

Erica Canizo Paola Lucero	Investigación Operativa 2002 Software Para Programación Lineal – LINGO/LINDO–
------------------------------	---

Presentaremos dos tipos de programas para resolver modelos de programación lineal.

## LINGO

### Generalidades

LINGO: (LINear Generalize Optimizer) es una herramienta simple para formular problemas lineales y no lineales, resolverlos y analizar su solución. El resultado que LINGO nos proporciona es la optimización que nos ayuda a encontrar el mejor resultado: la ganancia más alta, o el costo más bajo. A menudo estos problemas involucran el uso más eficiente de los recursos. Los problemas de optimización son clasificados a menudo como lineales o no lineales, dependiendo si las relaciones en el problema son lineales con respecto a las variables.

Uno de los rasgos más poderosos de LINGO es su aplicación en el lenguaje de modelo matemático. El cual permite expresar un problema de una manera muy similar a la anotación matemática normal pudiendo también, expresar una serie entera de restricciones en una declaración compacta. Esto lleva a modelos que son mucho más fáciles de mantener.

Otro aspecto es la sección de los datos, que le permite aislar los datos de la formulación del modelo. De hecho LINGO puede leer datos incluso de una hoja de cálculo separada, base de datos, o archivo de texto. Con datos independientes del modelo, es mucho más fácil de hacer cambios, y hay menos oportunidad de error cuando se realiza el modelo.

### Sintaxis de LINGO

La sintaxis que se utiliza en este programa es muy sencilla. Para el nombre de las variables se establece que deben tener 32 caracteres como máximo, Deben comenzar con una letra seguido de letras, dígitos o `_`. El compilador de LINGO no distingue entre mayúsculas y minúsculas.

Con respecto a las sentencias:

Todas las sentencias deben terminar en un punto y coma.

Para darle un nombre a la función objetivo o a las restricciones, estos se deben colocar entre corchetes.

Para declarar la función objetivo debemos colocar las palabras reservadas **MAX** o **MIN**, resaltadas en azul, seguidas del signo `=`.

Los comentarios deben comenzar con un signo `!`, los cuales son resaltados en verde.

Erica Canizo Paola Lucero	Investigación Operativa 2002 Software Para Programación Lineal – LINGO/LINDO–
------------------------------	---

Los archivos generados por LINGO tiene la extensión. LG4.

## Nombre de las restricciones

LINGO tiene la habilidad de nombrar las restricciones en su modelo. Ésta es una práctica buena por dos razones. Primero, los nombres de restricciones se usan en el reporte de las soluciones que los hacen más fácil interpretar. Segundo, muchos de los mensajes del error de LINGO se refieren a una restricción dada por nombre.

Dar nombre a una restricción es bastante simple. se inserta el nombre entre corchetes, adelante de una línea de código. El nombre debe obedecer los requisitos normales para un nombre de LINGO.

## Uso de funciones de dominio de variables

A menos que especificifique lo contrario, el valor de las variables por defecto en un modelo de LINGO son no-negativo y continuas. Más específicamente, las variables pueden asumir algún valor real desde cero a infinito positivo. En muchos casos, este dominio de valor por defecto puede ser impropio. Por ejemplo, usted puede querer una variable que asuma valores negativos, o se podría querer una variable restringida puramente a valores enteros. LINGO proporciona cuatro funciones de variables dominio que le permite sustituir el dominio predefinido de una variable. Los nombres de estas funciones y una descripción breve de su uso son:

- @GIN restringe una variable para comenzar con valores enteros,
- @BIN hace una variable binario (es decir, 0 o 1). Por ejemplo @BIN( X);
- @FREE permite que una variable pueda asumir algún valor real, positivo o negativa
- @BND limita una variable dentro de un rango finito.

## » Variables enteras y binarias

LINGO le da la posibilidad de definir dos tipos de variables enteras, una general y otra binaria. Una variable entera general requiere ser un número entero. Una variable entero binaria requiere ser cero o uno. Cualquier modelo que contiene uno o más variables enteras, es requerido para un modelo programación entera (IP)  
En muchos proyectos de modelos, usted se enfrentará con tipos de decisiones (si/no). Algunos ejemplos incluirían Produce/No Produce, Abre un Plan/Cierra un Plan, etc. Las variables binarias son el método normal usado por modelar estas decisiones de si/no.

Erica Canizo Paola Lucero	Investigación Operativa 2002 Software Para Programación Lineal – LINGO/LINDO–
------------------------------	---

## » Variables libres

Por defecto, las variables en LINGO tiene un límite inferior de cero y un límite superior de infinito. @FREE quita el límite inferior cero y permite que la variable tome valores negativos. La sintaxis es:

@FREE (variable\_name);

donde la variable\_name es el nombre de la variable libre.

La función de @FREE puede usarse en cualquier parte del modelo donde normalmente iría restricción. Algunos ejemplos de @FREE son:

Ejemplo 1: @FREE (X);

hace una variable X libre

## » Variables limitadas

Considerando que @FREE pone el límite superior e inferior de la variable especificada a más-menos infinito (quitando cualquier límite en la variable), la función de @BND le permite poner límite superior e inferior específicos en una variable. En otras palabras, @BND limita el rango de una variable dentro de algún intervalo. La sintaxis para @BND es:

@BND (lower\_bound, variable\_name, el upper\_bound);

donde el variable\_name es la variable a ser limitada debajo por el lower\_bound y limitado superiormente por el upper\_bound. Lower\_bound y " upper\_bound deben ser valores numéricos o variables cuyos valores han sido fijados en la sección de datos. @BND puede usarse en cualquier lugar que normalmente se usaría una restricción en el modelo. En términos matemáticos, LINGO interpreta que este @BND funcionan como:

lower\_bound <= variable\_name <= upper\_bound

Además, @BND no cuenta contra el límite con el número total de JERGA de constreñimiento impone en algunas versiones. En general, se usa @BND en lugar de una restricción siempre que sea posible.

Algunos ejemplos de @BND son:

Ejemplo 1: @BND (-1, X, 1);

## Operadores y Funciones

LINGO proporciona varias funciones y operadores al modelo matemático. Se muestran las categorías siguientes:

Los Operadores Normales: Aritmética, lógicos, y correlativos como +, -, =, <=.

PRIORITYLEVEL	OPERATOR(S)
Highest	#NOT# - (negation)
	^
	* /
	+ -
	#EQ# #NE# #GT# #GE# #LT# #LE#
	#AND# #OR#
Lowest	<= = >=

Mathematical - las funciones matemáticas Trigonométricas y generales.  
Ej @ABS( X)

Financial - las funciones financieras

Probabilidad de " - las Funciones utilizadas para determinar una gama amplia de probabilidad y las respuestas estadísticas. Poisson

Dominio de variables - son funciones que definen el rango de valores (dominio) que una variable puede asumir

Manejadores de conjuntos - las Funciones útil para manipular conjuntos.

