



6
21

**UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO**

**FACULTAD DE ESTUDIOS SUPERIORES
CUAUTITLAN**

**"DESARROLLO DEL SISTEMA DE
OPTIMIZACION DE RECURSOS"**

T E S I S

**QUE PARA OBTENER EL TITULO DE:
INGENIERO MECANICO ELECTRICISTA**

P R E S E N T A :

EDUARDO AGUILAR SANCHEZ

**TESIS CON
FALLA DE ORIGEN**

ASESOR: ING. SERGIO PEDRO ACOSTA TORRES



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



SECRETARÍA NACIONAL
DE EDUCACIÓN
MEXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN
UNIDAD DE LA ADMINISTRACION ESCOLAR
DEPARTAMENTO DE EXAMENES PROFESIONALES

FACULTAD DE ESTUDIOS
SUPERIORES CUAUTITLAN
DEPARTAMENTO DE
EXAMENES

ASUNTO: VOTOS APROBATORIOS

DR. JAIME KELLER TORRES
DIRECTOR DE LA FES-CUAUTITLAN
P R E S E N T E .

AT'N: Ing. Rafael Rodríguez Ceballos
Jefe del Departamento de Exámenes
Profesionales de la F.E.S. - C.

Con base en el art. 28 del Reglamento General de Exámenes, nos permitimos comunicar a usted que revisamos la TESIS:

DESARROLLO DEL SISTEMA DE OPTIMIZACION DE RECURSOS

que presenta EL pasante: EDUARDO AGUILAR SANCHEZ
con número de cuenta: 8810196-3 para obtener el TITULO de:
INGENIERO MECANICO ELECTRICISTA

Considerando que dicha tesis reúne los requisitos necesarios para ser discutida en el EXAMEN PROFESIONAL correspondiente, otorgamos nuestro VOTO APROBATORIO.

A T E N T A M E N T E .
"POR MI RAZA HABLARA EL ESPIRITU"
Cuautitlan Izcalli, Edo. de Mex.. a 8 de OCTUBRE de 1997.

PRESIDENTE	<u>ING. GLORIA VILLANUEVA AGUILAR</u>	
VOCAL	<u>L.C. VALENTIN ROLDAN VAZQUEZ</u>	
SECRETARIO	<u>ING. SERGIO PEDRO ACOSTA TORRES</u>	
PRIMER SUPLENTE	<u>ING. ROGELIO RAMOS CARRANZA</u>	
SEGUNDO SUPLENTE	<u>ING. MARTHA LILIA URRUTIA VARGAS</u>	

I. INDICE

	Pag.
I. Índice	1
II. Introducción	3
III. Antecedentes de Investigación de Operaciones.	5
1. Asignación y distribución de recursos	
1.1 Conceptos generales	11
1.2 Programación lineal	12
1.3 Transporte	28
1.4 Asignación	41
2. Planeación y control de proyectos	
2.1 Conceptos generales	50
2.2 Análisis de redes	53
2.3 Método de la ruta crítica	69
2.4 Técnica de evaluación y revisión de programas	77
3. Mantenimiento de equipo industrial	
3.1 Conceptos generales	84
3.2 Determinación de la vida económica del equipo	85
3.3 Análisis de reemplazo	91
3.4 Mantenimiento preventivo	99
3.5 Mantenimiento correctivo	105

	Pag.
4. Análisis de líneas de espera	
4.1 Conceptos generales	110
4.2 Teoría de colas	112
4.3 Líneas de espera de una fila - un servidor	117
4.4 Líneas de espera de una fila - servidores en paralelo	125
5. Aplicación del sistema en casos prácticos	
5.1 Conceptos generales	131
5.2 Mezclas de productos	132
5.3 Problemas de inventarios	139
5.4 Asignación de máquinas	144
5.5 Fabricación sobre diseño	146
5.6 Construcción de una planta eléctrica	153
5.7 Análisis de la vida económica de equipo para transportación	160
5.8 Análisis de reemplazo de fresadora para troqueles	162
5.9 Política de mantenimiento preventivo para equipo de cómputo	163
5.10 Mantenimiento de un sistema de procesamiento automático	165
5.11 Optimización de sistema de colas en planta troqueladora	167
5.12 Descripción del sistema de colas de un banco	171
5.13 Sistema de líneas de espera en serie (Línea de ensamble)	173
 Conclusiones	 176
Bibliografía	178

II. INTRODUCCION

Los modelos matemáticos de optimización (MMO) forman un conjunto de herramientas que se han diseñado para ayudar a hacer selecciones óptimas de entre un conjunto dado de alternativas. Proporcionan un método objetivo y lógico, para analizar los problemas en un esfuerzo que permite tomar las mejores decisiones.

Un MMO es la representación de un problema o un aspecto de la realidad en lenguaje matemático, y para el cual existe un algoritmo para encontrar la solución. Un algoritmo es simplemente un conjunto de procedimientos o reglas que, cuando se siguen en forma ordenada, proporcionan la mejor solución para un modelo determinado.

Los MMO pueden describir substancialmente un problema, son rápidamente computarizados y fácilmente manipulados para probar diferentes soluciones.

El objetivo de este trabajo, consiste en la implementación de algunos MMO en un lenguaje de programación de alto nivel, para poder obtener así, un sistema basado en computadora, que nos permita hacer uso de dichos modelos de manera rápida y eficiente. Con esto se impulsará la utilización de estos modelos ya que su uso ha sido limitado, por que una de las principales desventajas con que cuentan es que se requieren realizar una gran cantidad de operaciones aritméticas antes de obtener la solución óptima.

En general, existe una gran cantidad de MMO que se pueden utilizar en las diferentes áreas con las que cuenta una empresa de negocios, sin embargo, aquí sólo se consideraran aquellos modelos cuya utilización tienen mayor impacto en el área de Producción y en la Dirección de Operaciones, áreas que están generalmente bajo la responsabilidad de ingenieros.

El Sistema denominado "Sistema de Optimización de Recursos" se codificará en el lenguaje de alto nivel CA-Clipper versión 5.3 de la compañía Computers Associates, Inc., el cual es utilizado en el desarrollo de aplicaciones, bajo el modelo relacional para el manejo de bases de datos. Será utilizado este lenguaje, por que haciendo un análisis de los algoritmos aquí implementados, se observó que el modelo relacional para el manejo de bases de datos, es adecuado para poder almacenar a los MMO, ya que la estructura de éstos se puede representar por medio de tablas relacionadas entre sí. Sin embargo, y aunque se utilice este lenguaje, el análisis y el diseño de los programas son válidos para cualquier lenguaje de programación estructurada (como C, Pascal, Modula-2, etc.), de programación orientada a objetos (como C++, Visual Objects, etc.) y de programación visual (como Visual Basic, Visual C++, Delphi, etc.).

El trabajo se encuentra dividido en cinco capítulos. En los primeros cuatro se encuentra la descripción de los programas que fueron desarrollados en la implementación de cada modelo, para poder ser ejecutados en computadora. La forma en que han sido distribuidos, a lo largo de estos capítulos, es la siguiente: En el primer capítulo se encuentran los modelos de Asignación y Distribución de Recursos en el que se implementan los algoritmos de Programación Lineal (Simplex - Gran M), de Transporte (Aproximación de Vogel y Cruce del Arroyo) y de Asignación (Húngaro). En el segundo están los modelos de Planeación y Control de Proyectos, en el se encuentra el método de ruta crítica (CPM) y la técnica de evaluación y revisión de programas (PERT). En el tercero se incluyen los modelos de Mantenimiento de Equipo Industrial, tales como el de determinación de la Vida Económica del Equipo, el de Mantenimiento Preventivo, Mantenimiento Correctivo, así como el Análisis de Reemplazo. En el cuarto capítulo se analizan los algoritmos más importantes de Líneas de Espera, son los que describen líneas de espera de Una Fila - Un servidor y Una Fila - Servidores en Paralelo.

Finalmente, en el quinto capítulo se presenta un conjunto de problemas y su solución obtenida con el sistema. Algunos de estos problemas incluyen datos reales, con lo cual se pretende comprobar la utilidad del sistema en forma práctica.

III. ANTECEDENTES DE INVESTIGACION DE OPERACIONES

La Investigación de Operaciones es la aplicación, por grupos interdisciplinarios, del método científico a problemas relacionados con el control de las organizaciones o sistemas (hombres - máquinas) a fin de que se produzcan soluciones que mejor sirvan a los objetivos de toda la organización.

Una organización se puede interpretar como un sistema, pues así se facilita su entendimiento. Todo sistema tiene componentes e interacciones entre las mismas. Algunas interacciones son controlables, mientras que otras no lo son.

La Investigación de Operaciones es la aplicación de la metodología científica a través de modelos, primero para representar al problema real que se quiere resolver en un sistema y segunda para resolverlo. Los modelos que utiliza la Investigación de Operaciones son matemáticos y toman la forma de ecuaciones. Estos modelos son diferentes a otros modelos, por ejemplo de experimentación que se utilizan en las ciencias médico - biológicas, o las de representación que se utilizan en la Astronomía.

Los modelos matemáticos de decisión permiten calcular los valores exactos o aproximados de las componentes controlables del sistema para que pueda comportarse mejor, de acuerdo con ciertos criterios establecidos. Estos cálculos se realizan bajo el supuesto de que se conoce la información asociada al estado de aquellas componentes del sistema que no se pueden controlar. El acto de calcular el valor apropiado de estas componentes controlables, se conoce como derivar la solución al problema en cuestión, utilizando un modelo. La manera como se logra esta derivación de soluciones es muy variada y no existen reglas generales. Se puede lograr por simulación o emulación, o bien por un riguroso análisis matemático (algoritmo).

En la Investigación de Operaciones se utiliza en tres tipos de problemas: determinísticos, con riesgo, bajo incertidumbre.

Los problemas determinísticos son aquellos en los que cada alternativa del problema (hay más de una) tiene una y sólo una solución. Como hay varias alternativas, hay también varias soluciones, cada una con una diferente eficiencia y/o efectividad asociada a los objetivos del sistema. Por lo tanto, existe el problema de decisión.

Los problemas con riesgo son aquellos en los que cada alternativa del problema (hay más de una), tiene varias soluciones. Cada solución puede ocurrir con una cierta probabilidad. La distribución de estas probabilidades se conoce o se puede estimar.

Los problemas bajo incertidumbre son aquellos en los que cada alternativa del problema (hay más de una), tiene varias soluciones. Sin embargo se ignora con qué probabilidad o distribución probabilística ocurrirán estas soluciones.

A partir de la Revolución Industrial, el mundo ha tenido un notable crecimiento en la magnitud y complejidad de las organizaciones, los pequeños talleres de los artesanos del siglo pasado se han desarrollado hasta llegar a las grandes corporaciones de la actualidad. Los resultados de este cambio han sido un tremendo incremento en la división de la mano de obra y la segmentación de las responsabilidades de administración de las organizaciones. Así mismo los resultados han sido un tremendo incremento de la productividad, sin embargo, junto con ello, esta especialización ha creado nuevas situaciones adversas que aún se están presentando en muchas organizaciones. Uno de los problemas es la tendencia de muchos componentes de una organización a crecer en forma aislada y autónoma, con sus propias metas, perdiendo en consecuencia la visión de en que forma sus actividades y objetivos se entrelazan con los de la organización en conjunto. Lo que es mejor para uno de los componentes con frecuencia puede ir en detrimento de otro, de modo que pueden finalizar operando con propósitos cruzados. Una situación relacionada con lo anterior es que a medida que se incrementa la complejidad y especialización de una organización, se torna cada vez más difícil asignar los recursos disponibles a sus diversas actividades de manera que sea lo más efectivo para la organización

como un todo. Estos tipos de problemas y la necesidad de hallar la mejor manera de resolverlos dio lugar al medio necesario para que surgiera la Investigación de Operaciones.

Los inicios de lo que hoy se conoce como Investigación de Operaciones se remonta a los años 1759 cuando el economista Questnay empieza a utilizar modelos primitivos de programación matemática. Más tarde, otro economista de nombre Walras, hace uso, en 1874, de técnicas similares. Los modelos lineales de la Investigación de Operaciones, tienen como precursores a Jordan en 1873, Minkowsky en 1896 y a Farkas en 1903. Los modelos dinámicos probabilísticos tienen su origen con Markov a fines del siglo pasado. El desarrollo de los modelos inventarios, así como el de tiempos y movimientos, se lleva a cabo por los años veinte de este siglo, mientras que los modelos de líneas de espera se originan con los estudios de Erlang, a principios del siglo XX. Los problemas de asignación se estudian con métodos matemáticos por los húngaros Konig y Egervary en la segunda y tercera década de este siglo. Los problemas de distribución se estudian por el ruso Kantorovich en 1939. Von Neuman cimienta en 1937 lo que años más tarde culminará como la Teoría de Juegos y la Teoría de Preferencias (esta última desarrollada en conjunto con Morgenstern). Hay que hacer notar que los modelos de la Investigación de Operaciones que utilizaron estos precursores, estaban basados en el Cálculo Diferencial e Integral (Newton, Lagrange, Laplace, Lebesgue, Reimman, Stieltjes, por mencionar algunos), la Probabilidad y la Estadística (Bernoulli, Poisson, Gauss, Bayes, Gosset, Snedecor, etc.).

No fue sino hasta la Segunda Guerra Mundial, cuando la Investigación de Operaciones empezó a tomar auge. Primero se le utilizó en la logística estratégica para vencer al enemigo (Teoría de Juegos) y, más tarde al finalizar la guerra, en la logística de distribución de todos los recursos militares de los aliados dispersos por todo el mundo. Fue debido precisamente a este último problema, que la fuerza aérea norteamericana, a través de su centro de investigación Rand Corporation, comisionó a un grupo de matemáticos para que resolviera este problema que estaba consumiendo tantos recursos humanos, financieros y materiales. Fue el doctor George Dantzing, el que en 1947, resumiendo el trabajo de muchos de sus precursores, inventara el método Simplex, con lo cual dio inicio a la Programación Lineal. Con el avance de las computadoras digitales se empezó a extender la Investigación de Operaciones, durante la década de los

cincuenta en las áreas de Programación Dinámica (Bellman), Programación No Lineal (Kuhn y Trucker), Programación Entera (Gomory), Redes de Optimización (Ford y Fulkerson), Simulación (Markowitz), Inventarios (Arrow, Karlin, Scarf, Whitin), Análisis de Decisiones (Raiffa), Procesos Markovianos de Decisión (Howard). La generalización de la Investigación de Operaciones han tratado de darla Churchman, Ackoff y Arnoff.

A pesar de que la Investigación de Operaciones puede considerarse anterior a la Revolución Industrial, fue precisamente durante este movimiento cuando empezaron a desarrollarse los problemas que iba a resolver la Investigación de Operaciones. Hasta mediados del siglo pasado, muchas empresas industriales empleaban solamente unos pocos hombres. El advenimiento de los medios mecánicos - cuando la máquina sustituyó al hombre como fuente de energía - y el desarrollo de los sistemas de comunicación y transporte le dieron gran impulso a la industria, que comenzó a crecer hasta alcanzar el grado de avance que tiene en la actualidad. A medida que las empresas crecían, era cada vez más difícil que un hombre solo las manejara; en consecuencia el propietario dividió su trabajo en funciones y las asignó a otras personas. De este modo se crearon, por ejemplo, los cargos de gerentes de producción, finanzas, personal, mercadeo e investigación y desarrollo. Al continuar el crecimiento industrial, aún estas funciones se subdividieron: por ejemplo, la producción se dividía, a veces, en abastecimiento o compras, mantenimiento, tráfico, control de calidad y programación de producción. Conforme crecieron y se extendieron las poblaciones, se crearon nuevos mercados y se describieron nuevas fuentes de materias primas. Por consiguiente, las operaciones industriales se dispersaron geográficamente. La multiplicidad de instalaciones de producción y oficinas de venta se generalizó y cada una requería su propia administración. De esta manera, la segmentación funcional y geográfica de la administración tal como la conocemos hoy día, fue consecuencia natural del crecimiento industrial originado por la Revolución Industrial.

