



# MANUAL DEL USUARIO

**PROGRAMA PARA AJUSTAR DATOS EXPERIMENTALES**  
**CON MODELOS NO LINEALES**

DESARROLLADO POR

**PEDRO J. BAZÓ C.**

Escuela de Ingeniería Química, Facultad de Ingeniería, UNIVERSIDAD CENTRAL DE  
VENEZUELA, CARACAS, VENEZUELA.

# ÍNDICE

<b>INTRODUCCIÓN</b>	<b>2</b>
<b>CAPITULO I</b>	<b>4</b>
<b>INTRODUCCIÓN AL PROGRAMA</b>	<b>4</b>
1.1. LENGUAJE DE PROGRAMACIÓN UTILIZADO.	4
1.2. INSTALACIÓN DEL PROGRAMA OPTIMIZ.	4
1.3. ARCHIVOS PRINCIPALES DE LA APLICACIÓN.	6
1.4. ARCHIVOS A DESARROLLAR POR EL USUARIO.	7
1.5. ESQUEMA GENERAL DE FUNCIONAMIENTO.	7
1.6. INSTRUCCIONES PARA ESCRIBIR EL ARCHIVO TIPO TEXTO QUE CONTIENE LA DATA	8
1.7. ARGUMENTOS QUE DEBE RECIBIR LA DLL DE LA FUNCIÓN OBJETIVO.	9
1.8. ARGUMENTOS QUE DEBE RECIBIR LA DLL DEL MODELO MATEMÁTICO.	10
1.9. ARGUMENTOS QUE DEBE RECIBIR LA DLL DE LA DERIVADA DEL MODELO MATEMÁTICO.	11
1.10. INSTRUCCIONES PARA CREACIÓN DE LA DLL FUNOBJF.	12
1.11. INSTRUCCIONES PARA CREACIÓN DE LA DLL FUNOBJC.	14
1.12. INSTRUCCIONES PARA CREACIÓN DE LA DLL MODELOF.	17
1.13. INSTRUCCIONES PARA CREACIÓN DE LA DLL MODELOC.	18
1.14. INSTRUCCIONES PARA CREACIÓN DE LA DLL DERVF.	19
1.15. INSTRUCCIONES PARA CREACIÓN DE LA DLL DERVC.	19
1.16. PROBLEMAS QUE PUEDEN RESOLVERSE CON OPTIMIZ.	20
<b>CAPITULO II</b>	<b>25</b>
<b>INSTRUCCIONES PARA EL USO</b>	<b>25</b>
2.1. INTEGRACIÓN DE LAS DLL DESARROLLADAS POR EL USUARIO A LA APLICACIÓN.	25
2.2. INSTRUCCIONES PARA EL USO DE LA APLICACIÓN OPTIMIZ.	25
<b>CAPITULO III</b>	<b>51</b>
<b>RESULTADOS QUE REPORTA OPTIMIZ</b>	<b>51</b>
3.1. ANÁLISIS DE VARIANCIAS.	51
3.2. ANÁLISIS DE VARIANCIAS SOBRE LA REGRESIÓN.	52
3.3. ÍNDICE DE CORRELACIÓN, PENDIENTE Y CORTE CON EL EJE DE LAS ABSCISAS.	53
3.4. RESPUESTAS TEÓICAS.	55
<b>ANEXO</b>	<b>56</b>
<b>EJEMPLO DE APLICACIÓN</b>	<b>56</b>

## INTRODUCCIÓN

En el campo de la ingeniería química casi todo proceso de investigación involucra el desarrollo de modelos matemáticos que intentan aproximar lo mejor posible la realidad física del proceso en estudio. Estos modelos por lo general involucran a su vez parámetros que son necesarios ajustar. Esto indefectiblemente conduce a un problema de optimización.

En los cursos de pregrado de las distintas universidades se acostumbra a usar modelos de optimización muy sencillos que no necesitan de técnicas especiales para resolverlos. Ahora, cuando se hace investigación, con frecuencia se generan una serie de modelos que deben ajustados a los datos experimentales mediante técnicas de regresión especiales, bien sea por la complejidad del modelo (que muchas veces involucra la resolución de ecuaciones diferenciales), o bien sea por la cantidad de datos experimentales, que hace muy tediosa por no decir imposible, el ajuste de parámetros.

Una técnica de regresión frecuentemente utilizada es la de mínimos cuadrados, pero su versión standard no es aplicable a modelos no lineales.

Cuando se estudia la cinética de reacciones químicas en sistemas complejos, con regularidad se desarrollan modelos que no son lineales ni linealizables con respecto a sus parámetros, razón por la cual el método de mínimos cuadrados no puede utilizarse. Para el dimensionamiento de los equipos utilizados en operaciones y procesos químicos, invariablemente se requiere de información, muchas veces bajo la forma de modelos con parámetros y límites de confiabilidad conocidos. Estos modelos son el producto del análisis de datos experimentales y de su interpretación y ajuste mediante técnicas apropiadas.

Cuando los modelos de los procesos son no lineales, como los que incorporan reacciones químicas catalíticas, hacen falta programas de fácil adaptabilidad al sistema experimental utilizado.

OptiMiz es un programa de cálculo numérico que ha sido desarrollado para resolver dicho problema, para sistemas operativos Microsoft Windows 95/98/Me/NT/2000/XP. La velocidad de los cálculos y la adecuada ejecución del programa dependerá de la capacidad del computador a emplear. Se recomienda una amplia capacidad de memoria RAM (más de 64 MB) y un buen coprocesador matemático para resolver problemas que involucren gran cantidad de datos experimentales y modelos extremadamente complejos.

El programa hace uso de los siguientes métodos de optimización:

- Búsqueda directa desarrollado por Hooke y Jeeves.
- Búsqueda directa desarrollado por Buzzi Ferraris.
- Simplex flexible.
- Gauss Newton.
- Máxima pendiente.
- Levenberg Marquardt.

El método de búsqueda directa es de convergencia extremadamente lenta pero sustancialmente más estable y seguro que los métodos de tipo derivativo. El otro método de búsqueda directa, el de Buzzi Ferraris es de convergencia mucho más rápida y segura, y con gran estabilidad. El método de simplex flexible, también es un método de búsqueda tan robusto como los dos anteriores, por lo general converge mucho más rápido que el método de Hooke y Jeeves pero no más que el de Buzzi Ferraris. El método de Gauss Newton es de tipo derivativo y es el de convergencia más rápida, pero es muy inseguro e inestable. El método de máxima pendiente también es de tipo derivativo y es aun más inestable que el de Gauss Newton, y rara vez da resultados aceptables, pero es ideal para explorar parámetros cuando no se tiene un punto de partida en donde comenzar la búsqueda iterativa del método a emplear, ya que debido a que este método apunta en la dirección de máxima pendiente como su propio nombre lo indica, arroja un resultado que muchas veces apunta hacia algún mínimo o máximo local. Por último tenemos al método de Marquardt, que no es más que una alteración al método de Gauss Newton, tal que termina siendo un método intermedio entre los dos métodos de tipo derivativos expuestos anteriormente, este método es de convergencia más lenta que el de Gauss Newton pero sustancialmente más estable y seguro.

El método de Marquardt se puede considerar como uno de los más rápidos y seguros, pero cuando la superficie de los parámetros es extremadamente irregular es recomendable utilizar algún método de búsqueda directa, preferiblemente el desarrollado por Buzzi Ferraris o el de simplex flexible.

Entre las ventajas importante que presenta el programa debemos destacar las siguientes:

1. Permite imponer restricciones a la variables o parámetros a optimizar.
2. Posibilidad de poder optimizar las variables más importantes mientras se mantienen congeladas (fijas) las de menor importancia, o las que en ciertas etapas de cálculo conducen a situaciones críticas.
3. Resuelve problemas de estimación de parámetros en sistemas que involucren procesos con reacciones químicas complejas.
4. En caso de no disponer de las derivadas analíticas del modelo matemático con respecto a los parámetros, se puede optar por utilizar derivadas numéricas, lo cual es sumamente útil en sistemas complejos como lo son los casos de los reactores integrales con múltiples reacciones complejas que a su vez involucran ecuaciones diferenciales.
5. En caso de querer resolver problemas relativamente sencillo, permite la introducción de el modelo directamente desde el programa en lugar de crear la DLL que la contenga, pero esta opción reduce notablemente el tiempo de búsqueda.
6. Dispone de un soporte gráfico que permite representar algunos resultados en forma gráfica, lo cual permite hacer un análisis visual de los parámetros optimizados.

## CAPITULO I

### INTRODUCCIÓN AL PROGRAMA

#### 1.1. LENGUAJE DE PROGRAMACIÓN UTILIZADO.

La aplicación fue creado en el ambiente de desarrollo integrado de Microsoft Visual C++ 6.0, pero puede interactuar perfectamente con rutinas creadas desde Fortran 90/95, y enlazadas y compiladas en forma de DLL (dynamic-link library).

El programa esta dividido principalmente en tres partes. La primera contiene las rutinas de los distintos métodos de optimización a utilizar por la aplicación empaquetadas en una DLL creada desde C++. La segunda parte es la que contiene las rutinas de el modelo matemático, derivadas analíticas del modelo matemático con respecto a los parámetros y la función objetivo a emplear en forma de DLL (estas pueden ser creadas tanto en C/C++ como en Fortran 90/95), estas deben ser desarrolladas por el usuario. Y por último se encuentra el programa ejecutable, creado en Microsoft Visual C++ 6.0, que es el encargado de integrar las rutinas de las diversas funciones creadas en forma de DLL y evaluarlas en los distintos métodos de optimización creadas en esta primera parte.

Los compiladores para la creación de las distintas DLL a utilizar por la aplicación no se suministran, por lo que se recomienda su adquisición.

#### 1.2. INSTALACIÓN DEL PROGRAMA OPTIMIZ.

El programa viene en un CD-ROM que contiene el un archivo llamado OptiMiz.zip, para instalarlo debe disponer de algún software que permita descomprimir archivos zip<sup>1</sup>, el autor da por hecho de que el lector tiene conocimientos básicos del manejo de un PC.

A continuación se detalla la instalación del programa:

1. Introduzca el CD-ROM en la unidad correspondiente.
2. Navegue hasta la unidad de CD-ROM y descomprímalo en algún lugar de preferencia de su disco duro.
3. La aplicación estará contenida en una carpeta de nombre OptiMiz. Adentro de esta carpeta se encuentra el un archivo ejecutable llamado OptiMiz.exe cuyo icono es el siguiente:



---

<sup>1</sup> Generalmente se utiliza alguna versión del WinZip que esta disponible en forma gratuita en la siguiente dirección electrónica: [www.winzip.com](http://www.winzip.com)

Haga doble clic sobre este icono y a continuación se desplegará una ventana similar a la que se muestra en la figura 1.2.1.

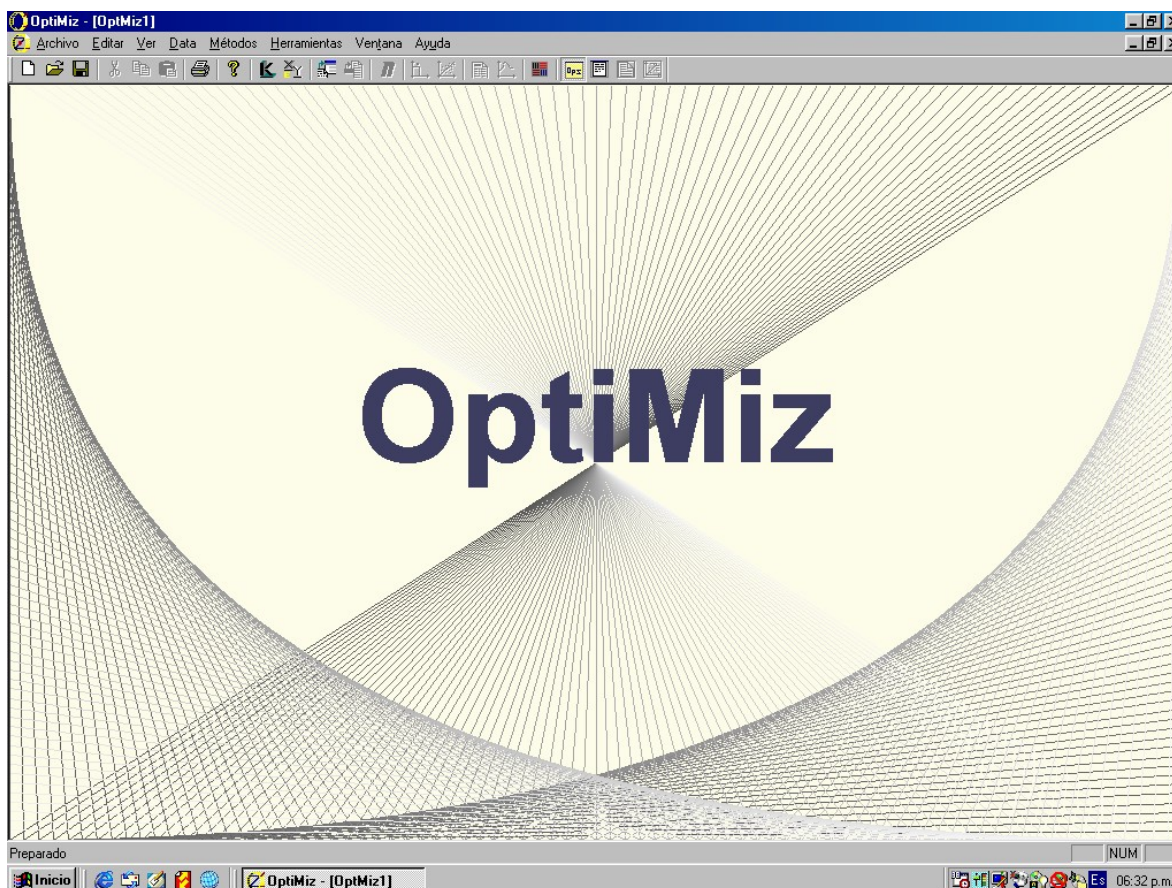


Figura 1.2.1. Ejecución del programa OptiMiz.

4. Debido a que es necesario registrar algunos controles ActiveX para el correcto funcionamiento del programa dirija el cursor del ratón a la barra de menú y haga clic sobre el menú **Herramientas**, y dentro del menú **Herramientas** seleccione el submenú **Configuración**, este a su vez desplegará una serie de opciones, seleccione **Registrar controles ActiveX**<sup>2</sup>, como se muestra en la figura 1.2.2.



Figura 1.2.2. Registro de los controles ActiveX.

<sup>2</sup> Si prefiere, puede utilizar el teclado y escribir las combinaciones de teclas ALT y CTRL y los cursores hasta llegar a la opción deseada.

5. A continuación deberá aparecer un mensaje como el que se muestra en la figura 1.2.3.



Figura 1.2.3. Finalización del registro de los controles ActiveX.

6. Para finalizar con el proceso de instalación haga clic en **Aceptar** y cierre la ventana **Finalizado – ActiveX** que apareció junto con este proceso.

### 1.3. ARCHIVOS PRINCIPALES DE LA APLICACIÓN.

El programa OptiMiz esta integrado por numerosos archivos<sup>3</sup>, pero entre ellos los más importantes son los siguientes:

- **OptiMiz.exe** Archivo ejecutable que inicia la aplicación.
- **Metodos.dll** Contiene todas las rutinas relacionadas con los diversos métodos de optimización.
- **FunObjC.dll** Contiene la rutina de la función objetivo a emplear por el programa, esta DLL deberá ser creada por el usuario en lenguaje C/C++.
- **FunObjF.dll** Contiene la rutina de la función objetivo a emplear por el programa, esta DLL deberá ser creada por el usuario en lenguaje Fortran 90/95.
- **ModeloC.dll** Contiene la rutina del modelo matemático a emplear por el programa, esta DLL deberá ser creada por el usuario en lenguaje C/C++.
- **ModeloF.dll** Contiene la rutina del modelo matemático a emplear por el programa, esta DLL deberá ser creada por el usuario en lenguaje Fortran 90/95.
- **DervC.dll** Contiene la rutina de las derivadas analíticas del modelo matemático con respecto a los parámetros a emplear por el programa, esta DLL deberá ser creada por el usuario en lenguaje C/C++.
- **DervF.dll** Contiene la rutina de las derivadas analíticas del modelo matemático con respecto a los parámetros a emplear por el programa, esta DLL deberá ser creada por el usuario en lenguaje Fortran 90/95.

<sup>3</sup> No extraiga o elimine ningún archivo de la carpeta de la aplicación ya que podría impedir que la aplicación se ejecute normalmente.

#### 1.4. ARCHIVOS A DESARROLLAR POR EL USUARIO.

Aunque el programa permite la introducción de la data directamente por el programa, también permite leer la data desde un archivo tipo texto, esto es muy útil cuando se desea introducir una data muy extensa y ya se dispone de dichos valores en archivos de Word, Excel o algún otro formato, en el que se pueda llevar con relativa facilidad a archivos tipo texto y alterar con suma facilidad para que pueda ser procesada por la aplicación. Por lo general el usuario también deberá crear las DLL que contendrán las rutinas de el modelo matemático, sus derivadas analíticas con respecto a los parámetros y la función objetivo. Cada una de estas DLL se deberá desarrollar de manera individual y su lenguaje de creación será de libre elección del usuario.

#### 1.5. ESQUEMA GENERAL DE FUNCIONAMIENTO.

La figura 1.5.1 muestra el esquema general de funcionamiento del programa.

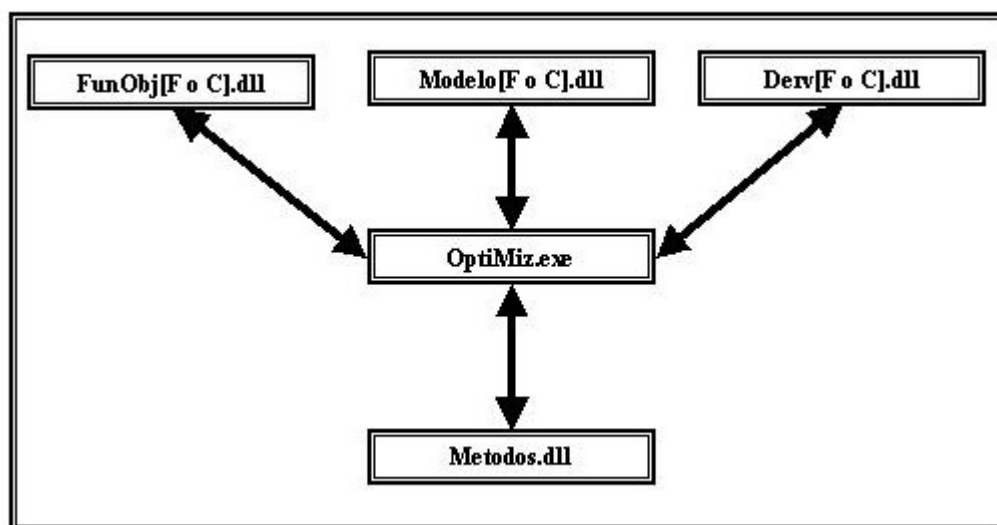


Figura 1.5.1. Esquema general de funcionamiento de la aplicación OptiMiz.

Desde la aplicación OptiMiz.exe se puede crear y actualizar el archivo de datos experimentales, en el mismo se guarda también toda la información necesaria para la llevar a cabo los procesos de optimización además de permitir almacenar igualmente los resultados obtenidos, este archivo de datos tendrá la extensión **opz**, cuyo icono es:



Este tipo de documento solo podrá ser cargado y procesado por la aplicación, sin embargo, la aplicación permite descargar los resultados en archivos tipo texto, cuyos valores se podrán pasar con relativa facilidad a otros formatos, como por ejemplo a



archivos de Microsoft Excel o algún otro manejador de hojas de cálculos, en donde podrán analizarse y graficarse los resultados según las preferencias y necesidades del usuario.

## 1.6. INSTRUCCIONES PARA ESCRIBIR EL ARCHIVO TIPO TEXTO QUE CONTIENE LA DATA

La escritura de este archivo no es obligatoria para la introducción de la data al programa, pero se recomienda su creación cuando los datos experimentales son muy extensos.

El archivo tipo texto a leer por la aplicación deberá prepararse con el siguiente orden separados por espacios en blanco, tabulaciones y/o líneas:

1. Número de parámetros<sup>4</sup>.
2. Variable booleana que indicará si los parámetros a optimizar se encuentran o no inicializados. Un valor de “1”, “TRUE” o “CIERTO” indicará que los parámetros se encuentran inicializados, y para indicar lo contrario deberá introducirse un valor de “0”, “FALSE” o “FALSO”.
3. Variable booleana que indicará si los parámetros a optimizar se encuentran o no acotados o restringidos. Un valor de “1”, “TRUE” o “CIERTO” indicará que los parámetros se encuentran acotados, y para indicar lo contrario deberá introducirse un valor de “0”, “FALSE” o “FALSO”.
4. Conjunto de parámetros iniciales. Si los parámetros no se encuentran inicializados no se deberá introducir ningún valor. En caso de que los parámetros estén acotados y no se desee restringir algunos valores de sus límites inferiores y/o superiores se deberá escribir en el lugar de su valor: “No” o “No Acotar”.
5. Conjunto de límites inferiores de los parámetros, seguidos de sus límites superiores. Si los parámetros no se encuentran acotados no se deberá introducir ningún valor.
6. Número de respuestas experimentales.
7. Número de puntos experimentales.
8. Número de variables independientes ligadas al modelo matemático.
9. Variable booleana que indicará si las respuestas experimentales presentan duplicados. Un valor de “1”, “TRUE” o “CIERTO” indicará que los parámetros se encuentran acotados, y para indicar lo contrario deberá introducirse un valor de “0”, “FALSE” o “FALSO”.
10. Conjunto de respuestas experimentales. Primero se escribirán las respuestas experimentales pertenecientes al primer punto experimental, luego las correspondientes al segundo punto y así hasta completar la cantidad de puntos experimentales.
11. Conjunto de variables independientes ligadas al modelo matemático. Primero se escribirán las variables independientes pertenecientes al primer punto experimental, luego las correspondientes al segundo punto y así hasta completar la cantidad de puntos experimentales.
12. Por ultimo se escribirá la data relacionada con los duplicados de las respuestas experimentales. En caso de no haber duplicados no se deberá introducir nada. Los

---

<sup>4</sup> Todo número real o entero a escribir deberá escribirse en forma de dígitos.

duplicados se escribirán de la siguiente manera, primero se escribirá un número entero que indique la cantidad de duplicados que presenta la primera respuestas del primer punto experimental seguidos de sus respectivos valores, luego se repetirá el paso anterior para la segunda respuesta experimental, y se seguirá así hasta completar la cantidad de respuestas experimentales. Todo lo expuesto anteriormente deberá cumplirse para cada punto experimental.