" Set Looping - Funciones loop que se utilizan para realizar operaciones en un conjunto de datos

" Import/Export - son funciones que crean conexiones con las fuentes de los datos externas.

" Miscellaneous - funciones de distintos tipos.

Erica Canizo Paola Lucero	Investigación Operativa 2002 Software Para Programación Lineal – LINGO/LINDO–
------------------------------	---

Para explicar el funcionamiento del programa tomaremos el siguiente ejemplo:

## Problema

### 1. ENUNCIADO

La empresa WINDOOR Ltda. fabrica puertas y ventanas de madera. Existen dos modelos de puertas: puertas dobles y puertas simples; y dos modelos de ventanas: ventanas dobles y ventanas simples. El insumo más importante es la madera. El proceso de corte de las partes se realiza en dos sierras eléctricas de precisión y el barnizado lo efectúa personal experimentado. Las cantidades de madera y los tiempos de corte y barnizado que requiere cada producto se muestran en la Tabla 1.

Tabla 1. Recursos requeridos.

	<i>Madera [m<sup>2</sup>]</i>	<i>Ensamblado [hrs-maq]</i>	<i>Barnizado [hrs-hom]</i>
<i>Puertas dobles</i>	4.0	1.5	2.0
<i>Puertas simples</i>	1.5	1.0	1.2
<i>Ventanas dobles</i>	3.0	2.0	1.5
<i>Ventanas simples</i>	1.8	0.8	0.8

Los proveedores de la madera pueden entregar hasta 800 [m<sup>2</sup>] en un mes. Además se pueden utilizar hasta 400 [hrs] de sierra para el corte y 400 [hrs-hom] para el barnizado.

La empresa está comprometida con una constructora para entregar 90 puertas simples y 50 ventanas dobles en el mes. Los precios de venta unitarios y costos unitarios de producción se muestran en la Tabla 2.

Tabla 2. Precios y costos.

	<i>Precio [um/unidad]</i>	<i>Costo [um/unidad]</i>
<i>Puertas dobles</i>	120	80
<i>Puertas simples</i>	80	50
<i>Ventanas dobles</i>	100	75
<i>Ventanas simples</i>	60	30

Se desea determinar el plan de producción para el mes que maximice el beneficio total, suponiendo que lo que se produce se vende.

Erica Canizo Paola Lucero	Investigación Operativa 2002 Software Para Programación Lineal – LINGO/LINDO–
------------------------------	---

## 2. FORMULACION DEL MODELO

### VARIABLES DE DECISION

Las decisiones que la empresa desea tomar se refieren a la cantidad de puertas y ventanas de cada tipo a producir en el período considerado.

$x_1$  = cantidad de puertas dobles a fabricar en el mes.  
 $x_2$  = cantidad de puertas simples a fabricar en el mes.  
 $x_3$  = cantidad de ventanas dobles a fabricar en el mes.  
 $x_4$  = cantidad de ventanas simples a fabricar en el mes.

### RESTRICCIONES

Las restricciones deben establecer las limitaciones existentes en cuanto a la disponibilidad de recursos y la necesidad de cumplir con los compromisos contraídos para el período.

#### A. Disponibilidad de recursos

- La cantidad de madera utilizada en la producción total no puede exceder la cantidad máxima que la empresa puede adquirir.

$$4.0 \cdot x_1 + 1.5 \cdot x_2 + 3.0 \cdot x_3 + 1.8 \cdot x_4 \leq 800$$

- La cantidad de horas máquina de sierra utilizada no puede exceder la cantidad máxima disponible.

$$1.5 \cdot x_1 + 1.0 \cdot x_2 + 2.0 \cdot x_3 + 0.8 \cdot x_4 \leq 400$$

- La cantidad de horas hombre para barnizado utilizada no puede exceder la cantidad máxima disponible.

$$2.0 \cdot x_1 + 1.2 \cdot x_2 + 1.5 \cdot x_3 + 0.8 \cdot x_4 \leq 400$$

#### B. Cumplimiento de compromisos de entrega

- La cantidad de puertas simples debe ser al menos suficiente para cumplir los compromisos contraídos.

$$x_2 \geq 90$$

- La cantidad producida de ventanas dobles debe ser al menos suficiente para cumplir con los compromisos contraídos.

$$x_3 \geq 50$$

#### C. No negatividad de las variables

$$x_1, x_2, x_3, x_4 \geq 0$$

### FUNCION OBJETIVO

Como se dijo en el enunciado el problema consiste en maximizar el beneficio durante el mes de operación.

$$\text{Beneficio total} = \text{ingreso total} - \text{costo total}$$

Hay que destacar que, tanto el beneficio como el costo total, dependen de las cantidades que finalmente se decida producir, es decir, dependen de las variables que se definieron.

$$B_{\text{total}} = 40 \cdot x_1 + 30 \cdot x_2 + 25 \cdot x_3 + 30 \cdot x_4$$

## RESUMEN

En resumen se tiene el siguiente *modelo matemático* para el problema planteado:



$$\begin{aligned}
 &\text{Max } 40 \cdot x_1 + 30 \cdot x_2 + 25 \cdot x_3 + 30 \cdot x_4 \\
 &4.0 \cdot x_1 + 1.5 \cdot x_2 + 3.0 \cdot x_3 + 1.8 \cdot x_4 \leq 800 \\
 &1.5 \cdot x_1 + 1.0 \cdot x_2 + 2.0 \cdot x_3 + 0.8 \cdot x_4 \leq 400 \\
 &2.0 \cdot x_1 + 1.2 \cdot x_2 + 1.5 \cdot x_3 + 0.8 \cdot x_4 \leq 400 \\
 &x_2 \geq 90 \\
 &x_3 \geq 50 \\
 &x_1, x_2, x_3, x_4 \geq 0
 \end{aligned}$$

