .MarkerBackgroundColorIndex = 5
                    .MarkerForegroundColorIndex = 5
                    .MarkerStyle = xlCircle
                End If
                .Smooth = True
                .MarkerSize = 5
                .Shadow = False
            End With

            .Legend.LegendEntries(.Legend.LegendEntries.Count).Delete

        Next i
    End If

    Set RangoDeDatos = RangoDeDatos.End(xlDown).End(xlDown).CurrentRegion
Loop

```



```

.Move Before:=Hoja
.Location WHERE:=xlLocationAsObject, Name:=Hoja.Name

End With

Set Figura = Hoja.Shapes(Hoja.Shapes.Count)

With Figura
.Left = 530
.Top = 210
.Height = 400
.Width = 500
End With

End Sub

```

```

Function TomaTipo(Variable) As String

Select Case Variable
Case "Fechas"
TomaTipo = "Fechas"
Case "Maximo", "Minimo", "Mediana", "Promedio", "Desv. Estan.", _
"Cuartil 50", "Cuartil 25", "Cuartil 75", "Alfa"
If Left(Application.VLookup("Datos", ActiveSheet.Cells(1, 1) _
.CurrentRegion, 2, False), 6) = "Precio" Then
TomaTipo = "Precio"
Else
TomaTipo = "Rendimiento"
End If
Case "Ren Esp", "Resid", "Ren Merc", "R EspTLR", "Ren Obs", "R ObsTLR"
TomaTipo = "Rendimiento"
Case "Varianza", "Var", "Var. Residual"
TomaTipo = "Riesgo"
Case "Sesgo"
TomaTipo = "Sesgo"
Case "Curtosis"
TomaTipo = "Curtosis"
Case "Normalidad"
TomaTipo = "Normalidad"
Case "Prueba"
TomaTipo = "Prueba"
Case "Beta"
TomaTipo = "Beta"
Case "Instrum"
TomaTipo = "Instrumento"
Case "<>Cero", "Fact", "Punto"
TomaTipo = "Entero"
Case "Acum Esp"
TomaTipo = "Indice"
Case "Parte"
TomaTipo = "Texto"
Case Else
TomaTipo = "Pesos"
End Select

End Function

```

Matriz de covarianzas y correlaciones

```
Function CreaMatCovCorr(Tipo As String, _
    Optional TipoDatos, Optional RangoDeDatos, Optional PreguntaHoja, _
    Optional RangoDeInstr, Optional RangoDeFechas, Optional ArrDeInstrSel, _
    Optional RangoDeDatosDesvStd, Optional Tiempo) As Range

Dim HojaCovCorr As Worksheet
Dim ArrHojas() As Worksheet
Dim Titulo As String
Dim ArrTipo
Dim RangoPrimColDesvStd
Dim RangoAuxNumeros
Dim ArrBoolInstrSel
Dim ArrDeInform
Dim ArrDeValores
Dim Hoja As String
Dim HojaDesvStd As String
Dim TipoPrimColDesvStd
Dim SaltaBorrar As Boolean
Dim UsaMatAddIn As Boolean
Dim Muestral As Boolean
Dim FechMuestra As Integer
Dim FechInicio As Integer
Dim FechTermino As Integer
Dim Muestra As Integer
Dim Periodo As Integer
Dim i As Integer
Dim j As Integer
Dim n As Integer
Dim m As Integer
Dim k As Integer
Dim Inicia As Integer
Dim TipoHoja As String
Dim RangoDeCambio 'No cambiar a rango
Dim NumFilaDesvStd As Integer
Dim RangoNvoDeDatos As Range
Dim NombreDatos As String
Dim ConFormulas As Boolean

Select Case Tipo
    Case "Covarianzas"
        TipoHoja = "MatCov"
        ArrTipo = Array("Precios", "Rendimientos", "Mat Correlaciones")
    Case "Correlaciones"
        TipoHoja = "MatCor"
        ArrTipo = Array("Precios", "Rendimientos", "Mat Covarianzas")
End Select

If IsMissing(RangoDeDatos) Then
    RepiteHojaMatriz:
        RangoDeDatos = ""
        If Not ObtenerHojas(Titulo:="Matriz de " & Tipo, ArrTipo:=ArrTipo, _
            Comentario:="para formar la Matriz de " & Tipo, MultipleHoja:=False, _
            ArrHojas:=ArrHojas) Then Exit Function
        If Not RegresaPreferencia(NombrePref:="Tipo", ValorPref:=TipoDatos, _
            Hoja:=ArrHojas(0)) Then Stop
        ArrDeInform = Array("Datos")
        Select Case TipoDatos
            Case "Mat Correlaciones"
                ReDim ArrDeInstrSel(RangoDeInstr.Count) As String
                ReDim ArrBoolIntrSel(RangoDeInstr.Count) As Boolean
                For j = 1 To RangoDeInstr.Count
                    ArrDeInstrSel(j) = RangoDeInstr(j).Value
```

```

    ArrBoolIntrSel(j) = True
Next j
Select Case TipoPrimColDesvStd
    Case "Fechas", "Periodo"
        Tiempo = RangoPrimColDesvStd(NumFilaDesvStd)
    Case "Inform", "Función"
        NumFilaDesvStd = Application.Match("Desv. Estan.", _
            RangoPrimColDesvStd, 0)
End Select
Set RangoDeDatosDesvStd = RangoDeDatosDesvStd(NumFilaDesvStd, 1) _
    .Resize(1, RangoDeInstr.Count)
Case "Mat Covarianzas"
    ReDim ArrDeInstrSel(RangoDeInstr.Count) As String
    ReDim ArrBoolIntrSel(RangoDeInstr.Count) As Boolean
    For j = 1 To RangoDeInstr.Count
        ArrDeInstrSel(j) = RangoDeInstr(j).Value
        ArrBoolIntrSel(j) = True
    Next j
    HojaDesvStd = RangoDeDatosDesvStd.Parent.Name
Case Else
    Set RangoDeFechas = ArrHojas(0).Range("Tiempo")
    Set RangoDeDatos = ArrHojas(0).Range("Series")
    Set RangoDeInstr = ArrHojas(0).Range("Titulos")
    With frm_Fecha
        Titulo = "Matriz de " & Tipo
        .Caption = Titulo & " [" & Hoja & "]"
        .CmbMuestra.RowSource = RangoDeFechas.Address(External:=True)
        .CmbInicio.RowSource = RangoDeFechas.Address(External:=True)
        .CmbTermino.RowSource = RangoDeFechas.Address(External:=True)
        .SpnNoMuestra.Max = RangoDeFechas.Count
        .SpnNoPeriodo.Min = 1
        .SpnNoPeriodo.Max = RangoDeFechas.Count
        .ValidaControles = False
        .CmbMuestra.ListIndex = 0
        .CmbInicio.ListIndex = RangoDeFechas.Count - 1
        .CmbTermino.ListIndex = RangoDeFechas.Count - 1
        .SpnNoMuestra.Max = RangoDeFechas.Count
        .SpnNoMuestra.Value = RangoDeFechas.Count
        .SpnNoPeriodo.Max = RangoDeFechas.Count
        .SpnNoPeriodo.Value = 1
        .TxtNoPeriodo.Value = .SpnNoPeriodo.Value
        .ValidaControles = True
        .UserCancel = False
    RepiteMatFecha:
        .Show
        If .UserCancel Then GoTo RepiteHojaMatriz
        Muestra = .SpnNoMuestra.Value
        FechMuestra = .CmbMuestra.ListIndex + 1
        FechInicio = .CmbInicio.ListIndex + 1
        FechTermino = .CmbTermino.ListIndex + 1
        Periodo = .SpnNoPeriodo.Value
        If Periodo > 5 Then If MsgBox(Prompt:="Se van a crear " & _
            Periodo & " hojas de " & Tipo & "." & Chr(13) & _
            "Está seguro de continuar?", Buttons:=vbYesNo, _
            Title:="Matriz de " & Tipo & " [" & Hoja & "]" ) = vbNo Then _
            GoTo RepiteMatFecha
    End With
    AgregaDatos Datos:=Array("Tipo", "Desde", "Hasta", "Muestra", _
        "Población", "Usa Add-In"), Arreglo:=ArrDeInform
End Select
SaltaBorrar = False
Else
    AgregaDatos Datos:=Array("Tipo", "Desde", "Hasta", "Muestra", _
        "Población", "Usa Add-In"), Arreglo:=ArrDeInform
    Muestra = RangoDeFechas.Count
    FechInicio = RangoDeFechas.Count
    FechMuestra = 1

```

```

n = RangoDeDatos.Columns.Count
Periodo = 1
SaltaBorrar = True
End If
Select Case Idioma
Case IdiomaIngles
UsaMatAddIn = MenuBars(xlWorksheet).Menus(NombreMenu) _
.MenuItems("&Matrix").MenuItems("&Excel Add-In").Checked
Case IdiomaEspanol
UsaMatAddIn = MenuBars(xlWorksheet).Menus(NombreMenu) _
.MenuItems("&Matriz").MenuItems("&Excel Add-In").Checked
Case Else
Stop
End Select
ReDim ArrHojas(0)
Set ArrHojas(0) = RangoDeDatos.Parent
If Not RegresaPreferencia(NombrePref:="Nombre", _
ValorPref:=NombreDatos, Hoja:=ArrHojas(0)) Then Stop

Hoja = RangoDeDatos.Parent.Name
ReDim ArrDeValores(UBound(ArrDeInform))
For i = LBound(ArrDeInform) To UBound(ArrDeInform)
Select Case ArrDeInform(i)
Case "Datos"
ArrDeValores(i) = Hoja
Case "Hoja Desv. Std."
ArrDeValores(i) = HojaDesvStd
Case "Tiempo"
ArrDeValores(i) = Tiempo
Case "Tipo"
ArrDeValores(i) = TipoDatos
Case "Muestra"
ArrDeValores(i) = Muestra
Case "Población"
If Not RegresaPreferencia(NombrePref:="Muestral", _
ValorPref:=Muestral, Hoja:=ArrHojas(0)) Then
Muestral = True
End If
If Muestral Then
ArrDeValores(i) = "Muestral"
Else
ArrDeValores(i) = "Total"
End If
Case "Usa Add-In"
ArrDeValores(i) = UsaMatAddIn
End Select
Next i
Select Case Idioma
Case IdiomaIngles
ConFormulas = MenuBars(xlWorksheet).Menus(NombreMenu) _
.MenuItems("F&ormulas").MenuItems("&Matrix").Checked
Case IdiomaEspanol
ConFormulas = MenuBars(xlWorksheet).Menus(NombreMenu) _
.MenuItems("F&ormulas").MenuItems("&Matriz").Checked
Case Else
Stop
End Select
Application.Calculation = xlCalculationManual
For k = 1 To Periodo
If k > 1 Then SaltaBorrar = True
If Not SigHoja(Libro:=ActiveWorkbook, TipoHoja:=TipoHoja, _
NombreDatos:=NombreDatos, NuevaHoja:=HojaCovCorr, _
SaltaBorrar:=SaltaBorrar) Then GoTo RepiteHojaMatriz
With HojaCovCorr
Set RangoNvoDeDatos = RangoDeDatos(FechMuestra + k - 1, 1) _
.Resize(Muestra, RangoDeInstr.Count)
ArrDeValores(Application.Match("Desde", ArrDeInform, 0) - 1) = _

```

