



UNIVERSIDAD DE BELGRANO

Las tesinas de Belgrano

**Facultad de Ingeniería y Tecnología Informática
Carrera de Licenciatura en Sistemas de Información**

**Programación Lineal: Desarrollo de un sistema
informático para la generación de modelos**

Nº 173

Rubén Diego Ivanoff

Tutor: Víctor M. Rodríguez

Departamento de Investigación
Abril 2005

Índice de contenidos

Índice de contenidos	3
1. Resumen Ejecutivo	5
2. Executive Summary	5
3. Introducción	6
4. Marco Teórico: La Programación Lineal	6
4.1. Un poco de historia... ..	6
4.2. Los problemas de programación lineal	7
4.3. El método Simplex	8
5. Aplicación en la Industria	10
6. La utilización de Solvers	11
7. El Lenguaje de Modelado MPL	14
8. Generación de Matrices	15
9. El sistema Optimat 32	16
9.1. Antecedentes: Optimat	16
9.2. Componentes de Optimat 32	16
9.2.1. Servicio de carga, modificación e importación de datos	16
9.2.2. Rutina generadora de matrices	18
9.2.3. Solvers enlazados a Optimat 32	19
9.2.4. Módulo de generación de reportes	20
10. Caso práctico: «La fábrica de fiambres»	21
10.1. El caso de la fábrica de fiambres	21
10.2. Resolución: Método 1. WinQSB	23
10.3. Resolución: Método 2. Optimat	24
10.4. Resolución: Método 3. Optimat 32	28
10.5. Conclusiones obtenidas del ensayo con el caso práctico	35
10.6. Limitaciones de Optimat 32	35
11. Bibliografía	36
11.1. Libros, Referencias y Manuales	36
11.2. Sitios de Internet	36
11.3. Software utilizado durante el desarrollo de esta Tesina	36
12. Anexos	37
12.1. Anexo 1: Flujo de Procesos	37

1. Resumen Ejecutivo

En este trabajo se exponen los resultados alcanzados mediante el desarrollo de una herramienta de software orientada a facilitar a los usuarios el proceso de modelización, resolución y utilización de la información producida por un sistema, en el marco teórico de la programación lineal.

Dicha herramienta consta de tres componentes fundamentales:

- a) Una interfase para la carga de datos
- b) Un programa generador del modelo que constituye el input para un programa *solver*. Este último puede ser uno de los disponibles en el mercado.
- c) Un programa para la exposición de los resultados en planillas, comprensibles para no matemáticos, con la posibilidad de exportarse a otro producto para Windows, Ej. Microsoft Excel.

La Tesina comienza con una breve introducción teórica acerca de la Programación Lineal; explayándose luego en la forma en que dichas rutinas matemáticas influyen positivamente en la administración de las organizaciones industriales.

El trabajo continúa con la explicación del modo de uso de los «solvers», como herramientas de optimización, y menciona las ventajas de utilizar generadores de matrices.

A partir de aquí, se comienza a tratar sobre el nuevo software, el cual ha sido denominado **Optimat 32**, realizando un desglose de sus partes componentes; y explicando el modo de uso de cada una y su integración en el sistema. Finalmente se presenta un caso práctico en el cual se compara el funcionamiento de Optimat 32 con otras aplicaciones de uso comercial; y se enuncian las conclusiones de dicha comparación.

La herramienta ha sido programada esencialmente en el lenguaje Delphi versión 6 de Borland Corporation; y el solver utilizado es el LP88 de Eastern Software.

Es posible utilizar otros programas solver del mercado, tales como XPRESS o LINDO.

El tamaño del problema no constituye un obstáculo, ya que para facilitar la validación de la herramienta se han utilizado modelos de la biblioteca de modelos desarrollados bajo Optimat. Algunos de ellos superan las 500 ecuaciones y 1000 variables¹.

Palabras clave:

Programación lineal; Generador de matrices; Solver; Modelización; Simplex.

2. Executive Summary

In this work the results are exposed reached by means of the development of a software tool guided to facilitate the users the modeling process, resolution and use of the information taken place by a system in the theoretical framework of the lineal programming.

This tool consists of three fundamental components:

- a) An interface for the load of data
- b) A generating program of the pattern that constitutes the input for a program solver. This program can be one of the available ones in the market.
- c) A program for the exhibition of the results in reports, comprehensible for not mathematical, with the possibility of being exported to another product for Windows, Example: Microsoft Excel.

The study begins with a brief theoretical introduction about the Lineal Programming; being extended then in the form in that this mathematical routines influence positively in the administration of the industrial organizations.

The work continues with the explanation in the way of use of the «solvers», as tools of optimization, and mentions the advantages of using matrix generators.

Starting from here, we begins to try on the new software, which has been denominated «**Optimat 32**», carrying out a breakdown on its component behalves; and explaining the use way of each one and their integration in the system. Finally a practical case is presented in which the operation of Optimat 32 is compared with other applications of commercial use; and the conclusions of this comparison are enunciated.

The tool has been programmed essentially in the language Delphi version 6 of Borland Corporation; and the used solver is the LP88 of Eastern Software.

It is possible to use other solver programs of the market, such as XPRESS or LINDO.

1. Ver Yacuzzi y Rodríguez, op. cit.

The size of the problem doesn't constitute an obstacle, since to facilitate the validation of the tool, models of the library of models have been used that has been developed under Optimat. Some of them overcome the 500 equations and 1000 variables.

Keywords:

Lineal programming; Matrix Generator; Solver; Modeling; Simplex.

3. Introducción

Hacia el año 1642, contando diecinueve años de edad, el matemático y filósofo francés Blaise Pascal (1623-1662) diseñó una máquina que tenía por destino ayudar a su padre en la administración de la superintendencia de la generalidad de Ruan. La creó con el fin de simplificar los interminables cálculos administrativos que efectuaba por medio de fichas en el ábaco de columnas. La máquina de Pascal consistía en una serie de cilindros con números, que rotaban de manera de permitir contar o realizar sumas o restas por complemento. Este fue el antecedente de las máquinas de calcular, desarrollo que siglos después derivó en las computadoras personales de las cuales disponemos hoy en día.

Sin saberlo, Pascal también desarrolló una de las primeras **herramientas de apoyo a las decisiones**².

Al tener mejor control de sus cuentas, el padre de Pascal contaba con mejores datos para realizar adecuadamente su tarea, lo que condujo al progreso de la superintendencia (y al de la familia Pascal). Varias copias de la *Pascalina*³ fueron vendidas en Europa.

Sin tratar de emular la genialidad de Blaise Pascal, el propósito de esta Tesina es desarrollar una aplicación de software que brinde apoyo a las decisiones de los administradores de organizaciones industriales.

La aplicación, debe ser capaz de generar modelos de programación lineal (consistentes en matrices de ecuaciones), de gran tamaño y a partir de bases de datos fácilmente administrables y mantenibles.

El sistema también debe enviar la información generada a una rutina matemática, y luego de obtener los resultados suministrados por la misma, generar una variedad de reportes e informes que ilustren en forma concisa y eficiente al usuario acerca de las maneras de optimizar sus niveles de producción o ganancias (de la empresa); o por ejemplo, minimizar gastos.

A fin de llevar este trabajo a buen término, disponemos de abundante información (detallada en la sección de bibliografía), amén de un sistema documentado; de similares características, pero fuertemente limitado por la tecnología de su época, ya que estaba desarrollado para el sistema operativo D.O.S. Este hecho obligó a resolver infinidad de cuestiones de índole teórico-práctica, para producir los resultados que se exponen como producto de la tesina.

El motivo de este desarrollo se fundamenta en la necesidad actual de disponer de aplicaciones con estas características que tienen las organizaciones industriales de nuestro país.

El software disponible no está adaptado a los recientes cambios en la tecnología de sistemas operativos más amigables. Cabe resaltar que un sistema de este tipo debe adaptarse a las necesidades específicas de cada empresa; lo que lo convierte en un «sistema a medida», en el mejor de los casos. Esto significa que no existe ningún sistema «enlatado» capaz de cubrir las necesidades de las organizaciones, por lo que estas utilizan planillas de cálculo, aplicaciones propietarias, lenguajes algebraicos de modelización, etc. para resolver estos problemas lo que conlleva un arduo trabajo para crear los modelos, trabajo para el cual los administradores en general no están específicamente preparados.

Optimat 32 es el prototipo de una aplicación adaptable, escalable y que pone especial énfasis en la amigabilidad para el usuario y en la precisión de sus resultados.

4. Marco Teórico: La Programación Lineal

4.1. Un poco de historia...

En el ámbito de los procesos industriales, es necesario aplicar, en la mayoría de los casos, una infinidad de recursos de diferente tipo. Así, por ejemplo, se utilizarán distintas y variadas materias primas, las cuales se combinarán de diferentes maneras, y serán tratadas en maquinarias, las que tienen distintas capacida-

2. Véase Georges IFRAH, *Historia Universal de las Cifras. La inteligencia de la Humanidad contada por los Números y el Cálculo*, Madrid, Espasa Calpe, 1997, 2ª edición, Pag. 1486

3. *Pascalina*, es la denominación que Pascal dio a su invento.

des de producción, y son operadas por personas que desarrollan su actividad en diversos horarios, cobran sumas diferentes, y trabajan en uno o varios sectores en la organización, los cuales realizan tareas distintas, con diferente velocidad, distinta capacidad productiva e incluso pueden residir en diferentes plantas...

Evidentemente, los procesos que involucra la producción, reflejan una gran complejidad en conjunto, y llevar a cabo el control de todas esas actividades es una tarea demasiado complicada para una sola persona. Muchas veces, el trabajo de quien esta a «cargo» consiste en maximizar las ganancias, y otras veces tiene que ver con minimizar los costos; pero siempre implica una gran cantidad de cálculos complicados.

En los siglos XVII y XVIII grandes matemáticos como Newton, Leibnitz, Bernouilli y sobre todo Lagrange, se ocuparon de obtener máximos y mínimos condicionados de determinadas funciones.

El matemático francés Jean Baptiste-Joseph Fourier (1768-1830), fue el primero en intuir, aunque de forma imprecisa, los métodos de lo que actualmente llamamos programación lineal, y la potencialidad que de ellos se deriva.

Recién en 1939 podemos encontrar nuevos estudios serios al respecto; en ese año, Leonodas Vitalyevich Kantarovitch publica una extensa monografía titulada «métodos matemáticos de organización y planificación de la producción».

En 1941-1942 se formula por primera vez el problema de transporte, estudiado por Koopmans y Kantarovich.

Tres años más tarde, G. Stigler plantea otro problema conocido como de régimen alimenticio optimal.

Pocos años después, en Estados Unidos, se asumió que la eficaz coordinación de todas las energías y recursos de la nación era un problema de tal complejidad que su resolución y simplificación pasaba necesariamente por los modelos de optimización que resuelve la programación lineal.

Paralelamente, se desarrollaban las técnicas de computación y los ordenadores; estos resultarían indispensables para solucionar este tipo de problemas.

En 1947, G.B. Dantzig, formula, en términos matemáticos muy precisos, el enunciado estándar al que cabe reducir todo problema de programación lineal. Junto con una serie de investigadores del United States Department of Air Force, formarían el grupo que se denominó SCOOP. (Scientific Computation of Optimum Programs).

Respecto al método Simplex, su estudio comenzó en el año 1951 y fue desarrollado por Dantzig en el U.S. Bureau of Standards SEAC COMPUTER, ayudándose con varios modelos de ordenadores IBM.

4.2. Los problemas de programación lineal

Un problema de programación lineal, básicamente, se resuelve obteniendo los valores de una serie de variables que componen una función, de manera que a través de esos valores resulte el máximo o el mínimo (según el caso) de esa función, y cuando dichas variables se encuentran sujetas a distintas restricciones, las cuales se expresan mediante inecuaciones lineales.

Para aclarar el concepto, veamos el siguiente ejemplo:

Una fábrica de pinturas produce esmaltes sintéticos para su venta.

La pintura se envasa en latas de 1, 4 y 20 litros, y sus precios de venta son \$5, \$18 y \$80 respectivamente.

Se desea producir al menos 3000 latas de pintura, y en el almacén hay lugar para colocar 50000 litros; cualquiera sea el tamaño de las latas.