The screenshot shows the LINGO software window titled "LINGO - [LINGO Model - EJEMPLO]". The interface includes a menu bar (File, Edit, LINGO, Window, Help) and a toolbar with various icons. The main text area contains the following model description:

```

!VARIABLES DE DESICIÓN
x1 = cantidad de puertas dobles a fabricar en el mes.
x2 = cantidad de puertas simples a fabricar en el mes.
x3 = cantidad de ventanas dobles a fabricar en el mes.
x4 = cantidad de ventanas simples a fabricar en el mes. ;

!FUNCIÓN OBJETIVO;
[Maximizar Ganancias] Max = 40*x1 + 30*x2 + 25*x3 + 30*x4;

!RESTRICCIONES;

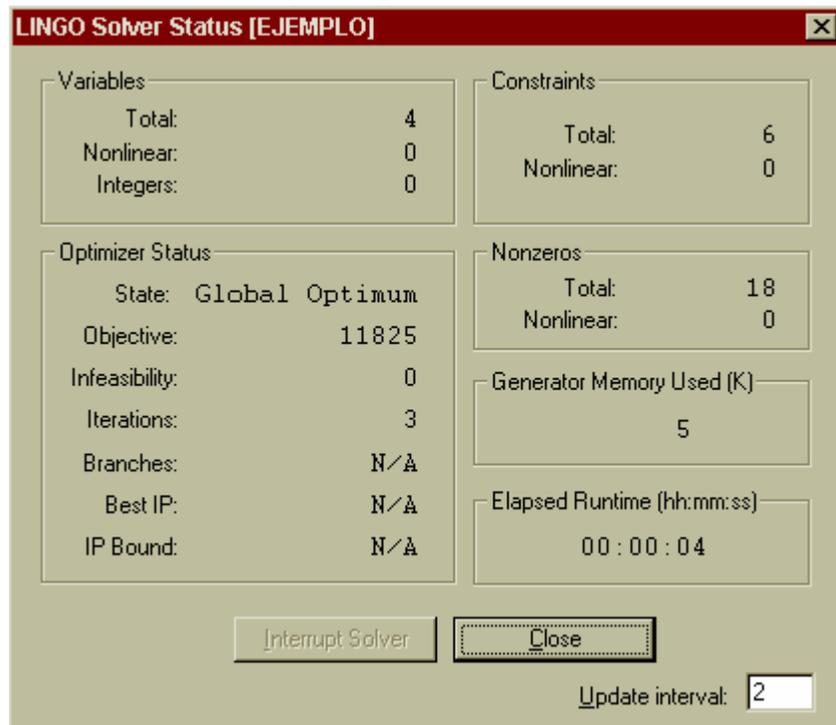
[Disponibilidad Recursos1 ] 4.0*x1 + 1.5*x2 + 3.0*x3 + 1.8*x4 <= 800; !Recurso: madera utilizada;
[Disponibilidad Recursos2 ] 1.5*x1 + 1.0*x2 + 2.0*x3 + 0.8*x4 <= 400; !Recurso: hs/máquina;
[Disponibilidad Recursos3 ] 2.0*x1 + 1.2*x2 + 1.5*x3 + 0.8*x4 <= 400; !Recurso: hs/hombre;
[Compromisos Entrega1 ] x2 >= 90; !cantidad de puertas simples;
[Compromisos Entrega2] x3 >= 50; !cantidad de ventanas dobles;
  
```

The status bar at the bottom shows "Ready", "CAP NUM", "Ln 21, Col 1", and "4:00 am".

## Estados del Solver: Solver Status

Si no hay errores de formulación durante la fase de compilación, LINGO invocará el Solver interno apropiado para comenzar a buscar la solución óptima del modelo.

La ventana de estado incluye lo siguiente:



La ventana de estado del Solver Status es útil para controlar el progreso del solver.

- [El box de Variables](#): muestra el número total de variables en el modelo, junto con el número de las variable no lineales. Una variable se considera no lineal cuando existe una relación no lineal en alguna restricción. También nos muestra el número de variable enteras en el modelo. Mientras más cantidad de variables no lineales y enteras exista en el modelo más tiempo tomará su resolución. Los modelos puramente lineales se resuelven con mayor velocidad.

Erica Canizo Paola Lucero	Investigación Operativa 2002 Software Para Programación Lineal – LINGO/LINDO–
------------------------------	---

### Aclaración “Clasificación de los modelos matemáticos”

- A nivel de problemas: Los problemas pueden ser irrestrictos (carecen de restricciones) o restringidos (son aquellos que tienen una o más restricciones)
- A nivel de restricciones: A la vez estas pueden ser lineales (satisfacen la regla de aditividad como de proporcionalidad) o no lineales (en la cual alguna restricción no satisface la regla de proporcionalidad y de aditividad).
  - ◆ Regla de aditividad: Se suman o restan las variables que conforman las restricciones.
  - ◆ Regla de proporcionalidad: se multiplican por una constante las variables que conforma las restricciones.
- A nivel de la función objetivo: A la vez la función puede ser lineal o no lineal.
- A nivel de variables: Por último estas pueden ser continuas (todas las variables satisfacen la divisibilidad) o enteras (en la que una o más variables deben tener valores enteros).

\* [El box de restricciones \(Constraints\)](#): Nos muestra el número total de restricciones y también el número de restricciones no lineales.

\* [El box Nonzeros](#): Muestra el número total de coeficientes no ceros en el modelo, como así también el número de estas que son de tipo no lineal. Generalmente en una restricción un pequeño subconjunto de las variables típicas aparece con el coeficiente implícito, en las variables que no aparecen es cero; mientras que en las variables que aparecen es distinto de cero.

\* [El box Optimizer Status muestra:](#)

- » [Campo Estado](#): Cuando LINGO comienza a resolver un modelo el estado inicial de la solución es “indeterminado” ya que todavía no se ha generado ninguna solución para el modelo. Una vez que se comienza a iterar el estado pasa a “no factible” en donde se han generado soluciones tentativas pero que no satisfacen las restricciones del modelo. Y luego se pasa a un estado “factible” cuando se encuentra una solución que satisface todas las restricciones pero todavía no se encuentra la mejor solución. Cuando el solver no encuentra una mejor solución para el modelo se llega al estado “óptimo global” u “óptimo local”. Esto se cumple para modelos con restricciones lineales y no lineales respectivamente.

Si un modelo termina en un estado “ilimitado” significa que se pudo mejorar la función objetivo sin límites y esto en la vida real correspondería a una situación de ganancias infinitas algo imposible lo cual significa que se ha omitido algún tipo de restricción en el modelo. Finalmente el estado “interrumpido” es cuando se interrumpe prematuramente el solver antes de que encuentre la mejor solución para el modelo.