```

RangoDeFechas(FechInicio + k - Muestra)
ArrDeValores(Application.Match("Hasta", ArrDeInform, 0) - 1) = _
RangoDeFechas(FechInicio + k - 1)
ReDim ArrDeInstrSel(RangoDeInstr.Columns.Count)
ArrDeInstrSel = RegresaVectorDeRango(RangoDeInstr)
m = RangoDeInstr.Count
n = RangoDeInstr.Count
FormatoHojas Hoja:=HojaCovCorr, ArrInfor:=ArrDeInform, _
ArrValInf:=ArrDeValores, ArrTitulos:=ArrDeInstrSel, _
TipoPrimCol:="Titulos", m:=m, n:=n, SigRen:=Inicia
Select Case TipoDatos
Case "Mat Correlaciones"
.Range("Series").FormulaArray = "=MMult(Transpose(" & _
RangoDeDatosDesvStd.Address(External:=True) & ")," & _
RangoDeDatosDesvStd.Address(External:=True) & ")*(" & _
RangoDeDatos.Address(External:=True) & ")"
Case "Mat Covarianzas"
.Range("Series").FormulaArray = "=" & _
RangoDeDatos.Address(External:=True) & "/MMult(Transpose(" & _
RangoDeDatosDesvStd.Address(External:=True) & ")," & _
RangoDeDatosDesvStd.Address(External:=True) & ")"
Case "Precios", "Rendimientos"
If UsaMatAddIn Then
Union(.Range("TipoPrimCol"), .Range("Titulos"), _
.Range("OtrosTitulos")).ClearContents
Select Case Application.International(xlCountryCode)
Case 1
AddIns("Analysis ToolPak - VBA").Installed = True
If Tipo = "Covarianzas" Then
Application.Run "ATPVBAEN.XLA!Mcovar", RangoNvoDeDatos, _
.Range("TipoPrimCol"), "C", False
ElseIf Tipo = "Correlaciones" Then
Application.Run "ATPVBAEN.XLA!Mcorrel", RangoNvoDeDatos, _
.Range("TipoPrimCol"), "C", False
Else
Stop
End If
Case 34
AddIns("Herramientas para análisis").Installed = True
Application.Run "Mcovar", Union(RangoDeDatos, RangoDeInstr), _
.Range("Tipo"), "C", True
End Select
.Range("TipoPrimCol").Value = "Titulos"
.Range("Titulos").Value = RangoDeInstr.Value
.Range("OtrosTitulos").Value = _
Application.Transpose(RangoDeInstr.Value)
For j = 1 To n - 1
For i = j + 1 To n
.Range("Series")(j, i).Formula = "=" & _
.Range("Series")(i, j).Address
Next i
If IsObject(RangoDeCambio) Then
Set RangoDeCambio = Union(RangoDeCambio, Range("Series") _
(j, j + 1).Resize(1, n - j))
Else
Set RangoDeCambio = Range("Series")(j, j + 1).Resize(1, n - j)
End If
Next j
RangoDeCambio.Locked = True
RangoDeCambio.Interior.ColorIndex = 40
RangoDeCambio = 0
Else
ActiveWorkbook.Names.Add NameLocal:="" & .Name & "!Muestra", _
RefersToR1C1Local:=.Range("Información")( _
Application.Match("Muestra", ArrDeInform, 0), 2)
Set RangoAuxNumeros = Union(.Range("Series")(n + 1, 1).Resize(1, n), _
.Range("Series")(1, n + 1).Resize(n, 1))

```

```

RangoAuxNumeros.Font.ColorIndex = 15
ReDim ArrNumeros(n) As Integer
j = 0
If IsArray(ArrBoolInstrSel) Then
    For i = 1 To RangoDeInstr.Count
        If ArrBoolInstrSel(i) Then
            j = j + 1
            ArrNumeros(j) = i
        End If
    Next i
Else
    For i = 1 To RangoDeInstr.Count
        ArrNumeros(i) = i
    Next i
End If
.Range("Series")(n + 1, 1).Resize(1, n).Value = ArrNumeros
.Range("Series")(1, n + 1).Resize(n, 1).Value = _
Application.Transpose(ArrNumeros)
Application.StatusBar = "Calculando " & Tipo & "..."
If Tipo = "Covarianzas" Then
    .Range("Series").Formula = "=Covar(Offset(" & _
        RangoNvoDeDatos(1, 1).Address(External:=True) & ",0," & _
        .Range("Series")(n + 1, 1).Address(ColumnAbsolute:=False) & _
        ",Muestra,1),Offset(" & RangoNvoDeDatos(1, 1) _
        .Address(External:=True) & ",0," & .Range("Series")(1, n + 1) _
        .Address(RowAbsolute:=False) & ",Muestra,1))"
Else
    .Range("Series").Formula = "=Correl(Offset(" & _
        RangoNvoDeDatos(1, 1).Address(External:=True) & ",0," & _
        .Range("Series")(n + 1, 1).Address(ColumnAbsolute:=False) & _
        ",Muestra,1),Offset(" & RangoNvoDeDatos(1, 1) _
        .Address(External:=True) & ",0," & .Range("Series")(1, n + 1) _
        .Address(RowAbsolute:=False) & ",Muestra,1))"
End If
If Muestral Then
    .Range("Series")(1, 1).Formula = .Range("Series")(1, 1).Formula & _
        "/(1-1/Min(Count(Offset(" & RangoNvoDeDatos(1, 1) _
        .Address(External:=True) & ",0," & .Range("Series")(n + 1, 1) _
        .Address(ColumnAbsolute:=False) & ",Muestra,1)),Count(Offset(" & _
        RangoNvoDeDatos(1, 1).Address(External:=True) & ",0," & _
        .Range("Series")(1, n + 1).Address(RowAbsolute:=False) & _
        ",Muestra,1))))"
    .Range("Series")(1, 1).Copy Destination:=.Range("Series")
    Calculate
End If
End If
Case Else
    Stop
End Select
If Not ConFormulas Then
    .Range("Series").Copy: .Range("Series").PasteSpecial Paste:=xlValues
    If Not UsaMatAddIn And TipoDatos <> "Mat Covarianzas" And _
        TipoDatos <> "Mat Correlaciones" Then RangoAuxNumeros.ClearContents
    On Error GoTo MnjErr
    Set RangoDeCambio = .Range("Series")
    .SpecialCells(Type:=xlConstants, Value:=xlErrors)
    On Error GoTo 0
Else
    On Error GoTo MnjErr
    Set RangoDeCambio = .Range("Series")
    .SpecialCells(Type:=xlFormulas, Value:=xlErrors)
    On Error GoTo 0
End If
If IsMissing(PreguntaHoja) Then
    RangoDeCambio.Value = "N.D."
Else
    RangoDeCambio.Value = 0

```

```

End If
SinErrores:
    Set CreaMatCovCorr = .Range("Series")
End With
Application.StatusBar = False
Next k
Application.Calculation = xlCalculationAutomatic

Exit Function

MnjErr:
Select Case Err
    Case 1004 'No se encontraron celdas.
        Resume SinErrores
    Case Else
        MsgBox Prompt:="Error, Title:="Error " & Err
    End Select

End Function

```

```

Sub GrafCalor()

Dim ArrTipo
Dim ArrHojas
Dim RangoDeMatriz As Range
Dim Max As Double
Dim Min As Double
Dim i As Integer
Dim j As Integer

ArrTipo = Array("MatCov", "MatCor")
If ObtenerHojas(Titulo:="Gráficas de Calor", ArrTipo:=ArrTipo, _
Comentario:="para formar la grafica de Calor", MultipleHoja:=False, _
ArrHojas:=ArrHojas) Then
    ArrHojas(0).Unprotect
    Set RangoDeMatriz = ArrHojas(0).Range("Series")
    Max = Application.Max(RangoDeMatriz)
    Min = Application.Min(RangoDeMatriz)
    For j = 1 To RangoDeMatriz.Columns.Count
        For i = 1 To RangoDeMatriz.Rows.Count
            Select Case RangoDeMatriz(i, j)
                Case Is < Min + 0.125 * (Max - Min)
                    RangoDeMatriz(i, j).Interior.ColorIndex = 3
                Case Is < Min + 0.25 * (Max - Min)
                    RangoDeMatriz(i, j).Interior.ColorIndex = 7
                Case Is < Min + 0.375 * (Max - Min)
                    RangoDeMatriz(i, j).Interior.ColorIndex = 38
                Case Is < Min + 0.5 * (Max - Min)
                    RangoDeMatriz(i, j).Interior.ColorIndex = 40
                Case Is < Min + 0.625 * (Max - Min)
                    RangoDeMatriz(i, j).Interior.ColorIndex = 15
                Case Is < Min + 0.75 * (Max - Min)
                    RangoDeMatriz(i, j).Interior.ColorIndex = 33
                Case Is < Min + 0.875 * (Max - Min)
                    RangoDeMatriz(i, j).Interior.ColorIndex = 17
                Case Else
                    RangoDeMatriz(i, j).Interior.ColorIndex = 32
            End Select
        Next i
    Next j
End If

```

```
End Sub
```

Portafolios

```
Global SolverEngine As String

Sub CreaPortafolio(Metodo As String)

Dim ArrHojas() As Worksheet
Dim TipoPort As String
Dim Pts As Integer
Dim RiesMax As Double
Dim RendMin As Double
Dim ArrDeHojasProy
Dim ArrDeInf
Dim ArrDeDatos
Dim ArrDatosHojas
Dim Muestra
Dim FechMuestra As Integer
Dim FechInicio As Integer
Dim FechTermino As Integer
Dim Periodo

Dim NoDeHojas As Integer
Dim NombreDatos As String
Dim TipoPrimCol
Dim RangoVars As Range

Dim i As Integer
Dim j As Integer
Dim l As Integer
Dim ColInstr As Integer
Dim HojaPort As Worksheet
Dim ArrTitulos
Dim ArrDeOtrosDatos
Dim n As Integer
Dim RenPortIni As Integer
Dim ColIni As Integer
Dim RangoCambio As Range
Dim FormaDeCalcular As Integer
Dim ConFormulas As Boolean
Const BorrarHojas = True

Dim ConLimites As Boolean
Dim TipoPosicion As String
Dim Tiempo As Integer
Dim Iteraciones As Long
Dim Precision As Single
Dim Estimado As Single

Dim ConTLR As Boolean
Dim TLR As Integer
Dim ConTLRCamb As Boolean
Dim RangoRends As Range
Dim RangoCov
Dim RangoActRend As Range
Dim ConRendObs As Boolean
Dim ArrHojasRendObs
```



```

Dim HojaRendObs As Worksheet
Dim RangoRendsObs As Range
Dim RangoActRendObs As Range
Dim PeriodoMas As Integer
Dim ArrPosClases
Dim Cotiza
Dim ConLimConjuntos As Boolean
Dim k As Integer
Dim ArrClases
Dim InstrFact As Integer
Dim ConAutoLimites As Boolean
Dim EnDifHojas As Boolean
Dim ArrDeDatosForm
Dim i_form As Integer
Dim j_form As Integer
Dim ConRendAcum As Boolean
Dim SaltaPregBorr As Boolean
Dim ArrHojasDef
Dim CurrentWB As Workbook

'SolverEngine = "Standard GRG Nonlinear"
SolverEngine = "Standard LP/Quadratic"
'SolverEngine = "Standard Evolutionary"
'SolverEngine = "Standard Interval Global"
'SolverEngine = "Standard SOCP Barrier"
'SolverEngine = "KNITRO Solver"
'SolverEngine = "LGO Global Solver"
'SolverEngine = "Large-Scale GRG Solver"
'SolverEngine = "Large-Scale LP Solver"
'SolverEngine = "Large-Scale SQP Solver"
'SolverEngine = "MOSEK Solver Engine"
'SolverEngine = "OptQuest Solver"
'SolverEngine = "XPRESS Solver Engine"