## **1.7. ARGUMENTOS QUE DEBE RECIBIR LA DLL DE LA FUNCIÓN OBJETIVO.**

Las DLL creadas desde Fortran 90/95 como las creadas en C/C++ deben recibir los siguientes tipos de argumentos:

### **1<sup>er</sup> argumento:**

Vector de números reales que contendrá los valores de las respuestas teóricas predichas por el modelo. Su dimensión será el producto del número de respuestas por el número de puntos experimentales, es decir, será el total del número de observaciones o experiencias realizadas. El orden en que deben almacenarse dichos valores será el siguiente, primero se almacenará el valor de la primera respuesta teórica perteneciente al primer punto experimental, seguido de el valor de la segunda respuesta teórica perteneciente al mismo punto, hasta el valor de la ultima respuesta nuevamente en el mismo punto. Lo expuesto anteriormente debe repetirse para cada punto experimental.

### **2<sup>do</sup> argumento:**

Vector de números reales que contendrá los valores de las respuestas experimentales. Su dimensión será igual al del argumento anterior en caso de no presentarse duplicados. En caso contrario su dimensión se incrementará en el número total de duplicados. El orden en que deben almacenarse dichos valores será similar al del argumento anterior, pero con sus respectivos valores experimentales en lugar de los teóricos. Ahora, cuando existe la presencia de duplicados se deberá seguir almacenando valores en dicho vector de acuerdo con el siguiente orden, primero de almacenará los valores de los duplicados de la primera respuesta y del primer punto experimental, seguido luego de los valores de la segunda respuesta del mismo punto experimental, hasta el valor de la ultima respuesta nuevamente en el mismo punto. Lo expuesto anteriormente debe repetirse para cada punto experimental.

### **3<sup>er</sup> argumento:**

Vector de números enteros que contendrá los valores de la cantidad de duplicados que presenta cada respuesta en cada punto experimental. Su dimensión debe ser igual a la del primer argumento en caso de presentarse duplicados, en caso contrario, este vector no será dimensionado y se le asignará un valor nulo. El orden en que deben almacenarse dichos valores será el siguiente, primero se almacenará el valor del número entero que corresponde a la cantidad de duplicados presentes en la primera respuesta del primer punto experimental, seguido de la cantidad de duplicados de la segunda respuesta del mismo punto experimental, hasta completar así la ultima respuesta. Lo expuesto anteriormente debe repetirse para cada punto experimental.

**4<sup>to</sup> argumento:**

Variable entera que contendrá el número de puntos experimentales.

**5<sup>to</sup> argumento:**

Variable entera que contendrá el número de respuestas presentes en el modelo.

**6<sup>to</sup> argumento:**

Variable entera que contendrá el número de parámetros.

**7<sup>mo</sup> argumento:**

Variable entera en donde se devolverá algún valor que pueda ser interpretada por la aplicación como un mensaje de error. Este argumento podrá devolver algunos de los valores que se listan en la tabla 1.7.1.

**Tabla 1.7.1. Valores que puede retornar el 7<sup>mo</sup> argumento de la DLL de la función objetivo.**

Valor	Interpretación
0	No hay error
1	No hay memoria suficiente
10	No esta definida la división por cero
11	No esta definido el logaritmo de cero
12	El logaritmo de un número negativo no retorna un valor real
13	La raíz de un número negativo no retorna un valor real
14	Valor no definido en los número reales.
15	Error no definido

## **1.8. ARGUMENTOS QUE DEBE RECIBIR LA DLL DEL MODELO MATEMÁTICO.**

Las DLL creadas desde Fortran 90/95 como las creadas en C/C++ deben recibir los siguientes tipos de argumentos:

**1<sup>er</sup> argumento:**

Vector de números reales que devolverá los valores de las respuestas teóricas predichas por el modelo. Su dimensión será igual al número de respuestas presentes en el modelo. El orden en que deben almacenarse dichos valores será siguiente, primero el valor que corresponde a la primera respuestas teórica, seguido del valor que corresponde a la segunda, hasta llegar a la ultima respuesta.

**2<sup>do</sup> argumento:**

Vector de números reales que contendrá los valores de los parámetros ajustados hasta el momento. Su dimensión será igual al número de parámetros. El orden en que deben almacenarse será desde el primer parámetro hasta el último.

**3<sup>er</sup> argumento:**

Vector de números reales que contendrá los valores de las variables independientes ligadas al modelo matemático. Su dimensión será igual al número de variables independientes ligadas al modelo matemático. El orden en que deben almacenarse será desde la primera variable independiente hasta la última.

**4<sup>to</sup> argumento:**

Variable entera en donde se devolverá algún valor que pueda ser interpretada por la aplicación como un mensaje de error. Este argumento podrá devolver algunos de los valores que se listan en la tabla 1.7.1. del ítem anterior.

**1.9. ARGUMENTOS QUE DEBE RECIBIR LA DLL DE LA DERIVADA DEL MODELO MATEMÁTICO.**

Las DLL creadas desde Fortran 90/95 como las creadas en C/C++ deben recibir los siguientes tipos de argumentos:

**1<sup>er</sup> argumento:**

Vector de números reales que devolverá los valores de las derivadas parciales de los parámetros con respecto al modelo. Su dimensión será igual al número de parámetros. El orden en que deben almacenarse será el siguiente, primero se introducirá el valor de la derivada del modelo con respecto al primer parámetro, luego con respecto al segundo, hasta llegar al último.

**2<sup>do</sup> argumento:**

Vector de números reales que contendrá los valores de los parámetros ajustados hasta el momento. Su dimensión será igual al número de parámetros. El orden en que deben almacenarse será desde el primer parámetro hasta el último.

**3<sup>er</sup> argumento:**

Vector de números reales que contendrá los valores de las variables independientes ligadas al modelo matemático. Su dimensión será igual al número de variables independientes ligadas al modelo matemático. El orden en que deben almacenarse será desde la primera variable independiente hasta la última.

**4<sup>to</sup> argumento:**

Vector de números enteros que indica si un parámetro se encuentra fijo o no, y tiene como finalidad evitar que se evalúe la derivada del modelo con respecto a dicho parámetro en algún posible punto crítico. Su dimensión será igual al número de parámetros. Los valores que contendrán dicho vector serán valores de 0 ó 1. Un valor de cero indica que el parámetro no se encuentra fijo, y un valor de 1 lo contrario.

**5<sup>to</sup> argumento:**

Variable entera en donde se devolverá algún valor que pueda ser interpretada por la aplicación como un mensaje de error. Este argumento podrá devolver algunos de los valores que se listan en la tabla 1.7.1. citada anteriormente.

## 1.10. INSTRUCCIONES PARA CREACIÓN DE LA DLL FUNOBJF.

Para la creación de la DLL FunObjF debe disponer de algún compilador de Fortran 90 ó 95. En este caso explicaremos como se crea este archivo a partir de el compilador de Digital Visual Fortran 6.0<sup>5</sup>.

Para la creación de esta DLL siga los siguientes pasos:

1. Inicie Digital Visual Fortran 6.0. Si aparece la ventana **Tip of the Day** haga clic en **Close** para que desaparezca la ventana.
2. En la barra de menú seleccione **File** y seguidamente seleccione el elemento de menú **New...** y aparecerá el cuadro de diálogo **New**, seleccione la pestaña **Projects** en caso de no estar seleccionada.
3. Ubique **Fortran Dynamic Link Library** como tipo de proyecto.
4. Asígnele como nombre del proyecto (Project name) FunObjF y ubíquelo (Location) en algún lugar de su preferencia, como se muestra en la figura 1.10.1.

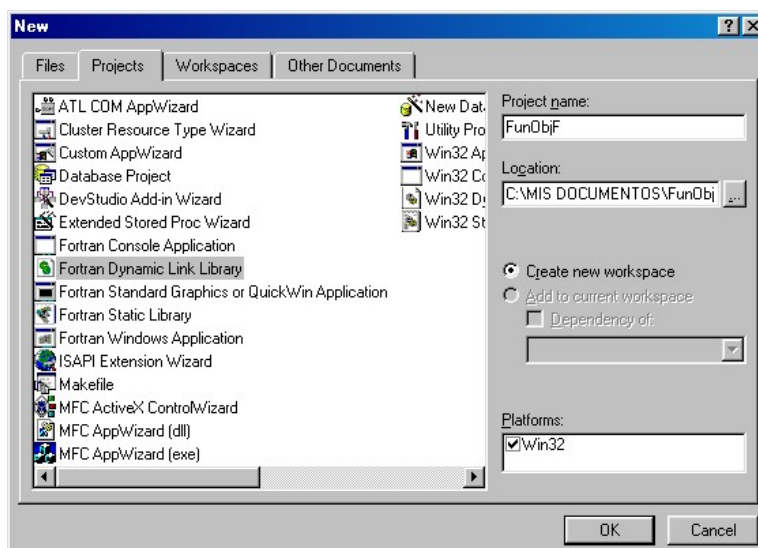


Figura 1.10.1. Cuadro de diálogo “New” de Digital Visual Fortran 6.0.

5. Haga clic en **OK**, para comenzar con el asistente.
6. Seleccione como tipo de DLL a crear la opción de **Una aplicación de DLL con exportación de símbolos** (A DLL application with exported symbols) como se muestra en la figura 1.10.2.
7. Haga clic en **Finish**.
8. Seguidamente aparecerá otro cuadro de diálogo que nos presentará un informe de las opciones seleccionadas en el asistente. Haga clic en OK para generar la aplicación.
9. Haga clic en la pestaña **File View** del panel **Workspace** del ambiente de desarrollo de Digital Visual Fortran 6.0, y navegue hasta el archivo FunObjF.f90 generado por la aplicación.

<sup>5</sup> Estas mismas instrucciones aplican también para los compiladores de Compac Visual Fortran 6.5 y 6.6.

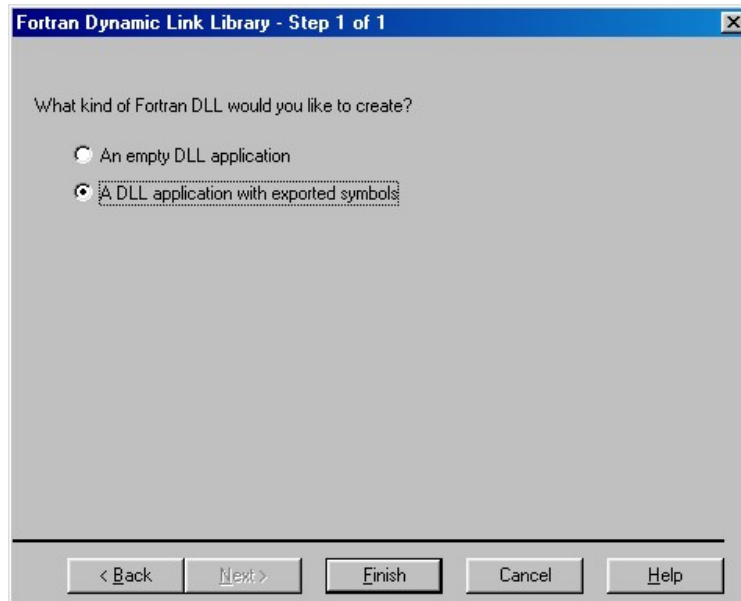


Figura 1.10.2. Paso 1 de 1 del asistente de Fortran Dynamic Link Library.

10. Haga doble clic sobre este archivo y modifíquelo como se muestra en el listado 1.10.1.

#### Listado 1.10.1. Escritura del código fuente FunObjF.f90

---

```
!      FunObjF.f90
!
!      FUNCTIONS/SUBROUTINES exported from FUNOBJF.dll:
!      FUNOBJF      - function
!
real(8) function FUNOBJF(mrResTeo, mrtRes, miNumDup, iNumRes, iNumPtsExp, iNumPar, iMensaje)

    ! Expose function FUNOBJF to users of this DLL
    !
    !DEC$ ATTRIBUTES DLLEXPORT::FUNOBJF
    implicit none
    ! Variables
    integer(4), intent(in) :: iNumRes          !Número de respuestas
    integer(4), intent(in) :: iNumPtsExp ! Número de puntos experimentales
    integer(4), intent(in) :: iNumPar          ! Número de parámetros
    real(8), intent(in), dimension(iNumRes * iNumPtsExp) :: mrResTeo ! Resp. teóricas
    real(8), intent(in), dimension(iNumRes * iNumPtsExp) :: mrtRes    ! Res. exp.
    Integer(4), intent(in), dimension(1) :: iNumDup                  ! Núm. de dupl.
    integer(4), intent(out) :: iMensaje ! Mensaje a devolver
    real(8) :: rFunObj          ! Función objetivo

    ! Aquí debe declarar todas las variables internas a usar por la función.

    ! Body of FUNOBJF

    ! Aquí debe escribir el código fuente de la función objetivo

    FUNOBJF = rFunObj          ! Retorno del valor de la función objetivo.

end function FUNOBJF
```

---

11. Una vez modificado este archivo se debe proceder a escribir el código de la función objetivo a emplear por la aplicación. Los argumentos de esta función se describen en el apartado 1.7<sup>6</sup>.
12. Haga clic en el menú **Build** y seguidamente vuelva a hacer clic sobre el elemento de menú **Batch Build...** y aparecerá la ventana de diálogo **Batch Build**.
13. Seleccione como configuración del proyecto (Project configuration) **FunObjF - Win32 Release** y haga clic en **Rebuild All** como se muestra en la figura 1.10.3.

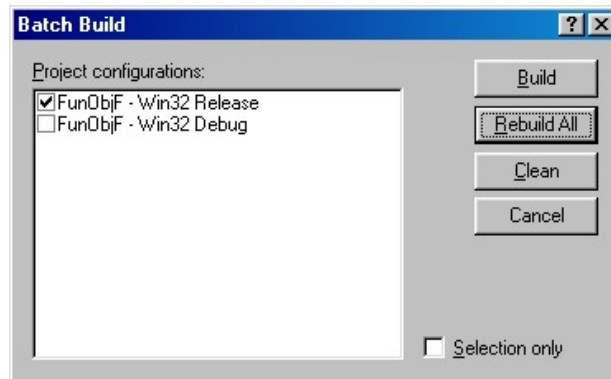


Figura 1.10.3. Ventana de diálogo “Batch Build”.

Una vez cumplido todos estos pasos, ya está creado el archivo FunObjF.dll, y puede ubicar dicho archivo, en una carpeta llamada **Release** dentro de la carpeta del proyecto.

## 1.11. INSTRUCCIONES PARA CREACIÓN DE LA DLL FUNOBJC.

Para la creación de la DLL FunObjC debe disponer de algún compilador de C/C++. En este caso explicaremos como se crea este archivo a partir de el compilador de Microsoft Visual C++ 6.0.

Para la creación de esta DLL siga los siguientes pasos:

1. Inicie Microsoft Visual C++ 6.0. Si aparece la ventana **Tip of the Day** haga clic en **Close** para que desaparezca la ventana.
2. En la barra de menú seleccione **File** y seguidamente seleccione el elemento de menú **New...** y aparecerá el cuadro de diálogo **New**, seleccione la pestaña **Projects** en caso de no estar seleccionada.
3. Ubique **Win32 Dynamic - Link Library** como tipo de proyecto.
4. Asígnele como nombre del proyecto (Project name) FunObjC y ubíquelo (Location) en algún lugar de su preferencia, como se muestra en la figura 1.11.1.
5. Haga clic en **OK**, para comenzar con el asistente.

<sup>6</sup> Los vectores que se reciben como argumento tienen como 1 el valor del primer subíndice.

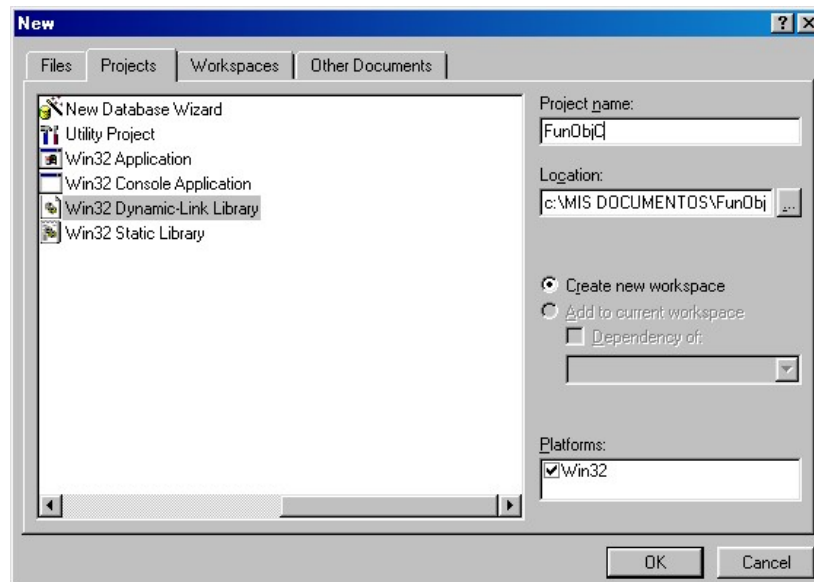


Figura 1.11.1. Cuadro de diálogo “New - Projects” de Microsoft Visual C++ 6.0.

6. Seleccione como tipo de DLL a crear la opción de **Un proyecto de DLL vacío** (An empty DLL project) como se muestra en la figura 1.11.2.

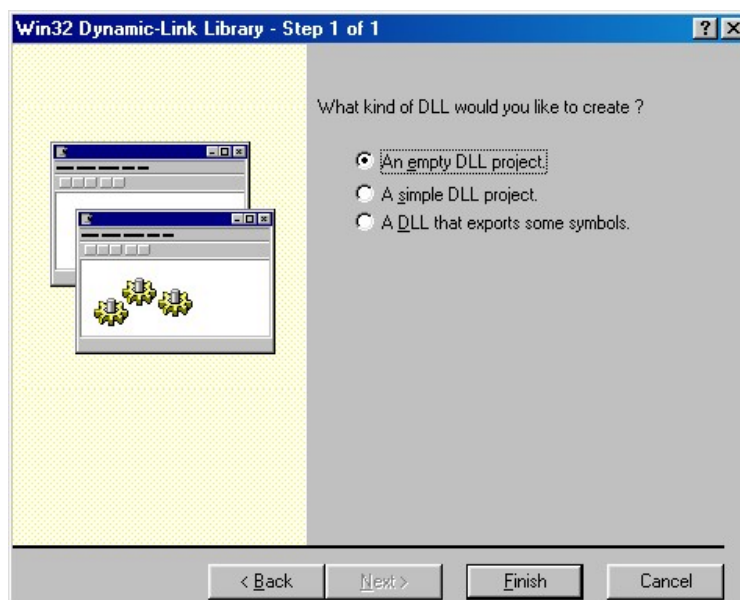


Figura 1.11.2. Paso 1 de 1 del asistente de Win32 Dynamic - Link Library.

7. Haga clic en **Finish**.
8. Seguidamente aparecerá otro cuadro de diálogo que nos presentará un informe de las opciones seleccionadas en el asistente. Haga clic en **OK** para generar la aplicación.
9. En la barra de menú seleccione nuevamente **File** y seguidamente seleccione el elemento de menú **New...** y aparecerá el cuadro de diálogo **New**, seleccione la pestaña **Files** en caso de no estar seleccionada.



10. Ubique **C/C++ Header File** como tipo de archivo y asegúrese que la casilla **Add to project** se encuentre seleccionada.
11. Asígnele como nombre del archivo (File name) FunObjC y ubíquelo (Location) en algún lugar de su preferencia<sup>7</sup>, como se muestra en la figura 1.11.3.

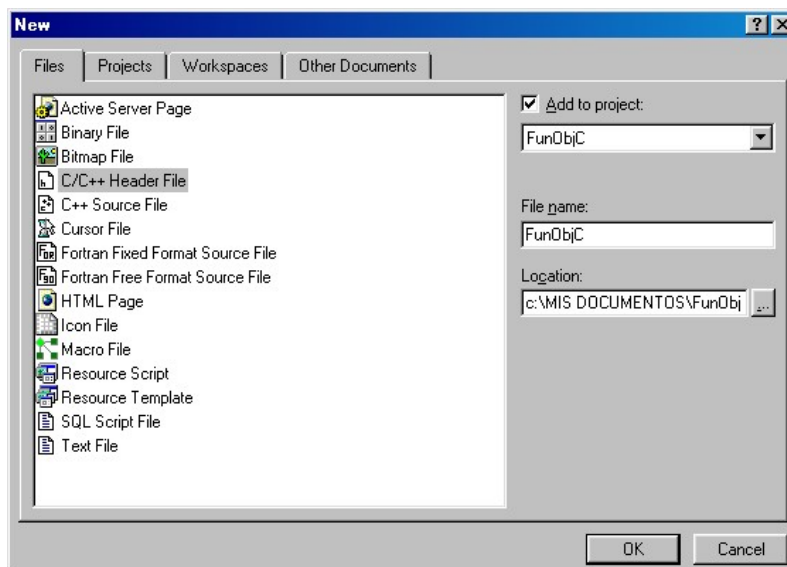


Figura 1.11.3. Cuadro de diálogo “New - Files” de Microsoft Visual C++ 6.0.