La función del ejecutivo, en la industria, se desarrolló gradualmente conforme lo hicieron las organizaciones. El ejecutivo no estuvo sujeto a los fuertes estímulos de la nueva tecnología, como le ocurrió al gerente de producción. El ejecutivo se arraigó con sus problemas y la solución de éstos aparentemente no

requería otra cosa que un buen criterio con base en la experiencia correspondiente. Debido a ello, el ejecutivo no sintió la necesidad de examinar sus problemas de una manera más estrictamente científica.

Desde sus comienzos, la Investigación de Operaciones se ha aplicado a gran variedad de problemas. Sin embargo, la mayoría de éstos, han sido de naturaleza táctica más que estratégica. La diferencia entre problemas tácticos y estratégicos no es sencilla, debido a que se basa, cuando menos, en tres características, cada una de las cuales implica una cuestión de magnitud.

Primero, un problema es más táctico que otro si el efecto de su solución es de menor solución, o lo que en sí es lo mismo si su solución puede modificarse o anularse fácilmente. Cuanto mayor es la duración del efecto de la solución del problema, éste más estratégico. Por lo tanto el problema consiste en lo que se producirá mañana es más táctico que el referente al lugar dónde se construirá una planta adicional. La Investigación de Operaciones se ha aplicado con más frecuencia a problemas a corto que a largo plazo. Podemos referirnos a esta característica del problema como a su rango.

En segundo lugar, un problema es más estratégico cuanto mayor sea la parte de la organización afectada directamente por su solución. Por tanto un problema que asocie la selección de un convenio contable es aparentemente más táctico que, por ejemplo, el presupuesto de la empresa. Esta característica de un problema puede referirse como su alcance.

Finalmente, un problema es más estratégico cuanto más implica la determinación de fines, metas u objetivos. Todos los problemas comprenden la selección de medios para alcanzar los resultados deseados, pero muchos consideran los resultados deseados como dados o proporcionados. En la medida que lo hacen son tácticos. De aquí que la planeación de la empresa, que debe establecer metas y objetivos en la organización, es más estratégica que un problema que trate de minimizar los costos de transporte, en la que dicha minimización se considera como resultado conveniente. Esta característica de un problema puede referirse como su orientación afines.

No hay puntos de separación definidos en las escalas que representan estas tres características que distinguen los problemas tácticos de los estratégicos. Por esta razón, lo más que podemos decir de un problema es que es, más o menos estratégico que otro, en función de esas características.

Como se mencionó antes, buena parte de la Investigación de Operaciones, pero de ninguna manera toda, se ha ocupado de problemas de naturaleza más bien táctica que estratégica.

Actualmente, la Investigación de Operaciones no sólo se aplica en el sector privado (industrias, sistemas de comercialización, sistemas financieros, transportes, sistemas de salud, etc.) sino también en el sector de los servicios públicos, tanto en los países desarrollados como en los países del tercer mundo. Precisamente en México, la Investigación de Operaciones se utiliza dentro del sector de servicios públicos, entre otros en la Compañía Nacional de Subsistencias Populares, la Secretaría de Comunicaciones y Transportes, la Secretaría de Recursos Hidráulicos, Petróleos Mexicanos, Comisión Federal de Electricidad, Instituto Mexicano del Seguro Social, Banco de México, Departamento del Distrito Federal, etc.

Existen varias asociaciones que agrupan a miembros que se dedican a la aplicación de la Investigación de Operaciones en todo el mundo. Solamente en los Estados Unidos, agrupaban desde 1972 a varios miles de miembros; la Operations Research Society of América, con aproximadamente 8,000 miembros y el Institute of Management Science, con 6,000 miembros. Además, hay asociaciones Canadienses, Europeas, Latinoamericanas y Asiáticas. Se publican mensualmente más de dos docenas de revistas diferentes en todo el mundo, relacionadas a los tópicos de Investigación de Operaciones.

Hoy en día se han diseñado programas profesionales y de posgrado (maestrías y doctorados) en la especialidad de Investigación de Operaciones. Se puede concluir que la Investigación de Operaciones todavía se encuentra en una edad incipiente y que hay mucho por hacer en el desarrollo de este campo tan fértil, tanto en su teoría como en su aplicación.

1. ASIGNACION Y DISTRIBUCION DE RECURSOS

1.1 CONCEPTOS GENERALES.

Los modelos de asignación y distribución, tienen como objetivo, el optimizar los recursos disponibles en una organización, ésto se hace ya sea maximizando los beneficios o minimizando los costos, en problemas de combinaciones, distribuciones y/o asignaciones.

Los recursos en una organización, son todos los elementos con los que cuenta para poder lograr sus objetivos, los recursos que generalmente se encuentran en la industria son: los recursos humanos (personas), los recursos materiales (equipo, edificios, materia prima etc.), los recursos financieros (dinero) y el tiempo.

Los modelos de asignación y distribución pueden aplicarse en una gran cantidad de situaciones, tales como en la ingeniería de producto, en el establecimiento de programas de producción, en la planeación y el control de inventarios, en la logística y la distribución de mercancías, etc.

Los modelos de asignación y distribución que aquí se desarrollan son los siguientes: Programación Lineal, que es un modelo para resolver aquellos problemas de combinaciones o mezclas, que se puedan plantear por medio de ecuaciones lineales; Transporte el cual realiza la distribución óptima en aquellos problemas, en los que se requiere enviar ciertos recursos desde un origen hasta un destino; y finalmente el modelo de Asignación, el cual se utiliza para realizar asignaciones óptimas de acuerdo al rendimiento global del problema.

1.2 PROGRAMACION LINEAL.

La programación lineal es un modelo de análisis, que sirve para elegir la mejor alternativa de entre varias. Cuando esta mejor alternativa incluye un conjunto coordinado de actividades, se le puede llamar plan o programa. La palabra "programa" se utiliza porque programar significa seleccionar la combinación de actividades a realizar.

Con frecuencia seleccionar una alternativa incluye satisfacer varios criterios al mismo tiempo. Estos criterios se pueden dividir en dos categorías: restricciones y objetivo. Las restricciones son las condiciones que deben satisfacer todas las relaciones entre las variables de decisión, el objetivo se usa para seleccionar una de entre todas las alternativas factibles. Existen muchos problemas que se ajustan a este modelo, el tratar de minimizar o maximizar un objetivo que está sujeto a una lista de restricciones. En la actualidad, la programación lineal encuentra aplicación en una gran cantidad de campos de actividades, pudiendo indicar concretamente:

- Mezclas de productos.
- Programación de inventarios.
- Distribuciones de planta.
- Determinación de estándares de tiempo.
- Asignación de gastos de operación.
- Planificación de la fuerza de trabajo.

La programación lineal es una técnica determinística, no admite probabilidades.

El objetivo y cada una de las restricciones se debe expresar como una relación lineal, de ahí su nombre de programación lineal.

Una característica más de la programación lineal es que carece de significado, el hecho de que alguna o algunas variables puedan tomar valores negativos.

El modelo general de programación lineal está formado por:

- a) Función objetivo. Puede ser de "Maximización" o de "Minimización".
- b) Restricciones funcionales. Pueden ser menor o igual que (\leq), igual que (=) o mayor o igual que (\geq). Únicamente un signo ocurre para cada restricción.
- c) Restricciones de no negatividad. Las variables de decisión no pueden ser negativas ya que no tienen ningún sentido real.

La forma general de un modelo de programación lineal se puede expresar de la siguiente forma:

Optimizar **FUNCION OBJETIVO** $Z = C_1X_1 + C_2X_2 + \dots + C_nX_n$

Sujeto a las **RESTRICCIONES**

$$a_{11}X_1 + a_{12}X_2 + \dots + a_{1n}X_n \geq b_1$$

$$a_{21}X_1 + a_{22}X_2 + \dots + a_{2n}X_n \leq b_2$$

$$\dots$$

$$a_{m1}X_1 + a_{m2}X_2 + \dots + a_{mn}X_n = b_m$$

y **RESTRICCIONES DE NO NEGATIVIDAD:**

$$X_1, X_2, \dots, X_n \geq 0$$

Ya que la programación lineal sirve para determinar la forma en que se deben combinar las variables en un problema para tener el valor óptimo de la función objetivo, la información que debe proporcionar este modelo para poder tomar decisiones es la siguiente:

- a) Para cada variable que forma parte de la solución, cuál es la cantidad en que debe ser utilizada.
- b) Considerando la cantidad de participación de cada una de las variables que integran la solución, cuál es el impacto que tiene sobre el objetivo del problema.
- c)Cuál es el valor total de llevar a cabo la combinación de variables que forman la solución.
- d) Qué cantidad de recursos disponibles no son utilizados en la solución.
- e) Finalmente, información adicional para realizar análisis posteriores, tales como el impacto de invertir en elevar una restricción del problema o en disminuir cierta cantidad de las variables que forman parte de la solución.

Existe un algoritmo, llamado "método simplex", que sirve para resolver problemas de programación lineal. Este algoritmo usa una ecuación de criterio (función objetivo) para seleccionar de manera sistemática una solución óptima de entre muchas soluciones factibles posibles, además, este es el método general que puede aplicarse a problemas de cualquier tamaño.

El procedimiento que se requiere para poder implementar el método simplex en computadora, consiste en lo siguiente:

1. Capturar los datos generales de la función objetivo.
2. Capturar el valor de cada variable que integra la función objetivo.
3. Capturar los datos generales de cada restricción que integran al modelo.
4. Normalizar el modelo, convirtiendo las desigualdades en igualdades y agregando variables de holgura.
5. Agregar variables no negativas a cada una de las expresiones correspondientes de los tipos (\geq) e ($=$). Estas variables se denominan variables artificiales y su adición hace que se infrinjan las restricciones correspondientes. Esta dificultad se elimina asegurando que las variables artificiales sean cero en la solución. Para lograr esto, se asigna una penalización grande por unidad a estas variables en la función objetivo. Tal penalización se designa como $-M$ para problemas de maximización y $+M$ para problemas de minimización y $M > 0$.
6. Construir la tabla simplex con los datos del problema, la cual incluye valores, variables de decisión, variables de holgura y puede o no contener variables artificiales. Esta incluye una solución básica factible, cuando se encuentra definido en la misma tabla la matriz unidad. Si no es así, efectuar las transformaciones correspondientes para obtenerla.
7. Seleccionar la variable que entra y que sale a la solución:
 - 7.1 Variable que entra a la solución: Para maximización entra la de mayor valor positivo en el renglón de la función objetivo. Para

minimizaci3n entra el valor m1s negativo en el rengl3n de la funci3n objetivo. Esto define la columna pivote.

7.2 Variable que sale de la soluci3n: En la columna pivote la relaci3n menor positiva, de la columna que contiene los valores entre los elementos correspondientes de la columna pivote.

8. Si para problemas de maximizaci3n, ya no existen coeficientes positivos o para problemas de minimizaci3n ya no existen coeficientes negativos en el rengl3n de la funci3n objetivo, 3sta es la soluci3n 3ptima y continuar con el punto 9. Si no pasar al punto 7.

9. Una vez encontrada la soluci3n 3ptima se organiza la soluci3n para poder ser presentada.

10. Se presenta la informaci3n en pantalla o en impresora.

El c3digo en donde se implementan en computadora, los tres primeros pasos del algoritmo simplex, es presentado a continuaci3n, en el se lleva ha cabo la lectura, desde la consola, de los datos que caracterizan al problema que se quiere resolver:

```
@ 05.02 TO 21.77
SETCOLOR(Gicolbri)
@ 05.16 SAY ' Descripci3n del Modelo [Ctrl-W para Salir] '
SETCOLOR(Gicolnor)
REPL A->DesMemo WITH MEMOEDIT(A->DesMemo,06.03,20.76,..T...73.3)
@ 05.02 CLEA TO 21.77
FOLDER(04,04,18.64,"      Modelo      ")
@ 09.05 SAY 'Tipo de optimizaci3n : '
@ 11.05 SAY 'Descripci3n : '
@ 13.05 SAY 'N1mero de variables : '
@ 15.05 SAY 'N1mero de restricciones:'
Maxmin:=VAL(A->TipOpti)
@ 09.29 PROMPT '1. Maximizar'
@ 09.45 PROMPT '2. Minimizar'
@ 11.29 GET A->DesGene Pict 'S35' VALID !EMPTY(A->DesGene)
@ 13.29 GET A->NumVari VALID A->NumVari > 0
```

ASIGNACION Y DISTRIBUCION DE RECURSOS

```

@ 15.29 GET A->NumRest VALID A->NumRest > 0
SETCOLOR(Gcolormen)
MENU TO Maxmin
SETCOLOR(Gcolornor)
READ
REPL A->TipOpti WITHI strf(Maxmin,1,0)
IF A->NumVari=0 AND A->NumRest=0
  folder(06,09,20,69," Funcion Objetivo ")
  @ 11.10 SAY 'No. de Variable:'
  @ 13.10 SAY 'Cantidad'
  @ 15.10 SAY 'Descripcion'
  @ 17.10 SAY 'Unidades'
  Lc_nuva = 1
  SELECT 2
  SET ESCAPE ON
  DO WHILE Lc_nuva <= A->NumVari AND LASTKEY() <= 27
    Lc_lla1:=Lc_lla+StrZero(Lc_nuva,6,0)
    SEEK(Lc_lla1)
    IF EOF()
      APPEND BLANK
      REPL B->NumMode WITHI Lc_lla1,;
        B->NumVari WITHI StrZero(Lc_nuva,6,0)
    ENDIF
    SETCOLOR(Gcolobr)
    @ 11.28 SAY LTRIM(STR(VAL(B->NumVari),6,0))
    SETCOLOR(Gcolornor)
    @ 13.28 GET B->ImpDeno Pict '9999999999.999';
    VALID B->ImpDeno <= 0
    @ 15.28 GET B->DesVari
    @ 17.28 GET B->UniVar
    READ
    Lc_tec1:=Lastkey()
    DO CASE
      CASE Lc_tec1=18
        IF Lc_nuva > 1
          SKIP -1
          Lc_nuva--
        ENDIF
      OTHERWISE
        SKIP
        Lc_nuva++
      ENDCASE
    ENDDO
    SET ESCAPE OFF
    folder(08,14,22,74," Restricciones ")
    @ 13.15 SAY 'Restriccion:'
    @ 16.15 SAY 'Descripcion:'
    @ 19.15 SAY 'Unidades'
    Lc_nuva = 1
    SELECT 3
    DO WHILE Lc_nuva <= A->NumRest
      Lc_lla1:=Lc_lla+StrZero(Lc_nuva,6,0)
      SEEK(Lc_lla1)
      IF EOF()
        APPEND BLANK
        REPL C->NumMode WITHI Lc_lla1,;
          C->NumRest WITHI StrZero(Lc_nuva,6,0)
      ENDIF
      Coeficie:=-0.00
      SETCOLOR(Gcolobr)
      @ 13.28 SAY LTRIM(STR(VAL(C->NumRest),6,0))
      SETCOLOR(Gcolornor)
      @ 16.28 GET C->DesRest
      @ 19.28 GET C->UnRest
      READ
      Lc_tec1:=LASTKEY()

```

ASIGNACION Y DISTRIBUCION DE RECURSOS

```

DO CASE
CASE Lc_tec1=18
IF Lc_nuva > 1
SKIP -1
Lc_nuva--
ENDIF
OTHERWISE
SELECT 4
Lc_cont=1
DO WHILE Lc_cont <= A->NumVari .AND. LASTKEY() <> 27
Lc_colu=Lc_tlav+C->NumRest+StrZero(Lc_cont,6,0)
SEEK(Lc_colu)
IF EOF()
APPEND BLANK
REPL D->NumMode WITH A->NumMode.;
D->NumReng WITH C->NumRest.;
D->NumColu WITH StrZero(Lc_cont,6,0)
ENDIF
SELECT 2
SEEK(D->NumMode+D->NumColu)
SELECT 4
Lc_pan3=-SAVESCREEN(18,26,22,74)
@ 18,26 CLEA TO 22,74
@ 18,26 TO 22,74
@ 19,28 SAY 'Variable:'
SETCOLOR(Gicolbn)
@ 19,38 SAY SUBS(LTRIM(STR(VAL(D->NumColu),6,0)));
'-' + B->DesVari,1,36)
SETCOLOR(Gicolnor)
@ 21,28 SAY 'Cantidad:' Get D->ImpDeno;
PICT '9999999999,999'
READ
RESTSCREEN(18,26,22,74,Lc_pan3)
Lc_tec1=LASTKEY()
DO CASE
CASE Lc_tec1=18
IF Lc_cont > 1
Skip -1
Lc_cont--
ENDIF
OTHERWISE
SKIP
Lc_cont++
ENDCASE
ENDDO
Lc_pan3=-SAVESCREEN(18,37,22,63)
Lc_cnop=VAL(C->TipRest)
@ 18,37 CLEA TO 22,63
@ 18,37 TO 22,63
@ 19,38 PROMPT '1. Menor o igual que (S)'
@ 20,38 PROMPT '2. Mayor o igual que (2)'
@ 21,38 PROMPT '3. Igual que (=)'
NETCOLOR(Gicolmen)
MENU TO Lc_cnop
SETCOLOR(Gicolnor)
RESTSCREEN(18,37,22,63,Lc_pan3)
REPL C->TipRest WITH Str(Lc_cnop,1,0)
Lc_pan3=-SAVESCREEN(18,35,22,66)
@ 18,33 CLEA TO 22,66
@ 18,33 TO 22,66
@ 20,37 SAY 'Cantidad' Get C->ImpDeno;
Pict '9999999999,999'
Read
RESTSCREEN(18,33,22,66,Lc_pan3)
SELE 3
SKIP

```

```
Le nuva**  
ENDCASE  
ENDDO
```

En el procedimiento anterior se capturan los datos en tres secciones diferentes. El procedimiento comienza con la captura de los datos generales referentes al problema planteado. Aquí se encuentra el código que permite que se capturen los datos como, la descripción del modelo, el tipo de optimización, el número de variables y el número de restricciones con que cuenta. Más adelante, continúa con la captura de las variables que conforman a la función objetivo, aquí además de permitir capturar el valor para cada variable, que integra la función objetivo, se encuentran las líneas de código para poder capturar una descripción y el tipo de unidades con que cuenta, con esto se podrá presentar la información en la solución final en un formato que sea fácil de leer.