Se desea saber cuántas latas de cada tipo deben fabricarse, a fin de maximizar las ventas.

Solución:

-Si llamamos x al número de latas de 1 litro, y al número de latas de 4 litros, y z al número de latas de 20 litros, se tiene que la función de ventas es:

$$Z = 5x + 18y + 80z \quad (\text{Función a maximizar}).$$

Y dicha función está sujeta a las siguientes restricciones:

$$x + y + z \geq 3000 \quad (\text{Se desea producir al menos 3000 latas})$$

$$x + 4y + 20z \leq 50000 \quad (\text{Límite del tamaño del depósito}).$$

$$x \geq 0; y \geq 0; z \geq 0 \quad (\text{No se puede producir un número negativo de unidades}).$$

A partir de aquí es cuando comienza a tener significado la utilización de computadoras; a pesar de lo pequeño que es el problema planteado, puede notarse el nivel de complejidad requerido para obtener su solución. Si el problema fuera el planeamiento de la producción de toda la planta, probablemente la cantidad de variables alcanzaría los cientos; y la cantidad de restricciones se contaría por miles.

Para ayudarnos en estos casos, existen muchas aplicaciones informáticas que corren en las más diversas plataformas. El componente más importante de todas ellas es el «solver», la rutina matemática que calcula los valores óptimos de las variables; combinación de valores también llamada «mix óptimo de producción».

Cabe aclarar que la programación lineal también se puede aplicar a otras áreas que no tienen que ver con producción; como por ejemplo los problemas de transporte, donde se tiene una cantidad de puntos de origen desde donde transportar una mercadería a otra cantidad de puntos de destino; y las maneras de hacerlo, constituyen diferentes alternativas, las cuales tienen diferente costo.

Los solvers son herramientas que trabajan a nivel matemático, utilizando métodos variados, como por ejemplo el «simplex», que consiste en obtener la solución a través de iteraciones con aproximaciones sucesivas.

Si bien en este trabajo nuestra herramienta ha sido diseñada para el tratamiento de programas lineales, existen posibilidades concretas de estudiar su ampliación a otros modelos más complejos (Casos de variables en números enteros y booleanos) que son soportados por el Solver.

4.3. El método Simplex

El método Simplex es un algoritmo iterativo, que obtiene una solución, y la mejora a cada paso. El proceso finaliza cuando ya no es posible seguir mejorando la solución obtenida.

Partiendo de la función objetivo en un vértice cualquiera, de un recinto convexo, el método consiste en buscar sucesivamente un nuevo vértice que mejore la solución obtenida para el anterior (o la mantenga).

Se basa en la siguiente propiedad: «Si la función objetivo f , no toma su valor máximo en el vértice A , entonces hay una arista que parte de A , a lo largo de la cual f aumenta»⁴.

Para mostrar su forma de funcionamiento, vamos a resolver el siguiente problema utilizando el método Simplex:

Maximizar:

$$Z = f(x,y) = 3x + 2y$$

Sujeto a las siguientes restricciones:

$$2x + y \leq 18$$

$$2x + 3y \leq 42$$

$$3x + y \leq 24$$

$$x \geq 0; y \geq 0$$

Consideraremos las siguientes fases:

1. Convertir las desigualdades en igualdades.

Se introduce una variable de holgura por cada una de las restricciones, para convertirlas en igualdades. Resulta el siguiente sistema de ecuaciones lineales:

$$2x + y + h = 18$$

$$2x + 3y + s = 42$$

$$3x + y + d = 24$$

2. Igualar la función objetivo a cero.

Simplemente se trata de despejar la ecuación objetivo para que quede igualada a cero:

$$-3x - 2y + Z = 0$$

3. Escribir la tabla inicial Simplex.

Este paso sirve para crear la tabla de valores que se utilizará a fin de resolver el problema. Se trata simplemente de escribir las ecuaciones en filas y columnas para poder trabajar más fácilmente con ellas. En las columnas aparecerán las variables del problema, y en las filas los coeficientes de las igualdades obtenidas en el paso 1; una fila para cada restricción, y la última fila con los coeficientes de la función objetivo. El resultado es el siguiente:

4. Ver <http://thales.cica.es/rd/Recursos/rd98/Matematicas/29/simplex.html>

Base	Variable de decisión		Variable de holgura			Valores Solución
	x	y	h	s	d	
h	2	1	1	0	0	18
s	2	3	0	1	0	42
d	3	1	0	0	1	24
Z	-3	-2	0	0	0	0

4. Encontrar la variable de decisión que entra en la base y la variable de holgura que sale.

A. Para seleccionar la variable de decisión que entra en la base, se debe observar la última fila, la que contiene los coeficientes de la función objetivo, y tomar entre las variables con el coeficiente negativo la de mayor valor absoluto.

En este caso, se trata de la variable x de coeficiente -3 .

Si existiesen dos o más coeficientes iguales que cumplan la condición anterior, entonces se puede seleccionar cualquiera de ellos.

Si en la última fila no existiese ningún coeficiente negativo, significaría que se ha alcanzado la solución óptima. Por lo tanto, lo que va a determinar el final del proceso de aplicación del método Simplex, será que en la última fila no queden coeficientes negativos.

La columna de la variable que entra en la base es llamada **columna pivote**.

B. Para encontrar la variable de holgura que debe salir de la base, se divide cada término de la última columna (valores solución), por el término correspondiente de la columna pivote, siempre que estos últimos sean mayores que cero. En el caso del ejemplo:

$$18/2 = 9$$

$$42/2 = 21$$

y

$$24/3 = 8$$

Si existiese algún elemento menor o igual que cero no se considera para dicho cociente. En el caso de que todos los elementos fuesen menores o iguales a cero, el resultado sería una función no acotada y es imposible continuar.

El término de la columna pivote que en la división anterior de lugar al menor cociente positivo, (en este caso el tercero, ya que 8 es el menor cociente positivo), indica la fila de la variable de holgura que sale de la base, (d en el ejemplo). Esta fila se llama **fila pivote**.

Si dos o más cocientes son iguales y a la vez son los menores positivos, significa que cualquiera de las variables correspondientes puede salir de la base (Caso de degeneración).

C. En la intersección de la fila pivote y columna pivote se encuentra el elemento **pivote operacional**, (3 para este caso).

5. Encontrar los coeficientes de la nueva tabla.

Los nuevos coeficientes de x se obtienen dividiendo todos los coeficientes de la fila d por el pivote operacional, 3, que es el que hay que convertir en 1. Luego, utilizando reducción Gaussiana, se convierten en 0 los restantes elementos de la columna, con lo que se obtienen los nuevos coeficientes de las otras filas, incluyendo los de la función objetivo Z.

Para ejemplificar, utilizaremos el siguiente esquema:

Para la fila del pivote:

$$\text{Nueva fila del pivote} = (\text{vieja fila del pivote}) / (\text{pivote})$$

Para el resto de las filas:

$$\text{Nueva fila} = (\text{vieja fila}) - (\text{coeficiente de la vieja fila en la columna de la variable entrante}) \times (\text{nueva fila del pivote})$$

En el caso del ejemplo:

Base	Variable de decisión		Variable de holgura			Valores Solución
	x	y	h	s	d	
h	0	1/3	1	0	-2/3	2
s	0	7/3	0	1	-2/3	26
d	1	1/3	0	0	1/3	8
Z	0	-1	0	0	1	24

Como en los elementos de la última fila todavía existe uno menor que cero (-1), significa que todavía la solución no es óptima. Hay que repetir el proceso.

Operando en forma análoga a la anterior, obtenemos la tabla:

Base	Variable de decisión		Variable de holgura			Valores Solución
	x	y	h	s	d	
y	0	1	3	0	-2	6
s	0	0	-7	0	4	12
x	1	0	-1	0	1	6
Z	0	0	3	0	-1	30

Y luego de una iteración más:

Base	Variable de decisión		Variable de holgura			Valores Solución
	x	y	h	s	d	
y	0	1	3	0	-2	6
s	0	0	-7	0	4	12
x	1	0	-1	0	1	6
Z	0	0	3	0	-1	30

Como todos los coeficientes de la fila de la función objetivo son positivos, nos encontramos ante una solución óptima.

La solución óptima viene dada por el valor de Z en la columna de los valores solución (33). En la misma columna puede observarse el vértice donde se alcanza, observando las filas correspondientes a las variables de decisión que han entrado en la base (D[3,12])⁵.

Si en lugar de maximizar, se tratara de minimizar, se utiliza el mismo proceso, pero cambiando el sentido del criterio: para entrar en la base se elige la variable cuyo valor, en la fila de la función objetivo sea el mayor de los positivos; y se finalizan las iteraciones cuando todos los coeficientes de la fila de la función objetivo son negativos.

5. Aplicación en la Industria

Como es evidente, puede obtenerse una gran complejidad en el planteo de un problema de producción, si se agregan muchas variables al modelo considerado.

Pero, sin embargo, la pregunta más frecuente que se hacen la mayoría de los fabricantes es: -» ¿Qué debo hacer para ganar mas dinero en mi empresa, invirtiendo el mismo esfuerzo o tal vez no MUCHO MÁS QUE AHORA»⁶

5. Una BASE del sistema $Ax=b$ son n columnas linealmente independientes en A .

6. Comentario personal.

La solución a esta pregunta está dada por el principio de la palanca, descubierto por Arquímedes y redefinido por Peter Senge en «La quinta disciplina», aplicándolo a la administración:

El pensamiento sistémico enseña que las soluciones obvias no funcionan. También enseña que los actos pequeños y bien focalizados a veces producen mejoras significativas y duraderas, si se realizan en el sitio apropiado. Los pensadores sistémicos lo denominan «Principio de la palanca».

Las zonas de alto apalancamiento no son evidentes para la mayoría de los integrantes del sistema. No están «próximas en el tiempo y el espacio» respecto de los síntomas.

Los cambios de alto apalancamiento en los sistemas humanos no resultan obvios para quien no comprende las fuerzas que actúan en esos sistemas. No hay reglas sencillas para efectuar cambios de alto apalancamiento, pero hay modos de pensar que los facilitan⁷.

O sea, que debemos encontrar aquellos factores que provocan cambios favorables y significativos, cuando se varían ligeramente.

Por ejemplo, tomemos el caso de una fábrica de productos alimenticios que tiene una gran capacidad productiva, cuyo producto terminado requiere mantenerse frío; y en dicha fábrica existe una limitación importante pero no obvia en cuanto a la capacidad de sus cámaras.

Un cálculo realizado con programación lineal podría determinar por ejemplo que si la empresa incorporara dos nuevas cámaras, se podría producir lo suficiente para recuperar el importe invertido en muy poco tiempo; y a la vez reducir el costo del producto, ingresando a nuevos mercados masivos de gran demanda; lo que le permitiría a la organización crecer en forma muy importante.

Desgraciadamente, en la mayoría de los casos, las soluciones no son tan evidentes, ni tan simples de implementar. Esto se debe principalmente a la complejidad de los sistemas de producción, y también a que las personas encargadas de administrarlos disponen de demasiada información incorrectamente organizada; por lo cual no pueden ver detalles importantes que los ayudarían en su negocio.

La programación matemática es una herramienta muy útil para organizar ese gran caudal de información; y a la vez combinarla, de modo de optimizar los recursos en beneficio de la organización.

6. La utilización de Solvers

En cuanto hayamos formulado el modelo matemático, lo natural es resolverlo. Para ello debemos encontrar métodos que produzcan los mejores valores para las variables de decisión, es decir, aquellos valores que satisfacen simultáneamente todas las limitaciones y además proporcionan el mejor valor para la función objetivo. Existen en el mercado muchos paquetes de computación («solvers») para resolver modelos previamente formulados, entre los que se puede incluir: Xpress, STORM, LP88, LINDO, RISK, QM y EXCEL. Muchos de estos paquetes muestran soluciones similares, aunque de hecho el formato específico es diferente.