12. Repita los pasos del 9 al 11, pero en lugar de seleccionar **C/C++ Header File** seleccione **C++ Source File** como tipo de archivo.
13. Haga clic en la pestaña **File View** del panel **Workspace**, y navegue hasta el archivo FunObjC.h que acaba de crear.
14. Haga doble clic sobre este archivo y modifíquelo como se muestra en el listado 1.11.1.

#### Listado 1.11.1. Escritura del archivo de cabecera FunObjC.h

```
#define DllExport __declspec( dllexport )
DllExport double FunObjC(double *mrResTeo, double *mrtRes, int *miNumDup, int iNumRes,
                        int iNumPtsExp, int iNumPar, int &iMensaje);
```

15. Haga clic nuevamente en la pestaña **File View** del panel **Workspace**, y navegue hasta el archivo FunObjC.cpp que creo anteriormente.
16. Haga doble clic sobre este archivo y modifíquelo como se muestra en el listado 1.11.2.
17. Una vez modificado este archivo se debe proceder a escribir el código de la función objetivo a emplear por la aplicación. Los argumentos de esta función se describen en el apartado 1.7<sup>8</sup>.

<sup>7</sup> Preferiblemente en la misma carpeta del proyecto de la DLL.

<sup>8</sup> Los vectores que se reciben como argumento tienen como 0 el valor del primer subíndice.

### Listado 1.11.2. Escritura del archivo de código fuente FunObjC.cpp

```
// Invoque aquí los archivos de cabecera necesarios
#include "FunObjC.h"

// Cuerpo de la función objetivo
DllExport double FunObjC(double *mrResTeo, double *mrtRes, int *miNumDup, int iNumRes,
                        int iNumPtsExp, int iNumPar, int &iMensaje)
{
    double rFunObj = 0;

    // Aquí debe escribir el código fuente de la función objetivo

    return rFunObj;    // Retorno del valor de la función objetivo
}
```

18. Haga clic en el menú **Build** y seguidamente vuelva a hacer clic sobre el elemento de menú **Batch Build...** y aparecerá la ventana de diálogo **Batch Build**.
19. Seleccione como configuración del proyecto (Project configuration) **FunObjC – Win32 Release** y haga clic en **Rebuild All** como se muestra en la figura 1.11.4.

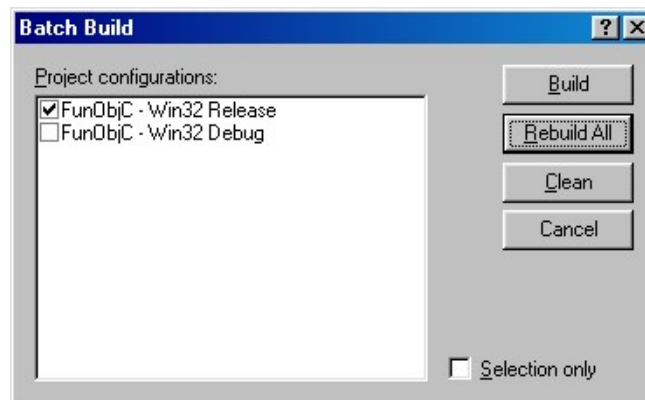


Figura 1.11.4. Ventana de diálogo “Batch Build”.

Una vez cumplido todos estos pasos, ya está creado el archivo FunObjC.dll, y puede ubicar dicho archivo, en una carpeta llamada **Release** dentro de la carpeta del proyecto.

## 1.12. INSTRUCCIONES PARA CREACIÓN DE LA DLL MODELOF.

Para la creación de la DLL ModeloF se seguirán los mismos pasos expuestos en el apartado 1.10, pero con las siguientes diferencias:

- El nombre del proyecto deberá ser cambiado a ModeloF.
- El archivo a editar será ahora ModeloF.f90, y se deberá modificar según el listado 1.12.1.

**Listado 1.12.1. Escritura del código fuente ModeloF.f90**


---

```

!      ModeloF.f90
!
!      FUNCTIONS/SUBROUTINES exported from MODELOF.dll:
!      MODELOF      - subroutine
!
subroutine MODELOF(vrResTeo, vrPar, vrVarInd, iMensaje)

    ! Expose subroutine MODELOF to users of this DLL
    !
    !DEC$ ATTRIBUTES DLLEXPORT::MODELOF

    implicit none
    ! Variables
    real(8), intent(out), dimension(1) :: vrResTeo      ! Respuestas teóricas
    real(8), intent(in), dimension(2)  :: vrPar         ! Parámetros
    real(8), intent(in), dimension(1)  :: vrVarInd      ! Var. independientes
    integer(4), intent(out) :: iMensaje                ! Mensaje a devolver

    ! Aquí debe declarar todas las variables internas a usar por la función.

    ! Body of MODELOF

    ! Aquí debe escribir el código fuente del modelo matemático

end subroutine MODELOF

```

---

**1.13. INSTRUCCIONES PARA CREACIÓN DE LA DLL MODELOC.**

Para la creación de la DLL ModeloC se seguirán los mismos pasos expuestos en el apartado 1.11, pero con las siguientes diferencias:

- El nombre del proyecto deberá ser cambiado a ModeloC.
- El archivo de cabecera (C/C++ Header File) deberá ser nombrado ModeloC.h y deberá ser modificado según el listado 1.13.1.

**Listado 1.13.1. Escritura del archivo de cabecera ModeloC.h**


---

```

#define DllExport __declspec( dllexport )
DllExport void ModeloC(double *vrResTeo, double *vrPar, double *vrVarInd, int &iMensaje);

```

---

- El archivo del código fuente (C++ Source File) deberá ser nombrado ModeloC.cpp y deberá ser modificado según el listado 1.13.2.

**Listado 1.13.2. Escritura del archivo de código fuente ModeloC.cpp**


---

```

// Invoque aquí los archivos de cabecera necesarios
#include "ModeloC.h"
// Cuerpo del modelo matemático
DllExport void ModeloC(double *vrResTeo, double *vrPar, double *vrVarInd, int &iMensaje)
{
    // Aquí debe escribir el código fuente del modelo matemático
}

```

---

## 1.14. INSTRUCCIONES PARA CREACIÓN DE LA DLL DERVF.

Para la creación de la DLL DervF se seguirán los mismos pasos expuestos en el apartado 1.10, pero con las siguientes diferencias:

- El nombre del proyecto deberá ser cambiado a DervF.
- El archivo a editar será ahora DervF.f90, y se deberá modificar según el listado 1.14.1.

### Listado 1.14.1. Escritura del código fuente DervF.f90

---

```
!      DervF.f90
!
!      FUNCTIONS/SUBROUTINES exported from DERVF.dll:
!      DERVF      - subroutine
!
subroutine DERVF(vrDerv, vrPar, vrVarInd, viParFijo, iMensaje)

    ! Expose subroutine DERVF to users of this DLL
    !
    !DEC$ ATTRIBUTES DLLEXPORT::DERVF
    implicit none
    ! Variables
    real(8), intent(out), dimension(1) :: vrDerv      ! Derivadas parciales
    real(8), intent(in), dimension(2)  :: vrPar       ! Parámetros
    real(8), intent(in), dimension(1)  :: vrVarInd    ! Var. independientes
    integer(4), intent(in), dimension(2) :: viParFijo  ! Parámetros fijos
    integer(4), intent(out) :: iMensaje              ; Mensaje a devolver

    ! Aquí debe declarar todas las variables internas a usar por la función.

    ! Body of DERVF

    ! Aquí debe escribir el código fuente del modelo matemático

end subroutine DERVF
```

---

## 1.15. INSTRUCCIONES PARA CREACIÓN DE LA DLL DERVC.

Para la creación de la DLL DervC se seguirán los mismos pasos expuestos en el apartado 1.11, pero con las siguientes diferencias:

- El nombre del proyecto deberá ser cambiado a DervC.
- El archivo de cabecera (C/C++ Header File) deberá ser nombrado DervC.h y deberá ser modificado según el listado 1.15.1.

### Listado 1.15.1. Escritura del archivo de cabecera DervC.h

---

```
#define DllExport __declspec( dllexport )
DllExport void DervC(double *vrDerv, double *vrPar, double *vrVarInd, int *viParFijo,
                    int &iMensaje);
```

---

- El archivo del código fuente (C++ Source File) deberá ser nombrado DervC.cpp y deberá ser modificado según el listado 1.15.2.

**Listado 1.15.2. Escritura del archivo de código fuente DervC.cpp**


---

```
// Invoque aquí los archivos de cabecera necesarios
#include "DervC.h"

// Cuerpo de las derivadas del modelo matemático
DllExport void DervC(double *vrDerv, double *vrPar, double *vrVarInd, int *viParFijo, int
                    &iMensaje)
{
    // Aquí debe escribir el código fuente de las derivadas del modelo matemático
}

```

---

**1.16. PROBLEMAS QUE PUEDEN RESOLVERSE CON OPTIMIZ.**

OptiMiz es un programa diseñado para resolver específicamente problemas de optimización de funciones con modelo no lineales.

**1.16.1. RESOLUCIÓN DE SISTEMAS DE ECUACIONES ALGEBRAICAS Y TRANSCENDENTES.**

Este tipo de problemas es muy frecuente en los cálculos de ingeniería. Matemáticamente el problema consiste en determinar los valores del vector de incógnitas del modelo  $\underline{\eta}(\underline{\beta})$  que satisfacen una serie de ecuaciones del tipo:

$$\eta_j(\underline{\beta}) = 0 \quad j = 1, 2, \dots, n$$

con las restricciones:

$$\beta \min_i \leq \beta_i \leq \beta \max_i \quad i = 1, 2, \dots, m$$

donde  $\underline{\beta \min}$  es el vector límites inferiores y  $\underline{\beta \max}$  es vector de límites superiores impuestos al vector de incógnitas o parámetros  $\underline{\beta}$ .

En este tipo de problemas se puede considerar indistintamente que se dispone de una sola experiencia o punto experimental con  $n$  variables dependientes nulas o de  $n$  experiencias con respuestas únicas también iguales a cero.

**1.16.2. REGRESIONES PARA SISTEMAS LINEALES Y NO LINEALES CON UNA ÚNICA RESPUESTA.**

Este es el problema típico que se presenta cuando se necesitan estimar los parámetros de un modelo. El mismo se resuelve generalmente por medio de una función objetivo de mínimos cuadrados que consiste en hallar el mínimo de dicha función:

$$\phi = \sum_{j=1}^n w_j \cdot (y_j - \eta_j)^2 = \sum_{j=1}^n w_j \cdot \delta_j^2$$

donde  $w_j$  es el peso estadístico asociado al  $j$ -ésimo punto experimental,  $y_j$  es la respuesta experimental  $j$ -ésima y  $\eta_j = \eta_j(\underline{\beta}, \underline{X})$  es la  $j$ -ésima respuesta teórica predicha por el modelo en función del vector de incógnitas o de parámetros  $\underline{\beta}$  y del vector de variables independientes  $\underline{X}$  ligadas al modelo matemático. Lo mismo que para las incógnitas del caso anterior, los elementos del vector de parámetros del modelo podrán estar sujetos a restricciones de límites inferiores y superiores.

### 1.16.3. REGRESIONES PARA SISTEMAS LINEALES Y NO LINEALES CON MÚLTIPLES RESPUESTAS.

En este caso, la estimación de los parámetros del modelo puede hacerse por medio de una función objetivo de máxima probabilidad o por una de mínimos cuadrados. La primera consiste en encontrar el mínimo mediante la elección de valores apropiados para el vector de parámetros o variables de optimización  $\underline{\beta}$ , y dicha función objetivo tiene la siguiente forma:

$$\phi = \sum_{z=1}^r \sum_{k=1}^r w_{z,k} \cdot \sum_{j=1}^n (y_{j,z} - \eta_{j,z}) \cdot (y_{j,k} - \eta_{j,k}) = \sum_{z=1}^r \sum_{k=1}^r w_{z,k} \cdot \sum_{j=1}^n \delta_{j,z} \cdot \delta_{j,k}$$

donde  $r$  es el número de respuestas del sistema y  $w_{z,k}$  son los elementos de la matriz  $\underline{W}$  definida como:

$$\underline{W} = \underline{\Gamma}^{-1}$$

donde

$$\underline{\Gamma} = \begin{bmatrix} \sigma_{1,1} & \sigma_{1,2} & \cdots & \sigma_{1,r} \\ \sigma_{2,1} & \sigma_{2,2} & \cdots & \sigma_{2,r} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{r,1} & \sigma_{r,2} & \cdots & \sigma_{r,r} \end{bmatrix}$$

Los elementos  $\sigma_{z,k}$  son estimados por repetición de experimentos con la ecuación:

$$s^2 = \sigma_{z,k} = \frac{\sum_{j=1}^n \sum_{l=1}^{d_j} (y_{j,z,l} - \eta_{j,z}) \cdot (y_{j,k,l} - \eta_{j,k})}{\sum_{j=1}^n (d_j - n)}$$

La segunda función objetivo, la de mínimos cuadrados, consiste en encontrar un conjunto de valores para los elementos del vector de parámetros del modelo, que haga mínima la función objetivo:

$$\phi = \sum_{j=1}^n \sum_{z=1}^r w_{j,z} \cdot (y_{j,z} - \eta_{j,z})^2 = \sum_{j=1}^n \sum_{z=1}^r w_{j,z} \cdot \delta_{j,z}^2$$

Con respecto a los factores de peso  $w_{j,z}$  que aparecen en esta ecuación, como a los  $w_j$  que se citan para el caso de una única variable, se pueden estimar estos valores de acuerdo a los siguientes criterios:

- Considerarlos constantes e igual a uno:

$$w_{j,z} = 1$$

- Considerarlos constantes y dados por la ecuación:

$$w_{j,z} = \frac{1}{\sum_{j=1}^n (y_{j,z} - \eta_z)}$$

- Suponerlos inversamente proporcionales al los valores experimentales de las variables dependientes o respuestas experimentales:

$$w_{j,z} = \frac{1}{y_{j,z}}$$

- Calcularlos usando los elementos diagonales del estimador de los errores experimentales arriba citados:

$$w_{j,z} = \frac{1}{\sigma_{z,z}}$$

#### 1.16.4. ESTIMACIÓN DE PARÁMETROS EN SISTEMAS EXPERIMENTALES QUE OPEREN COMO INTEGRADORES.

Cuando el análisis del sistema experimental hace necesario que el modelo contemple la predicción de las respuestas adoptando un modelo de reactor, surge la necesidad de definir cambios infinitesimales en las variables dependientes o respuestas con respecto de las coordenadas de posición y de tiempo, es decir:

$$\frac{\partial \eta_j}{\partial t} + \nabla \cdot (\eta_j \cdot \underline{V}) - \nabla \cdot \underline{J}_{\eta_j} = R(\underline{\eta}, \underline{X}, \underline{\beta})$$

El valor experimental de la respuesta o variable dependiente  $y_j$  debe compararse con el valor predicho por la integración de la ecuación diferencial a lo largo del volumen del equipo experimental, es decir:

$$\frac{\partial \eta_j}{\partial t} = - \int_S (\eta_j \cdot \underline{V} + \underline{J}_{\eta_j}) \cdot \underline{N} \cdot dS + \int_V R(\underline{\eta}, \underline{X}, \underline{\beta}) \cdot dV$$

La definición de error inobservable para la  $j$ -ésima respuesta en el  $j$ -ésimo experimento entonces será:

$$\delta_{j,z} = y_{j,z} - \eta_{j,z}$$

el cual puede utilizarse en las ecuaciones de las funciones objetivos de los dos puntos previos.

Las ecuaciones diferenciales aquí escritas son a título de ejemplo y por lo tanto no implican una visión restringida de reactores integrales continuos o por carga. Así, por ejemplo, si el reactor es no isotérmico deberá agregarse el correspondiente balance de energía.

Para la integración de las ecuaciones diferenciales que representan el modelo de reactor, pueden utilizarse los métodos numéricos que el usuario estime apropiados, y será su responsabilidad codificarlo para luego asociarlo con la rutinas de las DLL ModeloF o ModeloC.

## 1.1. CONSIDERACIONES FINALES DEL CAPITULO.

Haciendo uso del programa OptiMiz, se pueden resolver muchos problemas de modelado y estimación de parámetros en cinética de reacciones químicas complejas llevadas a cabo en reactores integrales del tipo tubular, tanto de lecho fijo como de lecho fluidizado, y en reactores tanque agitado continuos y discontinuos. Es de esperar que tampoco fracasará cuando se aplique a otros tipos de reactores.

Las razones por la que el programa OptiMiz es decididamente superior a otros sistemas son:

1. Funciona con PC que tengan como sistema operativos Microsoft Windows 95/98/Me/NT/2000/XP, y esta diseñado bajo los estándares comunes que hacen que su uso sea de fácil adaptabilidad.
2. Debido a que esta diseñado en un entorno 100 % gráfico, se han desarrollado módulos que son muy útil para:
  - a. La elección de un punto de partida apropiado para los parámetros a optimizar, lo cual no es siempre una tarea fácil.



- b. Visualizar el estado de avance del proceso de minimización sin salirse del sistema y con la posibilidad de seguir el proceso a partir del punto en que se suspendió.
  - c. Hacer un análisis visual de residuos, el cual es de mucha ayuda para apreciar si el modelo implementado es apropiado.
3. Permite congelar en forma transitoria la minimización sobre aquellos parámetros que el usuario estime conveniente. Ello contribuye a acelerar considerablemente el avance hacia el mínimo.

Finalmente OptiMiz es un programa de fácil modificación debido a su diseño modular, y en un futuro se le podría agregar opciones tales como:

- Implementación de métodos de optimización basados en algoritmos genéticos y/o redes neuronales.
- Implementación de otros métodos de optimización de tipo derivativos como lo puede ser el método de Powell.
- Implementación de módulos que ayuden a estimar los valores iniciales de los parámetros para ciertos tipos de modelos matemáticos.
- Implementación de editores de ecuaciones que eviten las creaciones de sus respectivas DLL.

## CAPITULO II

### INSTRUCCIONES PARA EL USO

Para usar el programa OptiMiz generalmente hay que crear previamente las DLL que contendrán las rutinas de la función objetivo, del modelo matemático y de las derivadas parciales del modelo matemático con respecto a los parámetros, según sea el caso del sistema a optimizar. Para problemas relativamente sencillos, OptiMiz dispone de un editor simple de ecuaciones que permite la introducción del modelo matemático, pero esta opción es muy limitada, y no permite la introducción de modelos que presenten múltiples respuestas, además de que tampoco permite la introducción de las derivadas parciales del modelo matemático, por lo que en estos casos solo se podrán utilizar derivadas numérica de las que ya dispone OptiMiz. La aplicación también dispone de una función objetivo de mínimos cuadrados sin la consideración del factor de peso, en caso de que el usuario no desee desarrollar su respectiva rutina en una DLL. Se espera que para futuras versiones estas limitaciones sean completamente superadas.

#### 2.1. INTEGRACIÓN DE LAS DLL DESARROLLADAS POR EL USUARIO A LA APLICACIÓN.

Para que la aplicación pueda reconocer las DLL de las diversas rutinas creadas por el usuario es necesario reemplazar las DLL presentes en la carpeta OptiMiz por las nuevas DLL que contienen las rutinas del sistema a optimizar<sup>9</sup>.

#### 2.2. INSTRUCCIONES PARA EL USO DE LA APLICACIÓN OPTIMIZ.

El inicio de la aplicación OptiMiz es sencilla. Si se esta utilizando un ratón puede hacer doble clic<sup>10</sup> sobre el icono OptiMiz:



que se encuentra dentro de la carpeta de la aplicación.

A pesar de que no existen estándares en la forma en que se diseñan las aplicaciones bajo ambiente Windows, si existen convenciones para su organización y diseño. Todas estas convenciones están disponibles en Windows Interface Guidelines for Software Design, una publicación de Microsoft destinada a los desarrolladores de aplicaciones bajo

---

<sup>9</sup> No elimine las DLL que no serán reemplazadas, ya que la aplicación no podrá ejecutarse.

<sup>10</sup> En caso de que su escritorio este configurado con el estilo Web debe hacer un solo clic sobre el icono OptiMiz.

ambiente Windows. El programa OptiMiz trata de ajustarse lo mejor posible a estos convenios, con la finalidad de permitirle al usuario una rápida y fácil adaptabilidad a dicha aplicación.

El entorno de desarrollo para aplicaciones OptiMiz permite de forma sencilla crear, abrir, editar, guardar, optimizar y generar informes sobre el modelo a estudiar.

### **2.2.1. FUNCIONAMIENTO DE LOS MENÚS.**

Antes de comenzar a explicar cada una de las utilidades de OptiMiz, se van a examinar algunas características comunes a todos los elementos de menú. Por ejemplo, existen dos formas de acceder a los elementos de un menú. La aproximación más habitual consiste en colocar el puntero del ratón sobre la opción deseada y pulsar el botón izquierdo del ratón. La segunda opción consiste en utilizar la tecla abreviada subrayada. Por ejemplo, se puede acceder directamente, desde el teclado, al menú Archivo pulsando al mismo tiempo la tecla ALT y la letra A.