En la tercera y última parte, se realiza la captura de los datos referentes a las restricciones. En esta parte se capturan los datos de cada variable que integra a cada una de las restricciones del problema, así como el tipo de restricción y el valor a que se está restringiendo la combinación de variables.

Los datos que son capturados desde la consola, se almacenan directamente en archivos de la base de datos del sistema, en estos archivos se guardan en forma de tablas relacionadas entre sí, haciendo así que el tamaño del modelo sea independiente del algoritmo y que sólo éste limitado por el medio de almacenamiento donde se ejecuta el sistema.

El procedimiento mostrado anteriormente, es la parte del programa que interactúa con el usuario en la entrada de datos, por lo cual, tiene incluido todas las posiciones, colores y manejos de pantalla necesarios para que el usuario capture de forma natural y en un ambiente elegante los datos que integran su problema.

Los pasos 4, 5 y 6 del algoritmo son codificados en el procedimiento **Snormaliza** que a continuación se muestra:

PROCEDURE Snormaliza

** Convierte un modelo en matriz.
 PARA aMatriz, nModelo

```

SELE 6
ZAP
RestMayor:=0
RestMenor:=0
RestIguar:=0
SELECT 1
SEEK(nModelo)
SELECT 3
SEEK(nModelo)
DO WHILE .NOT. EOF() .AND. nModelo=C->NUMMODE
DO CASE
CASE C->TipRest=='1'
RestMayor++
CASE C->TipRest=='2'
RestMayor++
CASE C->TipRest=='3'
RestIguar++
ENDCASE
SKIP
ENDDO
M->NUMRENG:=RestMayor+RestMenor+RestIguar+1
M->NUMCOLU:=a->NumVari+RestMenor+2*RestMayor+RestIguar+2
FOR J=1 TO M->NUMRENG
FOR I=1 TO M->NUMCOLU
aMatriz[J,I]:=(0.00,1)
NEXT
NEXT
SELECT 2
SEEK (nModelo)
SELECT 3
SEEK (nModelo)
FOR J=1 TO M->NUMRENG
FOR I=1 TO M->NUMCOLU
IF I==1
IF J==1
aMatriz[J,I]:=(0,1)
SELECT 6
APPEND BLANK
REPL F->NumColu WITH StrZero(1,6.0);
F->DesColu WITH 'VALORES'
ELSE
SELECT 3
SEEK (nModelo+StrZero(J-1,6.0))
aMatriz[J,I]:=ntof(c->ImpDeno)
ENDIF
ELSEIF I<=a->NumVari+1
aMatriz[J,I]:=(0.00,1)
IF J==1
SELECT 2
SEEK (nModelo+StrZero(I-1,6.0))
aMatriz[J,I]:=ntof(b->ImpDeno)
SELECT 6
APPEND BLANK
REPL F->NumColu WITH StrZero(1,6.0);
F->DesColu WITH B->DesVari.;
F->UniColu WITH B->UniVari
ELSE
SELECT 4

```

ASIGNACION Y DISTRIBUCION DE RECURSOS

```

SEEK (nModelo+StrZero(J-1,6,0)+StrZero(I-1,6,0))
aMatriz[J,I]:=not(d->ImpDeno)
ENDIF
ELSEIF I<=(M->NUMCOLU-M->NUMRENG)
IF I=1 AND I<(M->NUMCOLU-M->NUMRENG)<0]
SELECT 3
SEEK (nModelo+StrZero I;
-(M->NUMCOLU-M->NUMRENG)-1,6,0))
IF c->TipRest='2' or c->TipRest='3'
IF a->TipOpt='1'
aMatriz[J,I]:=(Gran_M,1)
ELSE
aMatriz[J,I]:=(-Gran_M,1)
ENDIF
SELECT 6
APPEND BLANK
REPL F->NumColu WITH StrZero(I,6,0);
F->DesColu WITH 'ARTIFICIAL'
ELSE
SELECT 6
APPEND BLANK
REPL F->NumColu WITH StrZero(I,6,0);
F->DesColu WITH C->DesRest;
F->UniColu WITH C->UniRest
ENDIF
ELSE
IF I<(M->NUMCOLU-M->NUMRENG)=J
aMatriz[J,I]:=(1,1)
ELSE
aMatriz[J,I]:=(0,1)
ENDIF
ENDIF
NEXT
NEXT
NEXT
I:=1
J:=2
DO WHILE I<=(M->NUMCOLU-M->NUMRENG-a->NumVari-1) AND;
J<=(M->NUMRENG
SELECT 3
SEEK (nModelo+StrZero(J-1,6,0))
SELECT 6
LOCA FOR F->NumColu=StrZero(I+A->NumVari+1,6,0)
IF EOF()
APPEN BLANK
ENDIF
SELECT 3
REPL F->NumColu WITH StrZero(I+A->NumVari+1,6,0);
F->DesColu WITH C->DesRest;
F->UniColu WITH C->UniRest
IF c->TipRest='2'
aMatriz[J,I+a->NumVari+1]:=(-1,1)
I++
ELSE
aMatriz[J,I+a->NumVari+1]:=(0,1)
ENDIF
J:=
ENDDO
RETURN (NIL)

```

El objetivo del procedimiento **Snormaliza** es convertir los datos, previamente capturados y almacenados en archivos, en una matriz, que incluya los datos del problema y que se encuentre normalizada, en forma de tabla simplex, para poder efectuar las operaciones necesarias. Primero convierte los datos almacenados en las estructuras de los archivos en una matriz, los cuales consisten en los coeficientes de la función objetivo y de todas las restricciones que conforman al problema. La matriz constituida es formada en dos partes, de tal forma que los datos numéricos son manejados en forma de un cociente de dos enteros y no como un número con punto decimal y una precisión limitada, evitando así al máximo todo tipo de error por redondeo dentro del proceso de solución. Con esto se va a garantizar que la solución sólo incluirá el error por redondeo inherente a la presentación de la solución, el cual es despreciable.

El procedimiento continúa realizando la adición de las variables de holgura y las variables artificiales, integrando así una solución básica factible, de ésta forma queda la matriz completa como una tabla simplex, lista para poder iniciar el algoritmo de optimización.

A continuación se presenta el procedimiento en el cual se lleva a cabo el algoritmo de optimización, el se encarga de obtener la mejor solución para un problema de programación lineal, si éste la tiene. Es un algoritmo iterativo y se efectúa tantas veces como sea necesario, para encontrar la solución o para determinar que el problema no la tiene, esto último, en algunas ocasiones, se debe a que los problemas están mal planteados. Es un procedimiento tipo función, es decir que regresa un valor, este valor es un indicador de tipo lógico, el cual toma una valor positivo para casos en que se pueda seguir mejorando la solución, y un valor negativo si se encontró la mejor solución o si no existe solución para el problema.

FUNCTION Soptimiza
PARA aMatriz, vMaxmin

EsOptimo := .F.
ColPivot := 0
ValPivot := 0
FOR J := 2 TO M -> NUMCOLU

ASIGNACION Y DISTRIBUCION DE RECURSOS

```

ValCelda := aMatriz[1,J][1]/aMatriz[1,J][2]
IF (vMaxmin=1 .AND. ValPivote >= ValCelda .AND.;
ValCelda != 0) .OR.;
(vMaxmin=2 .AND. ValPivote <= ValCelda .AND.;
ValCelda != 0 .AND.;
J=M->NUMCOLU:M->NUMRENG[1]-1)
ValPivote := aMatriz[1,J][1]/aMatriz[1,J][2]
ColPivote := J
ENDIF
NEXT
IF ColPivote > 0
RenPivote:=0
ValPivote:=(Gran M,1)
FOR I:=2 TO M->NUMRENG
DUMMY:=Div(aMatriz[1,1],aMatriz[1,ColPivote])
IF (ValPivote[1]/ValPivote[2]) >= DUMMY[1]/DUMMY[2];
AND (DUMMY[1]/DUMMY[2]>0 .OR. (DUMMY[1]/DUMMY[2])=0;
AND;
aMatriz[1,ColPivote][1]/aMatriz[1,ColPivote][2]>0))
ValPivote:=Div(aMatriz[1,1], aMatriz[1,ColPivote])
RenPivote:=1
ENDIF
NEXT
IF RenPivote > 0
Reduce (aMatriz, RenPivote, ColPivote)
ENDIF
ELSE
EsOptimo:=.T.
ENDIF
RETURN (EsOptimo)

```

Así mismo, en el procedimiento anterior se lleva a cabo la determinación de la variable que entra y la variable que sale de la solución en caso de que se pueda mejorar la solución. Para el caso de la iteración que ya no pueda ser mejorada, este mismo procedimiento la puede identificar. Para las iteraciones que aún se pueden mejorar, este procedimiento llama al procedimiento **Reduce** mandándole como parámetros tanto la columna como el renglón pivote, el cual se encarga de obtener una mejor solución del modelo. a continuación se presenta éste procedimiento:

```

PROCEDURE Reduce
** Reducción matricial.
PARA aMatriz, RenPivote, ColPivote
LOCAL ValPivote, I, J

ValPivote:=aMatriz[RenPivote,ColPivote]
FOR I:=1 TO M->NUMCOLU
aMatriz[RenPivote,I]:=Div(aMatriz[RenPivote,I],ValPivote)
NEXT
FOR J:=1 TO M->NUMRENG
IF J <> RenPivote
ValPivote:=aMatriz[J,ColPivote]
FOR I:=1 TO M->NUMCOLU
aMatriz[J,I]:=Suma(aMatriz[J,I];
Mulnp(aMatriz[RenPivote,I],Minus(ValPivote)))

```

```

NEXT
ENDIF
NEXT
RETURN (NIL)

```

Una vez que se ha obtenido la solución óptima del problema, de la matriz en la cual se encuentra la solución, es tomada la información para poder ser presentada tanto en pantalla como en impresora. A continuación se presentan los procedimientos con los cuales se puede obtener la información impresa y que es la última parte del procedimiento de implementación en computadora del método simplex. Esta es la parte que interactúa con el usuario en la salida de la información:

```

FUNCTION Resultado
** Resultado.
LOCA Lc_Resu
Lc_Resu:=.T.
SELE 5
ZAP
FOR I:=2 TO A->NumVar+1
Basica :=.T.
Todocero:=.T.
Renglon :=0
FOR J:=1 TO M->NumReng
IF (aMatriz[J,I]][1]/aMatriz[J,I][2])<=1 .AND.;
(aMatriz[J,I]][1]/aMatriz[J,I][2])<=0
Basica :=.F.
Todocero:=.F.
ELSEIF (aMatriz[J,I][1]/aMatriz[J,I][2])=1 .AND.;
Todocero
Basica :=.T.
Todocero:=.F.
Renglon :=J
ELSEIF (aMatriz[J,I][1]/aMatriz[J,I][2])=1
Basica :=.F.
Todocero:=.F.
ENDIF
NEXT
IF Basica
SELE 2
SEEK(A->NumMode+StrZero(1-1,6,0))
SELE 5
APPEND BLANK
REPL E->DesResu WITH B->DesVari.;
E->ImpResu WITH I;
(aMatriz[Renglon,1][1]/aMatriz[Renglon,1][2]);
E->ImpUnit WITH B->ImpDeno.;
E->ImpToia WITH E->ImpResu*E->ImpUnit.;
E->UniResu WITH B->UniVari.;
E->TipReg: WITH '1'
ENDIF
NEXT
FOR I:=1 TO M->NumColu-A->NumVari-1

```

```

Columna := A->NumVari-1+1
SELE 6
LOCA FOR StrZero(Columna,6,0)=F->NumColu
SELE 5
Basica := T.
Todocero := T.
Renglon := 0
FOR J:=1 TO M->NumReng
  IF (aMatriz[J,Columna][1]/aMatriz[J,Columna][2]) <= 1;
  -AND:
  (aMatriz[J,Columna][1]/aMatriz[J,Columna][2]) <= 0
  Basica := F.
  Todocero := F.
ELSEIF (aMatriz[J,Columna][1]/aMatriz[J,Columna][2]) == 1;
  -AND. Todocero
  Basica := T.
  Todocero := F.
  Renglon := J
ELSEIF (aMatriz[J,Columna][1]/aMatriz[J,Columna][2]) == 1
  Basica := F.
  Todocero := F.
ENDIF
NEXT
IF Basica .AND. !(F->(EOF())) .AND.;
F->DesColu <= 'ARTIFICIAL'
SELE 5
APPEND BLANK
REPL E->DesResu WITH F->DesColu.;
E->ImpResu WITH
Abs(aMatriz[Renglon,1][1]/aMatriz[Renglon,1][2]);
E->TipRegi WITH '2.';
E->UniResu WITH F->UniColu
ELSEIF Basica .AND. !(F->(EOF())) .AND.;
F->DesColu = 'ARTIFICIAL'
Lc Resu := F.
ENDIF
NEXT
FOR I:=2 TO M->NumColu
SELE 6
LOCA FOR StrZero(I,6,0)=F->NumColu
IF I <= A->NumVari+1
IF (aMatriz[I,I][1]/aMatriz[I,I][2]) <= 0
SELE 2
SBEK(A->NumMode+StrZero(I-1,6,0))
SELE 5
APPEND BLANK
REPL E->DesResu WITH B->DesVari.;
E->ImpResu WITH
Abs(aMatriz[I,I][1]/aMatriz[I,I][2]);
E->TipRegi WITH '3.';
E->UniResu WITH B->UniVar
IF A->TipOpt='1' .AND. E->ImpResu > 0
REPL E->SigResu WITH '1.'
ELSE
REPL E->SigResu WITH '4.'
ENDIF
ENDIF
ELSEIF (F->(EOF())) .AND. F->DesColu <= 'ARTIFICIAL'
IF (aMatriz[I,I][1]/aMatriz[I,I][2]) <= 0
SELE 5
IF EMPTY(F->DesColu)
APPEND BLANK
REPL E->DesResu WITH F->DesColu.;
E->ImpResu WITH
Abs(aMatriz[I,I][1]/aMatriz[I,I][2]);
E->TipRegi WITH '3.';