Existen muchos gerentes, que administran una variedad de modelos de producción. Sin embargo, son muy pocos los que saben como utilizar eficientemente los programas nombrados anteriormente, y por otro lado, las obligaciones naturales de su nivel gerencial no les dejan mucho tiempo como para elaborar modelos eficaces y detallados. Son aplicables pese a su antigüedad las palabras de Herbert A. Simon, cuando dice:

En el mundo real, usualmente no tenemos elección entre soluciones óptimas y satisfactorias, porque sólo raramente tenemos un método de encontrar el óptimo... No podemos, dentro de límites computacionales practicable, generar todas las alternativas admisibles y comparar sus méritos relativos. Ni podemos reconocer la mejor alternativa, aunque seamos lo suficientemente afortunados para generarla temprano, hasta que las hayamos visto a todas. Satisfacemos buscando alternativas de tal modo que podamos en general encontrar una aceptable después de sólo una búsqueda moderada⁸.

De modo que se hace necesario crear un software que funcione como un nexo entre los gerentes y los sistemas de resolución de ecuaciones (Solvers); de tal manera que mediante estos, sea posible obtener soluciones lo suficientemente buenas como para ser utilizadas con éxito, y que además, este trabajo no requiera la inversión de tiempo adicional, poniendo a disposición del usuario al mismo tiempo las más avanzadas herramientas de cálculo de los solvers, y una interfaz que «entienda» el lenguaje gerencial de la organización, traduciéndolo en forma transparente para el operador.

Un solver, es una herramienta extremadamente útil para todos aquellos que conocen su forma de utilización. Es indudable que son poderosísimas rutinas matemáticas, y que combinados con las veloces computadoras personales de hoy en día, se convierten en invaluable asistentes matemáticos a la hora de tomar decisiones importantes.

7. Peter SENGE, *La Quinta Disciplina*, capítulo 4 op.cit.

8. Simon (1969).P64. citado por Keen et.al. (1978). Yacuzzi y Rodríguez, op.cit. pág.17

Por desgracia, es necesaria una preparación previa y una experiencia en su utilización para poder aprovechar todo su potencial. Además tienen ciertas limitaciones en cuanto a la interfaz con el usuario, lo que hace un poco complicado realizar operaciones básicas, como por ejemplo el ingreso de datos. Así para ingresar un problema típico, en un solver tradicional, se deben tipear cientos o miles de ecuaciones, coeficiente por coeficiente, con la consiguiente y derivada alta probabilidad de cometer algún error involuntario.

```

LP88 - Version 7.03
Linear Programming for the IBM PC
(C) Copyright 1987 by ESP, Inc.
For Assistance with this Program
call the ESP Customer Hotline:
(703) 360 - 7600

```

KEY	MASTER MENU	SETUP MENU	EXECUTION MENU	OUTPUT MENU
F1	SETUP MENU	FILES LIST	PAUSE TOGGLE	PRIMAL VALUES
F2	SOLVE PROBLEM	NEW PROBLEM	STEP TOGGLE	DUAL VALUES
F3	RERUN w/BASIS	DISPLAY/EDIT	INVERT MATRIX	COST RANGES
F4	RESTART w/INU	SAVE PROBLEM	CHANGE CONTROLS	RHS RANGES
F5	GET SOLUTION	CONVERT to DUAL	PIVOT SELECT	INVERSE MATRIX
F6	OLD PROBLEM	OLD PROBLEM	DELTA SELECT	INU* A MATRIX
F7	BATCH MODE	INPUT FILE	SHORT PRINTOUT	SAVE SOLUTION
F8	OUTPUT MENU	PRINT PROBLEM	OUTPUT MENU	SAVE BASIS
F9	CONFIGURE	READ WORKSHEET	RECORD PIVOTS	WRITE WORKSHEET
F10	END SESSION	RETURN	RETURN	RETURN
Esc	HELP	HELP	HELP	HELP

```

Select and <Enter> to Begin Session
Destination for Listings (Printer or File Name)
Printer Carriage Width (Number of Columns)
Form Feed Before Printing a Table (Y/N)
Storage for Coefficients (Memory or File Name)
Storage for the Inverse (Memory or File Name)
Store the Inverse in Product Form (Y/N)

```

Ilustración 1 -Ejemplo de menú del solver LP88

```

EJEMPLO          OBJECTIVE:  MAX          VARIABLES:  6          DATE  02-26-2004
BASIS: NONE      CONSTRAINTS:  3          SLACKS:    0          TIME  14:53:37

```

	X.1	X.2	X.3	X.4	X.5	X.6	RHS
MAX							
RETURN	2.000	1.000	4.000	-2			.000000
Y.1	1.000	1.000	1.000				= .000000
Y.2	1.000		-2.000				= .000000
Y.3		5.000	1.000				= .000000

```

1GET RO 2GET CO 3RENAME 4EDIT L 5ROW OP 6COLUMN 7OVERLA 8FORMUL 9REPEAT 0RETURN

```

Ilustración 1 -Ejemplo de ingreso de datos en el solver LP88

Por esos motivos es que la mayoría de los administradores no esperan con agrado el momento de la utilización de este software. Para ellos, lo importante no es el medio sino el fin: Los resultados óptimos que se esperan. Ante esta situación, prefieren recurrir a otras soluciones, como por ejemplo los generadores de matrices o los lenguajes de modelado; estas aplicaciones son básicamente sistemas que «generan» automáticamente la matriz vectorial, a partir de una programación realizada por el usuario. De todas formas, tampoco es demasiado sencillo escribir en los generadores de matrices; y los lenguajes de modelado requieren del usuario conocimientos de programación.

A fin de ejemplificar el amplio uso de este tipo de software en grandes organizaciones, se adjunta una tabla de aplicaciones reales que han sido informadas en la bibliografía.

Organización	Naturaleza de la aplicación	Año de publicación*	Ahorros anuales
The Netherlands Rijkswaterstaat	Desarrollo de política nacional de administración del agua, incluye mezcla de nuevas instalaciones. Procedimientos de operación y costeo	1985	U\$S 15 millones
Monsanto Corp.	Optimización de operaciones de producción para cumplir metas con un costo mínimo	1985	U\$S 2 millones
United Airlines	Programación de turnos de trabajo en oficinas de reservaciones y aeropuertos para cumplir con las necesidades del cliente a un costo mínimo	1986	U\$S 6 millones
Citgo Petroleum Corp.	Optimización de las operaciones de refinación y de la oferta, distribución y comercialización de productos.	1987	U\$S 70 millones
San Francisco Police Department	Optimización de la programación y asignación de oficiales de patrulla con un sistema computarizado.	1989	U\$S 11 millones
Texaco Inc.	Optimización de la mezcla de ingredientes disponibles para que los productos de gasolina cumplieran con los requerimientos de de ventas y calidad.	1989	U\$S 30 millones
IBM	Integración de una red nacional de inventario de refacciones para mejorar el apoyo al servicio.	1990	U\$S 20 millones + U\$S 250 millones en mejor inventario
Yellow Freight System, Inc.	Optimización del diseño de una red nacional de transporte y la programación de rutas de envío.	1992	U\$S 17,3 millones
New Haven Health Dept.	Diseño de un programa efectivo de intercambio de agujas para combatir el contagio del SIDA.	1993	33% menos contagios
AT&T	Desarrollo de un sistema basado en PC para guiar a los clientes del negocio en el diseño del centro de llamadas.	1993	U\$S 750 millones
Delta Airlines	Maximización de ganancias a partir de la asignación de los tipos de aviones en 2.500	1994	U\$S 100 millones

- En la publicación se puede encontrar información adicional sobre estas aplicaciones⁹.

9. Frederick S. HILLIER, Gerald J. LIEBERMAN, *Investigación de Operaciones*, México, McGraw Hill, séptima edición, 2001.

7. El Lenguaje de Modelado MPL

MPL, es un lenguaje de modelización que interpreta una serie de instrucciones en una estructura similar a un programa, la cual representa el planteamiento de un modelo de programación lineal; y obtiene la resolución del mismo.

La estructura de un programa MPL si divide en dos sectores, la parte de definiciones y la parte del modelo:

Parte de Definiciones:

TITLE	- El nombre del problema.
INDEX	- Dimensiones del problema.
DATA	- Variables escalares, vectores de datos y archivos.
DECISION VARIABLES	- Variables de vector.
MACRO	- Macros para partes repetitivas.

Parte del modelo:

MODEL	
MAX o MIN	- La función objetivo.
SUBJECT TO	- Las restricciones.
BOUNDS	- Límites superiores e inferiores.
FREE	- Variables libres.
INTEGER	- Variables Enteras.
BINARY	- Variables Binarias.
END	

Ninguna de las secciones es estrictamente requerida en MPL, pero para obtener una optimización válida serán necesarias al menos la función objetivo y las restricciones. MPL permite colocar las secciones en cualquier orden, pero cualquier ítem debe ser declarado antes de utilizarse en el modelo. Las palabras MODEL, SUBJECT TO y END son opcionales, pero contribuyen a la claridad y legibilidad del modelo.

Lo siguiente es un ejemplo acerca de la forma en que los datos son ingresados en MPL; se trata de un modelo de planificación de la producción.

{Planificación de la Producción para 12 meses}

```

TITLE
  Planificacion_de_produccion;

INDEX
  producto = 1..3;
  mes = (Enero, Febrero, Marzo, Abril, Mayo, Junio, Julio,
        Agosto, Setiembre, Octubre, Noviembre, Diciembre);

DATA
  precio[producto]      := (105.09, 234.00, 800.00);
  Demanda[mes,producto] := 1000 DATAFILE(demanda.dat);
  CapacidadProduccion[producto] := 1000 (10, 42, 14);
  CostoProduccion[producto] := (64.30, 188.10, 653.20);
  CostoInventario      := 8.8 ;

DECISION VARIABLES
  Inventario[producto,mes] -> Invt
  Produccion[producto,mes] -> Prod
  Ventas[producto,mes]     -> Venta

MACRO
  Ingresos := SUM(producto,mes: precio * Ventas);
  CostoTotal := SUM(producto,mes: CostoInventario * inventario

```

```

+ CostoProduccion * Produccion);
MODEL
  MAX Beneficio = Ingresos - CostoTotal ;

SUBJECT TO
  BalanceInventario[producto,mes] -> IBal :
  Inventario = Inventario[mes-1] + Produccion - Ventas ;

BOUNDS
  Ventas < Demanda ;
  Produccion < CapacidadProduccion ;
  Inventario[mes=Enero..Noviembre] < 90000 ;
  Inventario[mes=Diciembre] = 20000 ;
END

```

Como podemos observar en el ejemplo anterior, se requieren ciertos conocimientos previos para escribir en MPL; asimismo, un usuario calificado de MPL también debería contar con conocimientos de modelización, y la capacidad de estructurar la información del modelo en un conjunto de datos compatibles con la sintaxis del lenguaje.

La diferencia básica entre un lenguaje como MPL y Optimat 32, consiste en que para utilizar Optimat 32 no es necesario poseer conocimientos avanzados de programación o modelización.

8. Generación de Matrices

La generación de matrices es una característica principal del sistema Optimat, y es en lo que se basa su importancia y su versatilidad.

Generar una matriz consiste básicamente en obtener una serie de coeficientes que son a la vez los coeficientes de un grupo de ecuaciones que componen el modelo de programación lineal a resolver.

Una matriz generada puede ser introducida en un solver para obtener su solución, y trabajar luego con los resultados obtenidos.

La matriz se obtiene a través de la compilación de los datos suministrados al sistema, y que están organizados de tal modo que los usuarios tengan una visión clara de lo que desean optimizar. En esto se destaca Optimat sobre otras aplicaciones disponibles: los datos son ingresados en el «lenguaje» de los administradores, pero son procesados por el sistema para convertirlos en algo potable para un solver.

Luego de realizar estos procesos, el sistema obtiene los resultados del problema, presentando esta información de manera eficiente, organizada y clara, para simplificar y explicar los datos obtenidos de la resolución del modelo; permitiendo al mismo tiempo la rápida modificación de datos o alternativas, para testear distintas posibilidades con agilidad, y precisión.

Estas características distintivas de Optimat son lo que lo convierten en un sistema único en su tipo. Si bien existen otros generadores de matrices, estos últimos requieren la escritura de rutinas de programación que generen los datos a través de instrucciones propias del solver. O sea, que para utilizarlos, no sólo hace falta conocer la manera de modelizar problemas, sino que también se deben poseer conceptos de programación estructurada, para crear el programa que generará la matriz al momento de su ejecución; o muchísima paciencia para ingresar el problema en un solver standard.