A los elementos de menú se puede acceder utilizando la misma secuencia descrita anteriormente y, a menudo, existe un mecanismo adicional de seleccionarlos. Algunos elementos de menú se pueden activar directamente desde cualquier lugar del entorno de la aplicación, utilizando sus combinaciones específicas de teclas abreviadas. Si un elemento de menú dispone de esta posibilidad, se muestra a la derecha del elemento en el menú, la combinación específica de teclas abreviadas para la opción determinada. Por ejemplo, la primera opción que aparece en el menú Archivo es Nuevo. Se puede seleccionar esta opción, de forma inmediata, pulsando simplemente CTRL. – N, sin la necesidad de seleccionar primero el menú Archivo.

A continuación, se muestran algunos comentarios adicionales que conciernen al funcionamiento de los menús: primero, si un elemento de menú aparece desactivado, el entorno de la aplicación está advirtiéndolo sobre el hecho de que esa opción particular no está disponible actualmente. Esto significa que el entorno de la aplicación ha detectado que falta algún prerequisite necesario para que esta opción particular sea válida.

Segundo, cualquier elemento de menú que aparezca seguido de tres puntos (...) indica que cuando esta opción sea seleccionada, se mostrará automáticamente un cuadro de diálogo.

Por último, indicar que se puede activar algunos elementos de menú pulsando sobre sus botones asociados en las barras de herramientas, que aparecen debajo de la barra del menú principal.

### **2.2.2. BARRAS DE HERRAMIENTAS ACOPLADAS O FLOTANTES.**

Se puede hacer que la barra de herramientas (situada justo debajo de la barra de menú) aparezca acoplada o flotante. En modo acoplado, una barra de herramientas aparece

fija a cualquiera de los cuatro bordes de la ventana de la aplicación. El tamaño de la barra de herramientas no se puede modificar cuando está acoplada.

En modo flotante, una barra de herramientas tiene una barra de título estrecha y puede parecer en cualquier lugar de la pantalla. Se puede modificar el tamaño o posición de la barra de herramientas cuando se encuentra en modo flotante.

Para modificar la barra de herramientas acoplada en una barra de herramientas flotante:

- Se pulsa (no se debe soltar el botón izquierdo del ratón) sobre la barra de título o sobre un área donde no haya botones de la barra de herramientas.
- Se arrastra la barra de herramientas desde la zona de acople hacia la posición deseada.

Para acoplar la barra de herramientas flotante:

- Se pulsa (no se debe soltar el botón izquierdo del ratón) sobre la barra de título o sobre un área donde no haya botones de la barra de herramientas.
- Se arrastra la barra de herramientas hacia cualquiera de los cuatro bordes de la ventana de la aplicación.

### 2.2.3. EL MENÚ ARCHIVO.

El menú **Archivo** de la aplicación engloba el conjunto estándar de ordenes habituales de manipulación de archivos para la mayoría de aplicaciones Windows. La figura 2.2.3.1 muestra las opciones de órdenes disponibles en el menú **Archivo**.

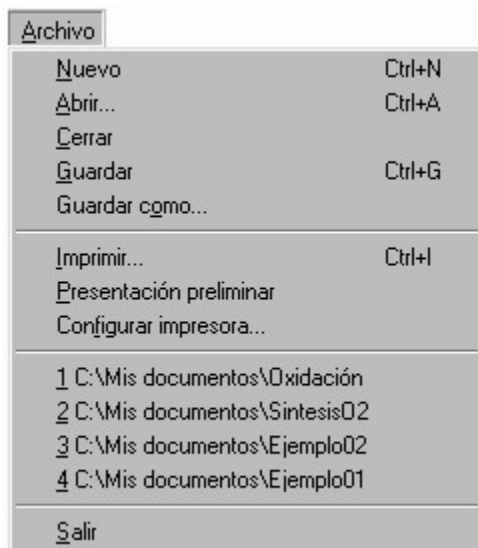



Figura 2.2.3.1. El menú Archivo de OptiMiz.


## **Nuevo**

El elemento de menú **Nuevo** abre un documento vacío. Normalmente al iniciarse la aplicación, esta ya presenta un documento nuevo. La aplicación OptiMiz automáticamente asigna un título y número a cada documento nuevo que se abre. La numeración comienza por 1 y, por tanto, el título del primer documento siempre será OptMiz1, el título del segundo documento OptMiz2 y así sucesivamente.

La forma más rápida de abrir un nuevo documento es pulsar sobre el botón de la barra de herramientas que tiene la siguiente forma: . También existe la opción de presionar la combinación de teclas CTRL. – N.

## **Abrir...**

A diferencia de **Nuevo**, que abre un documento vacío para un archivo que previamente no existe, el elemento de menú **Abrir...** abre un cuadro de diálogo que solicita información acerca de un archivo previamente guardado. Este cuadro de diálogo es el cuadro de diálogo estándar **Abrir Archivo**, que muestra la unidad por defecto, vía de acceso y los parámetros de búsqueda del archivo, permitiendo seleccionar nuestros propios parámetros.

La forma más rápida de abrir un documento existente es pulsar sobre el botón de la barra de herramientas que tiene la siguiente forma: . También existe la opción de presionar la combinación de teclas CTRL. – A.


## **Cerrar**

El elemento de menú **Cerrar** se utiliza para cerrar un documento abierto. Si se tienen múltiples archivos abiertos, esta orden solo cerrará la ventana activa o seleccionada. Se puede identificar la ventana activa, observando la barra de título de la ventana. Las ventanas activas o seleccionadas tienen el enfoque del ratón y teclado (esto significa que la ventana recibe actualmente entrada de ratón y teclado) y se muestran con los parámetros de configuración del color seleccionados por el sistema. Estos parámetros normalmente incluyen las barras de título con colores oscuros. Normalmente, en las ventanas inactivas aparecen en gris las barras de título de la ventana.

Si de una manera accidental se intenta cerrar un documento que ya ha sido modificado, no debe existir ninguna preocupación. La aplicación automáticamente protege al usuario de esta situación altamente peligrosa, avisándole que el archivo no se ha guardado anteriormente y preguntándole si desea guardar el archivo en ese momento.

## **Guardar**

El elemento de menú **Guardar** guarda el contenido del documento actualmente activo o seleccionado. Cuando se guarda un archivo que no se ha guardado previamente, se activará automáticamente el cuadro de diálogo **Guardar como**.


La forma más rápida de guardar un documento es pulsar sobre el botón de la barra de herramientas que tiene la siguiente forma: . También existe la opción de presionar la combinación de teclas CTRL. – G.

### **Guardar como...**

El elemento de menú **Guardar como...** permite guardar una copia de los contenidos de la ventana activa con un nuevo nombre. Esta opción es muy importante cuando se quieren llevar a cabo algunas modificaciones al documento y no se desea modificar su estado actual, así que en caso de que ocurriera un desastre, siempre se podría volver al documento original.

### **Imprimir...**

La obtención de una copia impresa de los contenidos de la ventana activa es tan sencillo como seleccionar el elemento de menú **Imprimir**. Esta opción despliega el cuadro de diálogo **Imprimir**, donde se puede seleccionar, por citar algunas opciones, la impresora a utilizar y configurar la impresora seleccionada eligiendo la opción **Configurar** (Setup).

La forma más rápida de invocar la opción **Imprimir** es pulsar sobre el botón de la barra de herramientas que tiene la siguiente forma: . También existe la opción de presionar la combinación de teclas CTRL. – I.

### **Presentación preliminar**

La opción de menú **Presentación preliminar** consiste justamente en lo que su nombre sugiere y ofrece la posibilidad de tener una presentación previa de la apariencia que tendrá la ventana activa antes de imprimirlas.

### **Configurar impresora...**

La opción **Configurar impresora...** desplegará el cuadro de diálogo **Configurar impresión**, el cual permite seleccionar algunas opciones de preferencia para imprimir.

### **Lista de archivos recientes**

Justo debajo de la opción **Configurar impresora...** del menú aparece una lista de los archivos más recientemente editados. La primera vez que se utiliza la aplicación, esta parte del menú **Archivo** aparece vacía, puesto que no existen archivos recientes.

### **Salir**

El elemento de menú **Salir** permite abandonar la aplicación. No es motivo de preocupación el hecho de olvidar guardar un documento ya modificado antes de seleccionar esta opción. La aplicación muestra un mensaje de advertencia para cada ventana o documento que contenga información no guardada, permitiendo guardar estas modificaciones antes de salir.

#### 2.2.4. EL MENÚ EDITAR.

La opción de menú Editar se encuentra actualmente inactiva, se espera que en futuras versiones de esta aplicación tenga ya alguna utilidad.

#### 2.2.5. EL MENÚ VER.

El menú **Ver** contiene las ordenes que permiten ocultar o aparecer las barra de estado y de herramientas, así como también las ordenes para intercambiar entre los distintos modos de vistas de que dispone la aplicación. La figura 2.2.5.1 muestra el modo en que se encuentran estos elementos de menú cuando se encuentra activado un documento nuevo.

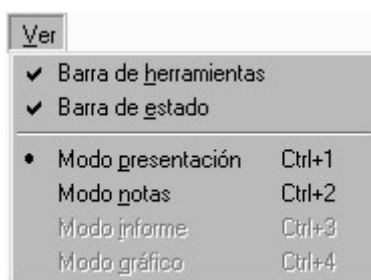


Figura 2.2.5.1. El menú Ver de OptiMiz.

##### Barra de herramientas


Si la opción de menú **Barra de herramientas** se encuentra activada esta estará visible de lo contrario estará oculta.

##### Barra de estado

Si la opción de menú **Barra de estado** se encuentra activada esta estará visible de lo contrario estará oculta.


##### **Modo presentación**

Si la opción de menú **Modo presentación** se encuentra seleccionada esta estará visible de lo contrario estará activa el último modo seleccionado.

La forma más rápida de seleccionar el **Modo presentación** es pulsar sobre el botón de la barra de herramientas que tiene la siguiente forma: . También existe la opción de presionar la combinación de teclas CTRL. – 1.


##### **Modo notas**

Si la opción de menú **Modo notas** se encuentra seleccionada esta estará visible de lo contrario estará activa el último modo seleccionado.

La forma más rápida de seleccionar el **Modo notas** es pulsar sobre el botón de la barra de herramientas que tiene la siguiente forma: . También existe la opción de presionar la combinación de teclas CTRL. – 2.


### Modo informe

La opción de menú **Modo informe** aparecerá activada cuando se haya generado al menos un informe. Si esta opción se encuentra seleccionada, entonces estará visible el último informe generado de lo contrario estará activa el último modo seleccionado.

La forma más rápida de seleccionar el **Modo informe** es pulsar sobre el botón de la barra de herramientas que tiene la siguiente forma: . También existe la opción de presionar la combinación de teclas CTRL. – 3.

### Modo gráfico

La opción de menú **Modo gráfico** aparecerá activada cuando se haya generado al menos un gráfico. Si esta opción se encuentra seleccionada, entonces estará visible el último gráfico generado de lo contrario estará activa el último modo seleccionado.

La forma más rápida de seleccionar el **Modo gráfico** es pulsar sobre el botón de la barra de herramientas que tiene la siguiente forma: . También existe la opción de presionar la combinación de teclas CTRL. – 4.

## 2.2.6. EL MENÚ DATA.

El menú **Data** contiene básicamente todas las opciones que se relacionan con la entrada de valores al documento. La figura 2.2.6.1 muestra las opciones de órdenes disponibles en el menú Data.

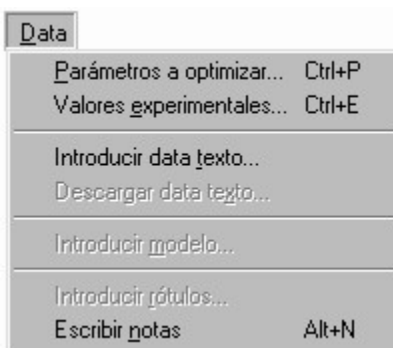


Figura 2.2.6.1. El menú Data de OptiMiz.

### Parámetros a optimizar...

La opción **Parámetros a optimizar...** despliega el cuadro de diálogo **Parámetros a optimizar** como se muestra en la figura 2.2.6.2.



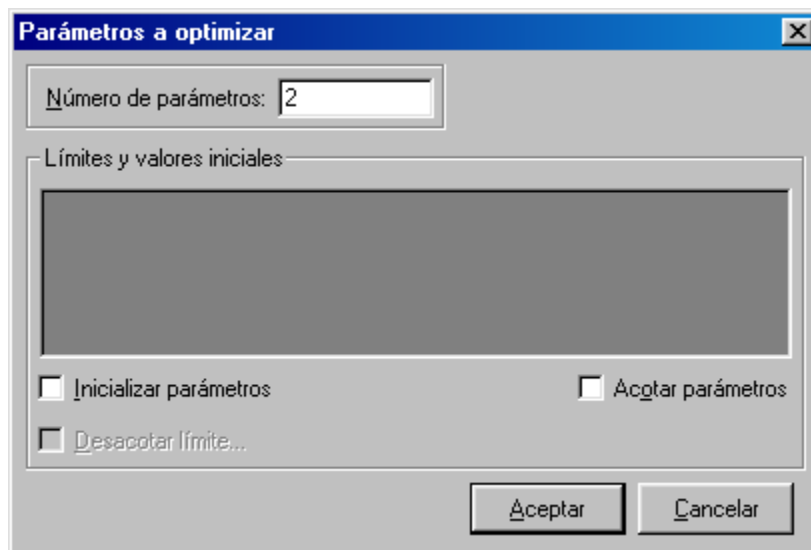


Figura 2.2.6.2. Cuadro de diálogo .Parámetros a optimizar.

Este cuadro de diálogo permite la entrada de algunos valores relacionados directamente con los parámetros.

Por omisión el número de parámetros a optimizar será 2, que es el valor mínimo permitido. También por omisión los parámetros no serán inicializados ni acotados. Si se desea inicializar los parámetros, bastará con activar la casilla de **Inicializar parámetros**, y el cuadro de diálogo cambiará como se muestra en la figura 2.2.6.3 para permitir la entrada de valores iniciales en sus celdas pertinentes.

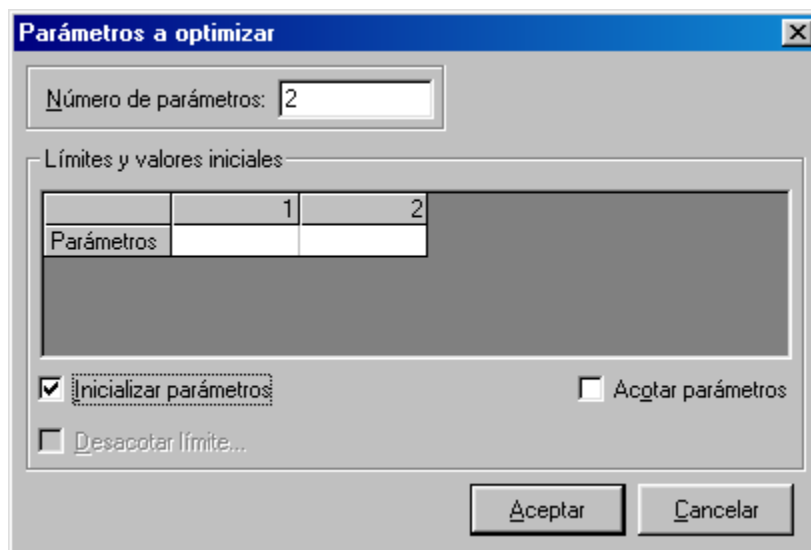



Figura 2.2.6.3. Cuadro de diálogo .Parámetros a optimizar con la opción Inicializar parámetros activada.

También si se desea acotar los parámetros, se podrá activar igualmente la casilla **Acotar parámetros**, y de manera similar a como ocurrió en caso anterior, el cuadro de

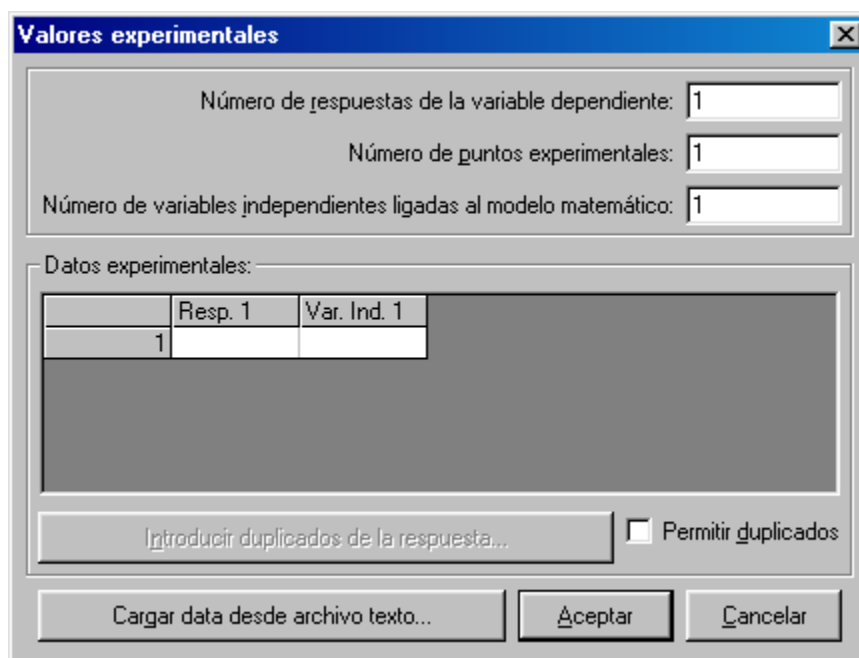
diálogo cambiará para permitir la entrada de los límites superiores e inferiores de los parámetros. En caso de querer liberar algún o algunos de los límites inferiores o superiores bastara ubicarse en la celda correspondiente, y en ese momento la casilla de desacotar límite estará disponible para activarla o no. Otra manera de lograr esto es hacer doble clic sobre la casilla que corresponde al límite que se desea liberar, y en caso de que esta ya se encuentre desacotada la casilla quedará en blanco para esperar la entrada de un nuevo valor.

Los botones de **Cancelar** y **Aceptar** actúan de la forma esperada, **Cancelar** para desechar todos los cambios realizados y **Aceptar** para guardar los valores establecidos.

La forma más rápida de abrir este cuadro de diálogo es pulsar sobre el botón de la barra de herramientas que tiene la siguiente forma: . También existe la opción de presionar la combinación de teclas CTRL. – P.

### Valores experimentales...

La opción **Valores experimentales...** despliega el cuadro de diálogo **Valores experimentales** como se muestra en la figura 2.2.6.4.



	Resp. 1	Var. Ind. 1
1		

Figura 2.2.6.4. Cuadro de diálogo Valores experimentales.

Este cuadro de diálogo permite la entrada de todos los valores relacionados con la data obtenida experimentalmente.

A medida que se establecen los valores del número de respuestas, puntos experimentales y variables independientes, la matriz de celdas que permiten la entrada de los valores de las respuestas experimentales y variables independientes se redimensionará para ajustarse a los parámetros establecidos.

En el caso de presentarse duplicados, se podrá activar la casilla **Permitir duplicados**, la activación de esta, ocasionará que el botón **Introducir duplicados de la respuesta...** este disponible. Por ejemplo, si se desea introducir duplicados de la respuesta 1, del punto experimental 1, bastará con ubicarse en la casilla respectiva y presionar el botón, que en este caso dirá, **Introducir duplicado de la respuesta 1, punto 1**, y se desplegará el cuadro de diálogo **Duplicados** como el que se muestra en la Figura 2.2.6.5.

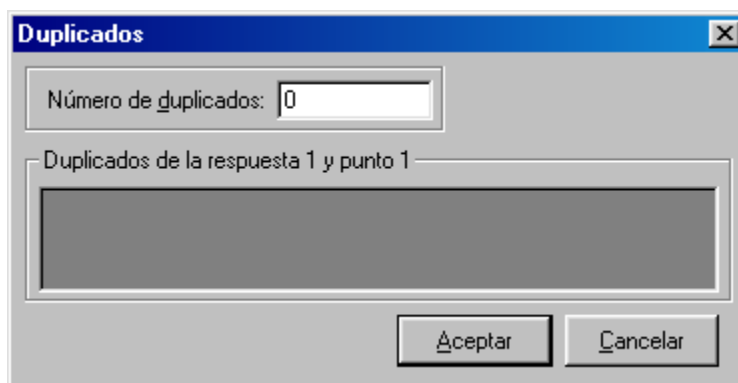



Figura 2.2.6.5. Cuadro de diálogo Duplicados.

Este cuadro de diálogo permite la introducción de duplicados de la  $r$ -ésima respuesta del  $j$ -ésimo punto experimental. En la parte superior se deberá colocar el número de duplicados de esta respuesta, y automáticamente, la matriz de celdas que permitirá la entrada de dichos valores, se redimensionará para permitir su entrada.

En el caso de disponer de la data de respuestas y valores experimentales en un archivo tipo texto, organizada de la misma forma a como se introducen en la matriz de casillas los valores de las respuestas experimentales y las variables independientes, se podrá presionar el botón **Cargar data desde archivo texto...**, esta opción abre un cuadro de diálogo que solicita información acerca del archivo tipo texto que contiene la data. Este cuadro de diálogo es el cuadro de diálogo estándar Abrir Archivo, que muestra la unidad por defecto, vía de acceso y los parámetros de búsqueda del archivo, permitiendo seleccionar nuestros propios parámetros.

Los botones de **Cancelar** y **Aceptar** actúan de la forma esperada, **Cancelar** para desechar todos los cambios realizados y **Aceptar** para guardar los valores establecidos.