```

ASIGNACION Y DISTRIBUCION DE RECURSOS

```

E->UniResu WITH F->UniColu
IF A->TipOpti='1' AND E->impResu > 0
  REPL E->SigResu WITH '-'
ELSE
  REPL E->SigResu WITH '.'
ENDIF
ENDIF
NEXT
RETURN (Lc_Resu)

```

PROCEDURE Impresion
 ** Impresion de resultados.

```

LcPan2 :=SAVE SCREEN(10,21,14,58)
@ 10,21 CL@A TO 14,58
@ 10,24 TO 14,55 DOUBLE
SETCOLOR(glc@lbr)
@ 12,27 SAY 'Imprimiendo resultados ...'
SETCOLOR(glc@lner)
SELE 3
INDEX ON E->TipRegi TO &gldirind.Oplresul
GO TOP
SET DEVI TO PRINT
SET CONSOL OFF
SET PRINT ON
impTot:=0.00
RENI :=60
LLAVE :=''
SELE 1
DO TITULOS
RENI:=RENI - 2
COLT:=INT((130-LEN(ALLTRIM(A->DesGene)))/2)
@ RENI,COLT SAY ALLTRIM(A->DesGene)
@ RENI,COLT SAY ALLTRIM(A->DesGene)
nNumin:=MILCOUNT(A->DesMemo,76)
RENI:=RENI - 2
FOR nLinea:=1 TO nNumin
  DO TITULOS
  @ RENI,27 SAY MEMOLINE(A->DesMemo,76,nLinea)
  RENI:=RENI-1
NEXT
RENI:=RENI-2
SELE 5
DO WHILE NOT. EOF()
  DO TITULOS
  IF LLAVE <> E->TipRegi
    LLAVE = E->TipRegi
  DO CASE
    CASE E->TipRegi='1'
      @ RENI,05 SAY 'SOLUCION'
      @ RENI,05 SAY 'SOLUCION'
    CASE E->TipRegi='2'
      @ RENI,05 SAY 'EXCEDENTES'
      @ RENI,05 SAY 'EXCEDENTES'
    CASE E->TipRegi='3'
      @ RENI,05 SAY 'PRECIOS SOMBRA'
      @ RENI,05 SAY 'PRECIOS SOMBRA'
    ENDCASE
  RENI:=RENI+1

```

```

ENDIF
IF LLAVE='1'
  @ REN1,08 SAY E->DesResu
  @ REN1,53 SAY E->ImpResu PICT '999,999,999,999'
  @ REN1,78 SAY E->UniResu
  @ REN1,94 SAY E->ImpResu*E->ImpUnit PICT,
    '999,999,999,999'
  M->ImpTota=M->ImpUnit
  (E->ImpResu*E->ImpUnit)
ELSEIF LLAVE='2'
  @ REN1,08 SAY E->DesResu
  @ REN1,53 SAY E->ImpResu PICT,
    '999,999,999,999'
  @ REN1,78 SAY E->UniResu
ELSEIF LLAVE='3'
  @ REN1,08 SAY E->DesResu
  @ REN1,71 SAY ' '
  @ REN1,78 SAY E->UniResu
  @ REN1,90 SAY ' ' E->SigResu='Y'
  @ REN1,94 SAY E->ImpResu PICT,
    '999,999,999,999'
ENDIF
REN1=REN1+1
SKIP
IF LLAVE <> E->TipRegi AND LLAVE='1'
  @ REN1,94 SAY REPL(' ',19)
  @ REN1, 94 SAY M->ImpTota;
  PICT '999,999,999,999'
  REN1=REN1+1
ENDIF
ENDDO
SPACE(20)
SET PRINT OFF
SET CONSOL OFF
SET DEVI TO SCREEN
RESTSCREEN(10,21,14,58,LcPan2)
RETURN

```

El procedimiento **Resultado** extrae la información de la matriz óptima y la combina con los datos del modelo y los almacena en los registros de un archivo. El procedimiento **Impresion** enciende la impresora, toma la información del archivo generado por **Resultado** y genera un reporte que incluye la solución final.

Al ejecutar los procedimientos que se mostraron, de manera conjunta, se obtiene el programa que resuelve los problemas de programación lineal, sin importar su tamaño, evitando errores internos de redondeo y presentando los resultados de una forma clara para poder interpretar fácilmente la solución de los problemas.

1.3 TRANSPORTE.

El modelo de transporte tiene que ver con la determinación de un plan de costo mínimo para transportar mercancía desde varias fuentes a varios destinos. El modelo se puede extender de una manera directa para abarcar situaciones prácticas de las áreas de control de inventarios, programación de empleos y asignación de personal, entre otros.

El método de transporte es básicamente un modelo de programación lineal que se puede resolver por el método simplex. Sin embargo, su estructura especial hace posible el desarrollo de un procedimiento de solución, conocido como técnica de transporte, que es más eficiente en términos de cálculo.

La manera más fácil de reconocer un problema de transporte es por su naturaleza o estructura "de - hacia": de un origen hacia un destino, de una fuente hacia un usuario, del presente hacia el futuro, de aquí hacia allá. Al enfrentar este tipo de problemas, la intuición dice que debe haber una manera de obtener una solución. Se conocen las fuentes y los destinos, las capacidades, demandas y los costos de cada trayectoria. Debe haber una combinación óptima que minimice el costo (o maximice la ganancia). La dificultad estriba en el gran número de combinaciones posibles.

En general, se especifica un problema de transporte mediante los siguientes datos:

1. Un conjunto determinado de puntos de oferta, desde los cuales se envía una utilidad y que sólo puede mandar cierta cantidad de unidades.
2. Un conjunto determinado de puntos de demanda hacia los cuales se envía la utilidad y que debe recibir una cierta cantidad de unidades.

3. El costo que incurre el envío de cada unidad desde el punto de oferta hacia el punto de demanda.

El problema consiste en determinar las cantidades transportadas, tal que los costos resulten mínimos, satisfaciendo los puntos de demanda y sin superar la capacidad de los puntos de oferta.

La información que debe proporcionar la técnica de transporte, para resolver de la manera óptima este tipo de problemas y que sirve para poder tomar las decisiones adecuadas, es la siguiente:

- a) De qué origen a que destino se debe mandar y cuáles son las cantidades óptimas.
- b)Cuál es el costo de enviar, de cada origen a cada destino, la cantidad óptima.
- c)Cuál es el costo total de llevar a cabo todos los envíos.

El modelo de transporte, en realidad no consiste de un sólo algoritmo, sino de varios. Sin embargo, existe una estrategia general: Primero se construye una matriz de transporte, la cual incluye, los orígenes y sus ofertas, los destinos y sus demandas, así como los costos de cada trayectoria origen-destino. Después se continúa encontrando una solución inicial. Esta solución inicial puede o no ser óptima. La única manera de saberlo es probándolo y para lo cual existen varias técnicas para hacerlo. Si la solución no es óptima, se revisa y la prueba se repite. Cada iteración la solución estará más cerca del óptimo.

En ocasiones, en el primer paso de la técnica, es necesario balancear la matriz de transporte. Esto se hace agregando orígenes o destinos ficticios, con un costo de envío igual a cero, para el caso de que el problema tenga un número diferente de la oferta y de la demanda, ya que esto último es indispensable para poder ejecutar los algoritmos de solución.

En general la metodología del modelo de transporte la podemos resumir de la siguiente manera:

1. Establecer la matriz de transporte y balancear el modelo. Las cantidades de las ofertas tienen que ser iguales a las cantidades de las demandas y establecer la matriz de transporte.
2. Determinar una solución factible básica. Para ello existen varios métodos como el de la esquina noroeste, el método de aproximación de Vogel, el costo mínimo, etc.
3. Determinar la variable que entra de entre las variables no básicas. Para ello existe el método del banquillo (o cruce del arroyo). Si tales variables satisfacen la condición de optimidad, éste es el modelo óptimo, de otra manera continuar en el punto.
4. Determinar la variable que sale de entre las variables de la solución básica real y definir la nueva solución básica. Regresar al punto 3.

Los programas que implementan a cada una de la etapas de la metodología se muestran a continuación.

La captura de los datos de cada modelo y su almacenamiento, en los archivos correspondientes, se hace de manera similar que como se hace en el método simplex para resolver los problemas de programación lineal.

El primer paso de la metodología del modelo de transporte, la cual consisten en la generación de la matriz de transporte así como su balanceo se realiza en el siguiente procedimiento:

```

PROCEDURE TNormalza
PARA nModelo, aMatriz, aOrigen, aDestin
LOCA mSumori, mSumdes, CosMaxi

SELECT 4
SEEK(nModelo)
SELECT 3
SEEK(nModelo)
SELECT 2
SEEK(nModelo)
SELECT 2
mSumori:=0.00
mSumdes:=0.00
CosMaxi:=0.00
FOR I:=1 TO M->NumReng
SEEK(nModelo+StrZero(I,6,0))
IF NOT EOF()
  aOrigen[I]:=b->CanOrig
  mSumori :=mSumori+b->CanOrig
ELSE
  aOrigen[I]:=0.00
ENDIF
NEXT
SELECT 3
FOR J:=1 TO M->NumColu
SEEK(nModelo+StrZero(J,6,0))
IF NOT EOF()
  aDestin[J]:=c->CanDest
  mSumdes :=mSumdes+c->CanDest
ELSE
  aDestin[J]:=0.00
ENDIF
NEXT
IF mSumori > mSumdes      && Balanceo.
  M->NumColu:=M->NumColu+1
  aDestin[M->NumColu]:=mSumori-mSumdes
FOR I:=1 TO M->NumReng
  aMatriz[LM->NumColu]:= {1,0}
NEXT
ELSEIF mSumdes > mSumori
  M->NumReng:=M->NumReng+1
  aOrigen[M->NumReng]:=mSumdes-mSumori
FOR J:=1 TO M->NumColu
  aMatriz[M->NumReng,J]:= {1,0}
NEXT
ENDIF
SELECT 4
FOR I:=1 TO M->NumReng
FOR J:=1 TO M->NumColu
SEEK(nModelo+StrZero(I,6,0)+StrZero(J,6,0))

```

```

IF .NOT. EOF()
  aMatriz[I,J]:= [d->ImpCost,0]
ELSE
  aMatriz[I,J]:= {0,0}
ENDIF
ENDIF
IF aMatriz[I,J][1] >= CosMaxi .AND. aMatriz[I,J][1];
  != GRAN_M
  CosMaxi:=aMatriz[I,J][1]
ENDIF
NEXT
NEXT
IF A->TipOpti='1'
  FOR I:=1 TO M->NumReng
    FOR J:=1 TO M->NumCola
      IF aMatriz[I,J][1]! =GRAN_M
        aMatriz[I,J][1]:=-CosMaxi-aMatriz[I,J][1]
      ENDIF
    NEXT
  NEXT
ENDIF
SELECT 1
RETURN (NIL)

```

En el procedimiento **Tnormaliza**, se crea la matriz de transporte, la cual está constituida de una matriz de costos de envío, dos vectores, uno con la oferta y el otro con la demanda y se reserva el espacio para almacenar las cantidades que van ha ser enviadas desde los orígenes hasta los destinos. Así mismo, en la parte señalada, se lleva a cabo el proceso de balanceo para el caso en que la cantidad de la demanda no sea igual al de la oferta. Finalmente el procedimiento concluye, transformando la matriz, para casos en que el problema sea de maximización, ya que cuando los destinos están relacionados con los orígenes por medio de utilidad en lugar de costos, el objetivo es maximizar y la forma más conveniente de resolver este problema es transformar la matriz en costos relativos, lo cual se logra restando la utilidad mayor de la matriz de utilidad de cada celda, quedando el costo relativo de la celda y proseguir con la metodología. Una vez lograda la solución óptima, los costos relativos en las celdas son reemplazados por las utilidades originales y así se procede a calcular la utilidad total.

El segundo paso de la metodología es el establecimiento de una solución inicial básica factible. Existen varios algoritmos para poder conseguir ésta solución inicial pero uno de los mejores es el método de

aproximación de Vogel (MAV) ya que proporciona una mejor solución de inicio, esto es debido a que el algoritmo toma en cuenta los costos unitarios. Este algoritmo es implementado con los siguientes procedimientos:

```

PROCEDURE AsignVogel
PARA aMatriz, aOrigen, aDestin
LOCA aOripas, aDespas, aOricos, aDescos

aOripas:=ACLONE(aOrigen)
aDespas:=ACLONE(aDestin)
aOricos:=ARRAY(30)
aDescos:=ARRAY(30)
NumAsig:=0
DO WHILE aSuma(aOripas,M->NumReng)+aSuma(aDespas,M->NumColu)>0
  NumAsig:=NumAsig+1
  MAVCASTIGO(aMatriz,aOripas, aDespas, aOricos, aDescos)
  TipCost:=""
  CosMaxi:=0.00
  FOR I:=1 TO M->NumReng-M->NumColu
    IF I<= M->NumReng
      IF CosMaxi < aOricos[I]
        CosMaxi := aOricos[I]
        Numero := I
        TipCost := 'O'
      ENDIF
    ELSE
      IF CosMaxi < aDescos[I-M->NumReng]
        CosMaxi := aDescos[I-M->NumReng]
        Numero := I-M->NumReng
        TipCost := 'D'
      ENDIF
    ENDIF
  NEXT I
  IF TipCost='O'
    CosMini:=GRAN_M
    CosColu:=0
    FOR I:=1 TO M->NumColu
      IF CosMini > aMatriz[Numero,I][1].AND. aDespas[I]>0
        CosMini := aMatriz[Numero,I][1]
        CosColu := I
      ENDIF
    NEXT I
  IF aOripas[Numero] > aDespas[CosColu]
    aMatriz[Numero,CosColu][2]:=aDespas[CosColu]
    aOripas[Numero] := aOripas[Numero]-aDespas[CosColu]
    aDespas[CosColu]:=0
  ELSE
    aMatriz[Numero,CosColu][2]:=aOripas[Numero]
    aDespas[CosColu]:=aDespas[CosColu] - aOripas[Numero]
  
```

ASIGNACION Y DISTRIBUCION DE RECURSOS

```

aOripas[Numero] :=0
ENDIF
ELSE
  CosMini:=GRAN_M
  CosReng:=0
  FOR I:=1 TO M->NumReng
    IF CosMini > aMatriz[1,Numero][1].AND.aOripas[1]>0
      CosMini := aMatriz[1,Numero][1]
      CosReng := 1
    ENDIF
  NEXT
  IF aDespas[Numero] > aOripas[CosReng]
    aMatriz[CosReng,Numero][2]:=aOripas[CosReng]
    aDespas[Numero] := aDespas[Numero] - aOripas[CosReng]
    aOripas[CosReng]:=0
  ELSE
    aMatriz[CosReng,Numero][2]:=aDespas[Numero]
    aOripas[CosReng]:=aOripas[CosReng] - aDespas[Numero]
    aDespas[Numero] :=0
  ENDIF
ENDIF
ENDDO
FOR K:=NumAsig+1 TO M->NumReng-M->NumColu-1
  FOR J:=1 TO M->NumReng
    FOR I:=1 TO M->NumColu
      IF aMatriz[J,I][2]==0 .AND. aMatriz[J,I][1]!=GRAN_M
        Renglon:=J
        Columna:=I
      ENDIF
    NEXT
    aMatriz[Renglon,Columna][2]:=-1      && Delta.
  NEXT
NEXT
RETURN (NIL)

```

PROCEDURE MAVCASTIGO
 PARA aMatriz, aOripas, aDespas, aOrcos, aDescos

```

FOR J:=1 TO M->NumReng
  IF aOripas[J]<<0
    nNumpeq1:=GRAN_M
    nNumpeq2:=GRAN_M
    FOR I:=1 TO M->NumColu
      IF aDespas[I]<<0
        IF nNumpeq1 > aMatriz[J,I][1]
          IF nNumpeq2 > nNumpeq1
            nNumpeq2 := nNumpeq1
          ENDIF
        nNumpeq1:=aMatriz[J,I][1]
      ELSEIF nNumpeq2 > aMatriz[J,I][1]
        nNumpeq2:=aMatriz[J,I][1]
      ENDIF
    ENDIF
  ENDIF

```

```
      NEXT
      aOricos[J]:=nNumpeq2-nNumpeq1
    ELSE
      aOricos[J]:=0.00
    ENDIF
  NEXT
  FOR J:=1 TO M->NumColu
    IF aDespas[J]<=0
      nNumpeq1:=GRAN_M
      nNumpeq2:=GRAN_M
      FOR I:=1 TO M->NumReng
        IF aOripas[I]<=0
          IF nNumpeq1 > aMatriz[I,J][1]
            IF nNumpeq2 > nNumpeq1
              nNumpeq2 := nNumpeq1
            ENDIF
            nNumpeq1:=aMatriz[I,J][1]
          ELSEIF nNumpeq2 > aMatriz[I,J][1]
            nNumpeq2:=aMatriz[I,J][1]
          ENDIF
        ENDIF
      NEXT
      aDescos[J]:=nNumpeq2-nNumpeq1
    ELSE
      aDescos[J]:=0.00
    ENDIF
  NEXT
  RETURN (NIL)
```

Lo primero que hace el procedimiento **AsignVogel** es seleccionar cada renglón el menor costo unitario y restarlo del que le sigue en magnitud, esto se denomina castigo unitario al envío, y el cual es calculado por el procedimiento **MavCastigo** el cual es llamado desde **AsignVogel**.