'EngineDesc Solver engine specified
'"Standard GRG Nonlinear" Nonlinear GRG Solver
'"Standard Simplex LP" Simplex LP Solver
'"Standard LP/Quadratic" LP/Quadratic Solver
'"Standard Evolutionary" Evolutionary Solver
'"Standard Interval Global" Interval Global Solver
'"Standard SOCP Barrier" SOCP Barrier Solver
'"KNITRO Solver" KNITRO Solver
'"LGO Global Solver" LGO Global Solver
'"Large-Scale GRG Solver" Large-Scale GRG Solver
'"Large-Scale LP Solver" Large-Scale LP Solver
'"Large-Scale SQP Solver" Large-Scale SQP Solver
'"MOSEK Solver Engine" MOSEK Solver Engine
'"OptQuest Solver" OptQuest Solver
'"XPRESS Solver Engine" XPRESS Solver Engine

If Not ObtenerHojas(Titulo:=Metodo, ArrTipo:=Array("Rendimientos"), _
  ArrSubTipo:=Array("Maximo", "Minimo", "Mediana", "Promedio"), _
  Comentario:="para formar portafolios", ArrHojas:=ArrHojas) Then Exit Sub
If ArrHojas(0).Range("Tiempo").Count < 3 Then
  MsgBox "No se puede tener menos de 3 datos para generar los portafolios", _
    vbCritical, TituloSistema
End If

Idioma = TomaIdioma

With MenuBars(xlWorksheet).Menus(NombreMenu)
  Select Case Idioma
    Case IdiomaIngles
      ConFormulas = .MenuItems("&ormulas").MenuItems("&Portfolio").Checked
      ConLimites = .MenuItems("&Portfolios").MenuItems("&Limits").Checked
    Case IdiomaEspañol

```

```

        ConFormulas = .MenuItems("F&ormulas").MenuItems("&Portafolio").Checked
        ConLimites = .MenuItems("&Portafolios").MenuItems("&Limites").Checked
    Case Else
        Stop
    End Select
End With

With frm_Port
    Select Case Idioma
        Case IdiomaIngles
            .BtnCancelar.Caption = "Cancel"
            .BtnContinuar.Caption = "OK"
            .OptPMV.Caption = "of Minimum Variance (PMV)"
            .OptRiesgo.Caption = "with Risk <= than..."
            .OptRend.Caption = "with Returns >= than..."
            .OptFrontera.Caption = "from the Frontier"
            .OptPuntosFactibles.Caption = "Feasible points"
            .OptGenLimites.Caption = "generate Limits"
        Case IdiomaEspanol
            .BtnCancelar.Caption = "Cancelar"
            .BtnContinuar.Caption = "Continuar"
            .OptPMV.Caption = "de Mnima Varianza (PMV)"
            .OptRiesgo.Caption = "con Riesgo <= a..."
            .OptRend.Caption = "con Rendimiento >= a..."
            .OptFrontera.Caption = "de la Frontera"
            .OptPuntosFactibles.Caption = "puntos Factibles"
            .OptGenLimites.Caption = "generar Limites"
        Case Else
            Stop
    End Select
    .OptFrontera.Enabled = True
    .OptGenLimites.Enabled = False
    .OptPMV.Value = True
    .UserCancel = False
DespliegaDial:
    .Show
    If .UserCancel Then Exit Sub
    ArrDeDatos = Array("Var", "Ren Esp")
    ArrDeOtrosDatos = Array("Filtro", "Lim Inf", "Lim Sup")
    ArrDeInf = Array("Datos", "Creado", "Mtodo", "Tipo", _
        "Desde", "Hasta", "Muestra", "Posiciones")
    Select Case True
        Case .OptPMV
            TipoPort = "PMV"
            Pts = 1
        Case .OptRiesgo
            TipoPort = "Riesgo"
            If .TxtRiesgoMax.Text = "" Then
                MsgBox Title:="Max Rendimiento dado un Riesgo", _
                    Prompt:="Indique el riesgo"
                GoTo DespliegaDial
            End If
            RiesMax = .TxtRiesgoMax.Text
            If Not RiesMax > 0 Then
                MsgBox Title:="Max Rendimiento dado un Riesgo", _
                    Prompt:="El riesgo no es vlido"
                GoTo DespliegaDial
            End If
            AgregaDatos Array("Riesgo Max"), ArrDeInf
            Pts = 1
        Case .OptRend
            TipoPort = "Rend"
            If .TxtRendMin.Text = "" Then
                MsgBox Title:="Min Riesgo dado un Rendimiento", _
                    Prompt:="Indique el rendimiento"
                GoTo DespliegaDial
            End If
    End Select
End With

```

```

    RendMin = .TxtRendMin.Text
    AgregaDatos Array("Rend. Min"), ArrDeInf
    Pts = 1
    Case .OptFrontera
        TipoPort = "Frontera"
        AgregaDatos Array("Punto", "Parte"), ArrDeDatos
        Pts = .SpnPtsFront
        AgregaDatos Array("Frontera", "Puntos"), ArrDeInf
    Case .OptPuntosFactibles
        TipoPort = "Ptos Factibles"
        AgregaDatos Array("Punto", "Parte"), ArrDeDatos
        Pts = .SpnPtsFact
        AgregaDatos Array("Portafolios", "Puntos"), ArrDeInf
    Case .OptGenLimites
        Stop
        TipoPort = "AutoLimites"
        ConAutoLimites = True
        AgregaDatos Array("Riesgos Max"), ArrDeInf
        AgregaDatos Array("Lim Min", "Lim Max"), ArrDeOtrosDatos
    End Select

End With

ArrHojas(0).Activate
n = ArrHojas(0).Range("Titulos").Count
ArrTitulos = RegresaVectorDeRango(ArrHojas(0).Range("Titulos"))
TipoPrimCol = ArrHojas(0).Range("TipoPrimCol")
If Not RegresaPreferencia(NombrePref:="Nombre", _
    ValorPref:=NombreDatos, Hoja:=ArrHojas(0)) Then Stop
ReDim ArrFilMaxMin(0 To UBound(ArrDeOtrosDatos), 1 To n) As String
For i = 1 To n
    ArrFilMaxMin(0, i) = True
Next i
If ConLimites Then
    If Not ObtienePosiciones(TipoPosicion, ArrTitulos) Then Exit Sub
Else 'Largas
    For i = 1 To n
        ArrFilMaxMin(2, i) = 0 'Lim Inf
    Next i
End If

If Not RegresaFechas(Metodo, ArrHojas(0), ArrHojas(0).Range("Tiempo"), 3, _
    Muestra, FechMuestra, FechInicio, FechTermino, Periodo, True) Then Exit Sub

With ArrHojas(0)
    Set RangoVars = CreaAnalisis(ArrDeFunc:=Array("Varianza"), _
        Tipo:="Rendimientos", Multiple:=False, FechMuestra:=FechMuestra, _
        Muestra:=Muestra, Periodo:=Periodo, RangoDeDatos:=.Range("Series"), _
        RangoDeTitulos:=.Range("Titulos"), RangoPrimCol:=.Range("Tiempo"), _
        TipoPrimCol:=TipoPrimCol)
End With

ReDim ArrDeValInf(UBound(ArrDeInf))
For i = LBound(ArrDeInf) To UBound(ArrDeInf)
    Select Case ArrDeInf(i)
        Case "Datos"
            ArrDeValInf(i) = ArrHojas(0).Name
        Case "Creado"
            ArrDeValInf(i) = Now
        Case "Método"
            ArrDeValInf(i) = Metodo
        Case "Tipo"
            ArrDeValInf(i) = TipoPort 'Port
        Case "Tiempo", "Desde"
            ArrDeValInf(i) = ArrHojas(0).Range("Tiempo")(FechInicio, 1)
        Case "Hasta"
    
```

```

    ArrDeValInf(i) = ArrHojas(0).Range("Tiempo")(FechTermino, 1)
Case "Muestra"
    ArrDeValInf(i) = Muestra
Case "Riesgo Max"
    ArrDeValInf(i) = RiesMax
Case "Riesgos Max"
    ArrDeValInf(i) = RiesMax 'ArrRiesgos
Case "Rend. Min."
    ArrDeValInf(i) = RendMin
Case "Puntos"
    ArrDeValInf(i) = Pts
Case "Capital"
    ArrDeValInf(i) = 100
Case "Frontera"
    ArrDeValInf(i) = "Eficiente"
Case "Posiciones"
    ArrDeValInf(i) = TipoPosicion
Case "Portafolios"
    ArrDeValInf(i) = "Factibles"
Case Else
    Stop
End Select
Next i
ArrHojasDef = Array("Muestra", "Muestra", "Muestra")
AgregaDatos Array("Suma"), ArrDeDatos
FormaDeCalcular = Application.Calculation
Application.Calculation = xlAutomatic
If Not ObtenerHojas(Titulo:=Metodo, ArrTipo:=Array("Rendimientos"), _
    Comentario:="con rendimientos observados", ArrHojas:=ArrHojasRendObs) Then _
    Exit Sub
Set HojaRendObs = ArrHojasRendObs(0)
NoDeHojas = 1

With ArrHojas(0)
    Set RangoRends = CreaAnalisis(ArrDeFunc:=Array("Promedio"), _
        Tipo:="Rendimientos", Multiple:=False, FechMuestra:=FechMuestra, _
        Muestra:=Muestra, Periodo:=Periodo + NoDeHojas - 1, _
        RangoDeDatos:=.Range("Series"), RangoDeTitulos:=.Range("Titulos"), _
        RangoPrimCol:=.Range("Tiempo"), TipoPrimCol:=TipoPrimCol)
End With

Set CurrentWB = ActiveWorkbook
CurrentWB.Activate

For l = 1 To NoDeHojas
    SigHoja Libro:=ArrHojas(0).Parent, TipoHoja:="Port", _
        NombreDatos:=NombreDatos, NuevaHoja:=HojaPort, SaltaBorrar:=False
    With HojaPort
        FormatoHojas Hoja:=HojaPort, ArrInfor:=ArrDeInf, ArrValInf:=ArrDeValInf, _
            ArrOtrosDatos:=ArrDeOtrosDatos, ArrTitulos:=ArrTitulos, _
            TipoPrimCol:=TipoPrimCol, ArrDatos:=ArrDeDatos, m:=1, n:=n, _
            SigRen:=RenPortIni, SigCol:=ColIni

        RenPortIni = RenPortIni + 1 'Cambiar lo que regresa formato para renini
        Set RangoCambio = .Range("Series")
        AddIns("Solver Add-in").Installed = True
        Tiempo = 600
        Iteraciones = 10000 '100
        Precision = 0.0000001 '0.000001
        Estimado = 2 'Quadratic
        If Metodo <> "Generar" Then
            SolverOptions MaxTime:=Tiempo, Iterations:=Iteraciones, _
                Precision:=Precision, AssumeLinear:=False, StepThru:=False, _
                Estimates:=Estimado
        End If
    End With
    For i = LBound(ArrDeDatos) To UBound(ArrDeDatos)

```