Esta información puede requerir su ingreso en diferentes formatos, de acuerdo al tipo de software utilizado.

Básicamente, en todos los casos, se trata de escribir la matriz a partir de los datos del problema previamente desmenuzados para su mejor estructuración. Para aclarar el concepto, podemos decir que la resolución de problemas en un generador de matrices o un solver requiere los siguientes pasos:

1. Comprender y plantear el problema a resolver: Esto significa, obtener los datos de un sistema real, en funcionamiento en una organización; y tratar de convertir esta información en un modelo. Muchas veces, los datos obtenidos no conducen necesariamente a la formulación del mismo, y debe realizarse un trabajo adicional de interpretación y organización de la información para lograrlo.
2. Identificar las restricciones y la función objetivo: Esto debe hacerse para obtener los datos que serán ingresados al solver o al generador de matrices.
3. Organizar la información «traduciéndola» al lenguaje correspondiente del generador o el solver: Algunos

lenguajes de modelado muy eficientes desarrollados en los últimos tiempos son: AMPL, MPL, GAMS y LINGO¹⁰.

4. Ejecutar el software para que resuelva el problema.
5. Interpretar los resultados obtenidos: Según el caso, puede ser que el resultado sea por ejemplo los valores resultantes de las variables, o alguna forma de reporte un poco mas completo.
En cambio utilizando Optimat 32, el problema se resuelve a través de los siguientes pasos:
 1. Creación de la estructura del modelo y definición de las actividades y su relacionamiento. Verificación si la estructura se adapta a la forma de trabajo de Optimat. Es decir si el sistema físico puede ser asimilado a una red donde se produce el flujo de las actividades, con restricciones en sus nodos. Estos pueden ser iniciales, intermedios o finales.
 2. Obtención y estructuración de los datos.
 3. Mantener actualizados los registros de Optimat 32 con los datos que se utilizarán habitualmente. (Productos, sectores, materias primas, gastos, etc.). Esto puede hacerse mediante el ingreso a través de pantallas o bien mediante importación de datos (desde Optimat). También puede realizarse integrando las bases de datos de Optimat 32 al sistema que se utilice en la organización. Así los datos se actualizarán en forma transparente para el usuario, quien sólo debe preocuparse por la utilización del sistema, y no por la carga de datos.
 4. Ordenar a Optimat 32 la resolución del modelo en cualquier momento que se desee; o bien almacenar distintos estados del modelo y comparar los resultados.
 5. Leer por pantalla o imprimir o simplemente exportar a un producto como Microsoft Excel una serie de detallados reportes, conteniendo toda la información necesaria, organizada de forma que sea fácilmente comprendida por operadores sin conocimientos de modelización.

9. El sistema Optimat 32

Como pudimos observar hasta ahora, Optimat 32 es un poderoso sistema de apoyo a las decisiones, que frente a otros sistemas de cálculo de óptimos, sobresale por sus características únicas en cuanto a su interfaz, su facilidad de uso, y su filosofía orientada al usuario.

Esta sección está dedicada a describir el sistema en sus partes componentes, detallando en cada caso las características más importantes que se han tenido en cuenta al momento del diseño del mismo.

9.1. Antecedentes: Optimat

El sistema Optimat para D.O.S. fue desarrollado por la empresa ADINSA S.R.L. en 1991-1994, con el fin de obtener una herramienta de apoyo a las decisiones que aprovechara las capacidades de procesamiento de las computadoras personales de ese entonces.

El resultado fue un sistema muy versátil, que permite modelizar tipos diferentes de organizaciones productivas, en tanto se puedan definir bajo la forma de un flujo de materiales en red, administrando la información en forma eficiente, y utilizando al máximo las capacidades de los procesadores disponibles, y de la interfaz utilizada.

Optimat es un interesante producto, y ha servido como una guía conceptual siendo la base sobre la cual se ha desarrollado Optimat 32.

9.2. Componentes de Optimat 32

El sistema puede dividirse en cuatro componentes perfectamente definidos, los cuales interactúan entre sí para obtener los resultados que se espera de ellos.

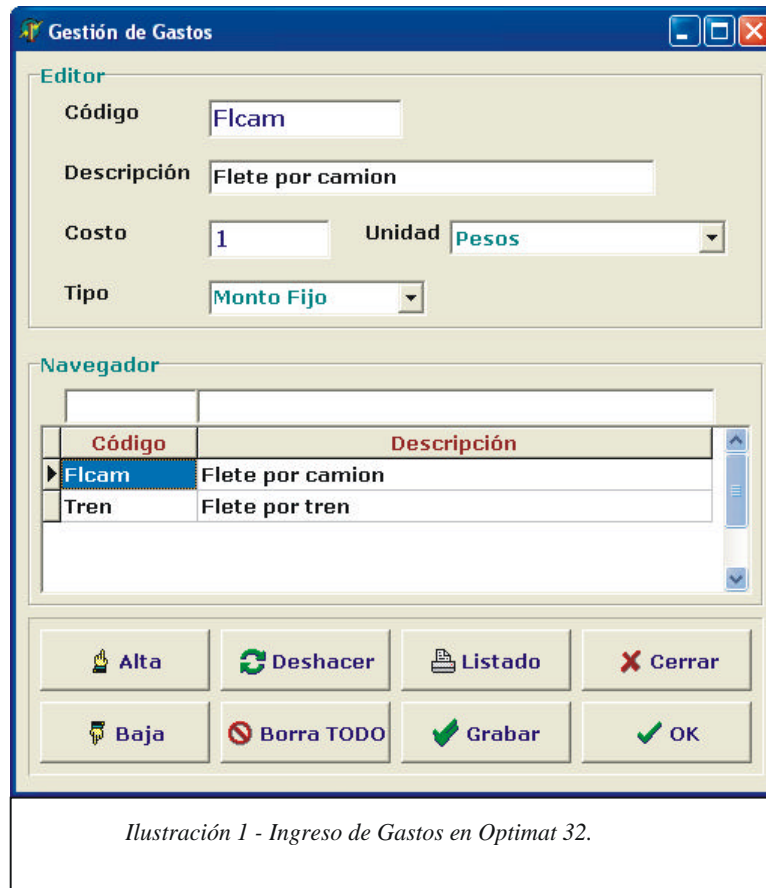
9.2.1. Servicio de carga, modificación e importación de datos

Este servicio, se basa en dos ítems fundamentales: las pantallas de carga de datos, y las pantallas de importación de información.

Las pantallas de carga de datos, fueron diseñadas con la doble intención de asistir al usuario en todo lo que pueda necesitar en cada una de ellas, y al mismo tiempo, obtener la simplicidad necesaria para que el usuario siempre sepa en forma intuitiva qué es lo que debe hacer para obtener lo que busca.

Todas las pantallas de carga de datos son similares. Por ejemplo:

10. Existen referencias en la Internet respecto a estos lenguajes de modelado en:
<http://www-fp.mcs.anl.gov/otc/Guide/SoftwareGuide/Categories/optsysmodlang.html>



Esta es una pantalla de ingreso de gastos. Pueden distinguirse tres zonas muy marcadas, las cuales se encuentran en todas y cada una de las pantallas de ingreso de datos. Estas zonas son: *Editor*, *Navegador*, y *Botonera*.

El *Editor*, es el lugar de la pantalla donde se ingresa la información que se pretende almacenar, como así también es el lugar donde se visualizan los datos previamente cargados, y además es donde se realizan las modificaciones. En síntesis, combina las pantallas clásicas de altas, modificaciones y consultas en una sola.

También en esta zona es donde pueden verse campos que tengan relación con otras tablas de la base de datos, y donde puede modificarse dicha relación.

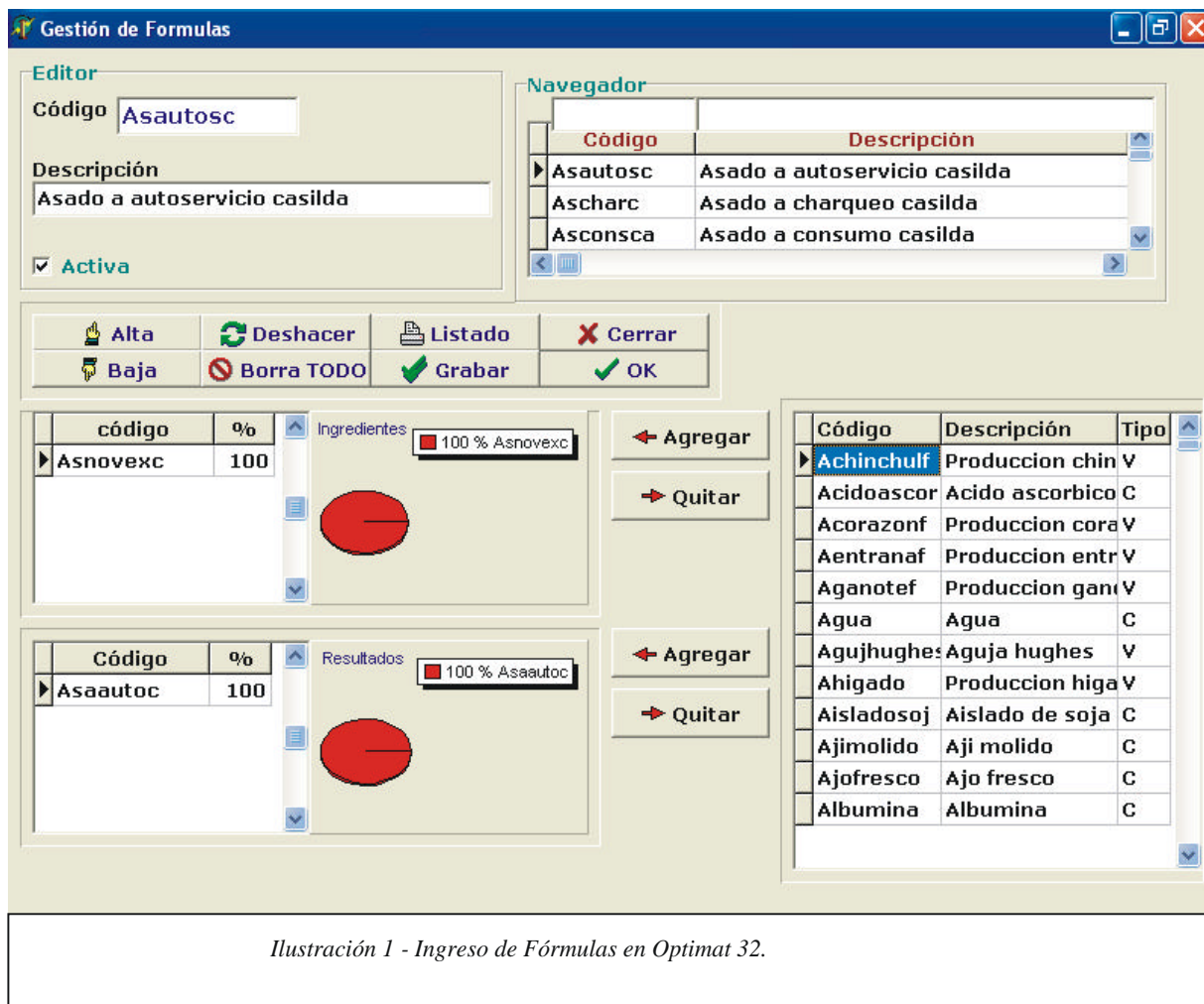
El *Editor*, funciona enlazado con el *Navegador*, el cual contiene una pequeña grilla con algunos de los datos más significativos de la tabla en cuestión. Esta grilla dispone de elementos de desplazamiento que permiten «moverse» entre los diferentes registros, viendo los cambios en el *Editor* al mismo tiempo.

También el *Navegador* utiliza campos de búsqueda rápida, que funcionan de la siguiente manera: Cuando el usuario escribe sobre ellos, el sistema busca la primera coincidencia entre los caracteres escritos y el campo correspondiente de la tabla, y posiciona en ese registro la pantalla. Si el usuario continúa escribiendo, el sistema sigue seleccionando, hasta que el usuario haya encontrado lo que buscaba.