Después se repite el mismo procedimiento, pero ahora para cada columna.

Una vez que se han calculado los castigos unitarios al envío, se procede hacer la asignación en aquella columna o renglón con mayor castigo unitario al envío, en la casilla de menor costo tomando en cuenta a la oferta y a la demanda. Se ajusta la oferta y la demanda y elimina el renglón o columna satisfecha. Todo esto lo lleva a cabo hasta que se tengan todas las asignaciones.

Finalmente termina con verificar el número de asignaciones, las cuales tienen que ser igual al número de renglones más el número de columnas menos uno. En caso de que no se cumpla esta condición se procede a completar las asignaciones con una constante "delta" que en este caso toma el valor de -1, pero que no tiene más que el propósito de cumplir con el número de asignaciones necesarias.

Una vez que se tiene la matriz con la solución inicial, se procede a optimizarla en caso de que se pueda. Esto se logra revisando cada una de las casillas no asignadas, para determinar que variable entra en la solución, en caso de existir, y luego se procede con la determinación de la variable que sale en caso de que exista. Esto se lleva a cabo mediante los siguientes procedimientos:

```

FUNCTION TopOptima
PARA aMatriz, NumReng, NumColu
LOCA nCosto, nRenviv, nColpiv, lresult, J, I, nCosto:=0.00

FOR J:=1 TO M->NumReng
FOR I:=1 TO M->NumColu
IF aMatriz[J,I][2]=0
nDummy:=Costo(J,I)
IF nDummy < nCosto
nCosto :=nDummy
nRenviv:=J
nColpiv:=I
ENDIF
NEXT I
NEXT J
nCostot:=0.00
FOR J:=1 TO M->NumReng
FOR I:=1 TO M->NumColu
nDummy:=-1IF(aMatriz[J,I][2]=-1,0,aMatriz[J,I][2])
IF a->TipOpu='1'
nCostot:=nCostot+(aMatriz[J,I][1]+nCostmax)*nDummy
ELSE
nCostot:=nCostot+aMatriz[J,I][1]*nDummy
ENDIF
NEXT I
NEXT J
IF nCosto < 0.00
Reasigna(nRenviv,nColpiv)

```

ASIGNACION Y DISTRIBUCION DE RECURSOS

```

IResult:=.F.
ELSE
IResult:=.T.
ENDIF
RETURN (IResult)

```

```

PROCEDURE Circuito
PARA RenCelda, ColCelda

```

```

aCircuit:=ARRAY(50,2)
aMatras:=aCLONE(aMatriz)
aMatras[RenCelda,ColCelda][2]=.1
FOR K:=1 TO (M->NumReng+M->NumColu)
  FOR J:=1 TO M->NumReng
    Element:=0
    FOR I:=1 TO M->NumColu
      IF aMatras[J,I][2]<>0
        Columna:=I
        Element:=Element+1
      ENDIF
    NEXT
    IF Element==1
      aMatras[J,Columna][2]=0
    ENDIF
  NEXT
  FOR I:=1 TO M->NumColu
    Element:=0
    FOR J:=1 TO M->NumReng
      IF aMatras[J,I][2]<>0
        Renglon:=J
        Element:=Element+1
      ENDIF
    NEXT
    IF Element==1
      aMatras[Renglon,I][2]=0
    ENDIF
  NEXT
  FOR J:=1 TO M->NumReng
    FOR I:=1 TO M->NumColu
      IF aMatras[J,I][2]<>0
        nElement:=nElement+1
        aCircuit[nElement,1]=J
        aCircuit[nElement,2]=I
      ENDIF
    NEXT
  NEXT
  aCelda[1,1]=RenCelda
  aCelda[1,2]=ColCelda
  aCelda[1,3]=1

```

ASIGNACION Y DISTRIBUCION DE RECURSOS

```

nCeldBus :=RenCelda
K:=1
DO WHILE RenCelda<>aCircuit[K,1].OR.ColCelda<>aCircuit[K,2]
  K++
ENDDO
aCircuit[K,1]:=0
aCircuit[K,2]:=0
FOR I:=2 TO nElement
  IF INT(I/2)==(I/2)
    K:=1
    DO WHILE nCeldBus<>aCircuit[K,1]
      K++
    ENDDO
    aCelda[1,1]:=aCircuit[K,1]
    aCelda[1,2]:=aCircuit[K,2]
    aCelda[1,3]:=-1
    aCircuit[K,1]:=0
    aCircuit[K,2]:=0
    nCeldBus:=aCelda[1,2]
  ELSE
    K:=1
    DO WHILE nCeldBus<>aCircuit[K,2]
      K++
    ENDDO
    aCelda[1,1]:=aCircuit[K,1]
    aCelda[1,2]:=aCircuit[K,2]
    aCelda[1,3]:=1
    aCircuit[K,1]:=0
    aCircuit[K,2]:=0
    nCeldBus:=aCelda[1,1]
  ENDF
NEXT
RETURN (NIL)

```

```

FUNCTION Costo
PARA RenCeld, ColCeld
LOCA LcResul
PRIV aCelda

aCelda:=ARRAY(50,3)
nElement:=0
Circuitos RenCeld,ColCeld)
LcResul:=0.00
FOR I:=1 TO nElement
  LcResul:=LcResul+aMatriz[aCelda[1,1],aCelda[1,2]][I]*;
  aCelda[1,3]
NEXT
RETURN (LcResul)

```

```

PROCEDURE Reasigna
PARA RenCeld, ColCeld

```

```

aCelda :=ARRAY(50,3)
nElement:=0
nCantida:=GRAN_M
Circuito(renCeld,ColCeld)
LcResul :=0.00
RenSale :=0
ColSale :=0
FOR I:=1 TO nElement
  IF aMatriz(aCelda[1,1],aCelda[1,2])[2] < nCantida .AND.:
    aCelda[1,3]==-1
    nCantida:=aMatriz(aCelda[1,1],aCelda[1,2])[2]
    RenSale :=aCelda[1,1]
    ColSale :=aCelda[1,2]
  ENDIF
NEXT
FOR I:=1 TO nElement
  R:=aCelda[1,1]
  C:=aCelda[1,2]
  IF nCantida==-1
    IF R==RenCeld .AND. C==ColCeld
      aMatriz[R,C][2]:=-1
    ELSEIF R==RenSale .AND. C==ColSale
      aMatriz[R,C][2]:=0
    ENDIF
  ELSE
    IF aMatriz[R,C][2]==-1
      aMatriz[R,C][2]:=aMatriz[R,C][2]+:
      nCantida*aCelda[1,3]-1
    ELSE
      aMatriz[R,C][2]:=aMatriz[R,C][2]+:
      nCantida*aCelda[1,3]
    ENDIF
  ENDIF
NEXT
NumAsig:=0
FOR I:=1 TO M->NumReng
  FOR J:=1 TO M->NumColu
    IF aMatriz[I,J][2]!=0
      NumAsig:=NumAsig+1
    ENDIF
  NEXT
NEXT
FOR K:=NumAsig+1 TO M->NumReng+M->NumColu-1
  FOR J:=1 TO M->NumReng
    FOR I:=1 TO M->NumColu
      IF aMatriz[J,I][2]==0 .AND.:
        aMatriz[J,I][1]:=GRAN_M
        Renglon:=J
        Columna:=I
      ENDIF
    NEXT
  NEXT
  aMatriz[Renglon,Columna][2]:=-1      && Delta.

```

NEXT
RETURN (NIL)

El procedimiento **Optimiza** se usa para determinar la variable que entra y la variable que sale, identifica un circuito cerrado para cada variable no básica, el circuito es generado por el procedimiento **Circuito**. El circuito inicia y termina en la variable no básica designada. Consiste en marcar segmentos horizontales y verticales sucesivos cuyos puntos extremos o vértices, deben ser variables básicas, excepto para los segmentos de inicio y término de la variable no básica, la metodología que se usa es la siguiente:

1. Para cada variable no básica (que no este en la solución), se obtiene su circuito cuyos extremos o vértices sean variables básicas, empezando y terminando en la variable no básicas que se está analizando.
2. Se definen las variables básicas que aumentan y las que disminuyen para representar las sumatorias por renglón y por columna, si la variable considerada pasa de no básica a básica.
3. Se efectúa una suma algebraica, cuyos términos serán los costos unitarios adicionando en aquellas variables que aumentan y restan en las que disminuyen. Esto se lleva a cabo en el procedimiento **Costo**.
4. La variable no básica que entra a la solución es aquella con mayor valor negativo.
5. La variable que sale, se elige de las variables en los vértices del circuito, las cuales disminuirán cuando entre la variable bajo consideración y es aquella que tenga la menor asignación.

6. El nuevo modelo se inicia con la variable que entra a la solución cuyo valor es el de la variable que sale de la misma. La reasignación se lleva a cabo con el procedimiento **Reasigna**.

Estos procedimientos son llamados por **Optimiza** el número de veces que sean necesarios, hasta que ya no existen celdas con costo negativo, lo cual indica que se ha llegado al modelo óptimo.

Una vez que se ha llegado al modelo óptimo, se procede a organizar los datos para poder ser presentados al usuario, similar al método simplex.

1.4 ASIGNACION.

Los problemas de asignación forman una subclase especial de los problemas de transporte. Para quedar clasificado como un problema de asignación, la capacidad en cada origen y la demanda en cada destino deber ser igual a uno. Como su nombre lo dice, el problema trata de decidir que origen se asigna a cada destino. Los problemas típicos de esta naturaleza incluyen el de asignar trabajadores a máquinas, equipos de trabajo a proyectos y agentes de ventas a distritos, etc.

Los problemas de asignación de pequeña escala pueden resolverse con la enumeración de todas las combinaciones y la selección de la mejor. Pero para un problema de $n \times n$, existen $n!$ soluciones posibles.

La condición necesaria y suficiente para que este tipo de problemas tenga solución, es que se encuentre balanceado o sea que los recursos totales iguale a las demandas totales.

Si tuviéramos que resolver un problema no balanceado procedemos como en transporte, adicionando renglones o columnas ficticias.

Se puede usar el método general de solución de programación lineal, o el de transporte. Sin embargo, un buen método es el Húngaro, una técnica fácil, desarrollada especialmente para este tipo de problemas.

El método Húngaro, utiliza una matriz de costos de oportunidad para encontrar la asignación óptima. El algoritmo es el siguiente:

1. Matriz de asignación. La matriz de asignación es similar a la matriz de transporte. No se necesitan condiciones de frontera puesto que siempre son igual a uno.

2. Desarrollo de la matriz de costo de oportunidad. El método de asignación busca minimizar el costo de oportunidad por no usar las celdas menos costosas. El primer paso es desarrollar los costos de oportunidad para cada celda. El costo más bajo en cada renglón se resta de cada celda de ese renglón. Después se lleva a cabo una reducción por columna. Se resta el menor costos de cada columna de todos los elementos en esa columna. Este paso siempre proporciona por lo menos una celda con costo cero en cada renglón y en cada columna. El costo de oportunidad de cero significa que el uso de esa celda para una asignación da la asignación de menor costo posible.

3. Prueba de optimalidad. Para realizar la prueba de optimalidad, se examina la matriz de costos de oportunidad para comprobar si se puede hacer una asignación óptima. Una asignación óptima emplea sólo celdas con costo cero. La regla del mínimo número de líneas consiste en que el número líneas debe ser igual que el número de renglones (o de columnas) para que se tenga una solución óptima. Cuando no puede hacerse una asignación óptima, la matriz debe revisarse.

4. Revisión de la matriz. El proceso de revisión opera sobre la matriz tachada que se usó para la prueba. Se localiza la celda con el menor costo de entre las celdas que no están cubiertas por líneas. También se suma ese costo a cada celda cubierta por una intersección. Estos ajustes llevan a una matriz revisada que tiene por lo menos una nueva celda con costo cero.

5. Asignación óptima. Al hacer la asignación óptima debe haber una celda con costo cero para cada par único de renglón y columna. Para encontrar el costo total de la solución debe regresar a la matriz original.

A continuación se presentan las líneas de código en donde se encuentran implementados todos los pasos del método Húngaro, para la solución de problemas de asignación.

Lo primero que se necesita es la generación de la tabla de asignación, es decir una matriz donde se encuentren los costos para cada asignación. Tal como en el algoritmo de transporte, es indispensable que la tabla esté balanceada, en éste caso, el balanceo consiste en que el número de orígenes sea igual al número de destinos, en caso de que no se cumpla con la condición de balanceo, se procede a adicionar orígenes o destinos ficticios.

```

PROCEDURE ANormaliza
PARA Pr_Mode

SELE 4
Le_cosmx:=0.00
FOR I:=1 TO A->NumOrig
  FOR J:=1 TO A->NumDest
    Le_llav:=Pr_Mode+StrZero(1,6,0)+StrZero(J,6,0)
    SEEK(Le_llav)
    aMatriz[LJ]:=D->ImpCost
    Le_cosmx:=IIF(D->ImpCost>Le_cosmx .AND. D->ImpCost!=-GRAN_M,D->ImpCost,Le_cosmx)
  NEXT
NEXT
IF A->NumOrig > A->NumDest      && Balanceo.
  FOR I:=1 TO A->NumOrig-A->NumDest
    M->NumColu:=M->NumColu+1
    FOR J:=1 TO M->NumReng
      aMatriz[J,1+A->NumDest]:=0.00

```

```
NEXT
NEXT
ELSEIF A->NumDest > A->NumOrig
FOR J:=1 TO A->NumDest-A->NumOrig
M->NumReng:=M->NumReng+1
FOR I:=1 TO M->NumColu
aMatriz[J+A->NumOrig,I]:=0.00
NEXT
NEXT
ENDIF
M->NumOrde:=M->NumReng
IF M->TipOpu=="1"
FOR I:=1 TO M->NumOrde
FOR J:=1 TO M->NumOrde
aMatriz[I,J]:=IF(aMatriz[I,J]==GRAN_M,GRAN_M,Lc_cosmx-aMatriz[I,J])
NEXT
ENDIF
FOR I:=1 TO M->NumOrde
Lc_Mini:=GRAN_M
FOR J:=1 TO M->NumOrde
IF Lc_Mini > aMatriz[I,J]
Lc_Mini:=aMatriz[I,J]
ENDIF
NEXT
IF Lc_Mini < GRAN_M
FOR J:=1 TO M->NumOrde
aMatriz[I,J]:=aMatriz[I,J]-Lc_Mini
NEXT
ENDIF
aOrigen[I]:=.F.
NEXT
FOR J:=1 TO M->NumOrde
Lc_Mini:=GRAN_M
FOR I:=1 TO M->NumOrde
IF Lc_Mini > aMatriz[I,J]
Lc_Mini:=aMatriz[I,J]
ENDIF
NEXT
IF Lc_Mini < GRAN_M
FOR I:=1 TO M->NumOrde
aMatriz[I,J]:=aMatriz[I,J]-Lc_Mini
NEXT
ENDIF
aDestin[J]:=.F.
NEXT
RETURN
```

El procedimiento **Anormaliza** comienza con recuperar los datos de los archivos de la base de datos del sistema y almacenarlos en una matriz, para después poder llevar a cabo el proceso de balanceo, que

consiste en agregar columnas o renglones, con costo igual a cero, según se necesiten, para poder tener así una matriz con el mismo número de renglones y de columnas.