```

Select Case ArrDeDatos(i)
Case "<>Cero"
    .Range("Datos")(1, Application.Match("<>Cero", ArrDeDatos, 0)) _
    .Formula = "=CountIf(" & RangoCambio(1, ColInstr + 1) _
    .Resize(1, ArrHojas(0).Range("Titulos").Columns.Count) _
    .Address(RowAbsolute:=False) & ", "<>0")"
Case "Suma"
    .Range("Datos")(1, Application.Match("Suma", ArrDeDatos, 0)) _
    .Formula = "=SUM(" & RangoCambio.Address(RowAbsolute:=False) & ")"
If Metodo <> "Generar" Then
    SolverAdd CellRef:=.Range("Datos")(1, Application.Match("Suma", _
    ArrDeDatos, 0)), Relation:=2, FormulaText:="=1"
End If
End Select
Next i
For j = 1 To n
Application.StatusBar = "Ingresando restricciones (" & j & "/" & n & _
") " & ArrHojas(0).Range("Titulos")(j) & "..."
If Metodo <> "Generar" Then
If ArrFilMaxMin(0, j) Then
    .Range("OtrosDatos")(Application.Match("Filtro", ArrDeOtrosDatos, _
    0), ColInstr + j).Value = "Si"
If ArrFilMaxMin(2, j) <> "" Then
    .Range("OtrosDatos")(Application.Match("Lim Inf", ArrDeOtrosDatos, _
    0), ColInstr + j).Value = ArrFilMaxMin(2, j)
    SolverAdd CellRef:=RangoCambio(1, ColInstr + j), Relation:=3, _
    FormulaText:=Range("OtrosDatos")(Application.Match("Lim Inf", _
    ArrDeOtrosDatos, 0), ColInstr + j).Address
End If
If ArrFilMaxMin(1, j) <> "" Then
    .Range("OtrosDatos")(Application.Match("Lim Sup", ArrDeOtrosDatos, _
    0), ColInstr + j).Value = ArrFilMaxMin(1, j)
    SolverAdd CellRef:=RangoCambio(1, ColInstr + j), Relation:=1, _
    FormulaText:=Range("OtrosDatos")(Application.Match("Lim Sup", _
    ArrDeOtrosDatos, 0), ColInstr + j).Address
End If
If ContLR Then
    RangoCambio(1, ColInstr + j) = 0
Else
    RangoCambio(1, ColInstr + j) = 1 / (UBound(ArrTitulos) + 1)
End If
Else
    .Range("OtrosDatos")(Application.Match("Filtro", ArrDeOtrosDatos, _
    0), ColInstr + j).Value = "No"
    SolverAdd CellRef:=RangoCambio(1, ColInstr + j), Relation:=2, _
    FormulaText:="0"
    RangoCambio(1, ColInstr + j).Font.ColorIndex = 3
    RangoCambio(1, ColInstr + j).NumberFormat = "General"
    RangoCambio(1, ColInstr + j) = 0
End If
End If
End If
Next j
Application.StatusBar = False
Select Case TipoPort
Case "Riesgo"
    SolverAdd CellRef:=.Range("Datos")(1, Application.Match("Var", _
    ArrDeDatos, 0)), Relation:=1, FormulaText:=CSng(RiesMax)
Case "Rend"
    Stop
    SolverAdd CellRef:=.Range("Datos")(1, Application.Match("Ren Esp", _
    ArrDeDatos, 0)), Relation:=3, FormulaText:=RendMin
End Select
.Select
Cotiza = EmpiezaTerminaCotizar(RangoDeTitulos:=ArrHojas(0) _
.Range("Titulos"), RangoDeDatos:=ArrHojas(0).Range("Series"))
For i = Periodo To 1 Step -1 '- NoDeHojas + 1
    Set RangoActRend = RangoRends(i + PeriodoMas, 1).Resize(1, n)

```

```

If Not EndDifHojas Then .Range("Tiempo").Value = RangoActRend(1, 0).Value
If ConTLR Then
.Range("Datos")(1, Application.Match("R EspTLR", ArrDeDatos, 0)) _
.Value = RangoActRend(1, TLR)
.Range("Datos")(1, Application.Match("Ren Esp", ArrDeDatos, 0)) _
.Formula = "=" & RangoCambio(1, Application.Match("TLR", _
RangoCambio(0, 1).Resize(1, RangoCambio.Count), 0)) _
.Address(RowAbsolute:=False) & "*" & Range("Datos") _
(1, Application.Match("R EspTLR", ArrDeDatos, 0)) _
.Address(RowAbsolute:=False)
.Range("Datos")(1, Application.Match("Ren Obs", ArrDeDatos, 0)) _
.Formula = "=" & RangoCambio(1, Application.Match("TLR", _
.Range("Titulos"), 0)).Address(RowAbsolute:=False) & _
"*" & .Range("Datos")(1, Application.Match("R ObsTLR", _
ArrDeDatos, 0)).Address(RowAbsolute:=False)
Else
.Range("Datos")(1, Application.Match("Ren Esp", ArrDeDatos, 0)) _
.Formula = "=0"
End If
If ConRendObs Then
If i < Periodo Or RangoRendsObs.Rows.Count = Periodo + 1 - 1 Then
Set RangoActRendObs = .Range(ArrDeHojasProy(i, _
Application.Match("Rend Observado", ArrDatosHojas, 0)) & _
"!Series")(Application.Match(ArrHojas(0).Range("Tiempo") _
(FechInicio + i - 1), Range(ArrDeHojasProy(i, _
Application.Match("Rend Observado", ArrDatosHojas, 0)) & _
"!Tiempo"), 0) + 1, 1).Resize(1, Range(ArrDeHojasProy(i, _
Application.Match("Rend Observado", _
ArrDatosHojas, 0)) & "!Titulos").Count)
If ConTLR Then If ConTLRCamb Then .Range("Datos")(1, _
Application.Match("R ObsTLR", ArrDeDatos, 0)).Value = _
RangoActRendObs.Cells(1, TLR)
Else
Set RangoActRendObs = .Range(.Name & "!OtrosDatos") _
(Application.Match("Rend Sig", ArrDeOtrosDatos, 0), ColInstr + 1). _
Resize(1, n)
End If
.Range("Datos")(1, Application.Match("Ren Obs", ArrDeDatos, 0)) _
.FormulaArray = .Range("Datos")(1, Application.Match("Ren Obs", _
ArrDeDatos, 0)).FormulaR1C1 & "+MMULT(" & RangoCambio(1, _
ColInstr + 1).Resize(1, n).Address(ReferenceStyle:=xlR1C1, _
RowAbsolute:=False, RelativeTo:=Range("Datos")(1, _
Application.Match("Ren Obs", ArrDeDatos, 0))) & _
",TRANSPOSE(" & RangoActRendObs.Address(ReferenceStyle:=xlR1C1, _
External:=True) & ")")
End If
Select Case Metodo
Case "Markowitz"
Set RangoCov = CreaMatCovCorr(Tipo:="Covarianzas", _
TipoDatos:="Rendimientos", RangoDeDatos:=ArrHojas(0) _
.Range("Series")(FechMuestra + i + PeriodoMas - 1, 1)) _
.Resize(Muestra, n), PreguntaHoja:=False, _
RangoDeInstr:=ArrHojas(0).Range("Titulos"), _
RangoDeFechas:=ArrHojas(0).Range("Tiempo") _
(FechMuestra + i - 1, 1).Resize(Muestra, 1), _
ArrDeInstrSel:=ArrTitulos)
.Select
If ConTLRCamb Then
.Range("Datos")(1, Application.Match("Ren Esp", ArrDeDatos, 0)) _
.FormulaArray = .Range("Datos")(1, Application.Match("Ren Esp", _
ArrDeDatos, 0)).FormulaR1C1 & "+MMULT(" & RangoCambio(1, _
ColInstr + 1).Resize(1, n).Address(RowAbsolute:=False, _
ReferenceStyle:=xlR1C1, RelativeTo:=RangoCambio(1, 1)) & _
",TRANSPOSE(" & RangoActRend.Address(ReferenceStyle:=xlR1C1, _
External:=True) & ")") ' , ReferenceStyle:=xlR1C1
Else
.Range("Datos")(1, Application.Match("Ren Esp", ArrDeDatos, 0)) _

```

```

        .FormulaArray = .Range("Datos")(1, Application.Match("Ren Esp", _
ArrDeDatos, 0)).FormulaR1C1 & "+MMULT(" & RangoCambio(1, _
ColInstr + 1).Resize(1, n).Address(RowAbsolute:=False, _
ReferenceStyle:=xlR1C1, RelativeTo:=RangoCambio(1, 1)) & _
",TRANSPOSE(" & RangoActRend.Address(ReferenceStyle:=xlR1C1, _
External:=True) & ")")
    End If
    .Range("Datos")(1, Application.Match("Var", ArrDeDatos, 0)) _
        .FormulaArray = "=MMult(MMult(" & RangoCambio(1, ColInstr + 1) _
        .Resize(1, n).Address(RowAbsolute:=False, ReferenceStyle:=xlR1C1, _
        RelativeTo:=RangoCambio(1, 1)) & "," & RangoCov _
        .Address(ReferenceStyle:=xlR1C1, External:=True) & ")," & _
        "TRANSPOSE(" & RangoCambio(1, ColInstr + 1).Resize(1, n) _
        .Address(RowAbsolute:=False, ReferenceStyle:=xlR1C1, _
        RelativeTo:=RangoCambio(1, 1)) & ")")
End Select
If ConLimConjuntos Then
    For k = 1 To UBound(ArrClases)
        .Range("Datos")(1, Application.Match("Restr" & k, ArrDeDatos, 0)) _
            .FormulaArray = "=MMult(" & RangoCambio(1, ColInstr + 1) _
            .Resize(1, n).Address(RowAbsolute:=False, RelativeTo:= _
            Range("Datos")(1, Application.Match("Restr" & k, ArrDeDatos, _
            0))) & ", Transpose(" & .Range("OtrosDatos") _
            (Application.Match(ArrClases(k), ArrDeOtrosDatos, 0), _
            ColInstr + 1).Resize(1, n).Address & ")")
        SolverAdd CellRef:=.Range("Datos")(1, Application:Match("Restr" & _
        k, ArrDeDatos, 0)), Relation:=3, _
        FormulaText:=Cdbl(ArrPosClases(1, k))
        If ArrPosClases(2, k) <> "" Then
            SolverAdd CellRef:=.Range("Datos")(1, Application.Match("Restr" & _
            k, ArrDeDatos, 0)), Relation:=1, _
            FormulaText:=Cdbl(ArrPosClases(2, k))
        End If
    Next k
End If
InstrFact = 0
For j = 1 To n
    Application.StatusBar = "Verificando cotización de instrumentos (" & _
    j & "/" & n & ") " & ArrHojas(0).Range("Titulos")(j) & "..."
    If ArrFilMaxMin(0, j) Then
        'Verificar cuando empieza a cotizar
        If Cotiza(j) > FechMuestra + i Then
            ArrFilMaxMin(0, j) = False
            If ArrFilMaxMin(2, j) <> "" Then
                SolverDelete CellRef:=RangoCambio(1, ColInstr + j), Relation:=3, _
                FormulaText:=.Range("OtrosDatos")(Application.Match("Lim Inf", _
                ArrDeOtrosDatos, 0), ColInstr + j).Address
            End If
            If ArrFilMaxMin(1, j) <> "" Then
                SolverDelete CellRef:=RangoCambio(1, ColInstr + j), Relation:=1, _
                FormulaText:=.Range("OtrosDatos")(Application.Match("Lim Sup", _
                ArrDeOtrosDatos, 0), ColInstr + j).Address
            End If
            SolverAdd CellRef:=RangoCambio(1, ColInstr + j), Relation:=2, _
            FormulaText:="0"
        Else
            InstrFact = InstrFact + 1
        End If
    End If
Next j
If TipoPort = "AutoLimites" And Metodo <> "Generar" Then
    MsgBox Prompt:="Crea Limites"
    GoTo TerminaOpt
End If
Application.Calculation = xlCalculationAutomatic
Select Case TipoPort
    Case "PMV", "Rend" ', "Frontera" Ya esta la funcion objetivo

```