La *Botonera*, contiene funciones de acceso inmediato con un clic. Para cada pantalla de ingreso de datos en particular, se ha diseñado una botonera acorde con las funciones a realizar, pero la mayoría contiene las siguientes:

- Alta: Este botón se utiliza para crear un nuevo registro.
- Baja: Su función consiste en eliminar datos. Antes de eliminar, el sistema pedirá una confirmación.
- Deshacer: Se utiliza cuando se desea retornar a los datos originales mostrados en pantalla, en caso de que hayan sido alterados en forma errónea o involuntaria.
- Borra Todo: Es una función de borrado de datos. Simplemente, elimina todos los datos de la tabla.
- Listados: Este botón se utiliza para generar listados de la base de datos. Estos listados pueden ser visualizados por pantalla e impresos a través del módulo de generación de reportes.
- Grabar: Este botón sirve para grabar o regrabar los datos que se visualizan.
- Cerrar: Cierra la ventana actual y descarta los cambios.
- OK: Cierra la ventana actual y guarda los cambios.

La pantalla de ingreso de datos más compleja es la de ingreso de fórmulas; pero puede observarse que mantiene el mismo patrón:



9.2.2. Rutina generadora de matrices

La rutina generadora de matrices es el corazón de Optimat 32. Esta rutina es un módulo perfectamente diferenciable, pero sería absolutamente inútil sin los demás.

Básicamente, funciona leyendo *todas las tablas cargadas por el usuario, las cuales contienen el modelo en forma implícita*. A medida que lee los registros de las tablas, obtiene los datos relacionados con las mismas, a fin de construir la ecuación a optimizar y las restricciones que se derivan de los datos almacenados.

Luego, organiza todos estos elementos de manera de crear un modelo matemático que sea comprensible para un solver; eso lo logra creando una matriz de coeficientes y ampliándola mientras los datos son leídos.

La rutina funciona en modo Thread; que bajo Microsoft Windows, significa que funciona en un proceso aparte de cualquier otro, y mientras se ejecuta, el sistema no espera; simplemente continúa la ejecución del resto del sistema hasta que el proceso de generación haya terminado. Esto se ha diseñado de esta manera porque esta característica permite realizar generaciones de modelos de tamaños extremadamente grandes (hablamos de miles de variables y restricciones)¹¹, sin dejar la máquina inactiva durante el proceso.

11. La cantidad de variables y restricciones que Optimat 32 puede procesar, sólo está limitada por la capacidad de memoria total bajo Windows de 32 bits; esto es la suma de la memoria RAM y la memoria virtual en disco rígido; además del espacio necesario en disco para almacenar el archivo que contiene el modelo. La mayoría de los solvers comerciales conocidos no pueden procesar una cantidad tan grande de ecuaciones, pero Optimat 32 está preparado para cuando esto sea posible).

9.2.3. Solvers enlazados a Optimat 32

Puede enlazarse a Optimat 32, cualquier solver que acepte realizar procesamientos batch.

Actualmente se está utilizando el solver **LP88 version 7.03** de la empresa **Eastern Software Products, Inc.**

Este producto acepta un archivo de entrada como por ejemplo el siguiente:

```
CONFIGURE,CON,80,NO,e:COEF,e:INVER,YES
INPUT FILE,PROB,MAX,FFF,CCC,ENTRADA.PL,NO
SOLVE PROBLEM
SAVE SOLUTION,2,NO,P,SALIDA
SAVE SOLUTION,3,NO,P,SALIDA2
END SESSION
```

CONFIGURE,CON,80,NO,e:COEF,e:INVER,YES

Esto se utiliza para configurar la consola; considerando que este programa fue diseñado para que funcione en terminales de baja resolución.

INPUT FILE,PROB,MAX,348,279,ENTRADA.PL,NO

Esta es la definición del problema. La entrada es por «**FILE**» (archivo), quiere decir que los datos de las restricciones y la ecuación a procesar serán obtenidos de un archivo. «**PROB**» es el nombre del problema. «**MAX**» quiere decir que se debe maximizar, no minimizar. «**FFF,CCC**» es el tamaño de la matriz de coeficientes en cantidad de filas por columnas. «**ENTRADA.PL**» es el nombre del archivo que contiene los valores de la matriz. «**NO**» significa que no se debe reinvertir la matriz al finalizar.

SOLVE PROBLEM

Este comando ordena a LP88 que resuelva el problema ingresado.

SAVE SOLUTION,2,NO,P,SALIDA

Aquí se indica que se debe guardar la solución (resultado de las variables) en un archivo llamado **SALIDA**.

SAVE SOLUTION,3,NO,P,SALIDA2

Este comando es para guardar el resultado de las variables slack en un archivo llamado **SALIDA2**

END SESSION

Finaliza la sesión batch.

Además, el archivo **ENTRADA.PL** mencionado en el código anterior, tiene el siguiente formato:

Fila, Columna, Valor.

```
Por ejemplo...
0,1,-432.730000
1,1,1.000000
1,REL,<=
1,0,14500.000000
2,1,1.000000
3,1,2.000000
4,1,2.000000
5,1,-550.000000
5,2,1.000000
0,3,-372.440000
6,3,1.000000
6,REL,<=
6,0,2000.000000
...
```

...es parte de un archivo con este tipo de formato. Como podemos observar, en algunos casos, el valor de la columna es «**REL**»; esto quiere decir que se debe escribir un valor en la columna de relación de la matriz.

(En estos casos, el valor es un signo matemático, por ejemplo «>=»).

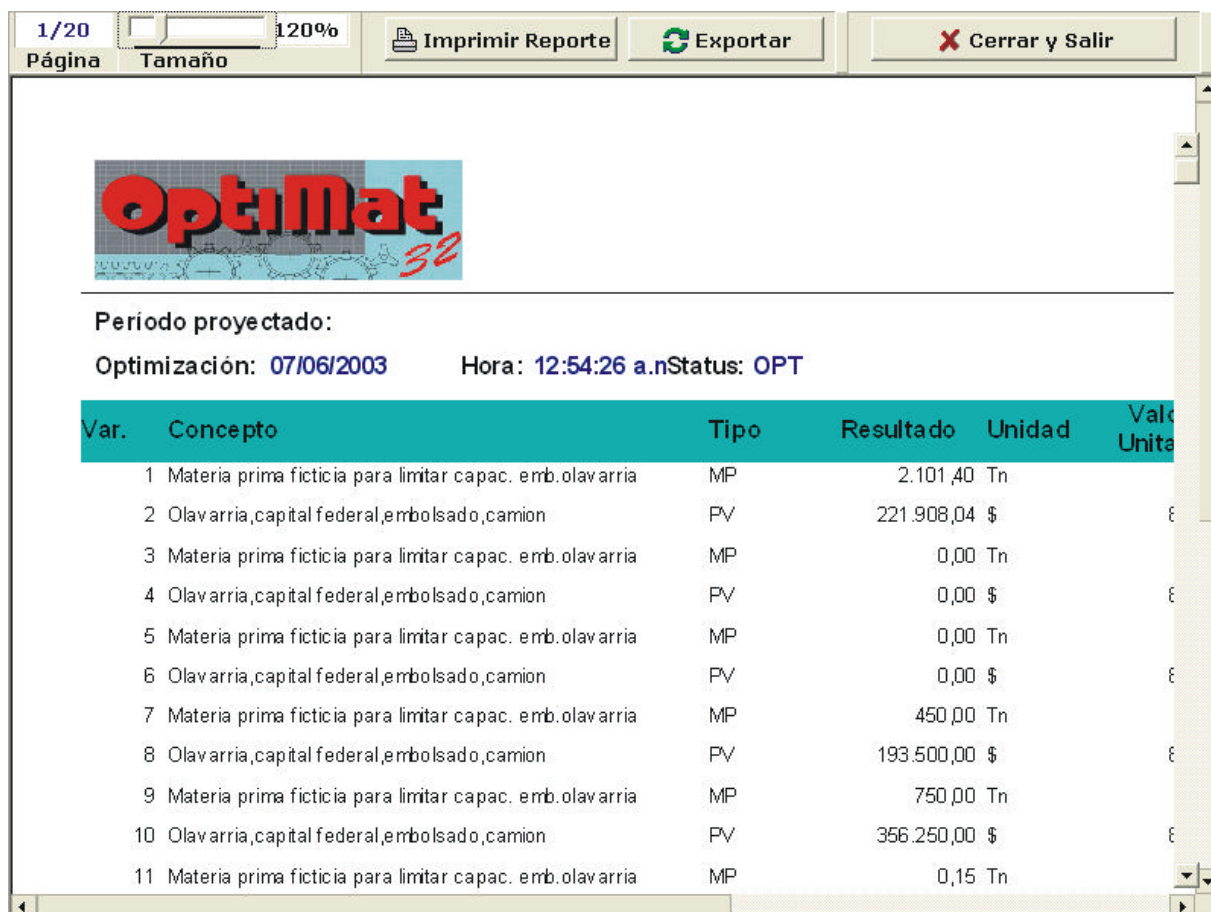
Las filas se corresponden con las restricciones, y las columnas con las variables.

En el caso de que la fila sea 0, estamos ingresando un valor en la función objetivo.

9.2.4. Módulo de generación de reportes

Este módulo permite la visualización de los listados y reportes generados por Optimat 32.

Se trata de una pantalla con los controles necesarios para manipular texto y gráficos en forma combinada. La misma, brinda la posibilidad de imprimir los reportes visualizados, en cualquier impresora disponible en el sistema, o bien en la red.



Var.	Concepto	Tipo	Resultado	Unidad	Valor Unita
1	Materia prima ficticia para limitar capac. emb.olavarria	MP	2.101,40	Tn	
2	Olavarria,capital federal,embolsado,camion	PV	221.908,04	\$	€
3	Materia prima ficticia para limitar capac. emb.olavarria	MP	0,00	Tn	
4	Olavarria,capital federal,embolsado,camion	PV	0,00	\$	€
5	Materia prima ficticia para limitar capac. emb.olavarria	MP	0,00	Tn	
6	Olavarria,capital federal,embolsado,camion	PV	0,00	\$	€
7	Materia prima ficticia para limitar capac. emb.olavarria	MP	450,00	Tn	
8	Olavarria,capital federal,embolsado,camion	PV	193.500,00	\$	€
9	Materia prima ficticia para limitar capac. emb.olavarria	MP	750,00	Tn	
10	Olavarria,capital federal,embolsado,camion	PV	356.250,00	\$	€
11	Materia prima ficticia para limitar capac. emb.olavarria	MP	0,15	Tn	

Ilustración 1 - Módulo de generación y muestreo de reportes en Optimat 32.

Número de página: n/N; donde n es la página actual, y N la cantidad total de páginas del documento.

Tamaño: Consiste en una barra de desplazamiento horizontal que modifica el tamaño de la imagen que se ve de la hoja de reporte. El tamaño que se utilice no tiene relación con el tamaño en la impresión, sólo es un control de pantalla para simplificar la lectura de letras pequeñas, o en caso contrario, para ver la distribución del texto en toda la hoja a la vez.

Imprimir reporte: Este botón se utiliza para realizar la impresión del reporte visualizado. Antes de imprimir, el sistema consultará sobre las características de la impresora a utilizar y las configuraciones deseadas en cuanto a calidad de impresión, cantidad de copias, etc.