Una vez balanceado el modelo, este procedimiento revisa si el problema es de maximización, para poder convertir la matriz del modelo a una matriz de costos, esto se hace porque el algoritmo Húngaro, tal como el modelo de transporte, considera que el problema consiste en reducir costos y de esta manera se pueden resolver problemas en que se requiera incrementar los beneficios.

Finalmente, el procedimiento resta el menor costo de cada columna de todos los elementos que la conforman y realiza la misma operación también para los renglones. Con esto se garantiza que existe por lo menos un cero en cada columna y un cero en cada renglón, de esta forma la matriz queda lista para la evaluación de los costos de oportunidad, que es el segundo paso del algoritmo de asignación.

El segundo y el tercer paso del algoritmo Húngaro de asignación se implementa en el siguiente procedimiento:

```

FUNCTION AOptimiza
PARA Pr_Mode

FOR I:=1 TO M->NumOrde
  aOrigen[I]:=-F.
  aDestin[I]:=-F.
NEXT
NumLine:=0
FOR I:=1 TO M->NumOrde
  MaxCeros:=0
  CerSimpl:=0
  Coremax:=0
  FOR J:=1 TO 2*M->NumOrde
    IF (J<=M->NumOrde .AND. !aOrigen[J]).OR.
      (J>M->NumOrde .AND. !aDestin[J-M->NumOrde])
      Ceros:=0
      FOR K:=1 TO M->NumOrde
        IF J <= M->NumOrde
          IF aMatriz[J,K]=0 .AND. !aDestin[K]
            Ceros:=-Ceros+1
          ENDF
        ELSE

```

ASIGNACION Y DISTRIBUCION DE RECURSOS

```

IF aMatriz[K,J-M->NumOrde]==0 .AND. !aOngen[K]
  Ceros:=Ceros+1
ENDIF
ENDIF
NEXT
IF J <= M->NumOrde
  CerSimp := CerSimp(J,1)
ELSE
  CerSimp := CerSimp(J-M->NumOrde,2)
ENDIF
IF MaxCeros < Ceros .OR. (MaxCeros==Ceros .AND. CerSimp1 < CerSimp)
  MaxCeros := Ceros
  Coremax := J
ENDIF
ENDIF
NEXT
IF MaxCeros > 0
  NumLine:=NumLine+1
  IF Coremax <= M->NumOrde
    aOngen[Coremax]:=.T.
  ELSE
    aDestin[Coremax-M->NumOrde]:=.T.
  ENDIF
ENDIF
NEXT
RETURN ((IF(NumLine==M->NumOrde,.T...F.))

```

El procedimiento **Aoptimiza** se encarga de determinar el menor número de líneas que se pueden trazar sobre la matriz de asignación, para tapar todos los ceros que existen. Esto se lleva a cabo mediante el uso de dos columnas adicionales de tipo lógico. El procedimiento consiste en encontrar la columna o el renglón con el mayor número de ceros y se marca así como se eliminan los ceros para que ya no participen en la determinación de columnas o renglones posteriores. Una vez que existen cubiertos todos los ceros de la matriz se calcula el número de líneas que se necesitan, este número es comparado con el orden de la matriz y si coinciden la matriz es la óptima. En caso de que el número de líneas sea menor se procede a hacer una revisión de la matriz.

```

PROCEDURE AReasigna
PARA Pr_Mode
  Lc_mni:=GRAN_M
  FOR I:=1 TO M->NumOrde

```

ASIGNACION Y DISTRIBUCION DE RECURSOS

```

FOR J:=1 TO M->NumOrde
  IF Lc_min1 > aMatriz[L,J] .AND. !aDestin[J] .AND. !aOrigen[I]
    Lc_min1:=aMatriz[L,J]
  ENDIF
NEXT J
NEXT L
IF Lc_min1 != GRAN_M
  FOR I:=1 TO M->NumOrde
    FOR J:=1 TO M->NumOrde
      IF aMatriz[L,J] != GRAN_M
        IF aDestin[J] .AND. aOrigen[I]
          aMatriz[L,J]:=aMatriz[L,J]-Lc_min1
        ELSEIF !aDestin[J] .AND. !aOrigen[I]
          aMatriz[L,J]:=aMatriz[L,J]-Lc_min1
        ENDIF
      ENDIF
    NEXT J
  NEXT I
NEXT L
ENDIF
RETURN

```

El procedimiento **Reasigna** comienza con determinar el mínimo valor que no este cubierto para después sumarlo en donde cruzan dos líneas dentro de la matriz, y restarlo de todos los elementos que se encuentran descubiertos. Los elementos cubiertos por una sola línea se dejan tal y como están, con ésto se obtiene una nueva matriz en la cual se tiene que volver a determinar el menor costo de oportunidad por no usar las celdas de menor costo, es decir tiene que volver a ser procesada por el procedimiento **Aoptimiza** así como el determinar el menor número de líneas que se necesitan para cubrir todos los ceros y si es necesario revisar una vez más la matriz con el procedimiento **Reasigna**.

El quinto paso del algoritmo Húngaro se encuentra codificado en el siguiente procedimiento, éste es el que se encargan de realizar las asignaciones adecuada una vez encontrada la matriz óptima.

PROCEDURE AResultado
 ** Asignación de resultados.

```

Lc_lla:=A->NumMode
SELE 5

```

ASIGNACION Y DISTRIBUCION DE RECURSOS

```

ZAP
aCrong:=ARRAY(g|plreng)
aCerde:=ARRAY(g|plcolu)
FOR J:=1 TO M->NumOrde
  FOR I:=1 TO M->NumOrde
    Crong:=0.00
    Cerde:=0.00

    FOR K:=1 TO M->NumOrde
      IF aMatriz[L,K]=0
        Crong:=Crong+1
      ENDIF
      IF aMatriz[K,I]=0
        Cerde:=Cerde+1
      ENDIF
    NEXT
    aCrong[I]:=Crong
    aCerde[I]:=Cerde
  NEXT
  L:=I
  X:=0
  Y:=0
  DO WHILE L <= M->NumOrde
    IF aCrong[L]=1
      M:=1
      DO WHILE aMatriz[L,M]=0
        M++
      ENDDO
      X:=L
      Y:=M
      L:=M->NumOrde+1
    ELSEIF aCerde[L]=1
      M:=1
      DO WHILE aMatriz[M,L]=0
        M++
      ENDDO
      X:=M
      Y:=L
      L:=M->NumOrde+1
    ENDIF
    L:=L+1
  ENDDO
  IF X<>0 .AND. Y<>0
    SELECT 2
    SEEK (Lc_lla+StrZero(X,6,0))
    SELECT 3
    SEEK (Lc_lla+StrZero(Y,6,0))
    SELECT 4
    SEEK (Lc_lla+StrZero(X,6,0)+StrZero(Y,6,0))

    IF !B->(EOF()) .AND. !C->(EOF()) .AND. !D->(EOF())
      SELECT 5
      APPEND BLANK
      REPL E->DesOrig WITH B->DesOrig,;
        E->DesDest WITH C->DesDest;
    
```

ASIGNACION Y DISTRIBUCION DE RECURSOS

```
E->CanEnvi WITH I.;
E->CosEnvi WITH E->CanEnvi*D->ImpCost.;
E->TipRegi WITH 'I'
ENDIF

FOR H:=-1 TO M->NumOrde
  aMatriz[X,H]:=-GRAN_M
  aMatriz[H,Y]:=-GRAN_M
NEXT
ENDIF
NEXT
SELECT I
RETURN
```

El proceso de presentación de la solución óptima, para problemas de asignación puede ser, como en los algoritmos simplex y de transporte, por impresora o en pantalla y se efectúa de manera similar que en los modelos anteriores.

Con estos procedimientos descritos, queda implementado todo el algoritmo Húngaro para la solución de problemas de asignación.

2. PLANEACION Y CONTROL DE PROYECTOS.

2.1 CONCEPTOS GENERALES.

Un proyecto es cualquier empresa humana con un claro principio y un claro final. Existen proyectos muy pequeños como el diseño de un nuevo producto, el lanzamiento de un programa de mercadotecnia o el desarrollo de un producto a la medida; unos más grandes como la construcción de una casa, la construcción de un edificio, o la pavimentación de una carretera. Y algunos que son muy grandes como el lanzamiento de naves espaciales, la instalación de pozos extractores de petróleo, la construcción de túneles subterráneos, etc. Sin embargo, todos los proyectos, pequeños y grandes, tienen características comunes, siempre hay:

- a) Una combinación de actividades.
- b) Una relación secuencial entre algunas actividades.
- c) Una preocupación por el tiempo: la terminación del proyecto a tiempo es importante.
- d) Una preocupación por los recursos: completar el proyecto dentro del presupuesto también es importante.

Cuando se trata de un proyecto pequeño, todas las fechas importantes y la información pueden ser guardados en la memoria. Sin embargo, cuando el proyecto es grande, los coordinadores necesitan poner en un papel los detalles del proyecto en caso de que quieran asegurar la planeación y el control apropiado.

La planeación de proyectos requiere desglosar el proyecto en actividades, estimar los recursos y el tiempo para cada actividad y describir las interrelaciones de las actividades. La programación requiere detallar las fechas de inicio y terminación para cada actividad. El control del proyecto no sólo requiere información sobre el estado actual sino analiza los posibles truces cuando surgen dificultades.

La planeación comienza con la definición de objetivos. El equipo del proyecto es integrado por varios departamentos de la organización y puede incluir personal de áreas como ingeniería, producción, mercadotecnia y contabilidad, según sea el proyecto. La definición del proyecto implica definir las variables controlables y no controlables, estableciendo las limitantes del proyecto.

El control del proyecto es el proceso que sirve para medir el estado de las actividades, transmitir estos datos a un centro de control donde esto es comparado con el plan e iniciar acciones correctivas cuando sea necesario.

Existen dos modelos, muy similares, los cuales sirven para planear y controlar todas las actividades que se requieren en el desarrollo de cualquier proyecto. Uno es utilizado cuando los tiempos de las actividades se conocen con precisión (CPM - Siglas en inglés de método de la ruta crítica) y el otro cuando existe cierta incertidumbre en la terminación de las actividades (PERT - Siglas en inglés de la técnica de evaluación y revisión de programas).

Se han usado CPM y PERT exitosamente en muchas aplicaciones, incluyendo:

1. La programación de proyectos de construcción, tales como edificios, oficinas, autopistas, etc.
2. La creación de un procedimiento para la cuenta regresiva y para la suspensión del lanzamiento de vuelos espaciales.
3. En el diseño y el distribución de un nuevo producto.
4. La conciliación de una unión corporativa.
5. La construcción de un barco.

Así mismo, las técnicas de planeación y control de proyectos, en fabricación, son especialmente útiles en la administración de la producción de artículos diseñados según modelos básicos para satisfacer las necesidades especiales de clientes. El ciclo de producción de esos artículos incluye la ingeniería de diseño y producción, así como los procesos de fabricación. Esta situación es común en la producción de artículos de equipos grandes como sistemas de manejo de materiales, carrocerías de camión, equipos de limpieza industrial, desarrollo de software, etc.

Además, éstas técnicas son también de ayuda en las siguientes actividades de la administración de la producción e inventarios:

1. Planeación a corto plazo - cambios de personal, instalación y desmantelamiento de equipo.
2. Aumento de la capacidad - reducción del tiempo planeado de mantenimiento.
3. Cálculo del tiempo de obtención de la producción - basado en la trayectoria de ruta crítica de compras, fabricación y ensamble.
4. Reducción del tiempo de proceso en la producción - al disminuir la trayectoria de ruta crítica de fabricación y ensamble.

Debido a que tanto el método de la ruta crítica (CPM) como la técnica de evaluación y revisión de programas (PERT) tienen la misma lógica, se presentará una primera sección, "Análisis de redes", en donde se describen aquellos procedimientos y funciones comunes para ambos modelos.

2.2 ANALISIS DE REDES.

Los modelos de redes (CPM y PERT) son una técnica eficaz en la planeación y control de todo tipo de proyectos. En esencia es la representación del plan de un proyecto en una red, que describe la secuencia e interrelación de todas las componentes del proyecto, así como el análisis lógico y la manipulación de esta red, para la completa determinación del mejor programa de operación; permite la evaluación y comparación rápida de distintos programas de trabajo, tipos de recursos a emplear, así como su magnitud. Una vez que el mejor plan ha sido elaborado en esta forma, la red de la ruta crítica indica claramente las operaciones que controlan y la ejecución fluida de los trabajos. Finalmente durante la ejecución del proyecto, la red provee al responsable una información precisa de los efectos de cada variación o retraso en el plan adoptado, permitiendo así identificar las operaciones que requieren cambios.

La metodología de desarrollo de los modelos de redes es la siguiente:

1. Desglosar el proyecto en las operaciones o procesos que son necesario para su terminación. El grado de descomposición de cada concepto, depende de cada proyecto y está sujeto a la naturaleza del trabajo y tipo de mano de obra involucrados, la localización del trabajo, la información de costos, etc. Cada una de estas operaciones o procesos se llama ACTIVIDAD y la terminación de una actividad, se llama EVENTO; las actividades consumen tiempo, mientras que los eventos no.
2. Después que se ha preparado una lista de todas las actividades que constituyen el proyecto, se procede a determinar las relaciones esenciales entre ellas. Aunque muchas de las actividades se pueden realizar simultáneamente, algunas deben ordenarse de acuerdo con una secuencia necesaria llamada CADENA. Se debe sujetar a cada una de las actividades del proyecto a las siguientes preguntas:

- ¿Cuáles son las actividades precedentes a ésta?
- ¿Qué actividades deben proseguir a ésta?
- ¿Qué actividades pueden realizarse simultáneamente con ésta?

En esta forma se examina cada actividad determinándose la secuencia necesaria de actividades. Cada actividad tienen definido un evento que señala su inicio y un evento que marca su término, lo mismo que cada proyecto tiene un sólo nodo de inicio y un sólo nodo de término.

3. Considerar las restricciones del proyecto tales como: seguridad, recursos, administrativas, etc.
4. Trazar la red en términos de la lógica del proyecto y enumerar los eventos.
5. Elaborar la tabla de los datos costo - tiempo para cada actividad. Para cada actividad existe un costo asociado que, generalmente depende de su tiempo específico de terminación. Si el tiempo varía, se esperará que también varíe el costo.
6. Asignar un tiempo a cada actividad en la red, usando la duración normal en el caso de CPM y el tiempo medio esperado para PERT. A continuación, siguiendo los eventos en orden numérico desde el principio, una simple adición nos dará el tiempo más próximo posible al que todas las actividades que llegan a cada evento, pueden iniciarse; esto es, el tiempo de INICIACION MAS PROXIMO (IMP) para el evento. Se continúa así, a través de la red, hasta encontrar el TMP del último evento; ésta es la TERMINACION MAS PROXIMA del proyecto y es la suma de las duraciones de

las actividades a través de la ruta que conduce a la duración más larga del proyecto, de principio a fin.

7. El siguiente paso es ir hacia atrás desde el último evento, restando la duración de cada actividad, para encontrar EL TIEMPO DE TERMINACION MAS TARDIO (TMT) permisible para cada evento, considerando que el proyecto debe ser terminado al TMP del evento final. El TMT esta controlado por todas las actividades que salen del evento en cuestión.

8. El cálculo del TIEMPO FLOTANTE u HOLGURA se lleva a cabo mediante la diferencia entre los dos números, que se calcularon anteriormente (TMP y TMT), de cada evento; éste es el margen de retraso de una actividad. En los eventos que el $TMP = TMT$, no hay tiempo flotante y son los EVENTOS CRITICOS, los cuales deben ser terminados dentro del programa si se quiere terminar el proyecto en el mínimo tiempo total. La ruta que une a estos eventos criticos es la RUTA CRITICA de la red.