```

        SolverOk SetCell:=.Range("Datos")(1, Application.Match("Var", _
        ArrDeDatos, 0)), MaxMinVal:=2, ValueOf:=0, _
        ByChange:=.Range("Series"), EngineDesc:=SolverEngine
    Case "Riesgo"
        Stop
        SolverOk SetCell:=.Range("Datos")(1, Application.Match("Ren Esp", _
        ArrDeDatos, 0)), MaxMinVal:=1, ValueOf:=0, _
        ByChange:=.Range("Series"), EngineDesc:=SolverEngine
    End Select
    Select Case TipoPort
        Case "Frontera", "Ptos Factibles"
            If Metodo = "Markowitz" Then
                CreaFrontera HojaPort, ArrFilMaxMin, RangoCambio, ColInstr, _
                ConTLR, RangoActRend, ArrDeDatos, Pts, ArrDeDatosForm, _
                ConFormulas, TipoPort ', ArrValClases:=ArrValClases, _
                ArrPosClases:=ArrPosClases
            End If
            SolverConfigurationExport
        Case Else
            SolverSolve UserFinish:=True
        End Select
    Select Case TipoPort
        Case "Riesgo"
            Select Case RiesMax - Range("Datos")(1, Application.Match("Var", _
            ArrDeDatos, 0))
                Case Is < -Precision
                    .Range("Datos")(1, Application.Match("Var", ArrDeDatos, 0)) _
                    .Font.ColorIndex = 3 'Rojo
                Case Is > Precision
                    .Range("Datos")(1, Application.Match("Var", ArrDeDatos, 0)) _
                    .Font.ColorIndex = 32 'Azul
                Case Else
                    .Range("Datos")(1, Application.Match("Var", ArrDeDatos, 0)) _
                    .Font.ColorIndex = xlAutomatic
            End Select
        Case "Rend"
            Select Case RendMin - Range("Datos")(1, Application.Match("Ren Esp", _
            ArrDeDatos, 0))
                Case Is > Precision
                    .Range("Datos")(1, Application.Match("Ren Esp", ArrDeDatos, 0)) _
                    .Font.ColorIndex = 3 'Rojo
                Case Is < -Precision
                    .Range("Datos")(1, Application.Match("Ren Esp", ArrDeDatos, 0)) _
                    .Font.ColorIndex = 32 'Azul
                Case Else
                    .Range("Datos")(1, Application.Match("Ren Esp", ArrDeDatos, 0)) _
                    .Font.ColorIndex = xlAutomatic
            End Select
        End Select
    End Select
TerminaOpt:
    If i > 1 And Not EndifHojas Then
        Rows(Range("Datos").Row + 1).Insert Shift:=xlDown
        If ConFormulas Then
            Rows(Range("Datos").Row).Copy Cells(Range("Datos").Row + 1, 1)
        Else
            Rows(RenPortIni).Copy
            .Cells(Range("Datos").Row + 1, 1).PasteSpecial Paste:=xlValues
        End If
        If IsArray(ArrDeDatosForm) Then
            For i_form = 1 To UBound(ArrDeDatosForm)
                Range("Datos")(1, Application.Match(ArrDeDatosForm(i_form), _
                ArrDeDatos, 0)).Copy Destination:=Range("Datos")(2, _
                Application.Match(ArrDeDatosForm(i_form), ArrDeDatos, 0))
            Next i_form
        End If
        If ConRendAcum Then
            .Range("Datos")(2, Application.Match("Acum Esp", ArrDeDatos, 0)) _

```



```

        .Formula = "=" & .Range("Datos")(1, Application.Match("Acum Esp", _
ArrDeDatos, 0)).Address(RowAbsolute:=False) & "*" & (1+ & _
.Range("Datos")(2, Application.Match("Ren Esp", ArrDeDatos, 0)) _
.Address(RowAbsolute:=False) & ")")
    If i < Periodo Then .Range("Datos")(2, Application.Match("Acum Esp", _
ArrDeDatos, 0)).Copy Destination:=.Range("Datos")(3, _
Application.Match("Acum Esp", ArrDeDatos, 0))
    If ConRendObs Then
        .Range("Datos")(2, Application.Match("Acum Obs", ArrDeDatos, 0)) _
        .Formula = "=" & .Range("Datos")(1, Application.Match( _
"Acum Obs", ArrDeDatos, 0)).Address(RowAbsolute:=False) & _
        "*" & (1+ & .Range("Datos")(2, Application.Match("Ren Obs", _
ArrDeDatos, 0)).Address(RowAbsolute:=False) & ")")
        If i < Periodo Then .Range("Datos")(2, Application.Match( _
"Acum Obs", ArrDeDatos, 0)).Copy Destination:=.Range("Datos") _
        (3, Application.Match("Acum Obs", ArrDeDatos, 0))
    End If
End If
Else
    If Not ConFormulas Then
        If IsArray(ArrDeDatosForm) Then
            For j_form = 1 To UBound(ArrDeDatos)
                For i_form = 1 To UBound(ArrDeDatosForm)
                    If ArrDeDatosForm(i_form) = ArrDeDatos(j_form) Then _
                        GoTo OtroDato
                Next i_form
                Range("Datos")(1, j_form).Copy
                Range("Datos")(1, j_form).PasteSpecial Paste:=xlValues
OtroDato:
                Next j_form
            Else
                .Cells(RenPortIni, 1).Resize(1, .Range("Series")(Range("Series") _
                .Columns.Count).Column).Copy
                .Cells(RenPortIni, 1).PasteSpecial Paste:=xlValues
            End If
        End If
    End If
    If ConRendObs Then
        If Not (i < Periodo Or RangoRendsObs.Rows.Count = Periodo) Then
            .Range("Datos")(1, Application.Match("Ren Obs", ArrDeDatos, 0)) _
            .FormulaArray = .Range("Datos")(1, Application.Match("Ren Obs", _
ArrDeDatos, 0)).FormulaR1C1 & "+MMult(" & RangoCambio(1, _
ColInstr + 1).Resize(1, n).Address(ReferenceStyle:=xlR1C1, _
RowAbsolute:=False, RelativeTo:=.Range("Datos")(1, _
Application.Match("Ren Obs", ArrDeDatos, 0))) & _
            ",TRANSPOSE(" & RangoActRendObs.Address(ReferenceStyle:=xlR1C1, _
            External:=True) & ")")
        End If
    End If
    If Metodo = "Markowitz" Then
        If Not ConFormulas And BorrarHojas Then
            Application.DisplayAlerts = False
            RangoCov.Parent.Delete
            Application.DisplayAlerts = True
        End If
    End If
Next i
If l < NoDeHojas Then
    ArrDeValInf(Application.Match("Tiempo", ArrDeInf, 0)) = _
    ArrHojas(0).Range("Tiempo")(FechInicio + 1, 1)
    PeriodoMas = PeriodoMas + 1
    SaltaPregBorr = True
End If
End With
TerminoOptHoja:
Next l
Application.Calculation = FormaDeCalcular

```

```

If Not ConFormulas And BorrarHojas Then
    Application.DisplayAlerts = False
    RangoVars.Parent.Delete
    RangoRends.Parent.Delete
    Application.DisplayAlerts = True
End If

End Sub

```

```

Sub CreaFrontera(Hoja As Worksheet, ArrFilMaxMin, _
    SelCambio, ColInstr, ConTLR, RangoActRend, ArrDeDatos, Pts, ArrDeDatosForm, _
    ConFormulas As Boolean, Tipo As String, Optional ArrValClases, _
    Optional ArrPosClases, Optional MaxRiesgo, Optional RiesgoMayor)

Dim InicioArr As Integer
Dim RendMayor As Double
Dim RendPMV As Double
Dim RendMenor As Double
Dim RendMenorSolver As Double
Dim NumCond As Integer
Dim incr As Double
Dim k As Integer
Dim i As Integer
Dim ArrMaxRend
Dim ArrPMV
Dim ArrMinRend
Dim ArrCondicionesSolver
Dim l As Integer
Dim m As Integer
Dim ActivoRendMenor As Integer
Dim ActivoRendMayor As Integer

If UBound(ArrFilMaxMin, 1) = 4 Then
    InicioArr = 3
Else
    InicioArr = 1
End If

With Hoja
    .Range("Datos")(1, Application.Match("Parte", ArrDeDatos, 0)).Value = _
        "PMaxRend"
    For k = 1 To UBound(ArrFilMaxMin, 2)
        If ArrFilMaxMin(0, k) Then
            If ArrFilMaxMin(InicioArr + 1, k) <> "" Then
                SelCambio.Cells(1, ColInstr + k) = ArrFilMaxMin(InicioArr + 1, k)
            Else
                SelCambio.Cells(1, ColInstr + k) = _
                    Application.Min(ArrFilMaxMin(InicioArr, k), -99)
            End If
        Else
            SelCambio.Cells(1, ColInstr + k) = 0
        End If
    Next k
    If ConTLR Then .Range("Series")(1).Value = 0
    k = 0
    Do
        k = k + 1
        If ArrFilMaxMin(0, Application.Match(Application.Large(RangoActRend, k), _
            RangoActRend, 0)) Then
            If ArrFilMaxMin(InicioArr, Application.Match( _
                Application.Large(RangoActRend, k), RangoActRend, 0)) = "" Then

```

```

SelCambio.Cells(1, ColInstr + Application.Match(
Application.Large(RangoActRend, k), RangoActRend, 0)) = _
1 - Application.Sum(SelCambio)
Else
SelCambio.Cells(1, ColInstr + Application.Match(
Application.Large(RangoActRend, k), RangoActRend, 0)) = _
Application.Min(1 - Application.Sum(SelCambio) + _
SelCambio.Cells(1, ColInstr + Application.Match(
Application.Large(RangoActRend, k), RangoActRend, 0)), _
ArrFilMaxMin(InicioArr, Application.Match(Application.Large(
RangoActRend, k), RangoActRend, 0)))
End If
End If
Loop Until Application.Sum(SelCambio) = 1 Or k = RangoActRend.Count
ReDim ArrMaxRend(SelCambio.Count) As Double
For i = 1 To SelCambio.Count
ArrMaxRend(i) = SelCambio(i).Value
Next i
If Not IsMissing(MaxRiesgo) Then MaxRiesgo = .Range("Datos")(1, _
Application.Match("Var", ArrDeDatos, 0))
RendMayor = .Range("Datos")(1, Application.Match("Ren Esp", ArrDeDatos, 0))
.Range("Datos")(1, Application.Match("Parte", ArrDeDatos, 0)).Value = "PMV"
SolverOk SetCell:=.Range("Datos")(1, Application.Match("Var", ArrDeDatos, _
0)), MaxMinVal:=2, ValueOf:="0", ByChange:=.Range("Series"), _
EngineDesc:=SolverEngine
SolverSolve UserFinish:=True
RendPMV = Application.Round(.Range("Datos")(1, Application.Match("Ren Esp", _
ArrDeDatos, 0)), 18)
ReDim ArrPMV(SelCambio.Count) As Double
For i = 1 To SelCambio.Count
ArrPMV(i) = SelCambio(i).Value
Next i