Erica Canizo Paola Lucero	Investigación Operativa 2002 Software Para Programación Lineal – LINGO/LINDO–
------------------------------	---

- » Campo Objetivo: El campo objetivo nos da el valor objetivo de la solución actual. Si el modelo no tiene una función objetivo entonces N/A aparece en este campo.
- » Campo No Factible: Muestra la cantidad por la cual todas las restricciones han sido violadas.
- » Campo de Iteraciones: El Campo de las Iteraciones despliega una cuenta del número de iteraciones completada por el solver de LINGO. La operación fundamental realizado por el este se llama iteración. Una iteración involucra una variable, actualmente cero a un valor que sería atractivo introducir en la solución a un valor distinto de cero. Esta variable se introduce entonces en la solución consecutivamente con valores más grandes hasta alguna restricción. En general, cuando un modelo es considerable, exigirá a más iteraciones a resolver y cada iteración requerirá más tiempo para completar.
- » Campo Branches: LINGO usa la estrategia llamada branch-and-bound para resolver modelos de programación entera
- » Mejor IP campo: Este campo muestra el valor objetivo de la mejor solución en un modelo de programación entera.

La ventana de estado del solver también provee un botón de interrupción que permite interrumpir el solver en la próxima iteración.

### Reporte de solución (Solution Report)

El primer informe del reporte nos dice el **número de iteraciones** para resolver el modelo. Segundo, **el valor que maximiza o minimiza función objetivo**. Tercero, **las cantidades de cada variable** que responden a la función objetivo.

Global optimal solution found at step: 3  
Objective value: 11825.00

Variable	Value	Reduced Cost
X1	0.0000000	16.25000
X2	90.00000	0.0000000
X3	50.00000	0.0000000
X4	262.5000	0.0000000

Row	Slack or Surplus	Dual Price
MAXIMIZARGANANCIAS	11825.00	1.0000000
DISPONIBILIDADRECURSOS1	42.50000	0.0000000
DISPONIBILIDADRECURSOS2	0.0000000	37.50000
DISPONIBILIDADRECURSOS3	7.0000000	0.0000000
COMPROMISOENTREGA1	0.0000000	-7.500000
COMPROMISOENTREGA2	0.0000000	-50.00000

Ready NUM MOD Ln 18, Col 1 3:31 am

- » **Costo reducido:** Primero, usted puede interpretar el costo reducido de una variable como la cantidad por la que el coeficiente objetivo de la variable tendría que mejorar antes de que se pusiera aprovechable para dar la variable en cuestión un valor positivo en la solución óptima. La variable  $X_1$  tiene un costo reducido de 16.25, el coeficiente objetivo de esta variable tiene que aumentar a través de 16.25 unidades en este problema. Si el problema fuera de minimización la variable debería disminuir en 16.25 unidades para que esta se vuelva una alternativa atractiva para entrar en la solución. Segundo, el costo reducido de una variable puede interpretarse como la cantidad de multa que se tendría que pagar para introducir  $x_1$  en la solución.

Erica Canizo Paola Lucero	Investigación Operativa 2002 Software Para Programación Lineal – LINGO/LINDO–
------------------------------	---

Variable	Value	Reduced Cost
X1	0.0000000	16.25000
X2	90.00000	0.0000000
X3	50.00000	0.0000000
X4	262.5000	0.0000000

- » **La columna de holgura o sobrante (slack o surplus):** Esta parte del informe de solución nos dice cuan cerca estamos de satisfacer una restricción como una igualdad. Si la restricción es de menor-igual nos referimos a variables de slack y si es de mayo-igual nos referimos a variables surplus.

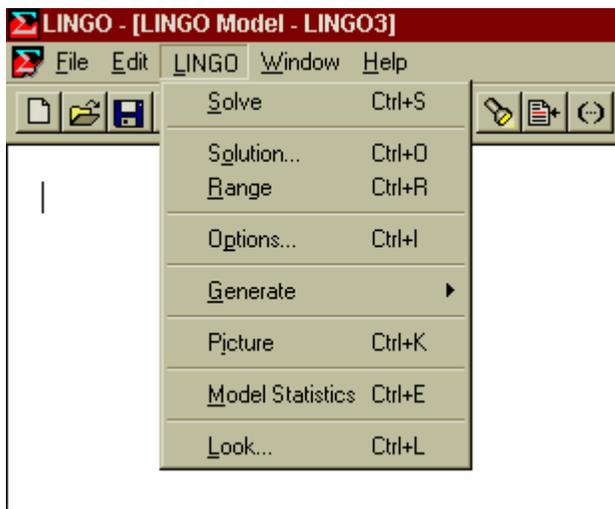
Row	Slack or Surplus
MAXIMIZARGANANCIAS	11825.00
DISPONIBILIDADRECURSOS1	42.50000
DISPONIBILIDADRECURSOS2	0.0000000
DISPONIBILIDADRECURSOS3	7.000000
COMPROMISOSENTREGA1	0.0000000
COMPROMISOSENTREGA2	0.0000000

Si la restricción está correctamente formulada se satisface como una igualdad, el valor de holgura o sobrante será cero. De lo contrario la solución será no factible, la holgura o valor sobrante será negativo. En nuestro ejemplo en la restricción de disponibilidad de recursos 1(madera); para que se satisfaga la igualdad debemos adicionar una variable slack con el valor de 42.5 y en la restricción de disponibilidad de recursos 3 (hs. Hombres); debemos adicionar una variable slack con valor 7.

- » **Precio dual:** El informe de solución también nos da el precio dual para cada restricción. Se puede interpretar el precio dual, como la cantidad en la que el objetivo mejoraría por el término constante de la restricción aumentado en una unidad. Retomando el ejemplo, en la solución del problema, el precio dual de 37.5 en la disponibilidad de recursos 2 (hs. Máquina) si agregamos una unidad más de trabajo causaríamos que el valor objetivo mejorara en 37.5 unidades, a un valor de 11.862,5. El término mejorar es relativo. En un problema de maximización, mejorar significa que el valor objetivo aumentaría. Sin embargo, en un problema del minimización, el valor objetivo disminuiría. A veces se llaman a los precios duales precios sombra.