9. Para cada actividad, se calcula lo siguiente:

FT.- Tiempo flotante total, es la suma total del tiempo en el que una actividad se puede retrasar sin aumentar la duración del proyecto.

$$FT = TMP - TMT$$

FL.- Tiempo flotante libre, es la suma del tiempo en el que el inicio de una actividad puede ser retrasado sin interferir con el inicio de la actividad que le sigue.

$$(FT \geq FL)$$

$FL = IMP$ (de la actividad siguiente) - TMP (de la actividad en cuestión)

IMP.- Tiempo de iniciación más próximo, es el momento al que una actividad del proyecto puede empezarse.

$TMP = IMP - Duración$

IMT.- Tiempo de iniciación más tardío, es aquel al que se puede comenzar, si se desea conservar la duración mínima del proyecto.

$IMT = TMT - Duración$

10. Con lo anterior se obtiene el modelo normal, después se obtienen el modelo óptimo y el de ruptura de la red, considerando el costo total del proyecto. El modelo óptimo es el modelo de menores costos totales de todos los modelos. El modelo de ruptura es el modelo de menor tiempo que todos los considerados. El costo total de un proyecto consiste en la suma de los costos variables de las actividades y los costos fijos.

El análisis del intercambio de costo de tiempo se inicia con una determinación de la pendiente estimada en el costo de tiempo de cada actividad; es una medida del costo de reducir la duración de una actividad.

11. Una vez obtenidos los modelos del proyecto (normal, óptimo y de ruptura) se puede realizar un análisis de nivelación de recursos, aunque esto ya no es propiamente parte del método de los modelos de redes, proporciona información útil, con la que se pueden administrar mejor los recursos que se necesitan en cada etapa del proyecto, así como también realizar análisis de flujo de efectivo ("Cash Flow").

El código de los programas en donde se implementan los modelos de redes, y que son comunes a los dos modelos de redes, son descritos a continuación.

Los puntos 1, 2 y 3 de los algoritmos de redes, deben ser realizados por el responsable del proyecto, esta información es obtenida a través de un análisis previo, el nivel de detalle depende del nivel de planeación y control que se quiera llevar y de la misma naturaleza del proyecto. Estos datos son muy importante, ya que son con los que se tendrá que alimentar a la computadora.

El punto 4 de la metodología es utilizado para la representación gráfica del proyecto, así como las relaciones entre las actividades que lo conforma, es de gran utilidad cuando el análisis se realiza de forma manual, para el análisis en computadora sólo es necesario para indicar las precedencias de las actividades en la captura inicial.

El punto 5 es realizado de manera separada por los programas de cada uno de los modelos ya que la información que se captura de diferente forma entre uno y el otro.

Con el punto 6 inicia los programas comunes para los modelos de redes, este punto se lleva a cabo por medio del procedimiento que se encuentra codificado a continuación:

```

PROCEDURE RECADEL
PARA Pr_Mode, Pr_Acti, Pr_Inic
LOCA Lc_Regi

SELE 3
SEEK(Pr_Mode+Pr_Acti)
IF .NOT. EOF()
DO WHILE .NOT. EOF() .AND. Pr_Mode+Pr_Acti=C->NumMode+StrZero(C->ActPrec,6,0)
  Lc_regi=RECN()
  SELE 2
  SEEK(Pr_Mode+C->NumActi)
  REPL B->ImpNorm WITH MAX(Pr_inic,B->ImpNorm)
  TmtMaxi:=MAX(TmtMaxi,B->ImpNorm+B->DurNorm)
  RECADEL(Pr_Mode,C->NumActi,Pr_Inic+B->DurNorm)
  SELE 3
  GOTO Lc_regi
  SKIP
ENDDO
ELSE
SELE 2
SEEK(Pr_Mode+Pr_Acti)
TmtMaxi:=MAX(TmtMaxi,B->ImpNorm+B->DurNorm)
SELE 3
ENDIF
RETURN

```

El procedimiento **RECADEL** se encarga de recorrer la red hacia adelante, calculando así, para cada una de las actividades que conforman la red, el tiempo más próximo de iniciación (IMP) del modelo normal para CPM o del medio esperado para PERT, el cual es el tiempo que más próximo se debe iniciar cada una de las actividades, sin retrasar el proyecto.

Debido a que la estructura de la red de un proyecto es en forma de árbol, con características diferentes para cada proyecto, con tamaño y número de ramificaciones diferente, es necesario que el procedimiento **RECADEL** sea de naturaleza recursiva.

Un procedimiento recursivo es aquel que ésta formado por sí mismo o se define en función de sí mismo. El poder de la recursión radica en la posibilidad de definir un conjunto infinito de objetos mediante una proposición finita. La herramienta necesaria y suficiente para poder expresar un programa recursivo es el procedimiento o subrutina, pues permite dar a una proposición un nombre con el cual puede ser usada.

Se acostumbra asociar un conjunto de objetos locales a un procedimiento, esto es un conjunto de variables, constantes y tipos que se definen localmente a este procedimiento y que carecen de existencia o significado fuera de este procedimiento. Cada vez que este procedimiento se activa de modo recursivo, se crea un nuevo conjunto de variables locales acotadas. Aunque tienen el mismo nombre que sus elementos correspondientes en el conjunto local al caso anterior del procedimiento, sus valores son específicos y se evita cualquier conflicto en la imposición del nombre por medio de las reglas de cobertura de los identificadores; éstos se refieren siempre al conjunto de variables de creación más reciente.

Como se puede apreciar, dentro del procedimiento **RECADEL**, existe una línea donde se invoca así mismo, sólo con parámetros diferentes, ya que entre llamadas, el registro del archivo de la base de datos de donde son tomados los parámetros es diferente.

El procedimiento **RECADEL** también calcula el tiempo más tardío de terminación de la última de las actividades.

Una vez que finaliza todo el recorrido del árbol y el procedimiento **RECADEL** es finalizado en todas las llamadas que se le realizaron los tiempos IMP de todas las actividades que forman parte del proyecto han sido calculados y es posible pasar al punto 7, para determinar el tiempo más tardío de finalización TMP, el cual es codificado en el siguiente procedimiento:

PROCEDURE RECATRA

PARA Pr_Mode, Pr_Acti, Pr_Inic, Pr_Imps, Pr_Tmts

LOCA Lc_Regi

SELE 3

SEEK(Pr_Mode+Pr_Acti)

DO WHILE .NOT. EOF() .AND. Pr_Mode+Pr_Acti=C->NumMode+C->NumActi

Lc_Regi:=REC�()

SELE 2

SEEK(Pr_Mode+StrZero(C->ActPrec,6,0))

SELE 3

REPL B->TmtNorm WITH MIN(Pr_Inic.B->TmtNorm);

B->TmpNorm WITH B->ImpNorm*B->DurNorm;

B->ImtNorm WITH B->TmtNorm-B->DurNorm;

B->FttNorm WITH B->TmtNorm-B->TmpNorm;

B->FltNorm WITH Pr_Imps.B->TmpNorm;

B->ImpOpti WITH B->ImpNorm;

B->ImpRupt WITH B->ImpNorm;

B->ImtOpti WITH B->ImtNorm;

B->ImtRupt WITH B->ImtNorm;

B->TmtOpti WITH B->TmtNorm;

B->TmtRupt WITH B->TmtNorm;

B->TmpOpti WITH B->TmpNorm;

B->TmpRupt WITH B->TmpNorm;

B->FttOpti WITH B->FttNorm;

B->FttRupt WITH B->FttNorm;

B->FltOpti WITH B->FltNorm;

B->FltRupt WITH B->FltNorm;

B->ActCrit WITH IIF(B->FttNorm=0,T...F.)

RECATRA(Pr_Mode, StrZero(C->ActPrec,6,0), Pr_Inic.B->DurNorm, B->ImpNorm, B->TmtNorm)

SELE 3

GOTO Lc_Regi

SKIP

ENDDO

RETURN

El procedimiento **RECATRA** se ejecuta una vez que se ha terminado de ejecutar el procedimiento **RECADEL**, en él se lleva a cabo un recorrido de la red hacia atrás, calculando de ésta forma el tiempo más tardío de terminación (TMT) para cada actividad. Con ésto queda cubierto el punto número 7 de la metodología.

Ya que en este procedimiento **RECATRA** , se cuenta con el IMP y el TMT de cada actividad, se puede realizar el cálculo del tiempo más tardío de iniciación (IMT) y el tiempo de terminación más próximo (TMP), que es lo que se tiene que realizar para cubrir el punto número 8.

El punto 9 de la metodología, también es llevado a cabo con el procedimiento **RECATRA**, es decir en el se calcula el tiempo flotante total (FT), que es el total del tiempo en el que una actividad puede ser retrasada sin aumentar la duración del proyecto, y el tiempo flotante libre (FL) que es la suma del tiempo en el que el inicio de una actividad puede ser retrasado sin interferir con el inicio de la actividad que le sigue.

El procedimiento **RECATRA** también , es de naturaleza recursiva, como puede observarse, dentro de él, se hace referencia a sí mismo, recorriendo desde el final hasta el principio toda la red que conforma al proyecto. Así mismo conforme se va recorriendo a la red, el procedimiento va marcando a aquellas actividades que conforman a la ruta crítica de éste primer modelo.

Una vez que se tiene completo el modelo normal, se procede a realizar el intercambio entre costo - tiempo, realizando una compresión de la red. Esto es lo que está comprendido en el punto 10 de la metodología. En el procedimiento **CPMÓPTI**, que se presenta a continuación, se lleva a cabo el proceso de compresión de la red de un proyecto, para poder encontrar el proyecto óptimo y el proyecto de ruptura.

PROCEDURE CPMOPTIM
PARA Pr_Mode

```
SeComprime:=T.  
DO WHILE SeComprime  
  SELE 2  
  SEEK(Pr_Mode)  
  Lc_Acts:=""  
  NumActs:=0  
  LimTota:=GRAN_M  
  
DO WHILE NOT. EOF() .AND. Pr_Mode=B->NumMode  
  Lc_Regi:=RECNO()  
  IF B->NumPrec=0
```

PLANEACION Y CONTROL DE PROYECTOS

```

DEFCOMPR(B->NumMode,B->NumActi)
ENDIF
SELE 2
GOTO Lc_Regi
LimTota:=IIF(B->FltRupt<LimTota .AND. B->FltAnte>B->FltRupt .AND. B->FltRupt/=0,B->FltRupt,LimTota)
SKIP
ENDDO

SELE 2
LimActi:=GRAN_M
FOR K:=1 TO NumActs
  Ps_Acti:=SUBS(Lc_Acts,K*7-6,6)
  SEEK(Pr_Mode+Ps_Acti)
  IF LimActi > (B->DurIter-B->DurFina) .AND. (B->DurIter-B->DurFina) > 0
    LimActi := (B->DurIter-B->DurFina)
  ENDF
NEXT
LimTota:=IIF(LimTota=0,GRAN_M,LimTota)
LimComp:=MIN(LimActi,LimTota)
IF LimComp > 0 .AND. LimActi != GRAN_M .AND. SeComprme
  SELE 2
  FOR K:=1 TO NumActs
    ActComp:=SUBS(Lc_Acts,K*7-6,6)
    SEEK(Pr_Mode+ActComp)
    REPL B->DurIter WITH B->DurIter - LimComp;
    B->CostIter WITH B->CostIter + B->ValPend * LimComp
  NEXT
  SELE 3
  SET ORDER TO 2
  SELE 2
  SEEK(Pr_Mode)
  DO WHILE .NOT. EOF() .AND. B->NumMode=Pr_Mode
    REPL B->ImpRupt WITH 0;
    B->TmtRupt WITH GRAN_M
  SKIP
  ENDDO
  SEEK(Pr_Mode)
  TmtMax:=0
  DO WHILE .NOT. EOF() .AND. B->NumMode=Pr_Mode
    Lc_Regi:=REC()
    IF B->NumPrec=0
      RECADELO(B->NumMode,B->NumActi,B->DurIter)
    ENDF
    SELE 2
    GOTO Lc_Regi
    SKIP
  ENDDO

  SELE 2
  SEEK(Pr_Mode)
  CostIter:=0
  DO WHILE .NOT. EOF() .AND. B->NumMode=Pr_Mode
    SELE 3
    SEEK(Pr_Mode+B->NumActi)
    SELE 2

```

PLANEACION Y CONTROL DE PROYECTOS

```

Lc_Regi:=RECNI()
IF C->(EOF())
  SELE 1
  REPL A->CosRupt WITH A->CosIter.;
  A->DurRupt WITH A->DurIter
  SELE 2
  REPL B->FltAnte WITH B->FltRupt.;
  B->TmtRupt WITH TmtMaxi.;
  B->TmtRupt WITH B->ImpRupt-B->DurIter.;
  B->ImtRupt WITH B->TmtRupt-B->DurIter.;
  B->FrtRupt WITH B->TmtRupt-B->TmtRupt.;
  B->FhtRupt WITH B->TmtRupt-B->TmtRupt.;
  B->ActCrit WITH IIF(B->FrtRupt=0..T..F.)
  SELE 3
  SET ORDER TO 1
  RECATRAO(Pr_Mode, B->NumActi, B->TmtRupt-B->DurIter, B->ImpRupt, B->TmtRupt)
  SET ORDER TO 2
ENDIF
SELE 2
GOTO Lc_Regi
REPL B->DurRupt WITH B->DurIter
CosIter:=CosIter+B->CosIter
SKIP
ENDDO
SELE 1
REPL A->CosIter WITH CosIter+A->CosFijo*TmtMaxi.;
A->DurIter WITH TmtMaxi;
IF A->CosOpti > A->CosIter
  REPL A->CosOpti WITH A->CosIter.;
  A->DurOpti WITH A->DurIter
  SELE 2
  SEEK (Pr_Mode)
  DO WHILE EOF() .AND. Pr_Mode=B->NumMode
    REPL B->DurOpti WITH B->DurIter.;
    B->CosOpti WITH B->CosIter.;
    B->ImpOpti WITH B->ImpRupt.;
    B->ImtOpti WITH B->ImtRupt.;
    B->TmtOpti WITH B->TmtRupt.;
    B->TmpOpti WITH B->TmtRupt.;
    B->FrtOpti WITH B->FrtRupt.;
    B->FhtOpti WITH B->FhtRupt
  SKIP
  ENDDO
ENDIF
SELE 3
SET ORDER TO 1
ELSE
  SeComprme:=-F.
ENDIF
ENDDO
RETURN

```

El procedimiento **CPMOPTI** consiste en un algoritmo iterativo, el cual se ejecuta el número de veces que sea necesario, hasta que se llega al proyecto de ruptura.

Lo primero que se tiene que realizar para comprimir la red, es determinar la actividad que se le va ha reducir su tiempo, ésta se determina como aquella a la que cueste menos realizar su compresión, es decir aquella actividad con menor pendiente de costo y que forme parte de la ruta crítica y que si se comprime un actividad crítica se reducirá el tiempo de terminación de todo el proyecto.

Para el caso en donde exista más de una ruta crítica, la determinación del tiempo de compresión se lleva ha cabo considerando, a todas las actividades críticas, y reduciendo a todas las rutas críticas en la misma proporción. Con ésto se garantiza que una ruta crítica encontrada permanecerá siendo crítica durante todo el proceso de compresión. Durante el proceso de compresión si es posible encontrar más rutas críticas de las que existían en la iteración anterior.

Una vez que se tiene la actividad bien definida, se procede a determinar el tiempo que es posible comprimirla, es decir el límite de compresión, el cual es determinado como el menor tiempo entre el tiempo flotante libre (FL) que tienda a reducirse y que sea mayor que cero y el tiempo flotante total (FT) de esa actividad, garantizando con ésto que no se rebasará el tiempo de ruptura de la actividad ni de otras actividades que se ejecuten simultáneamente.

Una vez que se ha reducido el tiempo de una actividad se incrementa el costo correspondiente a reducir el tiempo de ejecución de esa actividad, incrementado de esa manera el costo total del proyecto.

El procedimiento **CPMOPTI**, una vez que comprimió la red, procede a realizar un recorrido hacia adelante, con el procedimiento **RECADELO** y un recorrido hacia atrás, con el procedimiento **RECATRAO**, para volver a calcular los nuevos tiempos asociados a las actividades.