La forma más rápida de abrir este cuadro de diálogo es pulsar sobre el botón de la barra de herramientas que tiene la siguiente forma: . También existe la opción de presionar la combinación de teclas CTRL. – E.

### Introducir data texto...

La opción de menú **Introducir data texto...** es una forma alternativa de introducir la data al programa. Al ejecutar esta opción, se abre un cuadro de diálogo que solicita información acerca del archivo tipo texto que contiene la data. Este cuadro de diálogo es el cuadro de diálogo estándar **Abrir Archivo**, que muestra la unidad por defecto, vía de


acceso y los parámetros de búsqueda del archivo, permitiendo seleccionar nuestros propios parámetros.

El archivo tipo texto deberá estar organizado como se muestra en el apartado 1.6 del capítulo I.

La forma más rápida de seleccionar esta opción es pulsar sobre el botón de la barra de herramientas que tiene la siguiente forma: .

### **Descargar data texto...**

La opción de menú **Descargar data texto...** aparecerá activada solo si ya se ha introducido data al documento activo. En ocasiones, es deseable disponer de la data introducida al programa mediante sus respectivos cuadros de diálogo en un archivo tipo texto. Esta opción simplemente descarga la data contenida en el documento activo a un archivo tipo texto, y estos valores estarán organizados de manera idéntica a como están organizados los archivos citados en el apartado anterior.

La forma más rápida de seleccionar esta opción es pulsar sobre el botón de la barra de herramientas que tiene la siguiente forma: .

### **Introducir modelo...**

La aplicación OptiMiz, tiene una opción alternativa para introducir el modelo matemático, sin la necesidad de la creación de su respectiva DLL. El beneficio que se obtiene al no tener que crear esta DLL se paga con una demora mas o menos significativa en la obtención de los resultados, además esta opción es muy limitada como se mencionó al principio de este capítulo. La implementación de esta opción en la aplicación fue pensada para resolver problemas relativamente sencillos de Optimización, sin embargo se espera que para futuras versiones de OptiMiz, esta opción sea eficientemente desarrollada como para permitir la lectura de modelos con respuestas múltiples y sin importar la complejidad del mismo, además de permitir obtener un tiempo aceptable de computo.

La opción de menú **Introducir modelo...** aparecerá activada solo si ya se ha introducido data al documento activo. Esta opción despliega el cuadro de diálogo **Función matemática** como se muestra en la figura 2.2.6.6.

Esta opción permite la lectura de una función matemática, y la forma más simple de explicar de que forma se utiliza este cuadro de diálogo es con un ejemplo. Supongamos que el modelo matemático esta gobernado por la siguiente ecuación:

$$\eta = \frac{\beta_1 \cdot x_1^{\beta_2}}{x_2}$$

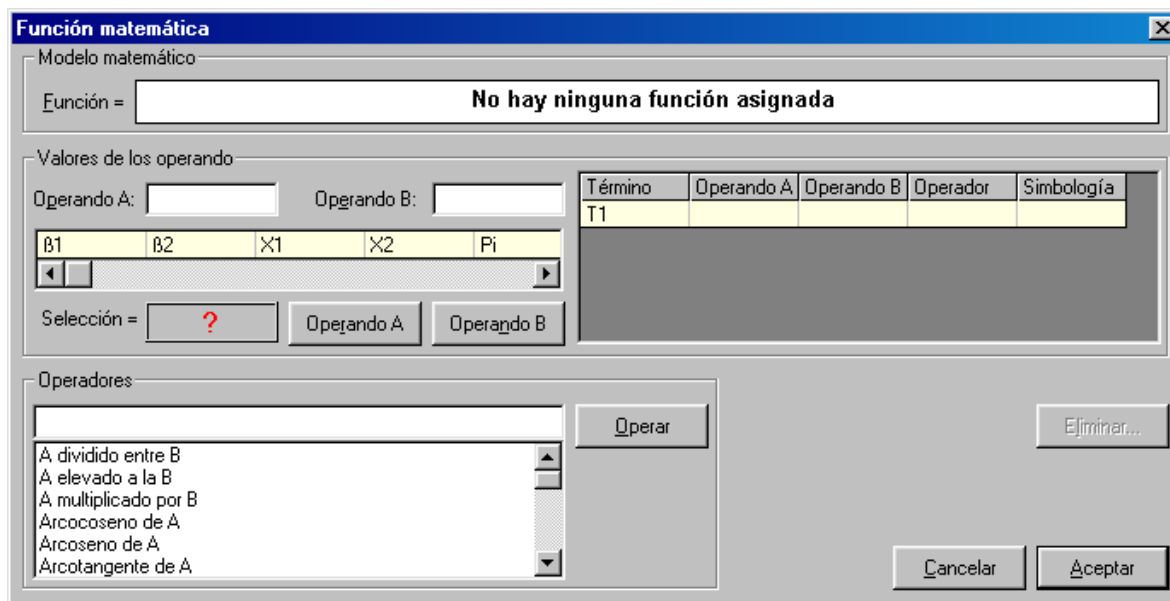


Figura 2.2.6.6. Cuadro de diálogo Función matemática.

Una forma de introducir este modelo en la aplicación sería la siguiente:

- Seleccione de la matriz de celdas que contienen las variables del modelo la que contiene la variable  $X_1$ , haga doble clic sobre esta o presione el botón **Operando A**.
- Seleccione nuevamente de la matriz de celdas que contienen las variables del modelo la que contiene la variable  $\beta_2$ , haga doble clic sobre esta o presione el botón **Operando B**.
- En el recuadro que contiene los operadores a aplicar a los operando seleccione la opción A elevado a la B, haga doble clic sobre esta o bien presione el botón **Operar**. Ahora en la matriz de celdas que contiene los términos de la función quedará definido el término T1 que acabamos de crear.
- Seleccione de la matriz de celdas que contienen las variables del modelo la que contiene la variable  $\beta_1$ , haga doble clic sobre esta o bien presione el botón **Operando A**.
- Ahora, en lugar de seleccionar el operando B de las matriz de celdas utilizadas en el caso anterior, seleccionaremos el término T1 de la matriz de términos, haga doble clic sobre esta o bien presione el botón **Operando B**.
- En el recuadro que contiene los operadores a aplicar a los operando seleccione la opción A multiplicado por B, haga doble clic sobre esta o bien presione el botón **operar**. Ahora en la matriz de celdas que contiene los términos de la función quedará definido el término T2 que acabamos de crear.
- Para que el modelo señalado anteriormente quede correctamente definido, bastara agregar el último término que definitivamente dejará asentado completamente el modelo matemático. Seleccionaremos el término T2 de la matriz de términos, haga doble clic sobre esta o bien presione el botón **Operando A**.

- Seleccione de la matriz de celdas que contienen las variables del modelo la que contiene la variable X2, haga doble clic sobre esta o presione el botón **Operando B**.
- En el recuadro que contiene los operadores a aplicar a los operando seleccione la opción A dividido entre B, haga doble clic sobre esta o bien presione el botón **Operar**. Ahora en la matriz de celdas que contiene los términos de la función quedará definido el término T3 que acabamos de crear. Ahora el cuadro de diálogo Función matemática debe verse como el que se muestra en la figura 2.2.6.7.

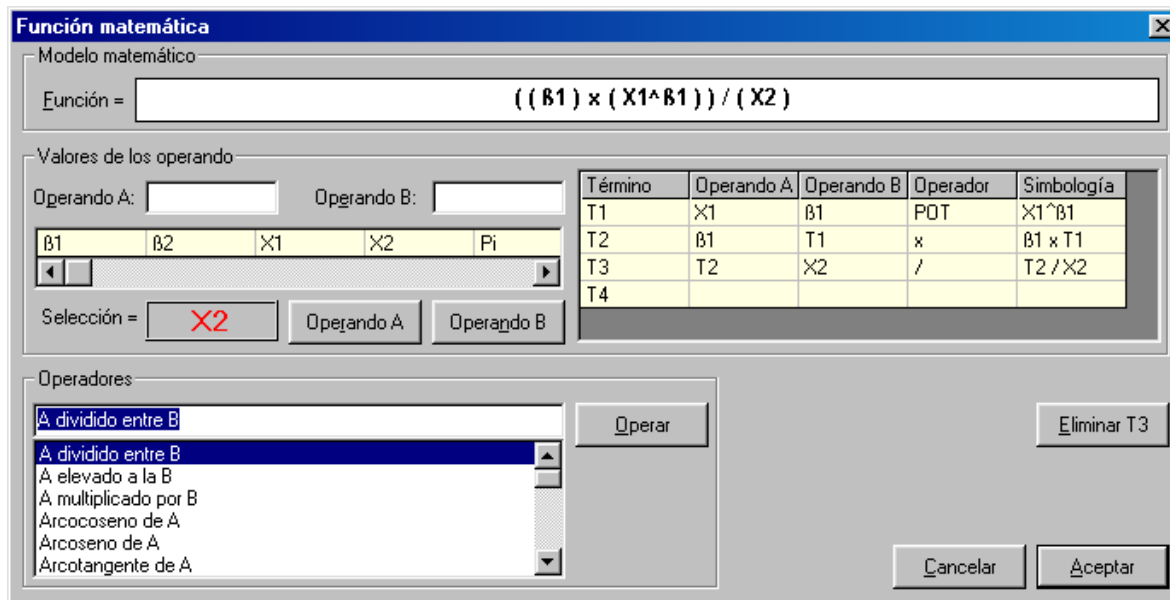



Figura 2.2.6.7. Cuadro de diálogo Función matemática con un modelo de ejemplo cargado.

Los botones de **Cancelar** y **Aceptar** actúan de la forma esperada, **Cancelar** para desechar todos los cambios realizados y **Aceptar** para guardar los valores establecidos.

La forma más rápida de seleccionar esta opción es pulsar sobre el botón de la barra de herramientas que tiene la siguiente forma: .

### Introducir rótulos...

La opción de menú **Introducir rótulos...** aparecerá activada solo si ya se ha introducido data al documento activo. Esta opción despliega el cuadro de diálogo **Rótulos** como se muestra en la figura 2.2.6.8.

Este cuadro de diálogo permite rotular o etiquetar el título con que deberán aparecer los informes y las gráficas generadas, el modelo matemático empleado, la función objetivo, las respuestas, las variables independientes ligadas al modelo matemático y los parámetros.

**Rótulos**

Título del informe:

Nombre del modelo:

Nombre de la función objetivo:

Nombre de las respuestas

Respuestas	1
Nombres	<input type="text"/>

Nombre de las variables independientes

Var. Ind.	1
Nombres	<input type="text"/>

Nombre de los parámetros

Parámetros	1	2
Nombres	<input type="text"/>	<input type="text"/>

Aceptar Cancelar

Figura 2.2.6.8. Cuadro de diálogo Rótulos.

Los botones de **Cancelar** y **Aceptar** actúan de la forma esperada, **Cancelar** para desechar todos los cambios realizados y **Aceptar** para guardar los valores establecidos.

### Escribir notas

La opción de menú **Escribir notas** activa la vista de la aplicación en modo de notas. Esta opción permite simplemente introducir al documento anotaciones particulares del mismo.

### 2.2.7. EL MENÚ MÉTODOS.

El menú **Métodos** contiene básicamente todas las opciones que se relacionan con los procesos de optimización o ajuste de parámetros. La figura 2.2.7.1 muestra las opciones de órdenes disponibles en el menú Métodos.

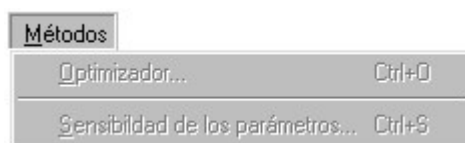


Figura 2.2.7.1. El menú Métodos de OptiMiz.

## Optimizador...

La opción de menú **Optimizador...** aparecerá activada solo si ya se ha introducido data al documento activo<sup>11</sup>. Esta opción de menú es el corazón de la aplicación, es la que permite seleccionar el método de optimización a emplear y algunas opciones más. La figura 2.2.7.2 muestra el cuadro de diálogo **Optimizador** para un problema de dos parámetros.

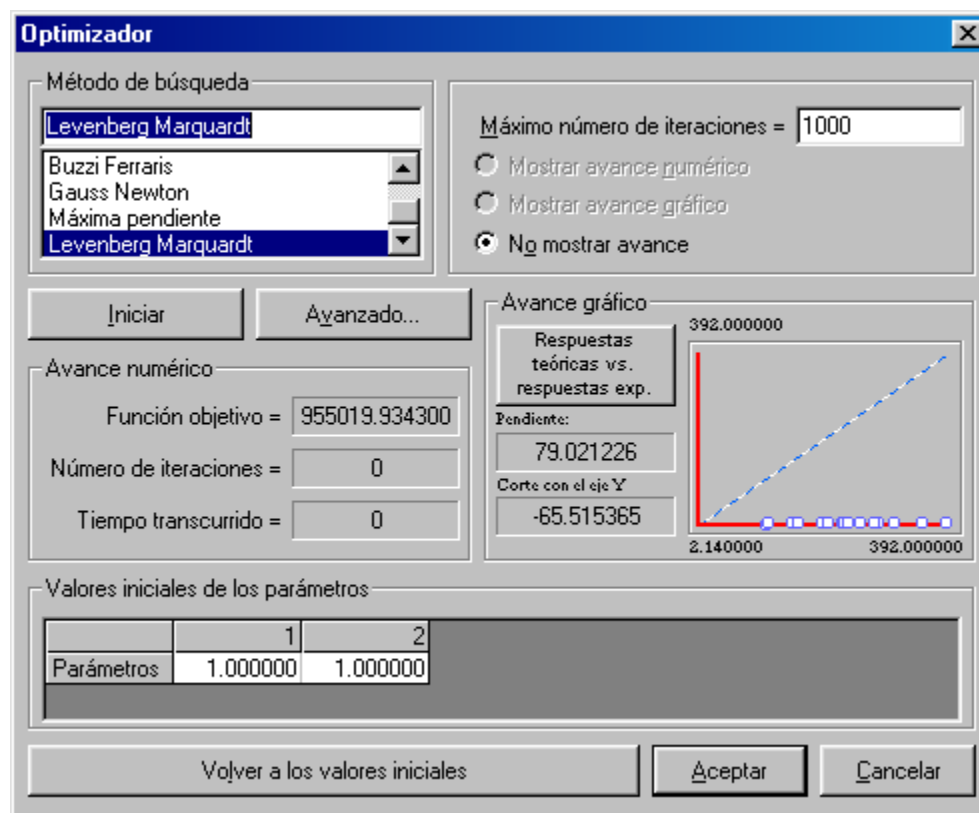


Figura 2.2.7.2. El cuadro de diálogo Optimizador.

En el recuadro superior izquierdo debe definir el método de optimización a implementar, por omisión el método de Levenberg Marquardt es el que aparece seleccionado.

En el recuadro superior derecho se debe definir el máximo número de iteraciones, que por omisión tiene un valor de 1000. También debe establecer que tipo de avance mostrará el proceso de búsqueda. Por omisión aparece seleccionada la opción **No mostrar avance**, esta opción es la más recomendada, debido a que como no actualiza el cuadro de diálogo a medida que se realiza cada iteración, el tiempo de computo es mucho menor. Las otras opciones no se encuentran disponibles actualmente pero lo que se espera que hará es lo siguiente: La opción **Mostrar avance numérico** actualizará los valores del recuadro

<sup>11</sup> Al seleccionar el elemento de menú **Optimizador**, aparte de tener ya toda la data definida en el documento, deberá preocuparse por definir correctamente la función objetivo, el modelo matemático y sus derivadas, ya que la aplicación no tiene forma de saber si las funciones definidas por el usuario son validas o no para el problema en particular. En caso de no considerar lo expuesto anteriormente podría provocar una salida intempestiva de la aplicación, y en el peor de los casos provocar la caída del sistema operativo.



**Avance numérico** en cada iteración, y la opción **Mostar avance gráfico** ocultará todos los controles del cuadro de diálogo y mostrará una gráfica de los ajuste de los parámetros y de la función objetivo a medida que se va realizando cada iteración.


El recuadro **Avance gráfico**, que se encuentra en la parte central derecha muestra una pequeña gráfica de las respuestas teóricas vs. las respuestas experimentales, en función de los parámetros actuales, para el caso de la figura 2.2.7.2. esta gráfica se muestra para los valores de los parámetros que se muestran en el recuadro inferior **Valores iniciales de los parámetros**. También se muestra una recta de pendiente 1. Esto se hace con la finalidad de obtener una idea gráfica de que tan cerca se encuentra el par ordenado (respuestas teóricas, respuestas experimentales) de esta. También se muestran los valores de la pendiente que tendría una recta trazada por el método de mínimos cuadrados al conjunto de puntos y se muestra igualmente el corte que tendrían con el eje de las Abscisas.

El botón **Iniciar** simplemente ejecuta el proceso de búsqueda con todas las opciones previamente seleccionada. Al finalizar el proceso de optimización con el método seleccionado, los valores del cuadro de diálogo se actualizarán con los valores obtenidos como resultado de la búsqueda.

Si se desea realizar una nueva exploración con otro método, partiendo de los valores obtenido en la búsqueda anterior, simplemente se selecciona el otro método, y en caso de ser necesario debe modificarse el valor del máximo número de iteraciones. En caso de querer realizar la nueva exploración partiendo de los valores iniciales de los parámetros, se debe presionar el botón **Volver a los valores iniciales**, e inmediatamente el cuadro de diálogo se actualizará con dichos valores.

El botón **Avanzado...** muestra el cuadro de diálogo **Opciones avanzadas**, el funcionamiento de este cuadro de diálogo se explicará en el apartado 2.2.8, por ahora solo será necesario adelantar, que las modificaciones realizadas a este cuadro de diálogo desde aquí, no quedarán registradas en el documento activo.

Los botones de **Cancelar** y **Aceptar** actúan de la forma esperada, **Cancelar** para desechar todos los cambios realizados y **Aceptar** para guardar los valores establecidos.

La forma más rápida de abrir este cuadro de diálogo es pulsar sobre el botón de la barra de herramientas que tiene la siguiente forma: . También existe la opción de presionar la combinación de teclas CTRL. – O

### **Sensibilidad de los parámetros...**

La opción de menú **Sensibilidad de los parámetros...** aparecerá activada solo si ya se ha introducido data al documento activo

En ocasiones, la elección de los valores iniciales de los parámetros se hace muy complicada, sobre todo cuando su rango de estudio es muy amplio. La opción **Sensibilidad de los parámetros...** proporciona una valiosa ayuda a la hora de seleccionar dichos valores.

Esta opción despliega el cuadro de diálogo **Sensibilidad de los parámetros** como se muestra en la figura 2.2.7.3.



Figura 2.2.7.3. El cuadro de diálogo Sensibilidad de los parámetros.

En la parte superior izquierda de este cuadro de diálogo aparecerán los valores iniciales de los parámetros y debajo de ellos unos valores que el programa denominará sensibilidad, dichos valores son gobernados por la siguiente ecuación:

$$s_i = \left| \frac{\phi_{m,i} - \phi_o}{\phi_o} \right| \times 100$$

donde

$s_i$ : Sensibilidad del  $i$ -ésimo parámetro.


$\phi_o$ : Función objetivo inicial.

$\phi_{m,i}$ : Función objetivo modificada en función del  $i$ -ésimo parámetro.

La finalidad de este cuadro de diálogo es que el usuario realice cambios en algunos de los parámetros, presione el botón **Ver cambios**, y luego se detenga a observar los cambios ocurridos en el recuadro inferior central de los **Valores modificados** y los compare con los del recuadro inferior izquierdo **Valores originales** (sin modificación), como también debe observar la actualización que se efectúa en la gráfica de respuestas experimentales vs. respuestas teóricas del recuadro derecho.

Si los cambios realizados a los parámetros son satisfactorios se podrá presionar el botón **Reemplazar los parámetros modificados por los originales**, y esto hará precisamente lo que dice dicho botón, y en caso de haber realizado cambios que no son nada alentadores, y desea regresar a sus valores iniciales de una forma rápida, entonces deberá presionar el botón **Restaurar**.

Los botones de **Cancelar** y **Aceptar** actúan de la forma esperada, **Cancelar** para desechar todos los cambios realizados y **Aceptar** para guardar los valores establecidos.

La forma más rápida de abrir este cuadro de diálogo es pulsar sobre el botón de la barra de herramientas que tiene la siguiente forma:  También existe la opción de presionar la combinación de teclas CTRL. – S

## 2.2.8. EL MENÚ HERRAMIENTAS.

El menú **Herramientas** contiene opciones variadas, las cuales se explicarán a continuación. La figura 2.2.8.1 muestra las opciones de órdenes disponibles en el menú **Herramientas**.

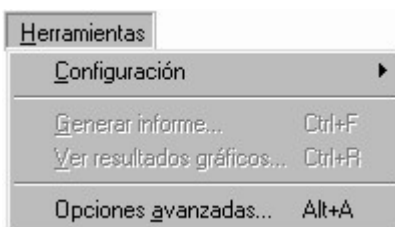


Figura 2.2.8.1. El menú Herramientas de OptiMiz.

### Configuración

El submenú **Configuración** permite seleccionar las DLL a utilizar y también permite registrar los controles ActiveX necesarios para que la aplicación funcione adecuadamente. La figura 2.2.8.2 muestra las opciones de órdenes disponibles en el submenú **Configuración**.



Figura 2.2.8.2. El submenú Configuración del menú Herramientas de OptiMiz.

### Generar informe...

La opción de menú **Generar informe...** aparecerá activada solo si ya se ha realizado algún proceso de optimización. Este elemento de menú despliega un cuadro de diálogo que contiene una serie de opciones que permite escoger que tipo de informe se desea generar. La figura 2.2.8.3 muestra el cuadro de diálogo que se despliega al seleccionar esta opción.