	Row	Dual Price
	MAXIMIZARGANANCIAS	1.000000
	DISPONIBILIDADRECURSOS1	0.000000
	DISPONIBILIDADRECURSOS2	37.50000
	DISPONIBILIDADRECURSOS3	0.000000
	COMPROMISOENTREGA1	-7.500000
	COMPROMISOENTREGA2	-50.00000

**El Menú de comandos LINGO:**



SOLVE	RESUELVE AL MODELO EN LA VENTANA ACTUAL
Solution	Genera una ventana de informe de solución para el modelo actual
Genera un informe de análisis de rango para la ventana actual	Genera un informe de análisis de rango para la ventana actual
Option	Pone opciones del sistema
Generate	Genera la representación algebraica para el modelo actual
Picture	Despliega una vista gráfica de la matriz del modelo

Erica Canizo Paola Lucero	Investigación Operativa 2002 Software Para Programación Lineal – LINGO/LINDO–
------------------------------	---

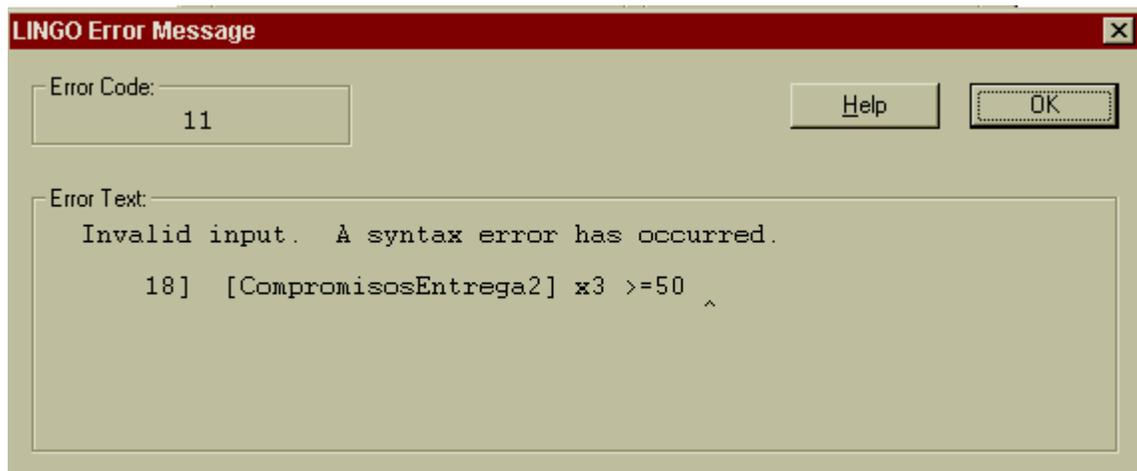
Model Statics	Despliega un informe breve con respecto a los detalles técnicos de un modelo
Look	Genera un informe de formulación para la ventana actual

## COMANDOS MÁS UTILIZADO EN EL MENU LINGO

### LINGO\Solve

El comando Solve se utiliza para resolver el modelo en la ventana activa.

Cuando se resuelve un modelo, LINGO examina la sintaxis del modelo primero para determinar si es válida. Si se encuentra un error de [sintaxis](#), se presentará un mensaje de error en donde se imprime el número de la línea en la que el error de la sintaxis ocurrió, el texto de la línea, y una marca (^) en donde LINGO piensa que el error ocurrió. En la mayoría de los casos, se apunta exactamente a donde el error ocurrió. Ejemplo

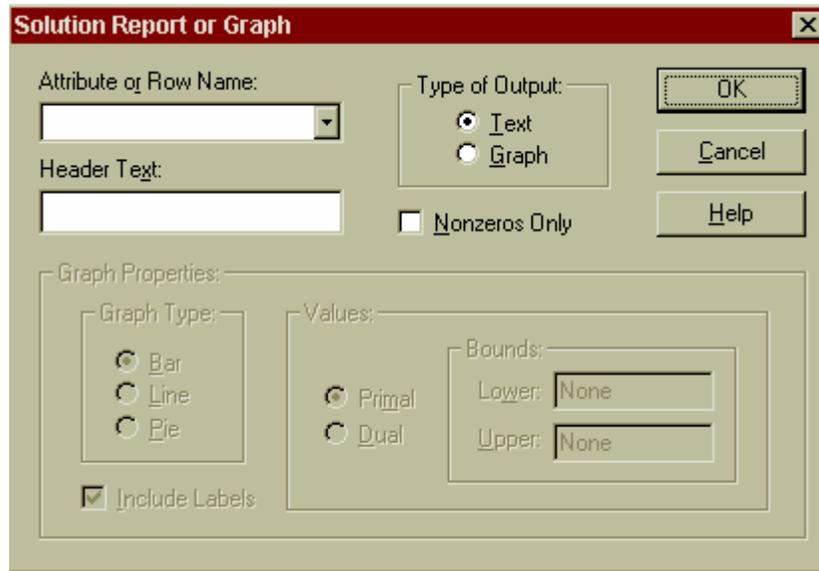


El error producido se debe a la falta de punto y coma al final de la sentencia; como en la pantalla lo podemos visualizar con la marca (^)

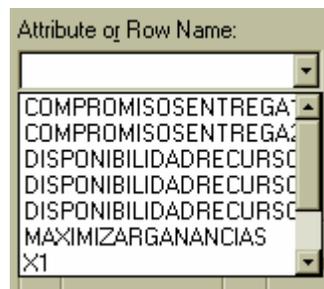
Cuando se emite el comando Solve (asumiendo que el modelo no tiene ningún error de sintaxis), se desplegará la ventana de estado Solver Status. Como ya se detalló anteriormente. Una vez que la opción Solve ha completado el proceso del modelo, creará una nueva ventana que contiene el informe de la solución para su modelo.

## LINGO|Solution

El comando Solution se usa para generar un informe de la solución para la ventana activa. El informe de la solución puede estar en formato de texto o gráfico. Luego de emitir el comando Solve y generado el reporte de solución se selecciona la ventana del modelo y se emite este comando.

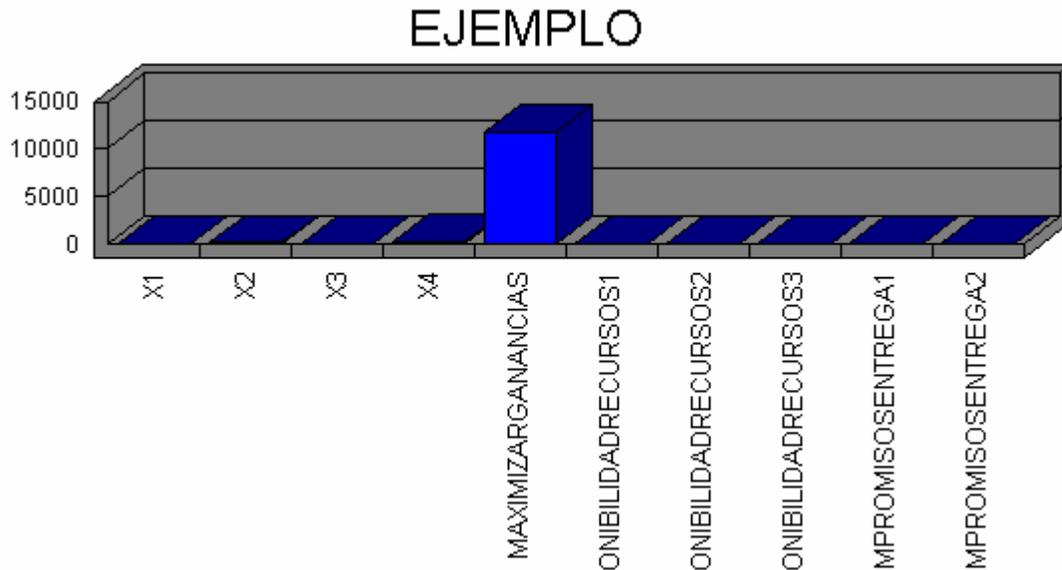


En el box de Attribute or Row Name, se selecciona la variable para la que deseamos el informe. Si no se selecciona un nombre en este box, LINGO generará un informe de la solución que incluye todos los atributos.