If Tipo = "Frontera" Then
RendMenor = RendPMV
Else ' "Ptos Factibles"
For k = 1 To UBound(ArrFilMaxMin, 2)
If ArrFilMaxMin(0, k) Then
If ArrFilMaxMin(InicioArr + 1, k) <> "" Then
SelCambio.Cells(1, ColInstr + k) = ArrFilMaxMin(InicioArr + 1, k)
Else
SelCambio.Cells(1, ColInstr + k) = _
Application.Max(ArrFilMaxMin(InicioArr, k), -99)
End If
Else
SelCambio.Cells(1, ColInstr + k) = 0
End If
Next k
k = 0
Do
k = k + 1
If ArrFilMaxMin(0, Application.Match(Application.Large(RangoActRend, k), _
RangoActRend, 0)) Then
If ArrFilMaxMin(InicioArr, Application.Match(
Application.Large(RangoActRend, k), RangoActRend, 0)) = "" Then
SelCambio.Cells(1, ColInstr + Application.Match(
Application.Small(RangoActRend, k), RangoActRend, 0)) = _
1 - Application.Sum(SelCambio)
Else
SelCambio.Cells(1, ColInstr + Application.Match(
Application.Small(RangoActRend, k), RangoActRend, 0)) = _
Application.Min(1 - Application.Sum(SelCambio) + _
SelCambio.Cells(1, ColInstr + Application.Match(
Application.Large(RangoActRend, k), RangoActRend, 0)), _
ArrFilMaxMin(InicioArr, Application.Match(
Application.Large(RangoActRend, k), RangoActRend, 0)))

```

```

.Range("Datos")(1, Application.Match("Parte", ArrDeDatos, 0)).Value = _
.Range("Titulos").Cells(Application.Match(Application.Small(_
RangoActRend, k), RangoActRend, 0)).Value
CopiaValoresFrontera Hoja, ArrDeDatos, ArrDeDatosForm, ConFormulas
Next k
End If
.Range("Datos")(1, Application.Match("Parte", ArrDeDatos, 0)).Value = _
"PMaxRend"
.Range("Datos")(1, Application.Match("Punto", ArrDeDatos, 0)).Value = _
"PMaxRend"
For i = 1 To SelCambio.Count
SelCambio(i).Value = ArrMaxRend(i)
Next i
CopiaValoresFrontera Hoja, ArrDeDatos, ArrDeDatosForm, ConFormulas
.Range("Datos")(1, Application.Match("Parte", ArrDeDatos, 0)).Value = _
"PMV"
.Range("Datos")(1, Application.Match("Punto", ArrDeDatos, 0)).Value = _
"PMV"
For i = 1 To SelCambio.Count
SelCambio(i).Value = ArrPMV(i)
Next i
CopiaValoresFrontera Hoja, ArrDeDatos, ArrDeDatosForm, ConFormulas
.Range("Datos")(1, Application.Match("Parte", ArrDeDatos, 0)).Value = _
"PMinRend"
.Range("Datos")(1, Application.Match("Punto", ArrDeDatos, 0)).Value = _
"PMinRend"
For i = 1 To SelCambio.Count
SelCambio(i).Value = ArrMinRend(i)
Next i
CopiaValoresFrontera Hoja, ArrDeDatos, ArrDeDatosForm, ConFormulas
.Rows(.Range("Datos").Row + 1).Insert Shift:=xlDown
If Not IsMissing(MaxRiesgo) Then
.Cells(.Range("Datos").Row + 1, 1).Resize(4).Rows.Group
Exit Sub
End If
.Cells(.Range("Datos").Row + 1, 1).Resize(RangoActRend.Count + 4).Rows.Group
.Range("Datos")(1, Application.Match("Parte", ArrDeDatos, 0)).Value = _
"PMinRend"
For i = 1 To SelCambio.Count
SelCambio(i).Value = ArrMinRend(i)
Next i
Select Case .Range("Datos")(1, Application.Match("Ren Esp", ArrDeDatos, 0)) _
- RendPMV
Case Is > 0.0000001
.Range("Datos")(1, Application.Match("Parte", ArrDeDatos, 0)).Value = _
"Efic"
Case Is < -0.0000001
.Range("Datos")(1, Application.Match("Parte", ArrDeDatos, 0)).Value = _
"Inf"
Case Else
.Range("Datos")(1, Application.Match("Parte", ArrDeDatos, 0)).Value = _
"PMV"
End Select
incr = (RendMayor - RendMenor) / (Pts - 1)
.Range("Datos")(1, Application.Match("Punto", ArrDeDatos, 0)).Value = 1
For k = 2 To Pts
CopiaValoresFrontera Hoja, ArrDeDatos, ArrDeDatosForm, ConFormulas
If RendMenor < RendPMV And RendMenor + incr > RendPMV Then
.Range("Datos")(1, Application.Match("Parte", ArrDeDatos, 0)).Value = _
"PMV"
For i = 1 To SelCambio.Count
SelCambio(i).Value = ArrPMV(i)
Next i
.Range("Datos")(1, Application.Match("Punto", ArrDeDatos, 0)).Value = _
"PMV"
.Range("Datos")(1, Application.Match("Parte", ArrDeDatos, 0)).Value = _
"PMV"

```

```

CopiaValoresFrontera Hoja, ArrDeDatos, ArrDeDatosForm, ConFormulas
.Cells(.Range("Datos").Row + 1, 1).Rows.Group
End If
.Range("Datos")(1, Application.Match("Punto", ArrDeDatos, 0)).Value = k
RendMenor = Application.Round(RendMenor + incr, 18)
RendMenorSolver = Round(RendMenor, 8)
SolverAdd CellRef:=.Range("Datos")(1, Application.Match("Ren Esp", _
ArrDeDatos, 0)), Relation:=2, FormulaText:=CStr(RendMenorSolver)
SolverSolve UserFinish:=True
If .Range("Datos")(1, Application.Match("Ren Esp", ArrDeDatos, 0)) _
> RendPMV Then
    .Range("Datos")(1, Application.Match("Parte", ArrDeDatos, 0)).Value = _
    "Efic"
Else
    .Range("Datos")(1, Application.Match("Parte", ArrDeDatos, 0)).Value = _
    "Inf"
End If
'TypeEnum:=8 Parte derecha de las condiciones
ArrCondicionesSolver = solverget(TypeNum:=8)
NumCond = UBound(ArrCondicionesSolver) + 1
'TypeEnum:=5 Numero de restricciones
SolverDelete CellRef:=.Range("Datos")(1, Application.Match("Ren Esp", _
ArrDeDatos, 0)), Relation:=2, FormulaText:=CStr(RendMenorSolver)
ArrCondicionesSolver = solverget(TypeNum:=8)
If NumCond = UBound(ArrCondicionesSolver) + 1 Then
    MsgBox "La condicion de rendimiento >= no se borro", vbExclamation
    Stop
End If
Next k
If Tipo = "Ptos Factibles" Then
    For i = 1 To .Range("Series").Count
        SolverAdd CellRef:=.Range("Series")(1, i), Relation:=1, _
        FormulaText:=.Range("OtrosDatos")(Application.Match("Lim Sup", _
        .Range("OtrosDatos")(1, 0).Resize(_
        .Range("OtrosDatos").Rows.Count, 1), 0), i).Address
    Next i
    For l = 1 To .Range("Series").Count
        For m = l + 1 To .Range("Series").Count
            CopiaValoresFrontera Hoja, ArrDeDatos, ArrDeDatosForm, ConFormulas
            .Rows(.Range("Datos").Row + 1).Insert Shift:=xlDown

            .Range("OtrosDatos")(3, 1).Resize(1, _
            .Range("OtrosDatos").Columns.Count) = "0"
            .Range("OtrosDatos")(3, 1) = "1"
            .Range("OtrosDatos")(3, m) = "1"

            SolverSolve UserFinish:=True
            RendPMV = .Range("Datos")(1, Application.Match("Ren Esp", _
            ArrDeDatos, 0)).Value
            For i = 1 To SelCambio.Count
                ArrPMV(i) = SelCambio(i).Value
            Next i

            .Range("Series") = "0"
            .Range("Series")(1) = 1
            RendMenor = .Range("Datos")(1, Application.Match("Ren Esp", _
            ArrDeDatos, 0)).Value
            .Range("Series") = "0"
            .Range("Series")(m) = 1
            If RendMenor > .Range("Datos")(1, Application.Match("Ren Esp", _
            ArrDeDatos, 0)).Value Then
                ActivoRendMenor = m
                ActivoRendMayor = l
                RendMayor = RendMenor
                RendMenor = .Range("Datos")(1, Application.Match("Ren Esp", _
                ArrDeDatos, 0)).Value
            Else

```

```

ActivoRendMenor = 1
ActivoRendMayor = m
RendMayor = .Range("Datos")(1, Application.Match("Ren Esp", _
ArrDeDatos, 0)).Value
.Range("Series") = "0"
.Range("Series")(ActivoRendMenor) = 1
End If

incr = (RendMayor - RendMenor) / (Pts - 1)
.Range("Datos")(1, Application.Match("Punto", ArrDeDatos, 0)).Value = 1
.Range("Datos")(1, Application.Match("Parte", ArrDeDatos, 0)).Value = _
"Inf" & 1 & ", " & m

For k = 2 To Pts
CopiaValoresFrontera Hoja, ArrDeDatos, ArrDeDatosForm, ConFormulas
If RendMenor < RendPMV And RendMenor + incr > RendPMV Then
.Range("Datos")(1, Application.Match(
"Punto", ArrDeDatos, 0)).Value = "PMV" & 1 & ", " & m
.Range("Datos")(1, Application.Match(
"Parte", ArrDeDatos, 0)).Value = "PMV" & 1 & ", " & m
For i = 1 To SelCambio.Count
SelCambio(i).Value = ArrPMV(i)
Next i
CopiaValoresFrontera Hoja, ArrDeDatos, ArrDeDatosForm, ConFormulas
.Cells(.Range("Datos").Row + 1, 1).Rows.Group
.Range("Series")(ActivoRendMenor) =
.Range("Series")(ActivoRendMenor).Offset(2) - 1 / (Pts - 1)
.Range("Series")(ActivoRendMayor) =
.Range("Series")(ActivoRendMayor).Offset(2) + 1 / (Pts - 1)
Else
.Range("Series")(ActivoRendMenor) =
.Range("Series")(ActivoRendMenor) - 1 / (Pts - 1)
.Range("Series")(ActivoRendMayor) =
.Range("Series")(ActivoRendMayor) + 1 / (Pts - 1)
End If
.Range("Datos")(1, Application.Match(
"Punto", ArrDeDatos, 0)).Value = k
RendMenor = .Range("Datos")(1,
Application.Match("Ren Esp", ArrDeDatos, 0))
If RendMenor > RendPMV Then
.Range("Datos")(1, Application.Match(
"Parte", ArrDeDatos, 0)).Value = "Efic" & 1 & ", " & m
Else
.Range("Datos")(1, Application.Match(
"Parte", ArrDeDatos, 0)).Value = "Inf" & 1 & ", " & m
End If
Next k
Next m
Next l
End If
End With