Las opciones que no se encuentran disponibles aparecerán desactivadas. En la figura 2.2.8.3 se puede apreciar que únicamente se encuentran activadas tres opciones para generar un informe, esto es debido a que el único método ejecutado hasta el momento es en este caso el de Levenberg Marquardt.

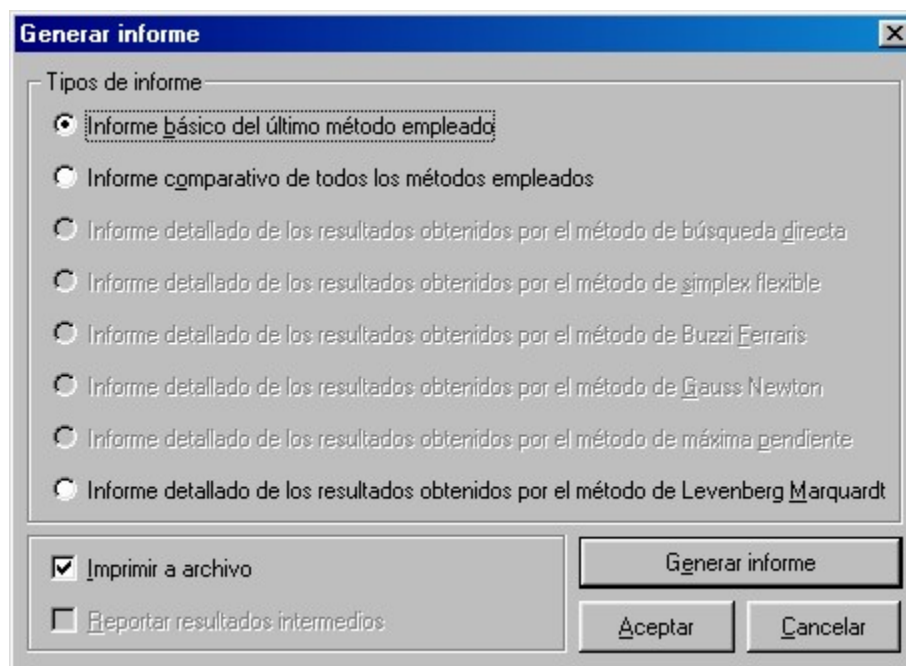



Figura 2.2.8.3. El cuadro de diálogo Generar informe.

Cuando la casilla **Imprimir a archivo** se encuentra activada, al presionar el botón **Generar informe**, se abrirá el cuadro de diálogo estándar **Guardar como**, este cuadro de diálogo permitirá seleccionar el nombre y la ubicación del archivo para almacenar los resultados del informe en formato texto, y al presionar el botón **Guardar** se cerrará este cuadro de diálogo e inmediatamente la aplicación nos preguntará si se desea abrir este archivo.

En el caso de que la casilla **Imprimir a archivo** se encuentre desactivada, y se presione el botón **Generar informe**, este cuadro de diálogo se cerrará y en la ventana principal de la aplicación se quedará impreso el tipo de informe seleccionado.

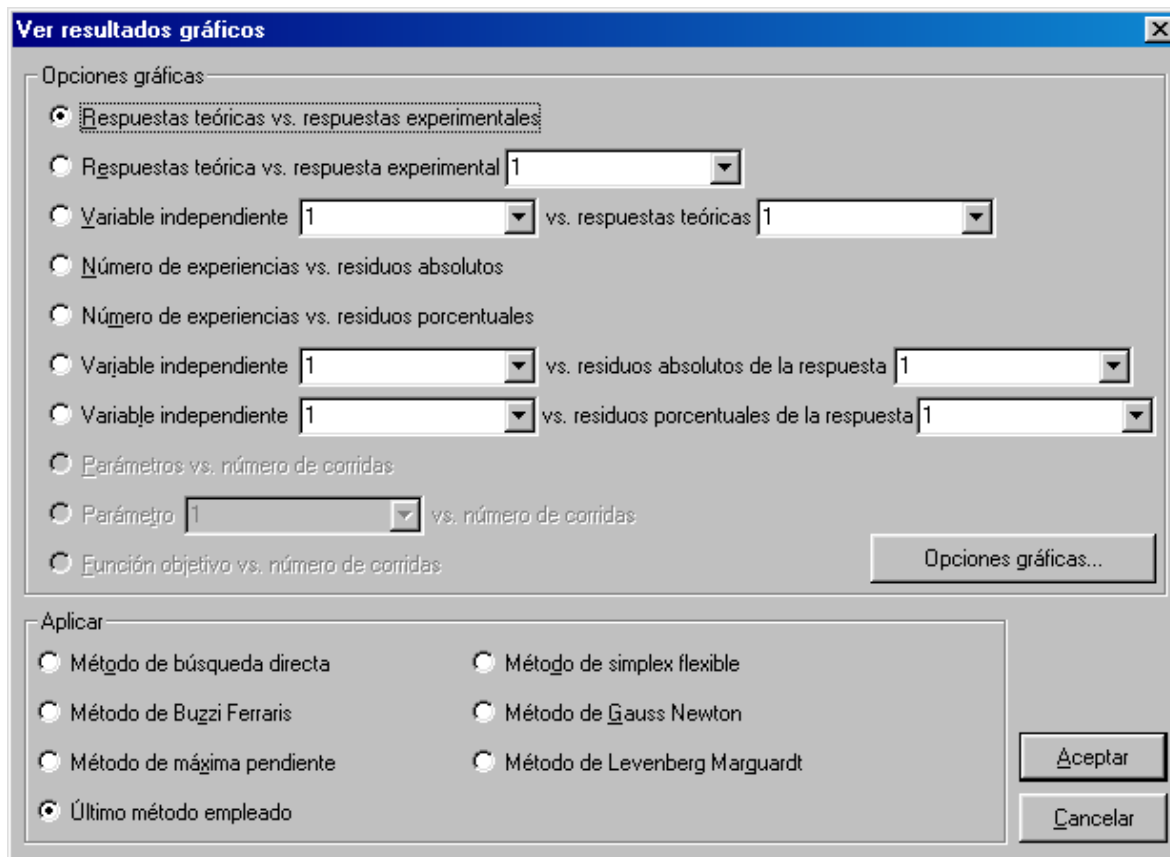
La casilla **Reportar resultados intermedios** actualmente no está disponible, pero se espera que hará lo siguiente: Reportará los resultados intermedios obtenidos en el proceso de optimización, esta opción deberá estar activa solo si el proceso de búsqueda fue efectuado con la opción **Registrar resultados intermedios** del cuadro de diálogo **Opciones avanzadas** que se estudiará en los próximos ítem.

Los botones de **Cancelar** y **Aceptar** actúan de la forma esperada, **Cancelar** para desechar todos los cambios realizados y **Aceptar** para guardar los valores establecidos.

La forma más rápida de abrir este cuadro de diálogo es pulsar sobre el botón de la barra de herramientas que tiene la siguiente forma: . También existe la opción de presionar la combinación de teclas CTRL. – F.

**Ver resultados gráficos...**

La opción de menú **Ver resultados gráficos...** aparecerá activada solo si ya se ha realizado algún proceso de optimización. Este elemento de menú despliega un cuadro de diálogo que contiene una serie de opciones que permite escoger que tipo de gráfico se desea generar. La figura 2.2.8.4 muestra el cuadro de diálogo que se despliega al seleccionar esta opción.



**Figura 2.2.8.4. El cuadro de diálogo Ver resultados gráficos.**

Este cuadro de diálogo se explica por si mismo, pero sin embargo se harán algunas aclaratorias.

Las tres últimas opciones del recuadro superior no se encuentran disponible para esta versión de OptiMiz.

Los residuos absolutos son calculados por la siguiente ecuación:

$$r_{abs,j} = y_j - \eta_j$$

donde

$r_{abs,j}$ : Residuo absoluto de la  $j$ -ésima experiencia.


$y_j$ : Respuesta experimental de la  $j$ -ésima experiencia.

$\eta_j$ : Respuesta teórica de la  $j$ -ésima experiencia.

Los residuos porcentuales son gobernados por una ecuación similar a la anterior, su diferencia es que el lado derecho de la ecuación es dividido por la respuesta experimental de la  $j$ -ésima experiencia y multiplicado luego por cien.

El botón **Opciones gráficas...** despliega el cuadro de diálogo Opciones gráficas, el cual permite agregar restricciones y/o condiciones a las variables que participan en el gráfico.

Los botones de **Cancelar** y **Aceptar** actúan de la forma esperada, **Cancelar** para desechar todos los cambios realizados y **Aceptar** para guardar los valores establecidos.

La forma más rápida de abrir este cuadro de diálogo es pulsar sobre el botón de la barra de herramientas que tiene la siguiente forma: . También existe la opción de presionar la combinación de teclas CTRL. – R.

### Opciones avanzadas...

La opción de menú **Opciones avanzadas...** del menú **Herramientas** permite ajustar algunas variables, con la finalidad de hacer más efectivo el proceso de optimización según las particularidades del problema en cuestión. La figura 2.2.8.5 muestra el cuadro de diálogo que se despliega al seleccionar esta opción.

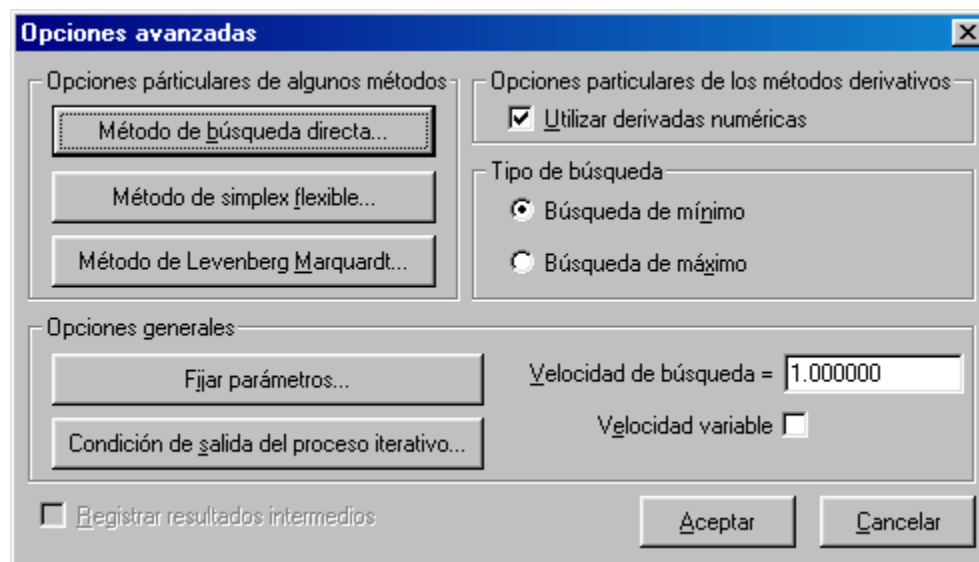


Figura 2.2.8.5. El cuadro de diálogo Opciones avanzadas.

En el recuadro superior izquierdo se puede acceder a las opciones particulares de los métodos de búsqueda directa, simplex flexible y Levenberg Marquardt.

Antes de presionar el botón **Método de búsqueda directa...** debe definir antes el número de parámetros. Este botón despliega un cuadro de diálogo como el que se muestra en la figura 2.2.8.6.

Este cuadro de diálogo permitirá la entrada del paso inicial de búsqueda para el método de búsqueda directa, como también los valores de tolerancia para finalizar con el proceso de búsqueda según el criterio propio del método.

	1	2
Paso	10.000000	10.000000
Tolerancia	0.000001	0.000001

Figura 2.2.8.6. El cuadro de diálogo Opciones del método de búsqueda directa.

Al presionar el botón **Método de simplex flexible...** aparecerá un cuadro de diálogo como el que se muestra en la figura 2.2.8.7.

	1	2
Distancia	1.000000	1.000000

Coeficiente de proyección = 1.000000

Coeficiente de expansión = 2.000000

Coeficiente de reducción = 0.500000

Figura 2.2.8.7. El cuadro de diálogo Opciones del método de simplex flexible.

En la matriz de celdas que se encuentra en la parte superior de cuadro de diálogo se deberá redefinir las distancias iniciales entre los vértices del simplex, en caso de que no se encuentre definido el número de parámetros, la matriz de celdas no aparecerá. Luego en la parte inferior se deberá también redefinir los coeficientes de proyección, expansión y reducción que mejor se adapten al problema en estudio.

Al presionar el botón **Método de Levenberg Marquardt...** aparecerá un cuadro de diálogo como el que se muestra en la figura 2.2.8.8.

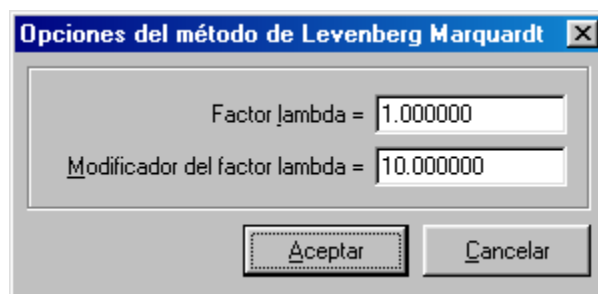


Figura 2.2.8.8. El cuadro de diálogo Opciones del método de Levenberg Marquardt.

Este cuadro de diálogo permite simplemente redefinir el valor inicial del factor lambda ( $\lambda$ ), como también su modificador.

En el recuadro superior derecho del cuadro de diálogo **Opciones avanzadas** se encuentra una opción particular de los métodos de tipo derivativo, que es el uso o no de derivadas numéricas, esta opción es muy útil en problemas, cuyo modelo es extremadamente complejo, cuya derivadas analíticas sean difíciles o imposibles de definir, por omisión esta opción se encuentra activada.

En el recuadro central derecho del cuadro de diálogo **Opciones avanzadas** se podrá escoger el tipo de búsqueda a realizar, si es de maximización o de minimización, por omisión aparece la opción de búsqueda de mínimo que es la que comúnmente se utiliza.

Finalmente tenemos en la parte inferior del cuadro de diálogo **Opciones avanzadas** se encuentran las características comunes a todos los métodos (Opciones generales).

Antes de presionar el botón **Fijar parámetros...** es necesario previamente definir el número de parámetros. Este botón despliega un cuadro de diálogo como el que se muestra en la figura 2.2.8.9.

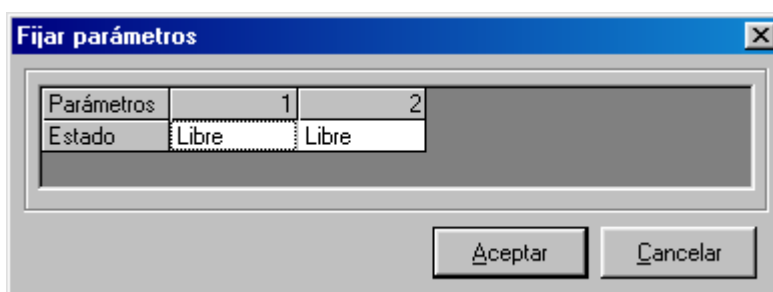


Figura 2.2.8.9. El cuadro de diálogo Fijar parámetros.

En este caso este cuadro de diálogo se invoca para un problema de optimización de dos parámetros, y el número mínimo de parámetros que deben quedar libres es 2, por lo tanto la aplicación no permitirá que cuando se presione el botón **Aceptar** con menos de 2 parámetros libres se almacene esta información. Para fijar o liberar un parámetro bastará ubicarse en la celda respectiva y presionar cualquier tecla o hacer doble clic sobre la misma.



Al presionar el botón **Condición de salida del proceso iterativo...** del cuadro de diálogo **Opciones avanzadas** aparecerá un cuadro de diálogo como el que se muestra en la figura 2.2.8.10.

Este cuadro de diálogo permite establecer el modo en que se desea que abandonar el proceso iterativo, las opciones de este cuadro de diálogo se explican por si misma.

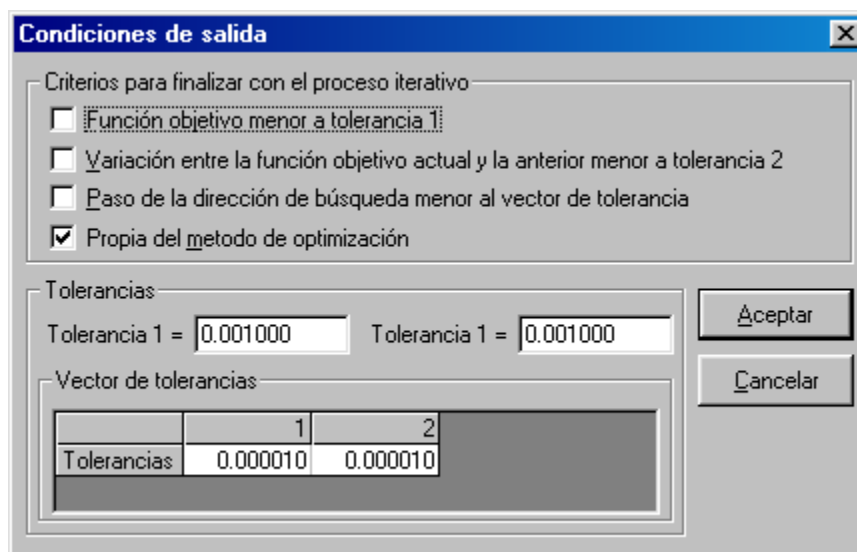



Figura 2.2.8.10. El cuadro de diálogo Condiciones de salida.

En la parte derecha del recuadro inferior del cuadro de diálogo **Opciones avanzadas** aparece editada la velocidad de búsqueda, su valor por omisión es 1, esta opción simplemente es un factor que aumenta o disminuye la velocidad de búsqueda del proceso iterativo, para valores mayores a 1 el paso será acelerado, y para valores menores a 1 lo contrario. La casilla que aparece justo debajo de esta opción indica si la velocidad de búsqueda se regulará en forma automática, y este criterio se fundamenta en la siguiente hipótesis, si el valor de la función objetivo con respecto al paso anterior es mejorada, entonces la velocidad de búsqueda disminuirá debido a que probablemente se este ya acercando a un máximo o a un mínimo, y en caso contrario la velocidad de búsqueda aumentará, ya que probablemente se este alejando de la solución y evitar así que el proceso iterativo oscile alrededor de un mínimo o máximo local.

La opción **Registrar resultados intermedios** actualmente no se encuentra disponible.

Los botones de **Cancelar** y **Aceptar** actúan de la forma esperada, **Cancelar** para desechar todos los cambios realizados y **Aceptar** para guardar los valores establecidos.

La forma más rápida de abrir este cuadro de diálogo es pulsar sobre el botón de la barra de herramientas que tiene la siguiente forma: . También existe la opción de presionar la combinación de teclas CTRL. – A.

### 2.2.1. EL MENÚ VENTANA.

Las opciones del menú **Ventana** son similares a las que aparecen normalmente en todos los productos estándares para Windows. La figura 2.2.9.1 muestra las opciones de órdenes disponibles en el menú **Ventana**.



Figura 2.2.9.1. El menú Ventana de OptiMiz.

#### Nueva ventana

La orden de menú **Nueva ventana** proporciona una de las muchas formas de comenzar a introducir y editar un documento nuevo.

#### Cascada

La orden de menú **Cascada** muestra todas las ventanas abiertas de forma similar a una sección desplegable de tarjetas. Esto permite visualizar fácilmente los títulos de las ventanas.

#### Mosaico

La orden de menú **Mosaico** muestra todas las ventanas abiertas al igual que la opción **Cascada**, pero con la diferencia de que todas las ventas se distribuyen de manera uniforme a lo largo de la aplicación sin solaparse una ventana con otra.

#### Lista de histórico

Justo debajo de la opción **Mosaico** del menú **Ventana** aparece una lista de archivos, se trata de una lista dinámica de los documentos o ventanas abiertas, por nombres, permitiendo seleccionar la ventana que desea activar.

### 2.2.2. EL MENÚ AYUDA.

Las opciones del menú **Ayuda** son similares a las que aparecen normalmente en todos los productos estándares para Windows. La figura 2.2.10.1 muestra las opciones de órdenes disponibles en el menú **Ayuda**.

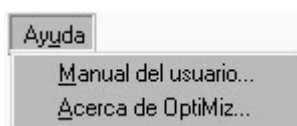


Figura 2.2.10.1. El menú Ayuda de OptiMiz.

### **Manual del usuario...**

La opción **Manual del usuario** abre un documento en formato HTML que contiene toda la información que se suministra en este manual.

### **Acerca de OptiMiz...**

Se trata de un cuadro **Acerca de...** estándar que muestra la versión del producto, la institución en donde se desarrolló, los nombres de los supervisores y del programador.

## CAPITULO III

### RESULTADOS QUE REPORTA OPTIMIZ

No basta siempre decir las cosas con firmeza y seguridad, lo más importante es que lo que se diga sea verdad, por esta razón la aplicación OptiMiz, no solamente reporta los valores de los parámetros optimizados, sino que también reporta otra serie de resultados, sobre todo estadísticos, que nos permitan analizar si los resultados obtenidos son confiables.

#### 3.1. ANÁLISIS DE VARIANCIAS.

La aplicación puede reportar dos tipos de variancias, la experimental<sup>12</sup> y la experimental pesada.

##### a. Variancia experimental:

Para la  $z$ -ésima respuesta:

$$V_{\text{exp},z} = \frac{\sum_{j=1}^n \sum_{h=1}^{d_{j,z}} (y_{j,z,h} - \bar{y}_{j,z})^2}{\sum_{j=1}^n (d_{j,z} - 1)}$$

donde:

$V_{\text{exp},z}$ : Variancia experimental para la  $z$ -ésima respuesta.