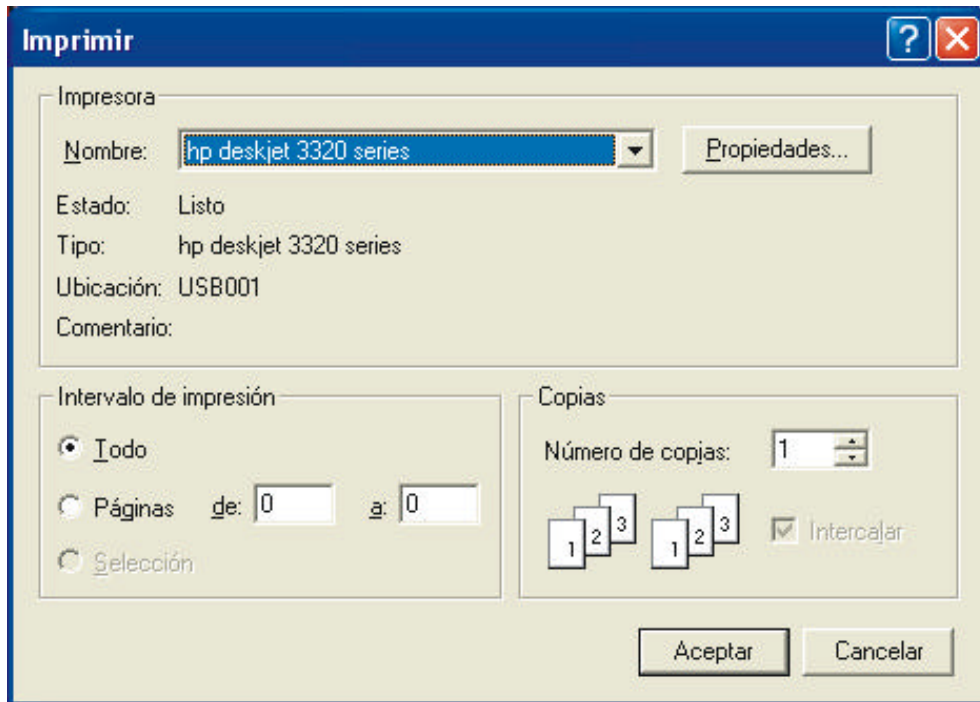


Ilustración 1 - Ventana de selección y configuración de impresoras de Microsoft Windows.

Cerrar y Salir: Sirve para cerrar la ventana actual, y abandonar la visualización del reporte.

10. Caso práctico: «La fábrica de fiambres»

Para este caso práctico, se ha utilizado el modelo de una pequeña fiambrería. El mismo consta de una reducida cantidad de productos, a efectos de simplificar la demostración. El sistema puede administrar una cantidad muchas veces superior de ítems.

10.1. El caso de la fábrica de fiambres

La fiambrería del ejemplo, fabrica siete productos para su comercialización, los cuales utilizan recursos compartidos y poseen materias primas en común.

Los productos son los siguientes:

Producto	Precio de Venta (\$/Kg.)	Demanda máxima (Kg./Mes)
Longaniza cantinera	4,00	10.000
Mortadela común vejiga	1,25	50.000
Salame Milán entero	3,50	35.000
Salame picado fino	3,90	30.000
Salamín picado grueso	3,90	30.000
Salchichas de viena x 12 (1)	2,30	100.000
Salchichas de viena x 36 (2)	2,25	20.000

(1) En cajas de 12 sachets por 6 unidades cada uno.

(2) En cajas de 36 sachets por 6 unidades cada uno.

Los siete productos se fabrican a partir de:

- . Insumos cárnicos vacunos (Ej. Salame, carne de entrecostilla, grasa vacuna, etc.)
- . Insumos cárnicos porcinos (Ej. Tocino, gordura de cerdo, etc.)
- . Aditivos no cárnicos (Ej. Leche en polvo, sal, harina, pimienta, etc.)

El proceso de fabricación depende de cada producto, haciendo uso de distintos sectores de planta, y teniendo cada uno distintas fórmulas de producción.

Además, cada producto tiene asociados su precio de venta y sus costos variables (Por ej. Materiales de empaque, - discriminados -, mano de obra, fletes, comisiones, energía, etc.).

Los productos tienen el siguiente flujo de procesos; el cual se puede observar gráficamente en el anexo 1.

a) Para productos secos: longaniza, salame Milán, salamín picado fino y grueso:

Picado -> Mezcla -> Embutido -> Secado -> Empaque

b) Para productos cocidos: salchichas y mortadela:

Picado -> Mezcla -> Embutido -> Cocción -> Cámaras -> Empaque

Se puede ver que los recursos de picado, mezcla, embutido y empaque son utilizados por todos los productos y por lo tanto «compiten» entre si en el uso de los mismos, pues son recursos con una capacidad limitada.

Por otra parte, la mortadela y las salchichas compiten en el uso de las cocinas.

Las capacidades de planta para un mes de producción (período de análisis) de 22 días por 8 horas (10.560 minutos/mes) son:

Sector	Máquinas/dotación	(Min/Mes)
Picadoras	6	63.360
Mezcladoras	4	42.240
Embutidoras	7	73.920
Cocinas	1	10.560
Empaque	2	21.120
Prep. Yogurt (*)	1	10.560

* El yogurt es un semielaborado que tiene su propia fórmula de producción y se utiliza en la fabricación de la mortadela.

Por otra parte, los otros recursos son los secaderos y las cámaras:

Sector	Capacidad (Kg.)	Permanencia (Días)	(Kg./Mes)
Secaderos	70.000	10	210.000
Cámaras	10.000	1	300.000

A título de ejemplo, mostramos a continuación la información de uno de los productos, la salchicha de viena en cajas de 36 sachets.

Proceso de fabricación:

1. Se realiza el picado fino en las PICADORAS consumiendo 0,3 minutos/Kg. con su fórmula:

- Salame de 2da 32,5 %
- Entrecostilla (carne de) 32,5 %
- Hielo 35,0 %

Se obtiene un preparado picado para salchichas que llamaremos SCHAPIC.

2. El siguiente paso es la mezcla en las MEZCLADORAS consumiendo 0,15 minutos/Kg. con su fórmula:

- SHAPIC 63,5 %
- Gordura de cerdo 25,75 %
- Fécula de trigo 8,00 %
- Sal 2,25 %

Polifosfato de sodio 0,25 %

Aroma para salchichas 0,25 %

Se obtiene una mezcla a granel que continúa en la siguiente etapa.

3. Se realiza el embutido de la mezcla en las EMBUTIDORAS consumiendo 0,20 minutos/Kg.
4. Se sigue con la cocción de las salchichas en las COCINAS consumiendo 0,10 minutos/Kg. Con una merma de un 10 %.
5. El siguiente paso es enviar el producto a CAMARAS en donde permanece 24 horas.
6. Finalmente la última operación es la de EMPAQUE: se guardan en bolsitas de 6 unidades y estos en cajas de 36 bolsitas. La velocidad de operación es de 0,05 Kg. por minuto.

Estructura de costos:

Si bien la unidad de venta es la caja de 36 bolsitas por 6 unidades, la unidad de trabajo es el kilogramo: los resultados estarán expresados en kilogramos.

La estructura de costos del producto es la siguiente:

Salchicha de viena en cajas de 36 bolsitas por 6 unidades

Precio: 2,25 \$/Kg.

Concepto de costo	Unidad	Precio	Consumo	Costo
Tripa sintética cristal calibre 23	MTR	0,020	3,7500	0,0750
Film liso fondo sachet	KGR	3,000	0,0095	0,0285
Film impreso tapa sachet	KGR	4,500	0,0055	0,0248
Caja p/salchichas X 36 sachets	UNI	0,550	0,1157	0,0637
Rótulo autoadhesivo	UNI	0,025	0,1157	0,0029
Mano de obra	HH	10,000	0,0100	0,1000
Energía eléctrica	KWH	0,100	1,2000	0,1200
Gas natural	M3	0,070	0,3500	0,0245
Fletes	KGR	0,100	1,0000	0,1000
Impuestos	%	1,500	0,0000	0,0338
Comisiones	%	12,000	0,0000	0,2700

Valor de realización:

Los productos «comparten» las materias primas, las que tienen un costo unitario de adquisición.

El sistema calcula para cada producto la diferencia entre el precio de venta y los costos unitarios, determinando así cuanto contribuyen individualmente para pagar su materia prima y su utilidad. Esta diferencia se llama *valor de realización*¹².

Objetivo que se busca lograr en la fábrica del ejemplo:

El objetivo consiste en hallar la mezcla de productos que hacen máxima la *contribución marginal total*, (diferencia entre el valor de realización total y las materias primas).

Esta mezcla debe cumplir con las restricciones impuestas por las capacidades de los distintos sectores de la fábrica y las correspondientes al mercado.

10.2. Resolución: Método 1. WinQSB¹³

El siguiente cuadro de WinQSB muestra los valores de resultado para cada variable, y también el resultado de la función maximizada. (señalado con el trazo rojo).

12. *Valor de Realización* es un término utilizado en la industria frigorífica, y representa el valor que le queda al empresario deducidos los costos directos de proceso, para pagar la materia prima, los gastos generales y su utilidad antes del impuesto a la renta.

13. WinQSB es un programa que engloba un conjunto de rutinas matemáticas para resolver diversos tipos de problemas. Para esta prueba se utilizó el módulo LP-ILP *Linear and Integer Programming*. Versión 1.0. Copyright de **YIH LONG CHANG**.

	00:27:15		Friday	August	20	2004		
	Decision Variable	Solution Value	Unit Cost or Profit c(j)	Total Contribution	Reduced Cost	Basis Status	Allowable Min. c(j)	Allowable Max. c(j)
1	1	15,916.6700	0	0	0	basic	-1.4605	M
2	Salame1	11,141.6700	-1.7500	-19,497.9200	0	basic	-3.8364	M
3	Tocino2	4,775.0000	-0.7000	-3,342.5000	0	basic	-5.5683	M
4	Sminpgpic	15,916.6700	0	0	0	basic	-1.4605	M
5	5	16,666.6700	0	0	0	basic	-1.3948	M
6	Sal	841.6667	-0.1250	-105.2083	0	basic	-17.1594	M
7	Lechpvo	141.6667	-3.2500	-460.4167	0	basic	-167.3412	M
8	Cerelose	91.6667	-0.8500	-77.9167	0	basic	-254.4455	M
9	Pimnga	33.3333	-2.0000	-66.6667	0	basic	-699.3876	M
10	Pimbca	16.6667	-3.0000	-50.0000	0	basic	-1,397.7750	M
11	Sminpgmez	16,666.6700	0	0	0	basic	-1.3948	M
12	12	16,666.6700	0	0	0	basic	-1.3948	M
13	Sminpgemb	16,666.6700	0	0	0	basic	-1.3948	M
14	14	16,666.6700	0	0	0	basic	-1.3948	M
15	Sminpgsec	12,000.0000	0	0	0	basic	-1.9372	M
16	Mersec	4,666.6670	0	0	0	basic	-4.9813	M
17	17	10,583.3300	0	0	0	basic	-0.6036	M
18	Salame2	3,439.5830	-1.6500	-5,675.3130	0	basic	-3.5072	M
19	Entrecos	3,439.5830	-0.7500	-2,579.6880	0	basic	-2.6072	M
20	Hielo	3,704.1670	0	0	0	basic	-1.7245	M
21	Schaptic	10,583.3300	0	0	0	basic	-0.6036	M
22	22	16,666.6700	0	0	0	basic	-0.3833	M
23	Gordcerd	4,291.6670	-0.7000	-3,004.1670	0	basic	-2.1884	M
24	Fectrig	1,333.3330	-0.5000	-666.6667	0	basic	-5.2909	M
25	Polsod	41.6667	-3.0000	-125.0000	0	basic	-156.3100	M
26	Arsalch	41.6667	-50.0000	-2,083.3340	0	basic	-203.3100	M
27	Schamez	16,666.6700	0	0	0	basic	-0.3833	M
28	28	16,666.6700	0	0	0	basic	-0.3833	M
29	Schaemb	16,666.6700	0	0	0	basic	-0.3833	M
30	30	16,666.6700	0	0	0	basic	-0.3833	M
31	Schacoc	15,000.0000	0	0	0	basic	-0.4259	M
32	Mercoc	1,666.6670	0	0	0	basic	-3.8328	M
33	33	15,000.0000	0	0	0	basic	-0.4259	M
34	Salchicha	15,000.0000	0	0	0	basic	-0.4259	M
35	35	10,000.0000	0	0	0	basic	-0.4259	M
36	Salch12	10,000.0000	1.3713	13,712.6400	0	basic	0.9454	M
37	37	5,000.0000	0	0	0	basic	-0.4615	M
38	Salch36	5,000.0000	1.4069	7,034.7450	0	basic	0.9454	M
39	39	12,000.0000	0	0	0	basic	-1.9372	M
40	Salpg	12,000.0000	3.9000	46,800.0000	0	basic	1.9628	M
	Objective	Function	(Max.) =	29,812.6000				

Ilustración 1 - Resolución del problema usando WinQSB.