End Sub

```

```

Function ObtienePosiciones _
(TipoPosicion As String, ArrTitulos) As Boolean

Dim i As Integer

With frm_Pos
.UserCancel = False
.Show

```

```

If .UserCancel Then Exit Function
Select Case True
Case .OptLargas
    TipoPosicion = "Largas"
    For i = 1 To UBound(ArrFilMaxMin, 2)
        ArrFilMaxMin(2, i) = 0 'Lim Inf
    Next i
Case .OptCortas
    TipoPosicion = "Cortas"
    For i = 1 To UBound(ArrFilMaxMin, 2)
        ArrFilMaxMin(1, i) = 0 'Lim Sup
    Next i
Case .OptLargasCortas
    TipoPosicion = "Largas y Cortas"
Case .OptMaxMin
    TipoPosicion = "Max y Min"
    For i = 1 To UBound(ArrFilMaxMin, 2)
        If .ChkMax Then
            ArrFilMaxMin(1, i) = .TxtMax 'Lim Sup
        End If
        If .ChkMin Then
            ArrFilMaxMin(2, i) = .TxtMin 'Lim Inf
        End If
    Next i
    If .ChkIndividuales Then
        With frm_PosInd
            .LstInstr.Clear
            For i = 0 To UBound(ArrTitulos)
                .LstInstr.AddItem ArrTitulos(i)
            Next i
            .Show
        End With
    End If
Case Else
    Exit Function
End Select
End With
ObtienePosiciones = True

End Function

```

```

Function EmpiezaTerminaCotizar _
    (RangoDeTitulos, RangoDeDatos)

Dim Cotizacion
Dim i As Integer
Dim j As Integer

ReDim Cotizacion(RangoDeTitulos.Count)
For j = 1 To UBound(Cotizacion)
    Application.StatusBar = "Verificando Cotización " & _
        RangoDeTitulos(j) & " (" & j & "/" & RangoDeTitulos.Count & ")"
    i = 0
    Do
        i = i + 1
    Loop Until (IsNumeric(RangoDeDatos.Cells(i, j)) And _
        Not IsEmpty(RangoDeDatos.Cells(i, j))) Or i > RangoDeDatos.Rows.Count
    Cotizacion(j) = i
Next j
EmpiezaTerminaCotizar = Cotizacion
Application.StatusBar = False

```

```
End Function
```

```
Sub CopiaValoresFrontera(Hoja As Worksheet, _  
    ArrDeDatos, ArrDeDatosForm, ConFormulas) _  
  
Dim i_form As Integer  
  
With Hoja  
    .Rows(.Range("Datos").Row + 1).Insert Shift:=xlDown  
    If ConFormulas Then  
        .Rows(.Range("Datos").Row).Copy .Cells(.Range("Datos").Row + 1, 1)  
    Else  
        .Rows(.Range("Datos").Row).Copy  
        .Cells(.Range("Datos").Row + 1, 1).PasteSpecial Paste:=xlValues  
        If IsArray(ArrDeDatosForm) Then  
            For i_form = 1 To UBound(ArrDeDatosForm)  
                .Range("Datos")(1, Application.Match(ArrDeDatosForm(i_form), _  
                    ArrDeDatos, 0)).Copy Destination:=.Range("Datos")  
                    (2, Application.Match(ArrDeDatosForm(i_form), ArrDeDatos, 0))  
            Next i_form  
        End If  
    End If  
End With  
  
End Sub
```

Idioma

```
Sub CambiaIdioma()  
  
Dim ArrIdioma  
  
ArrIdioma = Array(IdiomaEspanol, IdiomaIngles)  
Idioma = TomaIdioma  
With frmSelect  
    Select Case Idioma  
        Case IdiomaEspanol  
            .Caption = "Cambiar Idioma"  
            .LblSelect.Caption = "Idioma"  
            .BtnOk.Caption = "Seleccionar"  
            .BtnCancel.Caption = "Cancelar"  
        Case IdiomaIngles  
            .Caption = "Change Language"  
            .LblSelect.Caption = "Language"  
            .BtnOk.Caption = "Select"  
            .BtnCancel.Caption = "Cancel"  
    End Select  
    If ActiveWorkbook Is Nothing Then  
        .Caption = .Caption & " General"  
    Else  
        .Caption = .Caption & " [" & ActiveWorkbook.Name & "]"  
    End If  
    .LstLists.List() = ArrIdioma  
    .LstLists.MultiSelect = fmMultiSelectSingle
```



```

.LstLists.Value = Idioma
.UserCancel = False
.Show
If .UserCancel Or Idioma = .LstLists.Value Then Exit Sub
Idioma = .LstLists.Value
End With
Application.ScreenUpdating = False

CambiaPref "Idioma", Idioma

MenuBar(xlWorksheet).Menus(NombreMenu).Delete
CreaMenu
Application.ScreenUpdating = True

End Sub

```

```

Global Idioma As String
Global Const IdiomaEspanol = "Español"
Global Const IdiomaIngles = "English"

Function RegresaComentEnIdioma _
    (Comentario As String, NuevoComentario As String) As Boolean

Dim DB_Translate As New ADODB.Connection
Dim rst As New ADODB.Recordset

Dim SQLstr As String

Dim ComentarioIngles As String

Idioma = TomaIdioma

If Idioma = IdiomaEspanol Then
    NuevoComentario = Comentario
    RegresaComentEnIdioma = True
Else
    If Dir(ThisWorkbook.Path & "\Translate.mdb") = "" Then
        MsgBox "No existe el archivo:" & String(2, Chr(13)) & _
            ThisWorkbook.Path & "\Translate.mdb", vbCritical, TituloSistema
        Exit Function
    End If
    DB_Translate.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0; " & _
        "Data Source=" & ThisWorkbook.Path & "\Translate.mdb"
    DB_Translate.Open

    SQLstr = "SELECT Comment" & Idioma & " FROM [Translation] " & _
        "WHERE CommentSpanish = "" & Comentario & """"
    rst.Open SQLstr, DB_Translate, adOpenStatic

    If rst.EOF Then
        Select Case Idioma
            Case IdiomaEspanol
                MsgBox "Comentario no identificado" & Chr(13) & Comentario, _
                    vbCritical, TituloSistema
            Case IdiomaIngles
                MsgBox "Comment not identified" & Chr(13) & Comentario, _
                    vbCritical, TituloSistema
        End Select
        NuevoComentario = ""
    Else
        NuevoComentario = rst("Comment" & Idioma)
    End If
End Function

```

```

End If
If NuevoComentario = "" Then
    ComentarioIngles = InputBox("Cual es el comentario en Ingles de:" & _
        String(2, Chr(13)) & Comentario, TituloSistema, Comentario)
    If ComentarioIngles <> "" Then
        With frm_Select
            .Caption = "Comentario"
            .LblSelect.Caption = "Cual es el modulo del comentario"
            .BtnOk.Caption = "Continuar"
            .BtnCancel.Caption = "Cancelar"
            .LstLists.MultiSelect = fmMultiSelectSingle
            .LstLists.Clear
            .LstLists.ColumnCount = 1
            .LstLists.ColumnWidths = "180 pt"
            rst.Close
            SQLstr = "SELECT * FROM Procedures WHERE NOT RevTranslation"
            rst.Open SQLstr, DB_Translate, adOpenStatic
            Do Until rst.EOF
                .LstLists.AddItem rst("Procedure")
                rst.MoveNext
            Loop
            '.LstLists.List = ArrLibros
            .LstLists.ListIndex = 0
            .UserCancel = False
            .Show
            If .UserCancel Then Exit Function
            SQLstr = "INSERT INTO [Translation] SELECT "" & Comentario & _
                "" AS CommentSpanish, "" & ComentarioIngles & _
                "" AS CommentEnglish, "" & .LstLists.Value & "" AS Procedure1"
        End With

        DB_Translate.Execute SQLstr
        NuevoComentario = ComentarioIngles

        RegresaComentEnIdioma = True
    Else
        RegresaComentEnIdioma = False
    End If
Else
    RegresaComentEnIdioma = True
End If
rst.Close
End If
End Function

```

```

Function TomaIdioma(Optional NombreLibro) As String

    If Idioma = "" Then
        If Not TomaPrefIni(NombrePref:="Idioma", ValorPref:=Idioma) Then _
            Exit Function
        End If
    If Idioma <> IdiomaEspanol And Idioma <> IdiomaIngles Then
        MsgBox "El idioma "" & Idioma & "" no es válido." & String(2, Chr(13)) _
            & "Se va a cambiar a "" & IdiomaEspanol & _
            """. Si desea, podrá cambiarlo en el menú ""Idioma"", _
            vbExclamation, TituloSistema
        CambiaPref "Idioma", IdiomaEspanol
        Idioma = IdiomaEspanol
    End If
    TomaIdioma = Idioma

```

```
End Function
```

Salida

```
Sub SalirSistema()  
  
Auto_Close  
If ThisWorkbook.IsAddin Then  
    AddIns("Portprog").Installed = False  
Else  
    ThisWorkbook.Close  
End If  
  
End Sub
```

```
Sub Auto_Close()  
  
    MenuBars(xlWorksheet).Menus(NombreMenu).Delete  
  
End Sub
```

Rutinas generales

```
Sub SeleccionaBD ()  
  
Dim Comentario As String  
Dim Nombre_BD As String  
  
If Not RegresaComentEnIdioma("Seleccionar Base de Datos SAP preestablecida", _  
    Comentario) Then Exit Sub  
If Not TomaBD(Nombre_BD:=Nombre_BD, Mensaje:=Comentario) Then Exit Sub  
CambiaPref "DB_Name", Nombre_BD  
  
End Sub
```

```
Global Const vbSAPBD = 1  
Global Const vbSAPData = 2  
Global Const vbSAPPort = 3  
Global Const dir_DB = "DBase"  
Global Const dir_Data = "Data"  
Global Const dir_Port = "Portfolios"
```

```

Function RegresaPath(Directorio As Integer, _
    DirPath As String) As Boolean

Dim Comentario As String

Select Case Directorio
Case vbSAPBD
    DirPath = dir_DB
Case vbSAPData
    DirPath = dir_Data
Case vbSAPPort
    DirPath = dir_Port
Case Else
    MsgBox "Path no especificado", vbExclamation, TituloSistema
    Exit Function
End Select

DirPath = ThisWorkbook.Path & "\" & DirPath
If Dir(DirPath, vbDirectory) = "" Then
    If Not RegresaComentEnIdioma("Path no encontrado", Comentario) Then _
        Exit Function
    MsgBox Comentario & String(2, Chr(13)) & DirPath, _
        vbExclamation, TituloSistema
    Exit Function
End If

RegresaPath = True

End Function

```

```

Function TomaPrefIni _
    (NombrePref As String, ValorPref) As Boolean

Dim Pref As String

If Dir(ThisWorkbook.Path & "\" & ArchivoIni) = "" Then CreaArchivoIni
Open ThisWorkbook.Path & "\" & ArchivoIni For Input As #1
Do While Not EOF(1)
    Input #1, Pref, ValorPref
    If Pref = NombrePref Then
        TomaPrefIni = True
        Exit Do
    End If
Loop
Close #1

End Function

```

```

Sub CreaArchivoIni()

Open ThisWorkbook.Path & "\" & ArchivoIni For Output As #1
Print #1, "Idioma, " & IdiomaEspanol
Print #1, "DB_Name, BDSAP.mdb"
Close #1

```

```
End Sub
```

```
Sub CambiaPref(NombrePref As String, ValorPref)  
  
Dim PathArchivoIni As String  
Dim PathArchivoTmp As String  
Dim Pref As String  
Dim Valor  
  