$n$ : Número de puntos experimentales.

$d_{j,z}$ : Duplicados del  $j$ -ésimo punto experimental de la  $z$ -ésima respuesta.

$y_{j,z,h}$ : Respuesta experimental del  $j$ -ésimo punto experimental de la  $z$ -ésima respuesta del  $h$ -ésimo duplicado.

$\bar{y}_{j,z}$ : Promedio de los duplicados de la respuesta experimental del  $j$ -ésimo punto experimental de la  $z$ -ésima respuesta.

Para todas las respuestas (total):

<sup>12</sup> Si no existen duplicados, lo cual sucede con frecuencia, el denominador será cero, por lo que no podrá calcularse la variancia experimental, por lo que el programa indicará que no se puede calcular.

$$V_{\text{exp}} = \frac{\sum_{z=1}^r V_{\text{exp},z}}{r}$$

donde:

$V_{\text{exp}}$ : Variancia experimental total.

$r$ : Número de respuestas o variables dependientes del modelo.

**b. Variancia experimental pesada:**

Para la  $z$ -ésima respuesta:

$$V_{ep,z} = \frac{\sum_{j=1}^n w_{j,z} \sum_{h=1}^{d_{j,z}} (y_{j,z,h} - \bar{y}_{j,z})^2}{\sum_{j=1}^n (d_{j,z} - 1)}$$

donde:

$V_{ep,z}$ : Variancia experimental pesada para la  $z$ -ésima respuesta.

$w_{j,z}$ : Factor de peso del  $j$ -ésimo punto experimental de la  $z$ -ésima respuesta.

Para todas las respuestas (total):

$$V_{ep} = \frac{\sum_{z=1}^r V_{ep,z}}{r}$$

donde:

$V_{ep}$ : Variancia experimental pesada total.

### 3.2. ANÁLISIS DE VARIANCIAS SOBRE LA REGRESIÓN.

Para el análisis de variancias sobre la regresión se reportan dos tipos, la debida a la regresión y la de los residuos.

**a. Variancia debido a la regresión:**

Para la  $z$ -ésima respuesta:

$$V_{reg,z} = \frac{\sum_{j=1}^n w_{j,z} \cdot (\eta_{j,z} - \bar{y}_{j,z})^2}{n - ml}$$

donde:

$V_{reg,z}$ : Variancia debido a la regresión para la  $z$ -ésima respuesta.

$ml$ : Número de parámetros libres.

Para todas las respuestas (total):

$$V_{reg} = \frac{\sum_{z=1}^r V_{reg,z}}{r}$$

donde:

$V_{reg}$ : Variancia total debido a la regresión.

#### b. Variancia de los residuos:

Para la  $z$ -ésima respuesta:

$$V_{res,z} = \frac{\sum_{j=1}^n \sum_{h=1}^{d_{j,z}} (\eta_{j,z} - y_{j,z,h})^2}{\sum_{i=1}^n (d_{j,z} - ml)}$$

donde:

$V_{res,z}$ : Variancia de los residuos para la  $z$ -ésima respuesta.

Para todas las respuestas (total):

$$V_{res} = \frac{\sum_{z=1}^r V_{res,z}}{r}$$

donde:

$V_{res}$ : Variancia total de los residuos.

### 3.3. ÍNDICE DE CORRELACIÓN, PENDIENTE Y CORTE CON EL EJE DE LAS ABSCISAS.

A medida que la función objetivo disminuye, el índice de correlación debe tender a uno igualmente la pendiente que se obtiene al aplicar el método de mínimos cuadrados al conjunto de las respuestas teóricas vs. respuestas experimentales debe tender a uno y su corte con el eje de las abscisas a cero.

**a. Índice de correlación:**

Para la  $z$ -ésima respuesta:

$$R_z^2 = 1 - \frac{\sum_{j=1}^n \sum_{h=1}^{d_{j,z}} (y_{j,z,h} - \eta_{j,z})^2}{\sum_{j=1}^n \sum_{h=1}^{d_{j,z}} (y_{j,z,h} - \bar{y}_z)^2}$$

donde:

$R_z$ : Índice de correlación de la  $z$ -ésima respuesta.

$\bar{y}_z$ : Promedio de la  $z$ -ésima respuesta experimental.

Para todas las respuestas (total):

$$R^2 = 1 - \frac{\sum_{j=1}^n \sum_{z=1}^r \sum_{h=1}^{d_{j,z}} (y_{j,z,h} - \eta_{j,z})^2}{\sum_{j=1}^n \sum_{z=1}^r \sum_{h=1}^{d_{j,z}} (y_{j,z,h} - \bar{y})^2}$$

donde:

$R$ : Índice general de correlación.

$\bar{y}$ : Promedio de todas las respuestas experimentales, incluyendo los duplicados.

**b. Pendiente y corte con el eje de las abscisas:**

Cuando el ajuste de los parámetros es perfecto, los valores de las respuestas teóricas deberán ser exactamente igual a las experimentales, lo cual indica que la gráfica de las respuestas teóricas vs. las respuestas experimentales generará una recta de pendiente uno y el corte en el eje de las abscisas será igual a cero. Con la finalidad de obtener una idea de que tan bueno es el ajuste, se hace necesario el cálculo de la pendiente y corte con el eje de las abscisas.

Las ecuaciones para el cálculo de la pendiente y el corte con el eje de las abscisas serán respectivamente:

$$m = \frac{S_{xy} - S_x \cdot S_y}{S_{xx} - S_x^2} \quad [6.3.3]$$

$$b = \frac{S_{xx} \cdot S_y - S_{xy} \cdot S_x}{S_{xx} - S_x^2} \quad [6.3.4]$$

donde:

*m*: Pendiente del conjunto de par coordenado respuestas teóricas vs. respuestas experimentales.

*b*: Corte con el eje de las abscisas del conjunto de par coordenado respuestas teóricas vs. respuestas experimentales.

Los valores de  $S_x$ ,  $S_{xx}$ ,  $S_y$  y  $S_{xy}$  se calcularan por las siguientes ecuaciones respectivamente:

$$S_x = \frac{1}{n} \sum_{j=1}^n \sum_{z=1}^r \eta_{z,j} \quad S_{xx} = \frac{1}{n} \sum_{j=1}^n \sum_{z=1}^r \eta_{z,j}^2$$

$$S_y = \frac{1}{n} \sum_{j=1}^n \sum_{z=1}^r y_{z,j} \quad S_{yy} = \frac{1}{n} \sum_{j=1}^n \sum_{z=1}^r (y_{z,j} - \bar{y}_{z,j})^2$$

### 3.4. RESPUESTAS TEÓRICAS.

Las respuestas teóricas son las respuestas que reporta el modelo cuando lo evaluamos en los parámetros obtenidos por el proceso de optimización.



## ANEXO

## EJEMPLO DE APLICACIÓN

Existe un dicho popular que dice: *Un ejemplo vale más que mil palabras*. Con este escrito es mas que suficiente para comenzar con esta sección.

Supongamos que tenemos el siguiente modelo matemático:

$$\eta = \beta_1 + \beta_2 \cdot e^{\beta_3 \cdot x_1}$$

Con la siguiente tabla de valores experimentales:

Puntos experimentales	Variables independientes	Variables dependientes
	$x$	$y$
1	-5	127
2	-3	151
3	-1	379
4	1	421
5	3	460
6	5	426

Los parámetros presentan las siguientes restricciones:

Límite inferior	Parámetros	Límite superior
0	$\beta_1$	1000
-1000	$\beta_2$	500
-15	$\beta_3$	10

Se desea conocer entonces los valores de  $\beta_1$ ,  $\beta_2$  y  $\beta_3$  que mejor se adapten al modelo matemático. Los valores iniciales (semillas) de los parámetros podrán ser respectivamente los siguientes: 1000, 100 y  $-0.1$

Para resolver este problema utilizando la aplicación OptiMiz siga las siguientes pasos:

1. Primero que nada hay que desarrollar la DLL que contendrá el modelo matemático, esta podrá crearse tanto en Fortran 90/95 como en C/C++, para el caso del presente ejemplo realizaremos esta DLL en Fortran 90/95. Siga las instrucciones que se muestran en el capítulo I, ítem 1.12, y copie el siguiente código que se muestra en el listado A.1 para el archivo ModeloF.f90.

**Listado A.1. Escritura del código fuente ModeloF.f90**


---

```

!      ModeloF.f90
!
!      FUNCTIONS/SUBROUTINES exported from MODELOF.dll:
!      MODELOF      - subroutine
!
subroutine MODELOF(vrResTeo, vrPar, vrVarInd, iMensaje)

    ! Expose subroutine MODELOF to users of this DLL
    !
    !DEC$ ATTRIBUTES DLLEXPORT::MODELOF

    implicit none
    ! Variables
    real(8), intent(out), dimension(1) :: vrResTeo      ! Respuestas teóricas
    real(8), intent(in), dimension(2)  :: vrPar         ! Parámetros
    real(8), intent(in), dimension(1)  :: vrVarInd      ! Var. independientes
    integer(4), intent(out) :: iMensaje                 ! Mensaje a devolver

    ! Aquí debe declarar todas las variables internas a usar por la función.

    ! Body of MODELOF

    iMensaje = 0      ! Indica que no hay error
    vrResTeo(1) = vrPar(1) + (vrPar(2) * exp(vrPar(3) * vrVarInd(1)))

end subroutine MODELOF

```

---

2. Una vez creada esta DLL proceda a reemplazarla por la de la aplicación.
3. Ahora deberemos desarrollar la DLL que contendrá la rutina de la función objetivo, que igualmente podrá ser creada tanto en Fortran 90/95 como en C/C++, para el caso del presente ejemplo y para variar realizaremos esta DLL en C/C++. Ahora bien, tenemos que tomar otra decisión importante: ¿Qué tipo de función objetivo debemos definir?. Para este ejemplo consideraremos una función objetivo de mínimos cuadrados con un factor de peso igual a la unidad como se muestra en la siguiente ecuación:

$$\phi = \sum_{j=1}^n (y_j - \eta_j)^2 = \sum_{j=1}^n \delta_j^2$$

Siga las instrucciones que se muestran en el capítulo I, ítem 1.11, y copie el siguiente código que se muestra en el listado A.2 para el archivo FunObjC.cpp, el código de FunObjC.h se dejará tal cual como esta en el capítulo I.

4. Una vez creada esta DLL proceda a reemplazarla por la de la aplicación.
5. Ahora debemos decidir que tipo de derivadas parciales del modelo con respecto a los parámetros deberemos utilizar: ¿Analítica o numérica?. Debido a que el modelo matemático no es de mayor complejidad, podremos utilizar sin mayor problema derivadas analíticas. Para lograr esto, deberemos nuevamente desarrollar su respectiva DLL, y sin más que hablar, crearemos esta DLL en Fortran 90/95. Siga

las instrucciones que se muestran en el capítulo I, ítem 1.14, y copie el siguiente código que se muestra en el listado A.3 para el archivo DervF.f90.

### Listado A.2. Escritura del archivo de código fuente FunObjC.cpp

```
// Invoque aquí los archivos de cabecera necesarios
#include "FunObjC.h"

// Cuerpo de la función objetivo
DllExport double FunObjC(double *mrResTeo, double *mrtRes, int *miNumDup, int iNumRes,
                        int iNumPtsExp, int iNumPar, int &iMensaje)
{
    iMensaje = 0;          // Indica que no hay error

    double rFunObj = 0;    // Inicializa a cero el valor de la función objetivo
    for(int i = 0; i < iNumPtsExp * iNumRes; i++)
        rFunObj += pow(mrtRes[i] - mrResTeo[i], 2);

    return rFunObj;       // Retorno del valor de la función objetivo
}
```

### Listado A.3. Escritura del código fuente DervF.f90

```
!      DervF.f90
!
!      FUNCTIONS/SUBROUTINES exported from DERVF.dll:
!      DERVF      - subroutine
!
subroutine DERVF(vrDerv, vrPar, vrVarInd, viParFijo, iMensaje)
! Expose subroutine DERVF to users of this DLL
!
!DEC$ ATTRIBUTES DLLEXPORT::DERVF
implicit none
! Variables
real(8), intent(out), dimension(1) :: vrDerv          ! Derivadas parciales
real(8), intent(in), dimension(2)  :: vrPar           ! Parámetros
real(8), intent(in), dimension(1)  :: vrVarInd        ! Var. independientes
integer(4), intent(in), dimension(2) :: viParFijo      ! Parámetros fijos
integer(4), intent(out) :: iMensaje                   ! Mensaje a devolver

! Aquí debe declarar todas las variables internas a usar por la función.

! Body of DERVF

iMensaje = 0 ! Indica que no hay error
if(viParFijo(1) == 0) then !Chequea que el parámetro 1 no se encuentre fijo
    vrDerv(1) = 1
end if
if(viParFijo(2) == 0) then !Chequea que el parámetro 2 no se encuentre fijo
    vrDerv(2) = exp(vrPar(3) * vrVarInd(1))
end if
if(viParFijo(3) == 0) then !Chequea que el parámetro 3 no se encuentre fijo
    vrDerv(3) = vrPar(2) * vrVarInd(1) * exp(vrPar(3) * vrVarInd(1))
end if
end subroutine DERVF
```

6. Una vez creada esta DLL proceda a reemplazarla por la de la aplicación.
7. Ya cumplido todos estos pasos, ya estamos listos para ejecutar la aplicación e introducir luego la data por sus respectivas ventanas de diálogo. Pero en este ejemplo introduciremos la data en un archivo tipo texto y luego le pediremos a la aplicación que la cargue. Para hacer esto crearemos un archivo tipo texto en la carpeta **Mis documentos** y lo nombraremos EjmOpt.txt y guiándonos por el

capítulo I, ítem 1.6 escribiremos en EjmOpt.txt la información que se muestra en el listado A.4.

#### Listado A.4. Escritura del archivo EjmOpt.txt

3

CIERTO

CIERTO

1000 100 -0.1

0 -1000 -15

1000 200 10

1

6

1

FALSO

127

151

379

421

460

426

-5

-3

-1

1

3

5

8. Inicie la aplicación OptiMiz.
9. Seleccione la opción **Data** la barra de menú, y seguidamente seleccione el elemento de menú **Introducir data texto...**, como se indica en el capítulo II, ítem 2.2.6. e introduzca la data contenida en el archivo EjmOpt.txt que se creo anteriormente.
10. Como todo usuario precavido de PC, salve el documento antes de continuar, asígnele un nombre como el de EjmOpt<sup>13</sup>, como se indica en el capítulo II, ítem 2.2.3.
11. Antes de proceder a ejecutar algunos de los métodos de optimización, primero hay que indicarle a la aplicación que tipo de DLL utilizará para este caso. Seleccione la opción **Herramientas** de la barra de menú, y seguidamente seleccione el submenú **Configuración** y finalmente el elemento **Importación de DLL...**, como se muestra en el capítulo II, ítem 2.2.8, esto provocará que aparezca el cuadro de diálogo **Importación de las DLL**, establezca las opciones de este cuadro de diálogo como se muestra en la figura A.1.

<sup>13</sup> Procure salvar el documento cada vez que termine un nuevo paso.

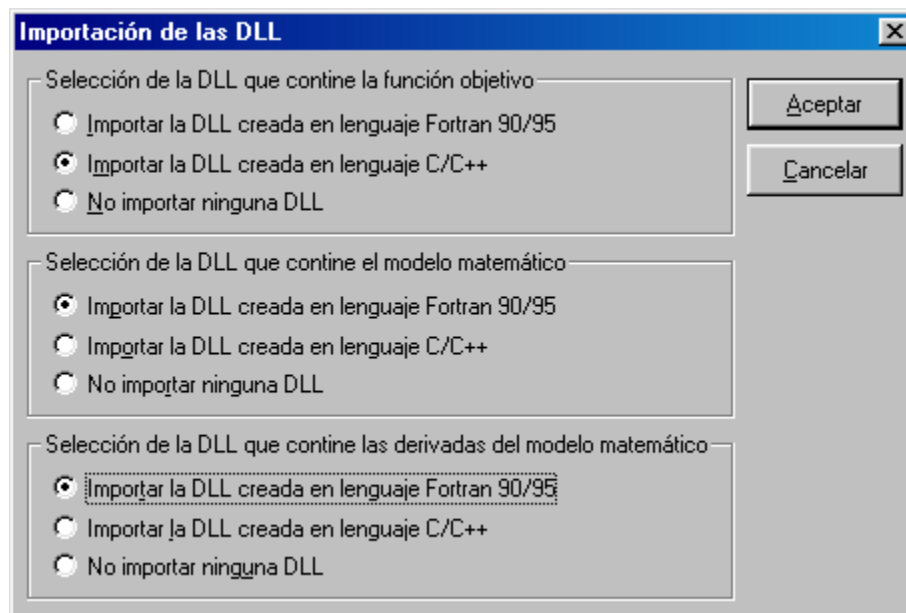


Figura A.1. El cuadro de diálogo Importación de las DLL.

12. Seleccione nuevamente la opción de menú **Herramientas**, y seguidamente accione el elemento **Opciones avanzadas...**, esto invocará el cuadro de diálogo **Opciones avanzadas**. Desactive la casilla **Utilizar derivadas numéricas** y deje las demás opciones como están como se muestra en la figura A.2.
13. Ahora si podemos proceder a ejecutar algunos de los métodos de optimización, pero antes trataremos de ajustar manualmente los valores de los parámetros iniciales, para tratar de garantizar aun mas su convergencia. Para esto seleccionaremos la opción **Métodos** de la barra de menú, y seguidamente accionaremos el elemento de menú **Sensibilidad de los parámetros...**, como se muestra en el capítulo II, ítem 2.2.7, esto provocará que se despliegue un cuadro de diálogo como el de la figura A.3.

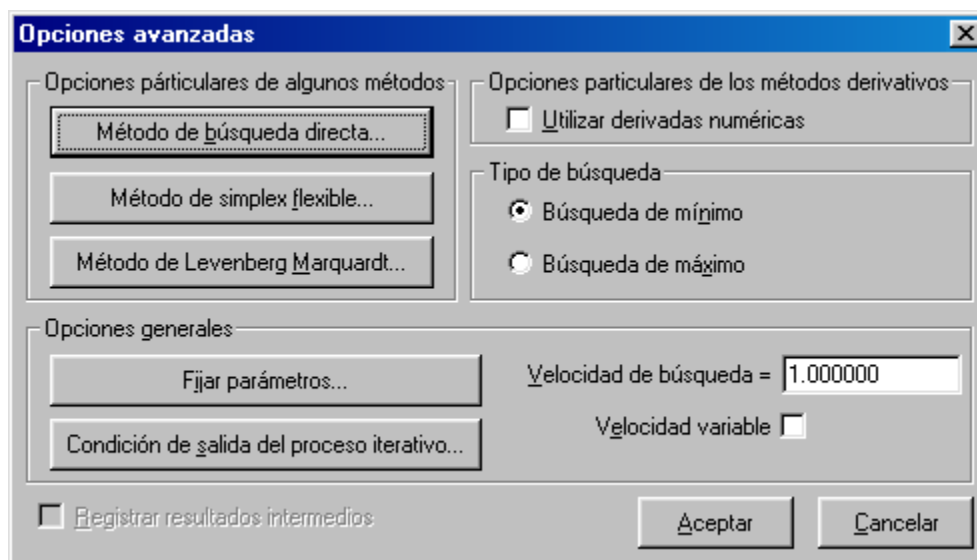


Figura A.2. El cuadro de diálogo Opciones avanzadas.

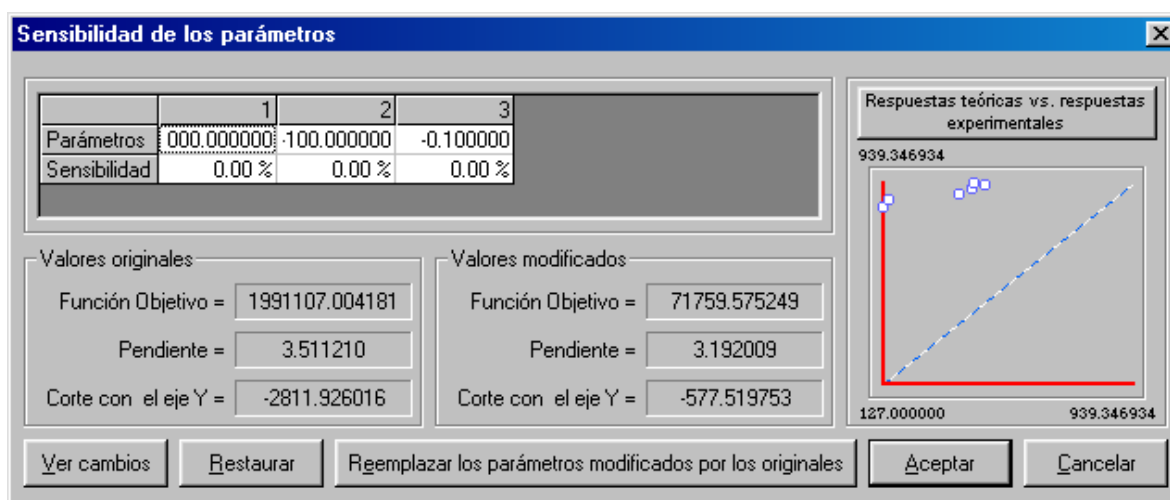


Figura A.3. El cuadro de diálogo inicial Sensibilidad de los parámetros.