10.3. Resolución: Método 2. Optimat

RESULTADOS DE LA OPTIMIZACION							
Periodo Proyectado : CASO A (3/93)							
Optimizacion: 2/2/2003 - Hora: 16:30 - Status: OPT							
Codigo	Tipo	Result.	Val.Unit	Total	Precio	Cota Min	Cota Max
PIMBCA	PC	17	3.0000	-50	0.0000	0	1000000
SMINPGMEZ	PI	16667	0.0000	0	0.0000	0	0
SMINPGEMB	PI	16667	0.0000	0	0.0000	0	0
SMINPGSEC	PI	12000	0.0000	0	0.0000	0	0
MERSEC	PU	4667	0.0000	0	0.0000	0	0
SALAME2	PC	3440	1.6500	-5675	0.0000	0	1000000
ENTRECOS	PC	3440	0.7500	-2580	0.0000	0	1000000
SALAME1	PC	11142	1.7500	-19498	0.0000	0	1000000
HIELO	PC	3704	0.0000	-0	0.0000	0	1000000
SCHAPIC	PI	10583	0.0000	0	0.0000	0	0
GORDCERD	PC	4292	0.7000	-3004	0.0000	0	1000000
FECTRIG	PC	1333	0.5000	-667	0.0000	0	1000000
POLSOD	PC	42	3.0000	-125	0.0000	0	1000000
ARSALCH	PC	42	50.0000	-2083	0.0000	0	1000000

Ilustración 1 - Resultados de la Optimización - Optimat.

RESULTADOS DE LA OPTIMIZACION							
Periodo Projectado : CASO A (3/93)							
Optimizacion: 2/2/2003 - Hora: 16:30 - Status: OPT							
Codigo	Tipo	Result.	Val.Unit	Total	Precio	Cota Min	Cota Max
ARSALCH	PC	42	50.0000	-2083	0.0000	0	1000000
SCHAMEZ	PI	16667	0.0000	0	0.0000	0	0
SCHAEMB	PI	16667	0.0000	0	0.0000	0	0
TOCINO2	PC	4775	0.7000	-3342	0.0000	0	1000000
SCHACOC	PI	15000	0.0000	0	0.0000	0	0
MERCOC	PU	1667	0.0000	0	0.0000	0	0
SALCHICHA	PI	15000	0.0000	0	0.0000	0	0
SALCH12	PU	10000	1.3713	13713	2.3000	0	10000
SALCH36	PU	5000	1.4069	7035	2.2500	0	5000
SMINPGPIC	PI	15917	0.0000	0	0.0000	0	0
SALPG	PU	12000	3.9000	46800	3.9000	0	12000
SAL	PC	842	0.1250	-105	0.0000	0	1000000
LECHPUO	PC	142	3.2500	-460	0.0000	0	1000000
CERELOSE	PC	92	0.8500	-78	0.0000	0	1000000

Ilustración 1 - Resultados de la Optimización – Optimat. Página 2

RESULTADOS DE LA OPTIMIZACION							
Periodo Projectado : CASO A (3/93)							
Optimizacion: 2/2/2003 - Hora: 16:30 - Status: OPT							
Codigo	Tipo	Result.	Val.Unit	Total	Precio	Cota Min	Cota Max
Contribucion Marginal : \$ 29879.2593 < Suma de Valor Total >							
Balance Entrada/Salida de Materiales:							
Entrada :		33300	< Suma de MP y de PC >				
Salida :		33333	< Suma de PU >				
Balance :		-33					

Ilustración 1 - Resultados de la Optimización – Optimat. Página 3

UTILIZACION DE SECTORES DE PRODUCCION				
Periodo Projectado : CASO A (3/93)				
Optimizacion: 2/2/2003 - Hora: 16:30 - Status: OPT				
Sector	Unidad	Utilizacion	Capacidad	Uso
CAMARAS	KGR	15000	300000	5.00%
COCINAS DE FIAMBRES	MIN	1667	10560	15.78%
EMBUTIDORAS	MIN	10000	73920	13.53%
EMPAQUE DE PRODUCTOS TERM.	MIN	1350	21120	6.39%
MEZCLADORAS	MIN	3500	42240	8.29%
PICADORAS	MIN	5562	63360	8.78%
SECADEROS	KGR	16667	210000	7.94%

Ilustración 1 - Utilización de Sectores de Producción – Optimat.

VENTAS PROYECTADAS					
Periodo proyectado : CASO A (3/93)					
Optimizacion: 2/2/2003 - Hora: 16:30 - Status: OPT					
Codigo	Precio	Volumen	Monto	Cota Min	Cota Max
SALCH12	2.3000	10000.00	23000.00	0	10000
SALCH36	2.2500	5000.00	11250.00	0	5000
SALPG	3.9000	12000.00	46800.00	0	12000

Ilustración 1 - Ventas proyectadas – Optimat.

VENTAS PROYECTADAS					
Periodo proyectado : CASO A (3/93)					
Optimizacion: 2/2/2003 - Hora: 16:30 - Status: OPT					
Codigo	Precio	Volumen	Monto	Cota Min	Cota Max
Total Volumen de Ventas :		27000.00	Kgs		
Total Monto de Ventas :		81050.00	\$		

Ilustración 1 - Ventas Proyectadas – Optimat. Página 2

REQUERIMIENTOS DE GASTOS VARIABLES				
Periodo proyectado : CASO A (3/93)				
Optimizacion: 2/2/2003 - Hora: 16:30 - Status: OPT				
Gasto	Unidad	Costo de Adq.	Requerim. de Cons.	Monto
COMISIONES	POR	12.0000		4110
CAJA PARA SALCHICHAS X 12 SCHT	UNI	0.2500	3472	868
CAJA PARA SALCHICHAS X 36 SCHT	UNI	0.5500	579	318
FILM IMPRESO TAPA DE SACHET	KGR	4.5000	82	371
FILM LISO FONDO DE SACHET	KGR	3.0000	142	428
ROTULO AUTOADHESIVO	UNI	0.0250	4051	101
TRIPA SINTETICA CRISTAL C.23	MTR	0.0200	56250	1125
ENERGIA ELECTRICA	KWH	0.1000	18000	1800
FLETES	KGR	0.1000	15000	1500
GAS NATURAL	M3	0.0700	5250	368
IMPUESTOS	POR	1.5000		514
MANO DE OBRA	HH	10.0000	200	2000

Ilustración 1 - Requerimientos de Gastos Variables – Optimat.

REQUERIMIENTOS DE GASTOS VARIABLES				
Periodo Projectado : CASO A (3/93)				
Optimizacion: 2/2/2003 - Hora: 16:30 - Status: OPT				
Gasto	Unidad	Costo de Adq.	Requerim. de Cons.	Monto
Total en Pesos de Gastos Variables :				13502.6157

Ilustración 1 - Requerimientos de Gastos Variables – Optimat. Página 2

COMPRAS PROYECTADAS					
Periodo Projectado : CASO A (3/93)					
Optimizacion: 2/2/2003 - Hora: 16:30 - Status: OPT					
Codigo	Precio	Volumen	Monto	Cota Min	Cota Max
PIMBCA	3.0000	16.67	50.00	0	1000000
SALAME2	1.6500	3439.58	5675.31	0	1000000
ENTRECOS	0.7500	3439.58	2579.69	0	1000000
SALAME1	1.7500	11141.67	19497.92	0	1000000
HIELO	0.0000	3704.17	0.00	0	1000000
GORDCERD	0.7000	4291.67	3004.17	0	1000000
PECTRIG	0.5000	1333.33	666.67	0	1000000
POLSOD	3.0000	41.67	125.00	0	1000000
ARSALCH	50.0000	41.67	2083.33	0	1000000
TOCINO2	0.7000	4775.00	3342.50	0	1000000
SAL	0.1250	841.67	105.21	0	1000000
LECHPUO	3.2500	141.67	460.42	0	1000000
CERELOSE	0.8500	91.67	77.92	0	1000000
PIMNGRA	2.0000	33.33	66.67	0	1000000

Ilustración 1 - Compras Proyectadas – Optimat.


COMPRAS PROYECTADAS					
Periodo Projectado : CASO A (3/93)					
Optimizacion: 2/2/2003 - Hora: 16:30 - Status: OPT					
Codigo	Precio	Volumen	Monto	Cota Min	Cota Max
Total Monto de Compras :		37734.79	\$		

Ilustración 1 - Compras Proyectadas – Optimat. Página 2

ESTADO DE RESULTADOS REDUCIDO		
Periodo proyectado : CASO A (3/93)		
Optimización: 2/2/2003 - Hora: 16:30 - Status: OPT		
	Monto	%
VENTAS TOTALES	81050.00	100.00
GASTOS VARIABLES TOTALES	13502.62	16.66
VALOR DE REALIZACION	67547.38	83.34
COMPRAS TOTALES	37734.79	46.56
CONTRIBUCION MARGINAL	29812.59	36.78
COSTOS FIJOS TOTALES	30000.00	37.01
RESULTADO DEL PERIODO	-187.41	-0.23

Ilustración 1 - Estado de Resultados Reducido - Optimat.

10.4. Resolución: Método 3. Optimat 32



29/08/2003 11:01:14 p.m.

Resultados de la Optimización

Periodo proyectado: 1
Optimización: 08/07/2003 Hora: 10:26:53 a.m. Status: OPT

Var.	Concepto	Tipo	Resultado	Unidad	Valor Unitario	Valor Total	Cota Inferior	Cota Superior	Cumplimiento %
2	Salsicha	FI	15,000.00	Kgr	0.00	0.00	0.00	0.00	0.00
3	Salsicha de viena en cajas de 12 salsichas	PV	10,000.00	Kgr	1.37	13,712.64	0.00	10,000.00	100.00
5	Salsicha de viena en cajas de 36 salsichas	PV	5,000.00	Kgr	1.41	7,034.74	0.00	5,000.00	100.00
7	Salsichas cocidas	FI	15,000.00	Kgr	0.00	0.00	0.00	0.00	0.00
9	Salsichas embudadas	FI	16,666.67	Kgr	0.00	0.00	0.00	0.00	0.00
10	Marra de corián	PV	1,666.67	Kgr	0.00	0.00	0.00	0.00	0.00
12	Preparado de salsichas	FI	16,666.67	Kgr	0.00	0.00	0.00	0.00	0.00
14	Picadillo fino para salsichas	FI	10,583.33	Kgr	0.00	0.00	0.00	0.00	0.00
15	Condura de cerdo	PC	4,201.67	Kgr	0.70	-3,004.17	0.00	1,000,000.00	0.43
16	Fecula de trigo	PC	1,333.33	Kgr	0.50	-666.67	0.00	1,000,000.00	0.13
17	Sal fina	PC	841.67	Kgr	0.13	-105.21	0.00	1,000,000.00	0.08
18	Poliostato de sodio	PC	41.67	Kgr	3.00	-125.00	0.00	1,000,000.00	0.00
19	Aroma para salsichas	PC	41.67	Kgr	50.00	-2,083.33	0.00	1,000,000.00	0.00
21	Salero de segunda	PC	3,438.58	Kgr	1.65	-5,675.31	0.00	1,000,000.00	0.34
22	Entrecostilla	PC	3,438.58	Kgr	0.75	-2,579.60	0.00	1,000,000.00	0.34
23	Hielo	PC	3,704.17	Kgr	0.00	0.00	0.00	1,000,000.00	0.37
25	Preparado de salamin picado grueso	FI	16,666.67	Kgr	0.00	0.00	0.00	0.00	0.00
26	Salamin picado grueso embudado	FI	16,666.67	Kgr	0.00	0.00	0.00	0.00	0.00
28	Salamin picado grueso secado	FI	12,000.00	Kgr	0.00	0.00	0.00	0.00	0.00
29	Salamin picado grueso	PV	12,000.00	Kgr	3.80	46,800.00	0.00	12,000.00	100.00
31	Picado grueso para salamin picado grueso	FI	15,916.67	Kgr	0.00	0.00	0.00	0.00	0.00
32	Leche en polvo	PC	141.67	Kgr	3.25	-460.42	0.00	1,000,000.00	0.01
33	Cereales	PC	91.67	Kgr	0.95	-77.92	0.00	1,000,000.00	0.01
34	Pimentón negro	PC	33.33	Kgr	2.00	-66.67	0.00	1,000,000.00	0.00

Página 1

Ilustración 1 - Resultados de la Optimización - Optimat 32.