PathArchivoIni = ThisWorkbook.Path & "\" & ArchivoIni  
PathArchivoTmp = ThisWorkbook.Path & "\" & _  
    Left(ArchivoIni, Len(ArchivoIni) - 4) & ".tmp"  
Open PathArchivoIni For Input As #1  
Open PathArchivoTmp For Output As #2  
Do While Not EOF(1)  
    Input #1, Pref, Valor  
    If Pref = NombrePref Then  
        Valor = ValorPref  
    End If  
    Print #2, Pref & ", " & Valor  
Loop  
Close #2  
Close #1  
Kill PathArchivoIni  
Name PathArchivoTmp As PathArchivoIni  
  
End Sub
```

```
Function RegresaPreferencia _  
    (NombrePref As String, ValorPref, Optional Hoja, Optional Libro, _  
    Optional DespliegaComentario) As Boolean  
  
Dim Nombre As Name  
  
If IsMissing(DespliegaComentario) Then DespliegaComentario = True  
If IsMissing(Hoja) Then  
    If IsMissing(Libro) Then  
        If ActiveWorkbook Is Nothing Then  
            If TomaPrefIni(NombrePref, ValorPref) Then  
                RegresaPreferencia = True  
                Exit Function  
            End If  
        Else  
            Set Libro = ActiveWorkbook  
        End If  
    End If  
    For Each Nombre In Libro.Names  
        If Nombre.Name = NombrePref Then  
            ValorPref = RegresaValorDePref(Nombre.Value)  
            RegresaPreferencia = True  
            Exit Function  
        End If  
    Next Nombre  
    If TomaPrefIni(NombrePref, ValorPref) Then  
        RegresaPreferencia = True  
    End If
```

```

Else
  For Each Nombre In Hoja.Names
    If Right(Nombre.Name, Len(Nombre.Name) - InStr(Nombre.Name, "!")) = _
      NombrePref Then
      If InStr(Nombre.Value, "=") > 0 Then
        If InStr(Nombre.Value, "\"") > 0 Then
          ValorPref = Mid(Nombre.Value, 3, Len(Nombre.Value) - 3)
        Else
          ValorPref = Right(Nombre.Value, Len(Nombre.Value) - 1)
        End If
      End If
      RegresaPreferencia = True
      Exit Function
    End If
  Next Nombre

End If
If Not RegresaPreferencia And DespliegaComentario Then
  Stop
  MsgBox "Falta la preferencia " & NombrePref, vbExclamation, TituloSistema
End If

End Function

```

```

Function RegresaValorDePref(Valor As String)

If InStr(Valor, "=") > 0 Then
  If InStr(Valor, "\"") > 0 Then
    RegresaValorDePref = Mid(Valor, 3, Len(Valor) - 3)
  Else
    RegresaValorDePref = Right(Valor, Len(Valor) - 1)
  End If
End If

End Function

```

```

Function TomaTxt(ArrTxt() As String) As Boolean

Dim i As Integer
Dim cont_sel As Integer
Dim Comentario As String

With frm_Select
  If Not RegresaComentEnIdioma("Carga de Series de Archivos Txt", Comentario) _
    Then Exit Function
  .Caption = Comentario
  If Not RegresaComentEnIdioma("Selecciona los archivos de texto a cargar", _
    Comentario) Then Exit Function
  .LblSelect.Caption = Comentario
  If Not RegresaComentEnIdioma("Seleccionar", Comentario) Then Exit Function
  .BtnOk.Caption = Comentario
  If Not RegresaComentEnIdioma("Cancelar", Comentario) Then Exit Function
  .BtnCancel.Caption = Comentario

  .LstLists.Clear
  .LstLists.ColumnCount = 1

```

```

.LstLists.ColumnWidths = "180 pt"
.LstLists.MultiSelect = fmMultiSelectExtended

If ListaArchivos(TipoArchivo:=vbSAPData, ArrArchivos:=ArrTxt) Then
    .BtnOk.Enabled = True
Else
    .BtnOk.Enabled = False
End If

.LstLists.List() = ArrTxt

For i = 0 To .LstLists.ListCount - 1
    .LstLists.Selected(i) = True
Next i
.UserCancel = False
.Show
If Not .UserCancel Then
    cont_sel = -1
    For i = 0 To .LstLists.ListCount - 1
        If .LstLists.Selected(i) Then
            TomaTxt = True
            cont_sel = cont_sel + 1
            ArrTxt(cont_sel) = .LstLists.List(i)
        End If
    Next i
    ReDim Preserve ArrTxt(cont_sel)
End If
End With
Application.Cursor = xlDefault
End Function

```

```

Function ListaArchivos(TipoArchivo As Integer, _
    ArrArchivos() As String) As Boolean

Dim DirFiles As String
Dim FileExt As String
Dim i As Integer
Dim FileName As String

If Not RegresaPath(Directorio:=TipoArchivo, DirPath:=DirFiles) Then _
    Exit Function

Select Case TipoArchivo
    Case vbSAPBD
        FileExt = "*.mdb"
    Case vbSAPData
        FileExt = "*.Txt"
    Case Else
        Stop
End Select

With Application.FileSearch
    .NewSearch
    .LookIn = DirFiles
    .FileName = FileExt
    If .Execute() > 0 Then
        ReDim ArrArchivos(.FoundFiles.Count - 1) As String
        ListaArchivos = True
        For i = 1 To .FoundFiles.Count
            FileName = Dir(.FoundFiles(i))
            ArrArchivos(i - 1) = FileName
        Next i
    End If
End With

```

```

Next i
Else
ReDim ArrArchivos(0) As String
End If
End With

End Function

```

```

Function TomaBD_Default _
(Nombre_BD As String) As Boolean

If Not RegresaPreferencia(NombrePref:="DB_Name", ValorPref:=Nombre_BD) Then
If Not RegresaPreferencia(NombrePref:="DB_Name", ValorPref:=Nombre_BD, _
Libro:=ThisWorkbook) Then Exit Function
End If
TomaBD_Default = True

End Function

```

```

Function TomaBD (Nombre_BD As String, _
Mensaje As String, Optional NombreBoton) As Boolean

Dim ArrBD() As String
Dim i As Integer
Dim Comentario As String
Dim Comentario1 As String

If Not ListaArchivos(TipoArchivo:=vbSAPBD, ArrArchivos:=ArrBD) Then
If Not RegresaComentEnIdioma("No existen Bases de Datos. " & _
"Para continuar cree una.", Comentario) Then Exit Function
MsgBox Prompt:=Comentario, Buttons:=vbExclamation, Title:=TituloSistema
Exit Function
End If

With frm_Select
If Not RegresaComentEnIdioma("Base de Datos SAP", Comentario) Then _
Exit Function
.Caption = Comentario
If IsMissing(Mensaje) Then
If Not RegresaComentEnIdioma("Bases de Datos", Comentario) Then _
Exit Function
Else
Comentario = Mensaje
End If
.LblSelect.Caption = Comentario
If IsMissing(NombreBoton) Then
If Not RegresaComentEnIdioma("Seleccionar", Comentario) Then Exit Function
Else
Comentario = NombreBoton
End If
.BtnOk.Caption = Comentario
If Not RegresaComentEnIdioma("Cancelar", Comentario) Then Exit Function
.BtnCancel.Caption = Comentario

.LstLists.Clear
.LstLists.MultiSelect = fmMultiSelectSingle

```



```

.LstLists.ColumnCount = 1
.LstLists.ColumnWidths = "180 pt"
.LstLists.List() = ArrBD
If Not RegresaPreferencia(NombrePref:="DB_Name", ValorPref:=Nombre_BD) Then
    If Not RegresaPreferencia(NombrePref:="DB_Name", _
        ValorPref:=Nombre_BD, Libro:=ThisWorkbook) Then Exit Function
End If
Do Until i = UBound(ArrBD) Or ArrBD(i) = Nombre_BD
    i = i + 1
Loop
If ArrBD(i) <> Nombre_BD Then
    If Not RegresaComentEnIdioma("La base de datos predeterminada " & _
        "no fue encontrada", Comentario) Then Exit Function
    If Not RegresaComentEnIdioma("Escoja otra de la lista", Comentario1) Then _
        Exit Function
    MsgBox Comentario & ": " & Nombre_BD & String(2, Chr(13)) & Comentario1, _
        vbExclamation, TituloSistema
Else
    .LstLists.ListIndex = i
End If
.UserCancel = False
.Show
If .UserCancel Then Exit Function
Nombre_BD = .LstLists.Value
End With
TomaBD = True
End Function

```

```

Function GeneraStrDeArreglo(Arreglo, Dimension As Integer) As String

Dim i As Integer
Dim str As String
For i = LBound(Arreglo, 2) To UBound(Arreglo, 2) - 1
    If Arreglo(Dimension, i) <> Arreglo(Dimension, i + 1) Then _
        str = str & "'" & Arreglo(Dimension, i) & "',"
Next i
GeneraStrDeArreglo = str & "'" & Arreglo(Dimension, UBound(Arreglo, 2)) & "'"

End Function

```

```

Function RegresaFechas(Titulo As String, _
    Hoja As Worksheet, RangoDeFechas As Range, _
    MinNumMuestra As Integer, Muestra, FechMuestra, _
    FechInicio As Integer, FechTermino As Integer, Periodo) As Boolean

With frm_Fecha
    .Caption = Titulo & " [" & Hoja.Name & "]"
    .ValidaControles = False
    .CmbMuestra.RowSource = RangoDeFechas.Address(external:=True)
    .CmbMuestra.ListIndex = 0
    .CmbInicio.RowSource = RangoDeFechas.Address(external:=True)
    .CmbInicio.ListIndex = RangoDeFechas.Count - 1
    .CmbTermino.RowSource = RangoDeFechas.Address(external:=True)
    .CmbTermino.ListIndex = RangoDeFechas.Count - 1
    .SpnNoMuestra.Min = MinNumMuestra
    .SpnNoMuestra.Max = RangoDeFechas.Count

```

```

.SpnNoMuestra.Value = RangoDeFechas.Count
.TxtNoMuestra.Value = .SpnNoMuestra.Value
.SpnNoPeriodo.Max = RangoDeFechas.Count
.SpnNoPeriodo.Value = 1
.TxtNoPeriodo.Value = .SpnNoPeriodo.Value
.ValidaControles = True
.UserCancel = False
RepiteMatFecha:
.Show
Application.StatusBar = ""
If Not .UserCancel Then
    Muestra = .SpnNoMuestra.Value
    FechMuestra = .CmbMuestra.ListIndex + 1
    FechInicio = .CmbInicio.ListIndex + 1
    FechTermino = .CmbTermino.ListIndex + 1
    Periodo = .SpnNoPeriodo.Value
    RegresaFechas = True
End If
End With

End Function

```

```

Function RegresaVectorDeRango(Rango)

Dim n As Integer
Dim Arreglo
Dim i As Integer

If Rango.Rows.Count = 1 Then
    n = Rango.Columns.Count
ElseIf Rango.Columns.Count = 1 Then
    n = Rango.Rows.Count
Else
    MsgBox Prompt:="Error el vector debe de ser de un " & _
        "renglon o una columna", Buttons:=vbExclamation
End If
ReDim Arreglo(n - 1)
For i = 1 To n
    Arreglo(i - 1) = Rango(i)
Next i
RegresaVectorDeRango = Arreglo

End Function

```

```

Sub AgregaDatos(Datos, Arreglo)

Dim i As Integer

If IsArray(Arreglo) Then
    ReDim Preserve Arreglo(UBound(Arreglo) + UBound(Datos) - LBound(Datos) + 1)
    For i = LBound(Datos) To UBound(Datos)
        Arreglo(UBound(Arreglo) - UBound(Datos) + i) = Datos(i)
    Next i
Else
    Arreglo = Datos
End If

```

End Sub