En el box de Header Text, se ingresa el texto que se desea que aparezca en el informe.

En el box Type of Output se puede seleccionar la salida de texto o de gráfico. Si se selecciona texto, LINGO creará una nueva ventana que contiene la solución en formato del texto. Si se selecciona gráfico se creará una nueva ventana que contiene la solución en uno de varios formatos gráficos diferentes. Se soporta formato de gráficos de barra, línea, y de círculos. Esta opción se selecciona en el box de Graphics properties, en graphics type.



En el box de nonzeros Only se selecciona si las variables que aparecen en el reporte tienen valores distintos de cero.

En el Values box se puede seleccionar valores dual u primal.

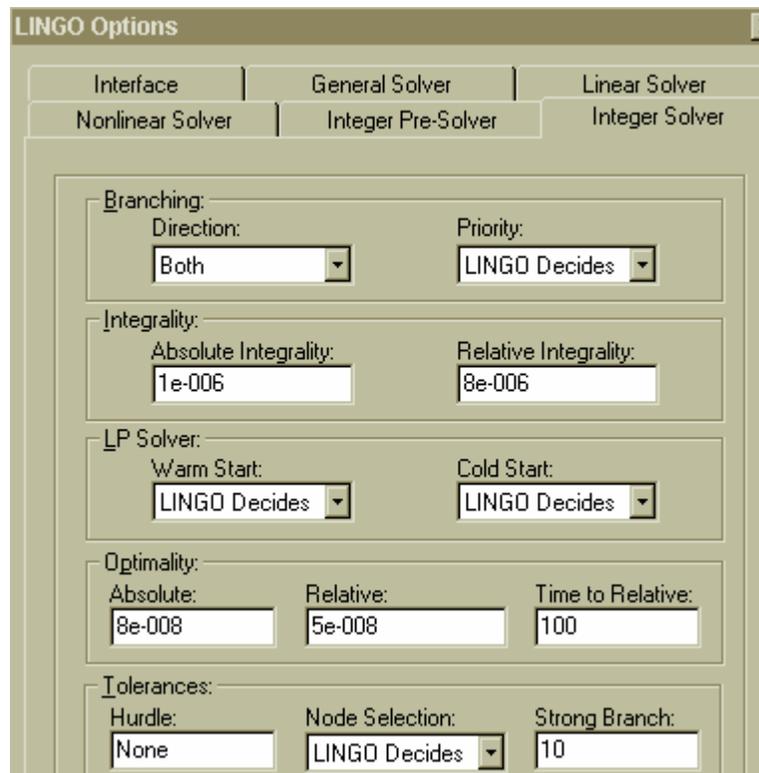
El box Bounds da la opción de poner límites en los valores desplegados en el gráfico.

## **LINGO|Range**

El comando Range sirve para generar un informe del rango para el modelo en la ventana activa. Un informe del rango muestra encima de qué rangos que usted puede, 1) el cambio un coeficiente en el objetivo sin causar cualquiera de los valores óptimos de las variables de decisión puedan cambiar, o, 2) el cambio el término constante de una fila (también llamado el coeficiente lateral diestro) sin causar cualquiera de los valores óptimos de los precios duales o costos reducidos puedan cambiar.

## LINGO|Options

El comando Options se utiliza para cambiarlos parámetros que afectan la interface del usuario de LINGO, como así también la manera en que se resuelve el modelo

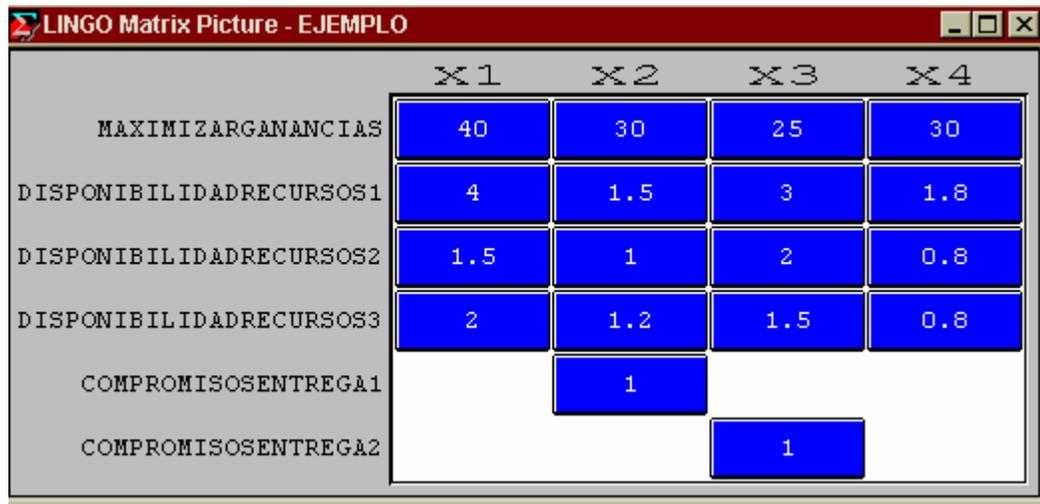


Hay seis etiquetas en las Opciones:

- Interface,
- Solver General,
- Solver Lineal,
- Nonlinear Solver,
- Integer " Pre-Solver
- Integer " Solver.

## LINGO|Picture

Este comando despliega el modelo en forma de matriz



The screenshot shows a window titled "LINGO Matrix Picture - EJEMPLO". The window displays a matrix with the following structure:

	X1	X2	X3	X4
MAXIMIZARGANANCIAS	40	30	25	30
DISPONIBILIDADRECURSOS1	4	1.5	3	1.8
DISPONIBILIDADRECURSOS2	1.5	1	2	0.8
DISPONIBILIDADRECURSOS3	2	1.2	1.5	0.8
COMPROMISOSENTREGA1		1		
COMPROMISOSENTREGA2			1	

## LINDO

### Generalidades

Lindo (Linear Interactive Discrete Optimizer) es un software muy básico que permite realizar optimización de problemas de programación lineal (PL), y cuadrática, definidos sobre variables reales y/o binarias. Debido a su simplicidad se utiliza con fines educativos, ya que es de fácil acceso y puede correr hasta en un viejo 286. El problema es que su interfaz es muy primitiva y no es amigable.