Vat.	Concepto	Tipo	Resultado	Unidad	Valor Unitario	Valor Total	Cota Inferior	Cota Superior	Cumplimiento %
35	Pimenta blanca	PC	16.07	Kgr	3.00	-50.00	0.00	1,000,000.00	0.00
37	Salame de primera	PC	11,141.67	Kgr	1.75	-19,497.92	0.00	1,000,000.00	1.11
38	Tocino de segunda	PC	4,775.00	Kgr	0.70	-3,342.50	0.00	1,000,000.00	0.48
49	Morera de secado	PV	4,686.67	Kgr	0.00	0.00	0.00	0.00	0.00

Contribución Marginal: 29,812.59

Balance de Entrada/Salida de Materiales

Entrada: 33,333.33

Salida: 33,333.33

Balance: 0.00



29/08/2003 23:12:08

Utilización de Sectores de Producción

Período proyectado: 1


Optimización: 08/07/2003

Hora: 10:26:53 a.m. Status: OPT

Sector	Unidad	Utilización	Capacidad	Uso %
Camaras	Kgr	15,000.00	300,000.00	5.00
Cocinas de fiambres	Mín	1,666.67	10,560.00	15.78
Embutidoras	Mín	10,000.00	73,920.00	13.53
Empaque de productos term.	Mín	1,350.00	21,120.00	6.39
Mezcladoras	Mín	3,500.00	42,240.00	8.29
Picadoras	Mín	5,562.50	63,360.00	8.78
Preparadoras de yogurt	Mín	0.00	0.00	0.00
Secaderos	Kgr	16,666.67	210,000.00	7.94

Página 1

Ilustración 1 - Utilización de Sectores de Producción – Optimat 32.



29/08/2003 23:15:26

Ventas Proyectadas

Periodo proyectado: 1
 Optimización: 08/07/2003 Hora: 10:26:53 a.nStatus: OPT

Producto	Precio de Venta	Volumen de Venta	Monto de Venta	Cota Inferior	Cota Superior	Cumplimiento %
Salt hicha de viena en cajas de 12 sachets	2.30	10,000.00	23,000.00	0.00	10,000.00	100.00
Salt hicha de viena en cajas de 36 sachets	2.25	5,000.00	11,250.00	0.00	5,000.00	100.00
Mermá de cocción	0.00	1,688.67	0.00	0.00	0.00	0.00
Salami picado grueso	3.90	12,000.00	46,800.00	0.00	12,000.00	100.00
Mermá de secado	0.00	4,666.67	0.00	0.00	0.00	0.00
		33,333.33	81,050.00			

Página: 1

Ilustración 1 - Ventas Proyectadas – Optimat 32.



29/08/2003 23:17:47

Requerimiento de Gastos Variables

Período proyectado: 1


Optimización: 08/07/2003

Hora: 10:26:53 a.rStatus: OPT

Gasto	Unidad	Costo de Adquisición	Requerimiento de Consumo	Monto
Comisiones	Por	12.00	0.00	4,110.00
Caja para fiambres	Uni	0.80	0.00	0.00
Caja para salsichas x 12 s oht	Uni	0.25	3,472.22	868.05
Caja para salsichas x 36 s oht	Uni	0.55	578.70	318.29
Faja identificatoria	Uni	0.03	0.00	0.00
Film impreso tapa de sachet	Kgr	4.50	82.50	371.25
Film liso fondo de sachet	Kgr	3.00	142.50	427.50
Hilo choricero	Kgr	7.35	0.00	0.00
Rotulo autoadhesivo	Uni	0.03	4,050.92	101.27
Tripa vacuna calibre 44/46	Mtr	0.10	0.00	0.00
Tripa sintetica cristal c.110	Mtr	0.10	0.00	0.00
Tripa sintetica cristal c.23	Mtr	0.02	56,250.00	1,125.00
Vejiga	Uni	0.01	0.00	0.00
Energia electrica	Kwh	0.10	18,000.00	1,800.00
Fletes	Kgr	0.10	15,000.00	1,500.00
Gas natural	MB	0.07	5,250.00	367.50
Impuestos	Por	1.50	0.00	513.75
Mano de obra	Hh	10.00	200.00	2,000.00
				13,502.62

Página 1

Ilustración 1 - Requerimientos de Gastos Variables – Optimat 32.



29/08/2003 23:20:10

Compras Proyectadas

Periodo proyectado: 1
 Optimización: 08/07/2003 Hora: 10:26:53 a.m. Status: OPT

Producto	Precio de Compra	Volumen de Compra	Monto de Compra	Cota Inferior	Cota Superior	Cumplimiento %
Gradura de cerdo	0.70	4,291.67	3,004.17	0.00	1,000,000.00	0.43
Fecula de trigo	0.60	1,333.33	800.00	0.00	1,000,000.00	0.13
Salfina	0.13	841.67	105.21	0.00	1,000,000.00	0.08
Polifosfato de sodio	3.00	41.67	125.00	0.00	1,000,000.00	0.00
Aroma para salchichas	90.00	41.67	3,750.00	0.00	1,000,000.00	0.00
Salame de segunda	1.65	3,438.68	5,673.92	0.00	1,000,000.00	0.34
Entrecostilla	0.75	3,438.68	2,579.01	0.00	1,000,000.00	0.34
Halo	0.00	3,704.17	0.00	0.00	1,000,000.00	0.37
Leche en polvo	3.25	141.67	460.42	0.00	1,000,000.00	0.01
Cereales	0.65	81.67	53.18	0.00	1,000,000.00	0.01
Pericostilla	2.00	33.33	66.67	0.00	1,000,000.00	0.00
Primita blanca	3.00	16.67	50.00	0.00	1,000,000.00	0.00
Salame de primera	1.75	11,141.67	19,497.92	0.00	1,000,000.00	1.11
Tocino de segunda	0.70	4,775.00	3,342.50	0.00	1,000,000.00	0.48
		33,333.33	37,734.79			

Página 1

Ilustración 1 - Compras Proyectadas – Optimat 32.



20/08/2004 00:15:39

Estado de Resultados Reducido

Período proyectado: 1

Optimización: 20/08/2004

Hora: 00:14:44

Status: OPT

	Monto	%
Ventas Totales	81,050.00	100.00
- Gastos Variables Totales	13,502.62	16.66
- Compras Totales	37,734.79	46.56
Contribución Marginal	29,812.59	36.78
Costos Fijos Totales	30,000.00	37.01
Resultado del Período	-187.41	-0.23

Página 1

Ilustración 1 - Estado de Resultados Reducido – Optimat 32.

10.5. Conclusiones obtenidas del ensayo con el caso práctico

Puede observarse claramente en las pantallas anteriores que los resultados suministrados por Optimat 32 son igualmente precisos que los obtenidos mediante Optimat.

En cuanto al primer ensayo, podemos observar que si bien existen pequeñas diferencias en los decimales, se obtuvieron los mismos resultados para cada variable. Obviamente, se trata de solvers diferentes en ambos casos. Al utilizar Optimat y Optimat 32, se ha enlazado a ambos sistemas el solver LP88, y el mismo ha obtenido los mismos resultados ante el mismo modelo, presentado por dos sistemas diferentes.

Este ejemplo además de ser una demostración del funcionamiento del sistema, sirve para comprobar que efectivamente Optimat 32 ha generado correctamente la matriz correspondiente al problema, dado que los datos eran los mismos y los resultados obtenidos han sido análogos.

Otras particularidades que podemos señalar a partir de este ensayo, indican que Optimat 32 posee una interfaz gráfica eficiente, y que resultó mucho más sencillo utilizarlo para resolver este problema. Los resultados fueron presentados en elegantes pantallas factibles de ser impresas con la simplicidad de presionar un botón del ratón; y los datos se han ingresado simplemente «importándolos» desde la base de datos de Optimat.

La velocidad de resolución del problema ha sido acortada respecto a Optimat, dado que se han utilizado rutinas directamente desarrolladas para procesadores de 32 bits, las cuales insumen un tiempo menor para su ejecución, y generan más rápidamente matrices que pueden ser mucho más grandes en cantidad de componentes.

Se han realizado diferentes pruebas modificando algunos valores para observar el comportamiento del modelo ante diversas circunstancias. En el caso de Optimat 32, estos ensayos han podido ser almacenados por separado, para tener la posibilidad de compararlos volviendo a cargar ensayos anteriores.

En definitiva, Optimat 32 es un sistema amigable, confiable, veloz y eficiente; que combina la potencia de un solver comercial con la versatilidad de un generador de matrices capaz de generar modelos a partir de bases de datos escritas en formatos amigables para el usuario; a través de una interfaz rápida y de fácil utilización y aprendizaje, que aprovecha todas las ventajas de los entornos gráficos de los sistemas operativos comerciales más utilizados mundialmente.

10.6. Limitaciones de Optimat 32

- a) Necesidad de expresar el sistema productivo como una red. Esto significa asociar a los productos, procesos que consumen recursos y se realizan en sectores. (Restricciones).
- b) Los nodos se tratan mediante las fórmulas que mantienen el equilibrio input/output.
- c) Homogeneidad de las unidades de flujo de producto. Se puede realizar un cambio de unidades mediante las fórmulas de materias primas.

11. Bibliografía

11.1. Libros, Referencias y Manuales

- ESP INC., **LP88 Versión 7.03 - User Manual**, 1987.
- HILLIER, Frederick S., LIEBERMAN, Gerald J., **Introducción a la investigación de operaciones**, tercera edición, McGraw-Hill, México, 1991.
- HILLIER, Frederick S., LIEBERMAN, Gerald J., **Investigación de operaciones**, séptima edición, McGraw-Hill, México, 2001.
- IFRAH, Georges, **Historia Universal de las Cifras. La inteligencia de la Humanidad contada por los Números y el Cálculo**, Segunda edición, Espasa Calpe, Madrid, 1997.
- RODRIGUEZ, Víctor M., **Optimat. Documentación del sistema**, Impresión propia, Argentina, 1994.
- SENGE, Peter, **La Quinta Disciplina**, Granica, Buenos Aires, 1992.
- WILLIAMS, H.P., **Model Building in mathematical programming**, third edition, John Wiley & Sons, Great Britain, 1991.
- YACUZZI, Enrique, RODRÍGUEZ, Víctor M. **Diseño e implementación de un sistema de apoyo a las decisiones basado en el modelo de transporte**, Universidad del CEMA, Serie documentos de trabajo, N° 214, Buenos Aires, Abril 2002.

11.2. Sitios de Internet

- † Blaise Pascal (1623-1662).
<http://www-gap.dcs.st-and.ac.uk/~history/Mathematicians/Pascal.html>
- † Blaise Pascal (1623-1662).
http://www.maths.tcd.ie/pub/HistMath/People/Pascal/RouseBall/RB_Pascal.html
- † Blaise Pascal's Arithmetic Machine – 1640AD.
<http://www.maxmon.com/1640ad.htm>
- † NEOS Guide
<http://www.fp.mcs.anl.gov/otc/Guide/SoftwareGuide/Categories/optsysmodlang.html>
<http://www.fp-mcs.anl.gov/otc/Guide/SoftwareGuide/Blurbs>
- † Programación Lineal
<http://orbita.starmedia.com/~aribera/lineal.htm>
- † Programación Lineal. Teo Coronado.
<http://thales.cica.es/rd/Recursos/rd98/Matematicas/29/simplex.html>
- † Maximal Software, Inc. – MPL for Windows Manual.
<http://www.maximal-usa.com/mplman/mplwtoc.html>

11.3. Software utilizado durante el desarrollo de esta Tesina

- **ADOBE PHOTOSHOP**. Versión 7.0. De Adobe Systems Incorporated. (2002)
- **BORLAND DELPHI 6 ENTERPRISE**. Versión 6.0. de Borland Software Corporation. (2001)
- **LP88**. Versión 7.03. De ESP INC. (1987)
- **MICROSOFT OFFICE XP**. De Microsoft Corporation. (2002)
- **MICROSOFT WINDOWS XP PROFESSIONAL**. De Microsoft Corporation. (2002)
- **OPTIMAT**. Versión 1.0. De ADINSA S.A. (1994)
- **WINQSB. Módulo LP-ILP Linear and Integer Programming**. Versión 1.0. De Yih Long Chang. (1997)

12. Anexos

Anexo 1: Flujo de Procesos