14. Como se muestra en la figura A.3 podemos apreciar que los puntos de la gráfica de respuestas teóricas vs. respuestas experimentales tienden a recostarse un poco sobre el eje de las abscisas, y esto indica que las semillas no son un buen punto de partida. Un punto de partida aceptable es cuando los pares ordenados se encuentran esparcidos alrededor de la recta de pendiente uno, y no necesariamente deben ser cercanas a esta<sup>14</sup>. Finalmente, para seguir a la par con este ejemplo tomemos respectivamente los siguientes valores iniciales de los parámetros: 400, -110 y -0.1 y presionemos el botón **Reemplazar los parámetros modificados por los originales**. Ahora el cuadro de diálogo de verse como el de la figura A.4.

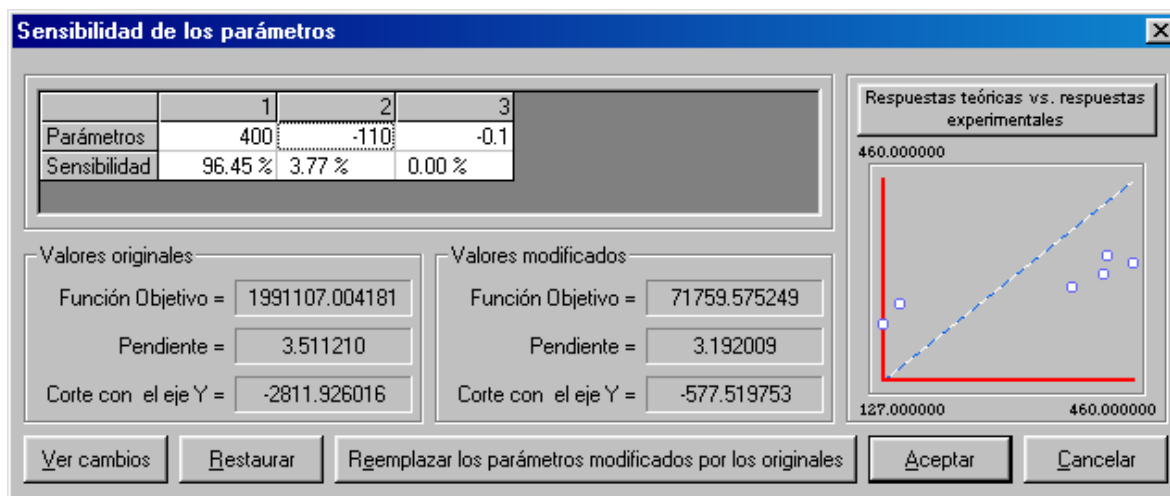


Figura A.4. El cuadro de diálogo final Sensibilidad de los parámetros.

Presione finalmente el botón **Aceptar**.

<sup>14</sup> En el caso que desee afinar aun mas los parámetros iniciales, trate a la vez que analiza la gráfica que la pendiente del ajuste tienda a uno y su corte a cero.

15. Ahora si que estamos listos para comenzar a aplicar algún proceso de optimización. Seleccione nuevamente la opción **Métodos** de la barra de menú, y seguidamente accione el elemento **Optimizador**, esto provocará que se despliegue el cuadro de diálogo **Optimizador** como se muestra en la figura A.5.

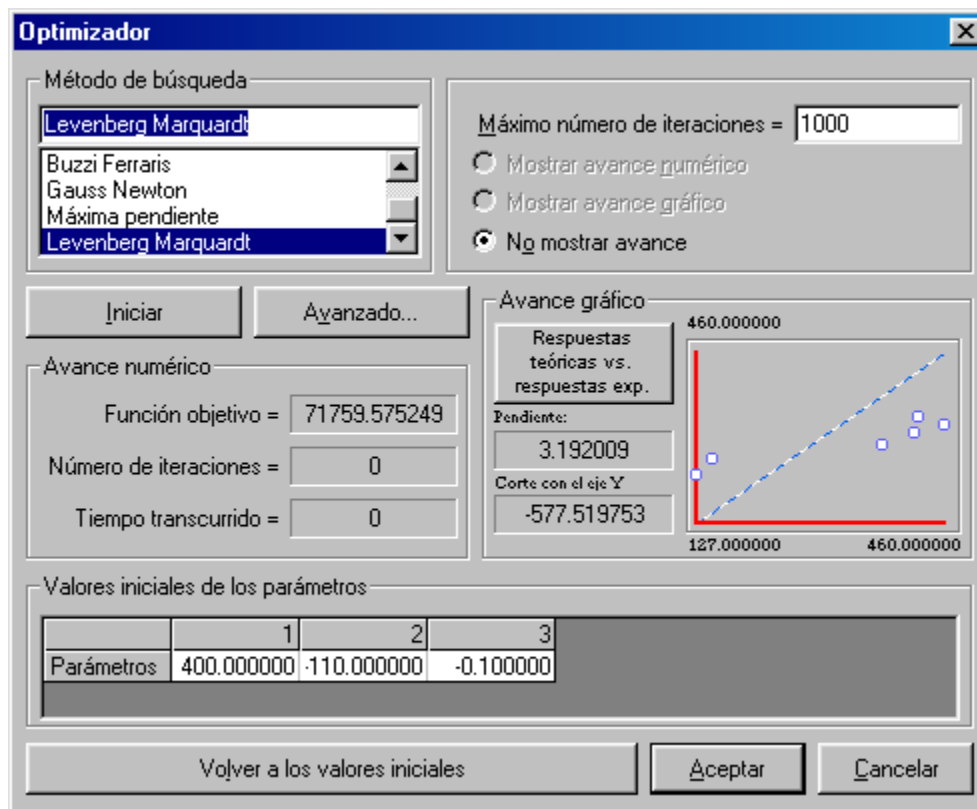


Figura A.5. El cuadro de diálogo inicial Optimizador.

La opción **Método de búsqueda** que aparece por defecto es **Levenberg Marquardt**, deje todas estas opciones como se encuentran inicialmente para el caso de este ejemplo y presione el botón **Iniciar**. El cuadro de diálogo que se muestra en la figura A.5 cambiará entonces a como se muestra en la figura A.6.

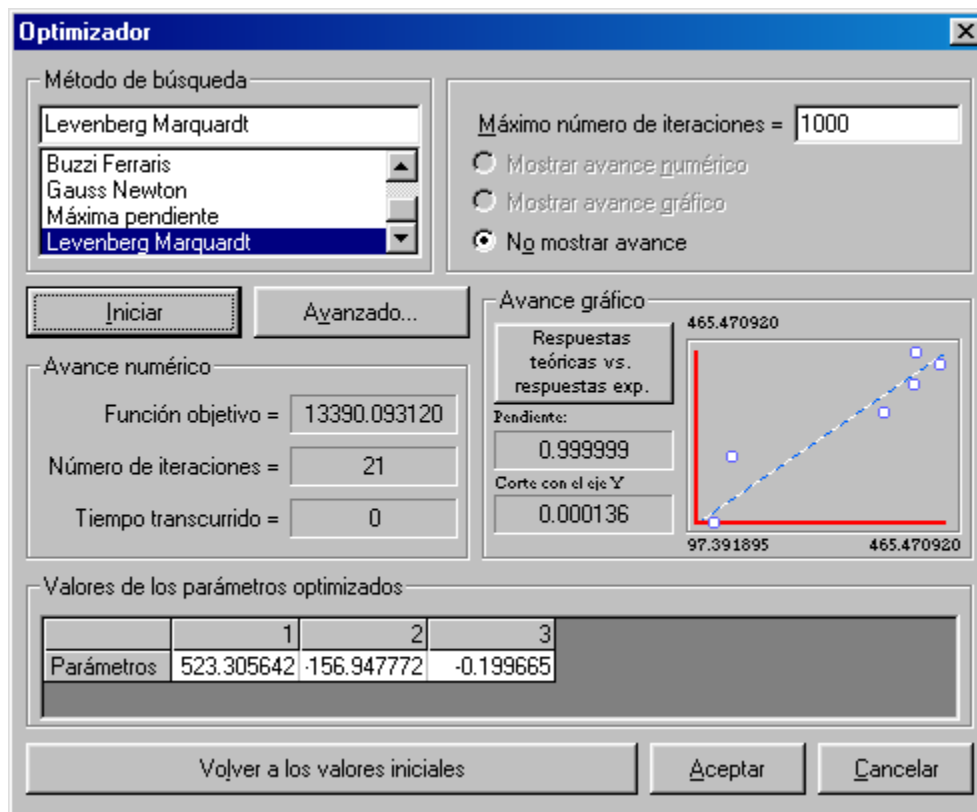


Figura A.6. El cuadro de diálogo final Optimizador.

Presione el botón **Aceptar**.

16. Ahora los elementos de menú **Generar informe...** y **Ver resultados gráficos...** aparecerán activados. Seleccione la opción **Herramienta** de la barra de menú y seguidamente accione el elemento **Generar informe...**, esto provocará que se despliegue el cuadro de diálogo **Generar informe**.
17. Por defecto aparecerá seleccionado como **Tipo de informe** la opción **Informe básico del último método empleado**, deje esta opción como se encuentra y desactive la casilla **Imprimir a archivo**, como se muestra en la figura A.7.

Presione el botón **Generar informe**.



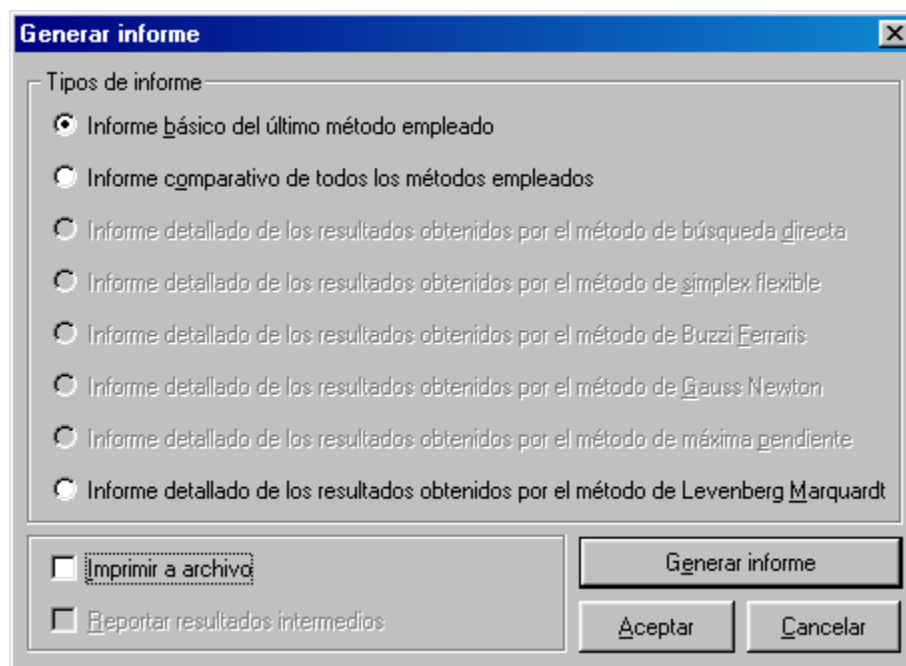


Figura A.7. El cuadro de diálogo Generar informe.

18. Una vez generado el informe la vista del documento cambiará de su modo presentación a su modo informe como se muestra en la figura A.8.

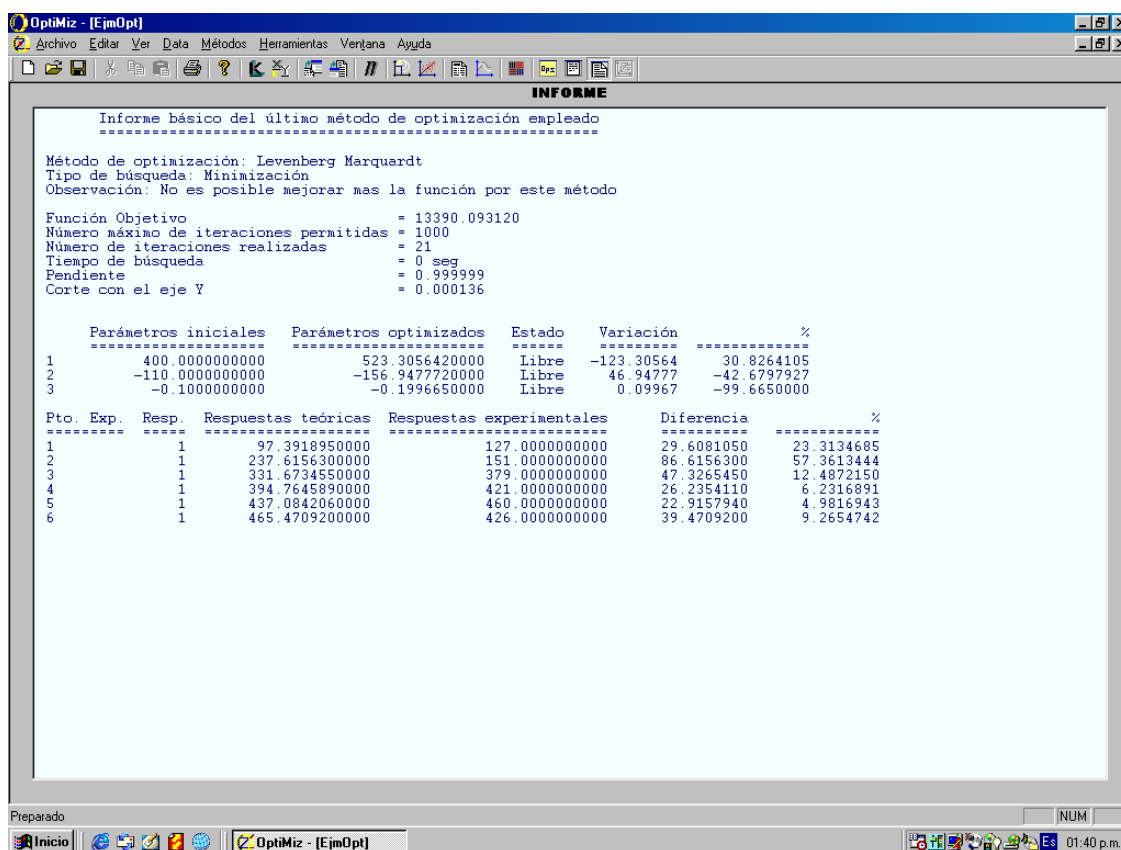


Figura A.8. Documento de la aplicación OptiMiz con el modo de vista informe.

Con este modo de vista puede apreciar los resultados obtenidos y analizar si estos son confiables o no.

19. Seleccione nuevamente la opción **Herramientas** de la barra de menú y accione el elemento **Ver resultados gráficos...**, esto provocará que se despliegue el cuadro de diálogo **Ver resultados gráficos**. Una de las opciones gráficas que dice mucho de la confiabilidad del ajuste es la opción **Número de experiencias vs. residuos porcentuales**, seleccione entonces esta opción y deje todo lo demás como esta, como se muestra en la figura A.9.

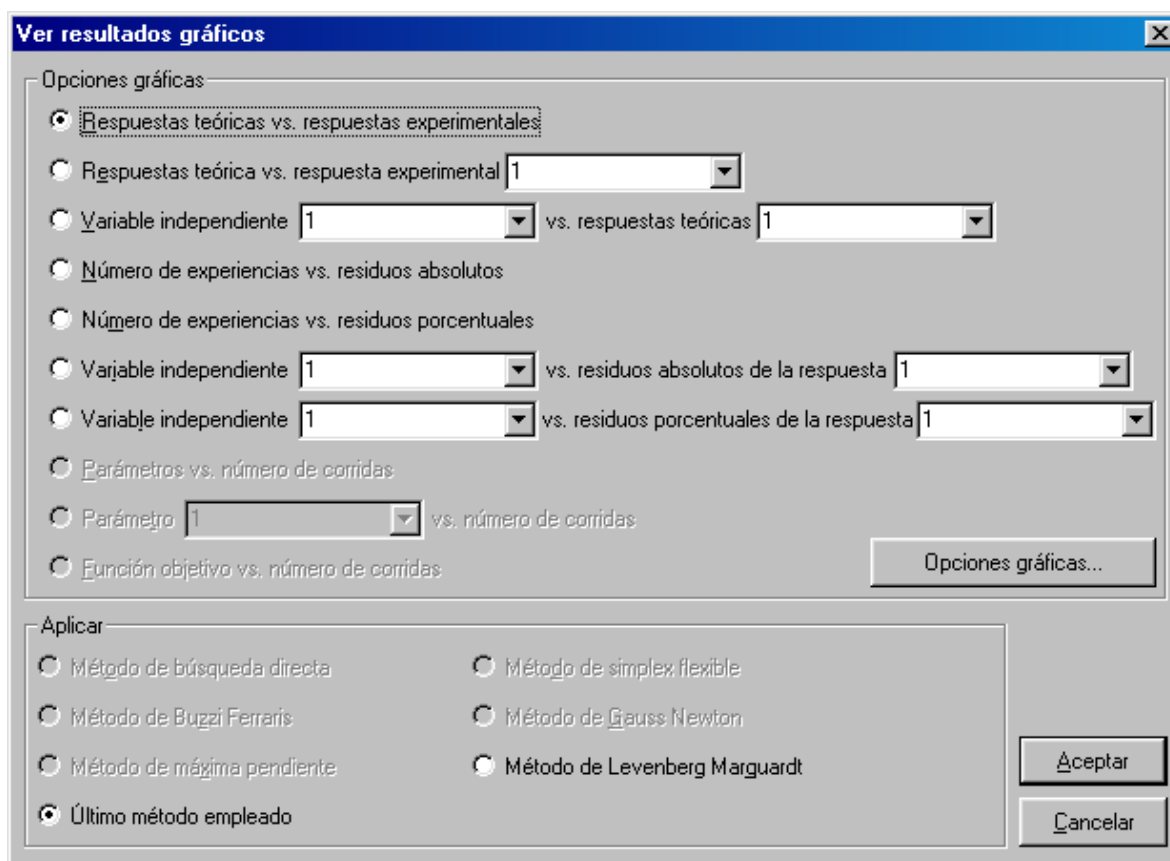
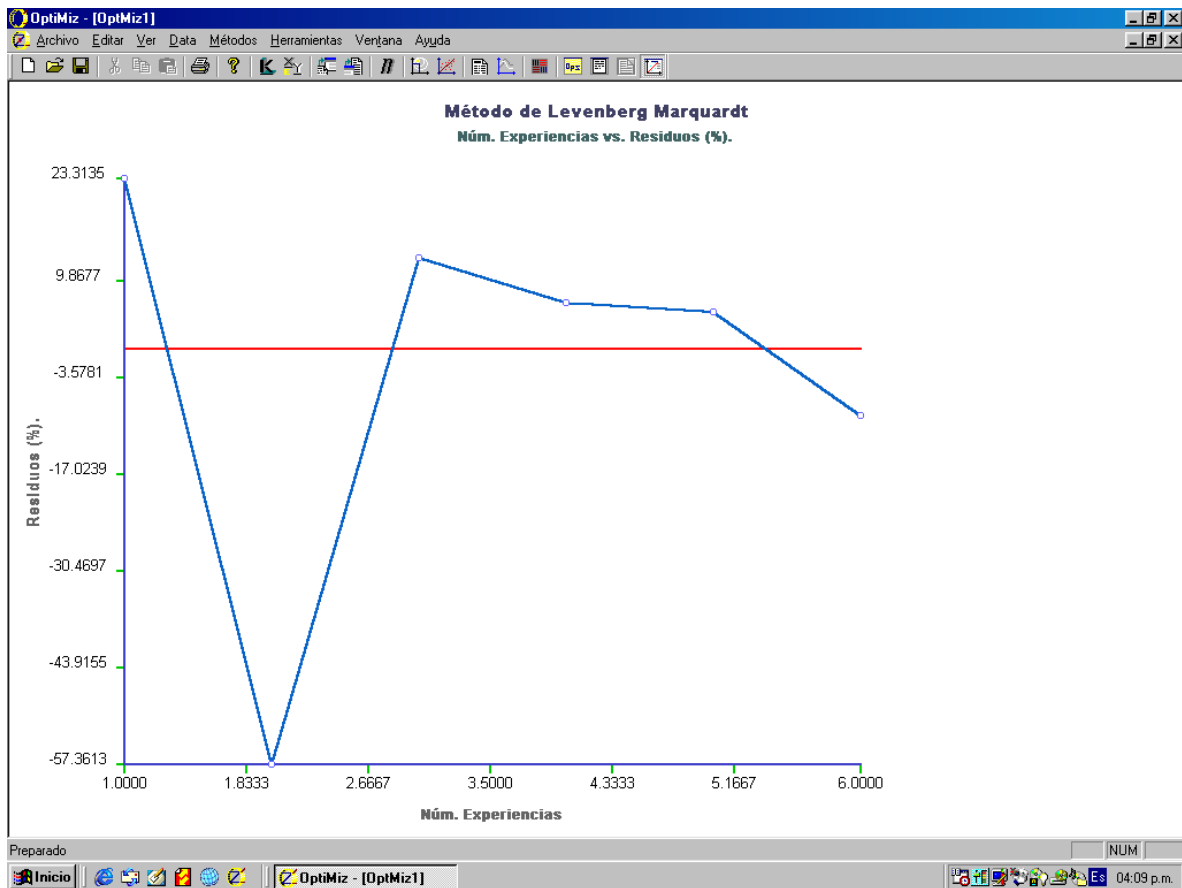


Figura A.9. El cuadro de diálogo Ver resultados gráficos.

Presione el botón Aceptar

20. Una vez generado el gráfico la vista del documento cambiará de su modo informe a su modo gráfico como se muestra en la figura A.10.



**Figura A.10. Documento de la aplicación OptiMiz con el modo de vista gráfico.**

Con este modo de vista puede apreciar los resultados obtenidos y analizar si estos son confiables o no, pero ahora desde un punto de vista gráfico.