### Sintaxis

La Sintaxis requerida para LINDO tiene requisitos mínimos, se deben diferenciar 3 partes: la función objetivo, las variables de decisión y las restricciones.

La función objetivo debe estar siempre al comienzo del modelo y debe comenzar con **MAX** para maximizar o **MIN** para minimizar. Al finalizar de formular la función objetivo y antes de comenzar con la formulación de las restricciones debe colocarse la sentencia **SUBJECT TO** o **SUCH THAT** o **S.T.** o **ST.** Luego se formulan las restricciones el final de las restricciones debe ir seguida de la palabra **END.**

Erica Canizo Paola Lucero	Investigación Operativa 2002 Software Para Programación Lineal – LINGO/LINDO–
------------------------------	---

En lo que respecta al nombre de las variables LINDO tiene un límite de 8 caracteres para nombrarlas. El nombre debe ser un carácter alfabético (A-Z) seguido de cualquier carácter excepto ! ) + - = < >.

Opcionalmente también se les puede dar nombres a las restricciones lo que hacen más sencillo leer el reporte de solución. Para los nombres de las restricciones se sigue las mismas reglas que para las variables. Para hacerlo se debe comenzar la restricción con su nombre, seguido de un paréntesis luego del mismo se comienza con la formulación de la restricción. Solo constantes pueden ser definidas del lado derecho de las restricciones.

Hay distintas sentencias opcionales que se incluyen en el texto del programa luego de la sentencia **END**. Estas sentencias y sus funciones aparecen en la tabla siguiente

**FREE** <Variable> remueve todos los límites en <Variable>, permitiendo a al <Variable> tomar cualquier valor real positivo o negativo

**GIN** <Variable> hace la <Variable> una entera (restringida a un conjunto de enteros no negativos).

**INT** <Variable> hace la <Variable> binaria (restringida a 0 ó 1)

**SLB** <Variable> <Value> pone un límite inferior en <Variable> de <Value>.

**SUB** <Variable> <Value> pone un límite superior en <Variable> de <Value>.

**TITLE** <Title> hace <Title> el título del modelo.

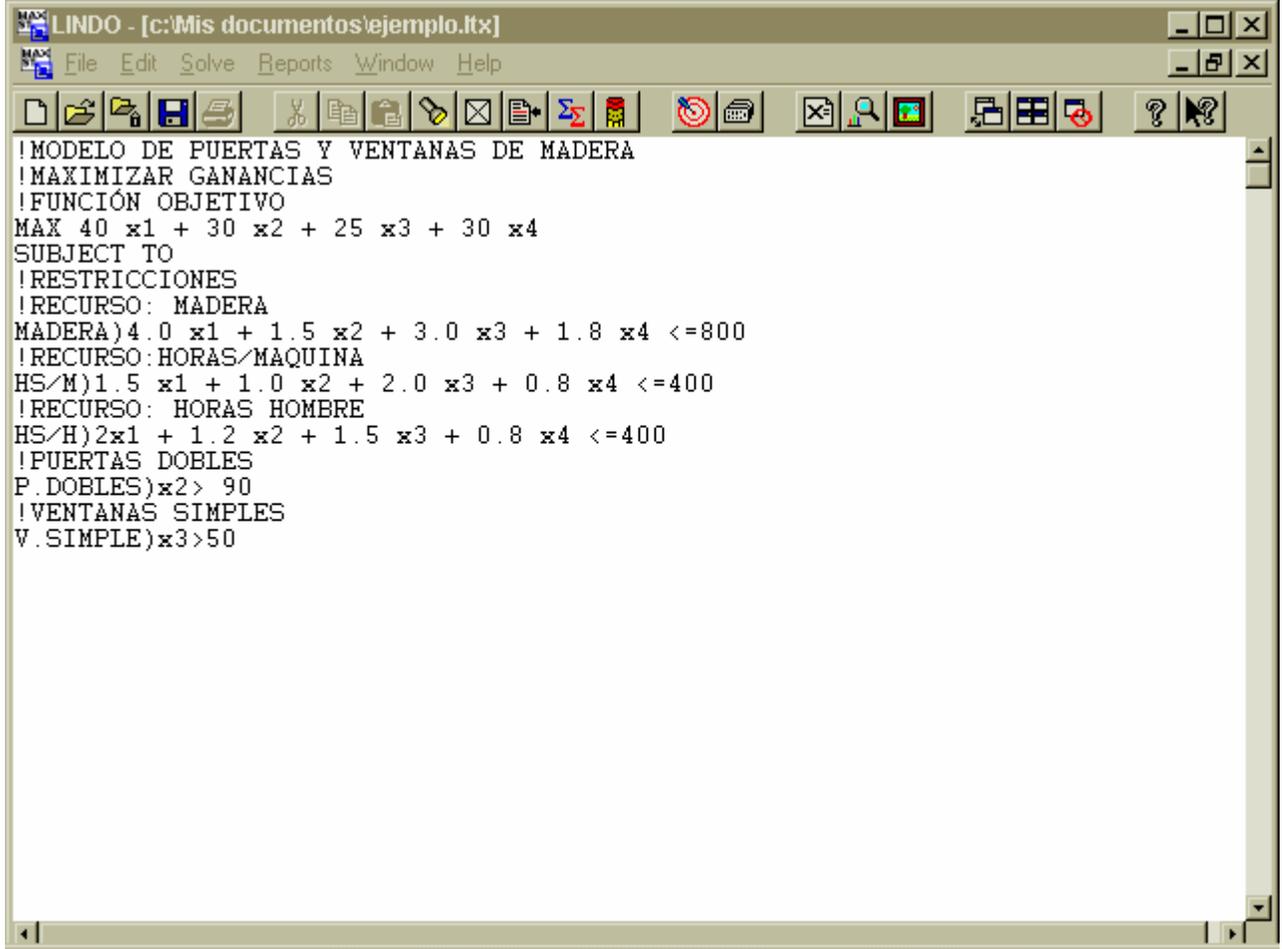
LINDO solo reconoce 5 operadores: +, -, >, <, =. No se aceptan paréntesis para indicar precedencias y se toman las operaciones de izquierda a derecha.

Los comentarios pueden estar situados en cualquier lugar en el modelo precedido del signo ! , todo lo que sigue al signo de admiración en la misma línea será considerado como un comentario.

Tomaremos el mismo ejemplo resuelto en LINGO para comparar resultados:

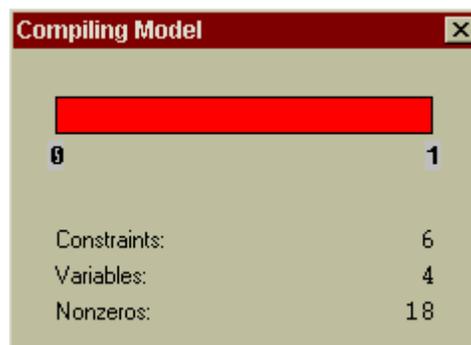
### PROBLEMA EJEMPLO



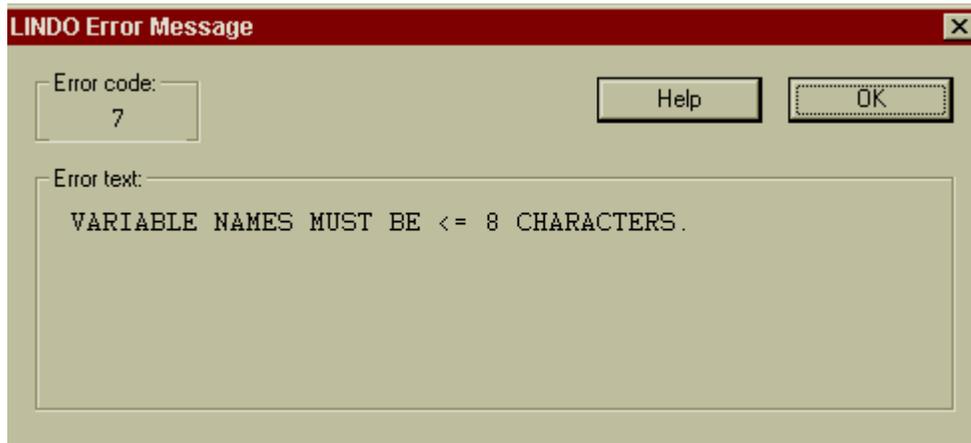


```
LINDO - [c:\Mis documentos\ejemplo.ltx]
File Edit Solve Reports Window Help
! MODELO DE PUERTAS Y VENTANAS DE MADERA
! MAXIMIZAR GANANCIAS
! FUNCIÓN OBJETIVO
MAX 40 x1 + 30 x2 + 25 x3 + 30 x4
SUBJECT TO
! RESTRICCIONES
! RECURSO: MADERA
MADERA) 4.0 x1 + 1.5 x2 + 3.0 x3 + 1.8 x4 <= 800
! RECURSO: HORAS/MAQUINA
HS/M) 1.5 x1 + 1.0 x2 + 2.0 x3 + 0.8 x4 <= 400
! RECURSO: HORAS HOMBRE
HS/H) 2x1 + 1.2 x2 + 1.5 x3 + 0.8 x4 <= 400
! PUERTAS DOBLES
P.DOBLES) x2 > 90
! VENTANAS SIMPLES
V.SIMPLE) x3 > 50
```

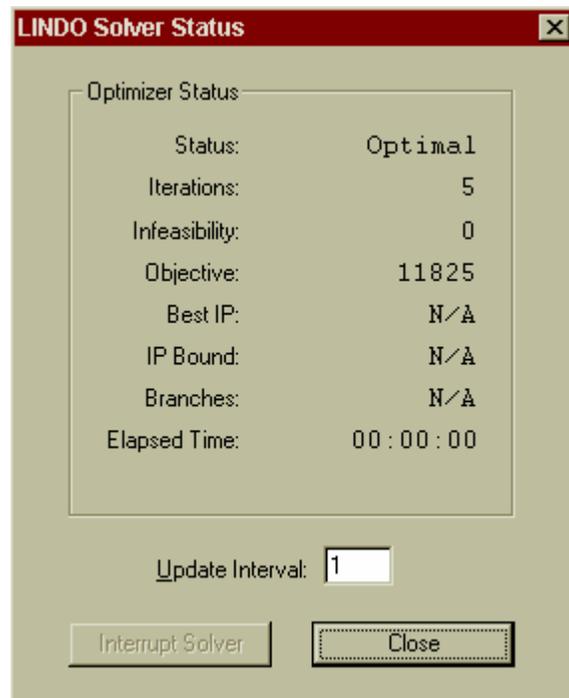
Una vez finalizada la escritura del programa se debe seleccionar en el menú Solve el comando Compile Model. Mientras el modelo se compila aparece la siguiente pantalla que muestra el progreso de la compilación



Aquí se muestra el número de restricciones, la cantidad de variables y la cantidad de coeficientes distintos de cero. Si ocurre un error durante la compilación se mostrará un mensaje y se posicionará el cursor en la línea donde ocurrió.



Si luego de la compilación no existe ningún error se debe proceder a resolver el modelo. Para esto debemos elegir en el menú Solve el comando del mismo nombre. En este momento tenemos la opción de seleccionar si queremos realizar un análisis de riesgo de nuestro problema. Esto nos da como primer resultado la pantalla de estado del solver. Para nuestro ejemplo;



La información que nos entrega es la siguiente:

A diferencia de LINGO el estado (Status) se puede clasificar como: Optimal, Feasible (Factible), Infeasible (no factible), Unbounded (Ilimitado). El resto de la información es similar a la dada por LINGO.

A continuación se despliega la pantalla de reporte de la solución que también es similar a la de LINGO.

```

Reports Window
LP OPTIMUM FOUND AT STEP      5

      OBJECTIVE FUNCTION VALUE
    1)      11825.00

      VARIABLE            VALUE            REDUCED COST
      X1                0.000000            16.250000
      X2                90.000000            0.000000
      X3                50.000000            0.000000
      X4               262.500000            0.000000

      ROW      SLACK OR SURPLUS      DUAL PRICES
MADERA)      42.500000            0.000000
HS/M)        0.000000            37.500000
HS/H)        7.000000            0.000000
P.DOBLES)    0.000000           -7.500000
V.SIMPLE)    0.000000           -50.000000

NO. ITERATIONS=      5

```

También podemos ver el valor de las variables luego de cada iteración para esto vamos al menú Reports y seleccionamos el comando Tableau.

Continuando con el análisis del programa el resto de las funciones que podemos realizar en LINDO también son similares a las de LINGO