El proceso se repite hasta que todas las actividades de una ruta crítica, ya no se puedan seguir comprimiendo, es decir que se ha llegado al modelo de ruptura.

A continuación se presentan los procedimientos **RECADELO** y **RECATRAO**, que como puede apreciarse, tienen la misma lógica de los procedimientos de recorrido normal, presentados anteriormente, sólo que utilizan diferentes campos para almacenar los tiempos que van calculando.

PROCEDURE RECADELO

PARA Pr_Mode, Pr_Acti, Pr_Inic
LOCA Lc_Regi

```

SELE 3
SEEK(Pr_Mode+Pr_Acti)
DO WHILE .NOT. EOF() .AND. Pr_Mode+Pr_Acti=C->NumMode+StrZero(C->ActPrec,6.0)
  Lc_Regi=RECNC()
  SELE 2
  SEEK(Pr_Mode+C->NumActi)
  REPL B->ImpRupt WITH MAX(Pr_inic,B->ImpRupt)
  TmtMaxi:=MAX(TmtMaxi,B->ImpRupt+B->DurIter)
  SELE 3
  RECADELO(Pr_Mode,C->NumActi,Pr_Inic-B->DurIter)
  GOTO Lc_Regi
SKIP
ENDDO
RETURN

```

PROCEDURE RECATRAO

PARA Pr_Mode, Pr_Acti, Pr_Inic, Pr_Imps, Pr_Tmts
LOCA Lc_Regi

```

SELE 3
SEEK(Pr_Mode+Pr_Acti)
DO WHILE .NOT. EOF() .AND. Pr_Mode+Pr_Acti=C->NumMode+C->NumActi
  Lc_Regi=RECNC()
  SELE 2
  SEEK(Pr_Mode+StrZero(C->ActPrec,6.0))
  SELE 3
  REPL B->TmtRupt WITH MIN(Pr_Inic,B->TmtRupt);
  B->TmptRupt WITH B->ImpRupt+B->DurIter.;
  B->ImtRupt WITH B->TmtRupt-B->DurIter.;
  B->FrtRupt WITH B->TmtRupt-B->TmptRupt.;
  B->FrtRupt WITH Pr_Imps-B->TmptRupt.;
  B->ActCrit WITH IIF(B->FrtRupt=0,.T.,.F.)
  RECATRAO(Pr_Mode, StrZero(C->ActPrec,6.0), Pr_Inic-B->DurIter, B->ImpRupt, B->TmtRupt)

```

```

SELE 3
GOTO Lc_Regi
SKIP
ENDDO
RETURN

```

El procedimiento **CPMOPTI** realiza una llamada al procedimiento **DEFCOMPR** el cual es el encargado de determinar la actividad con menor pendiente de costo:

```

PROCEDURE DEFCONPR
PARA Pr_Mode, Pr_Acti

```

```

SELE 3
SET ORDER TO 2
Lc_Acum:=IIF(B->DurIter-B->DurFina>0,Pr_Acti,")
Lc_Pend:=IIF(B->DurIter-B->DurFina>0,B->ValPend,GRAN_M)
ASIGCOMP(Pr_Mode, Pr_Acti, Pr_Acti, .T., Lc_Acum, Lc_Pend)
SELE 3
SET ORDER TO 1
SELE 2
RETURN

```

```

PROCEDURE ASIGCOMP
PARA Pr_Mode, Pr_Acti, Pr_Cade, Pr_Crit, Pr_Acmi, Pr_Pend
LOCA Lc_Regi, Lc_Cade

```

```

Lc_Cade:=Pr_Cade
SELE 3
SEEK(Pr_Mode+Pr_Acti)
IF .NOT. EOF()
DO WHILE .NOT. EOF() .AND. Pr_Mode+Pr_Acti=C->NumMode+StrZero(C->ActPrec,6.0)
Lc_regi=RECNC()
SELE 2
SEEK(Pr_Mode+C->NumActi)
IF Pr_Crit .AND. B->ActCrit .AND. B->ValPend < Pr_Pend .AND. (B->DurIter-B->DurFina)>0
Pr_Acmi:=B->NumActi
Pr_Pend:=B->ValPend
ENDIF
SELE 3
ASIGCOMP(Pr_Mode, C->NumActi, Lc_Cade+','->C->NumActi, Pr_Crit .AND. B->ActCrit, Pr_Acmi, Pr_Pend)
GOTO Lc_regi
SKIP
ENDDO

```

```

ELSE
SELE 2
SEEK(Pr_Mode>Pr_Act)
IF Pr_Crit .AND. B->ActCrit
  IF B->ValPend < Pr_Pend .AND. (B->DurIter-B->DurFina)>0
    Pr_Acml:=-B->NumActi
    Pr_Pend:=-B->ValPend
  ENDIF
ELSE
  Pr_Crit:=F.
ENDIF
IF Pr_Crit .AND. !EMPTY(Pr_Acml) .AND. AT(Pr_Acml,Lc_Acts)=0
  Lc_Acts:=Lc_Acts+Pr_Acml+'
  NumActs:=NumActs+1
ELSEIF Pr_Crit .AND. EMPTY(Pr_Acml)
  SeComprime:=F.
  REPL A->CosRupt WITH A->CosIter.;
  A->DurRupt WITH A->DurIter
ENDIF
ENDIF
SELE 3
RETURN

```

Con los algoritmos anteriores queda terminado el punto 10 de la metodología, teniendo así los resultados de los tres modelos de la red del proyecto. El punto 11 determina que se debe realizar un balanceo de los recursos, esto se lleva a cabo por medio de un algoritmo de acumulación de recursos y con los tiempos calculado anteriormente, es posible determina cual es el total de los recursos que se están usando en cada momento. El procedimiento que lo lleva a cabo es el siguiente:

```

PROCEDURE CPMBALANC
PARA Pr_Mode
  SELE 4
  ZAP
  SELE 2
  SEEK(Pr_Mode)
  IF A->CosNorm==A->CosOpti .AND. A->CosOpti==A->CosRupt .AND.;
    A->DurNorm==A->DurOpti .AND. A->DurOpti==A->DurRuot
    NI:=1
  ELSE
    NI:=3
  ENDIF
  DO WHILE .NOT. EOF() .AND. B->NumMode=Pr_Mode
    FOR J=1 TO NI

```

PLANARIZATION & CONTROL OF REACTION

```

TIPK =SLBS(NormOpaRup,J=4,3,4)
TIPK =STR(1,1,0)
FOR K =B->(MP&TIPO-1 TO B->TMP&TIP)
  Llave =TIPK+StrZero(K,6,0)
  SELE 4
  FIND &Llave
  IF EOF()
    APPE BLANK
    REPL D->NumPens WITH K.;
    D->CanRecu WITH B->RecReqs.;
    D->TipMode WITH TIPK
  ELSE
    REPL D->CanRecu WITH D->CanRecu + B->RecReqs
  ENDDIF
  NEXT
  SELE 2
  SKIP
  ENDDO
RETURN
  
```

El procedimiento CPMBALAN^o genera los datos necesarios para el control de la reacción de planarización.

Los datos de salida se generan en un archivo de texto que puede ser utilizado para el control de la reacción.

El archivo de salida se genera en el directorio de trabajo del usuario.

Los datos de salida se generan en un archivo de texto que puede ser utilizado para el control de la reacción.

Los datos de salida se generan en un archivo de texto que puede ser utilizado para el control de la reacción.

Los datos de salida se generan en un archivo de texto que puede ser utilizado para el control de la reacción.

2.3 METODO DE LA RUTA CRITICA.

El método de la ruta crítica (CPM) se desarrolló para resolver el problema de los intercambios entre el tiempo y los costos. El método supone que las duraciones y los costos de las actividades se pueden predecir bastante bien para poder usar estimaciones deterministas. Sin embargo, el CPM requiere dos estimaciones de tiempo y costo para cada una de las actividades. Se toma en cuenta la posibilidad de que el esfuerzo extra (costo) puede reducir el tiempo de terminación de una actividad.

Cuando existe un proyecto donde se conocen las duraciones de las actividades y sus costos con bastante precisión así como las relaciones y secuencias que existen entre ellas, es posible determinar la información que se requiere para planear y controlar todo el proyecto, tal información consiste en:

- a) Cuando se inicia y cuando se termina cada actividad que forma parte del proyecto, si se esta realizando de manera normal, de manera mas rápida, o con los costos más bajos.
- b) Cuanto dura el proyecto en su totalidad, ya sea si es llevado a cabo de forma normal, con costos más bajos o de la forma más rápida.
- c) El tiempo que se puede retrasar una actividad específica sin que se retrase el proyecto.
- d) Que actividades son las que determinan la terminación del proyecto y que se debe prestar mayor atención, para que sean realizadas dentro del programa..

- e) Que cantidad de recursos se están ocupando en un momento determinado a lo largo del proyecto.
- f) Cual es la diferencia en costos de llevar a cabo uno de los tres modelos del proyecto: normal, ruptura u óptimo.
- g) Que actividades se están realizando y cual ya se debieron haber realizado.
- h) El establecimiento del programa de actividades y su seguimiento.

Toda esta información puede ser obtenida a través de CPM el cuál es implementado, por medio de los algoritmos de análisis de redes, sin embargo para poder ser ejecutados los algoritmos de redes, es necesario primero que se lleven a cabo los tres primeros puntos de la metodología, como se menciona anteriormente, ésto lo tiene que hacer el responsable del proyecto y una vez que tiene los datos de las actividades, es necesario llevar a cabo la captura en la computadora, para poder ser ejecutado todo el proceso de análisis, ésto se realiza a través de un programa que a continuación se muestra y que permite al usuario capturar la descripción del modelo y las características de las actividades y una vez que se termina la captura es ejecutado el procedimiento que controla los programas de análisis de redes:

```
Lc_pant:=SAVESCREEN(00,00,24,79)
lc_name:=1
lc_tpo:=1
Lc_Otro:='S'
DO WHILE Lc_Otro='S'
  CLS
  @ 00,00 TO 24,79 DOUBLE
  SETCOLOR(glc=blm)
  @ 01,26 SAY 'METODO DE RUTA CRITICA (CPM)'
  SETCOLOR(glc=bor)
  @ 02,01 SAY REFL('-',78)
```

PLANEACION Y CONTROL DE PROYECTOS

```

SELECT 1
MODELOS:=ARRAY(RECC()+1)
MODELOS[1]:='NUEVO'
ULTMODE:=""
CONT:=2
GO TOP
DO WHILE .NOT. EOF()
  MODELOS[CONT]:=A->DesGene
  ULTMODE:=A->NumMode
  CONT:=CONT+1
  SKIP
ENDDO
RENPAN :=IIF(CONT>15,15,CONT)
RENINIC:=INT((24-RENPAN)/2)+1
RENFINA:=RENINIC+RENPAN
@ RENINIC,19 TO RENFINA,59
SETCOLOR(GIcolbr)
@ RENINIC,28 SAY ' Selección del Modelo '
SETCOLOR(GIcolnr)
SET KEY 19 TO NADA()
SET KEY 04 TO NADA()
MODEL:=ACHOICE(RENINIC+1,20,RENFINA-1,58,MODELOS)
SET KEY 04 TO
SET KEY 19 TO
GO TOP
@ RENINIC,19 CLEA TO RENFINA,59
IF MODEL==1
  Lc_num:=VAL(ULTMODE)+1
ELSEIF MODEL==0
  CLOSE DATA
  RESTSCREEN(00,00,24,79,lc_pant)
  RETURN
ELSE
  SKIP MODEL-2
  Lc_num:=VAL(A->NumMode)
ENDIF
Lc_llav:=StrZero(Lc_num,6,0)
SELE 1
SEEK (Lc_llav)
IF EOF()
  SELE 1
  APPEND BLANK
  REPL A->NumMode WITH Lc_llav
ENDIF
@ 05,02 TO 21,77
SETCOLOR(GIcolbr)
@ 05,16 SAY ' Descripción del Modelo [Ctrl-W para Salir] '
SETCOLOR(GIcolnr)
REPL A->DesMemo WITH MEMOEDIT(A->DesMemo,06,03,20,76, T,,,120,3)
@ 05,02 CLEA TO 21,77
FOLDER(04,04,18,67," Modelo ")
@ 10,05 SAY 'Descripción '
@ 11,05 SAY 'Número de actividades '
@ 12,05 SAY 'Costos fijos por período:'
@ 13,05 SAY 'Descripción de recursos:'

```

PLANEACION Y CONTROL DE PROYECTOS

```

@ 14.05 SAY 'Unidades de periodos'
@ 10.30 GET A->DesGene Pict '@S37' VALID !EMPTY(A->DesGene)
@ 11.30 GET A->NumActi
@ 12.30 GET A->CosFijo
@ 13.30 GET A->DesRecu
@ 14.30 GET A->UniPeri VALID !EMPTY(A->UniPeri)
READ
IF A->NumActi = 0
  folder06.09.20.69."  Actividades      ")
  ly:=UPPER(SUBS(A->DesRecu.1.1))-LOWER(SUBS(A->DesRecu.2.15))
  @ 11.10 SAY 'No. de actividad'
  @ 12.10 SAY 'Descripcion'
  @ 13.10 SAY 'Duracion (normal)'
  @ 14.10 SAY 'Costo (normal)'
  @ 15.10 SAY 'Duración (ruptura)'
  @ 16.10 SAY 'Costo (ruptura)'
  @ 17.10 SAY 'Número de precedencias'
  IF !EMPTY(ly)
    @ 18.10 SAY '&ly'
  ENDIF
  Lc_nuva:=1
  SELECT 2
  SET ESCAPE ON
  DO WHILE Lc_nuva <= A->NumActi .AND. LASTKEY() <= 27
    Lc_llaj:=Lc_llav+StrZero(lc_nuva,6,0)
    SEEK(Lc_llaj)
    IF EOF()
      APPEND BLANK
      REPL B->NumMode WITH Lc_llav;
      B->NumActi WITH StrZero(Lc_nuva,6,0)
    ENDIF
    SETCOLOR(GlcColbrn)
    @ 11.33 SAY LTRIM(STR(VAL(B->NumActi),6,0))
    SETCOLOR(GlcColnro)
    @ 12.33 GET B->DesActi PICT '@S36'
    @ 13.33 GET B->DurNorm
    @ 14.33 GET B->CosNorm
    @ 15.33 GET B->DurFina
    @ 16.33 GET B->CosRupt
    @ 17.33 GET B->NumPrec
    IF !EMPTY(LY)
      @ 18.33 GET B->RecReqe
    ENDIF
    READ
    IF B->CosRupt = 0 .AND. B->DurFina = 0
      REPL B->CosRupt WITH B->CosNorm;
      B->DurFina WITH B->DurFina
    ENDIF
    Lc_tec1:=Lastkey()
    DO CASE
    CASE Lc_tec1=18
      IF Lc_nuva > 1
        SKIP 1
        Lc_nuva--
      ENDIF

```

PLANEACION Y CONTROL DE PROYECTOS

```

OTHERWISE
SELECT 3
Lc_cont:=1
DO WHILE Lc_cont <= B->NumPrec .AND. LASTKEY() <> 27
  Lc_colu:=Lc_llav+B->NumActi+StrZero(Lc_cont,6,0)
  SEEK(Lc_colu)
  IF EOF()
    APPEND BLANK
    REPL C->NumMode WITH A->NumMode.;
    C->NumActi WITH B->NumActi.;
    C->NumPrec WITH StrZero(Lc_Cont,6,0)
  ENDIF
  Lc_pan3:=SAVESCREEEN(17,41,21,70)
  @ 17,41 CLEA TO 21,70
  @ 17,41 TO 21,70
  @ 18,42 SAY 'No. de precedencia: '
  SETCOLOR(Gicolbrn)
  @ 18,64 SAY LTRIM(STR(VAL(C->NumPrec),6,0))
  SETCOLOR(Gicolnor)
  @ 20,42 SAY 'Actividad precedente: ' GET C->ActPrec
  READ
  RESTSCREEN(17,41,21,70,Lc_pan3)
  Lc_tec1:=LASTKEY()
  DO CASE
  CASE Lc_tec1=18
    IF Lc_cont > 1
      Skip -1
      Lc_cont--
    ENDIF
  OTHERWISE
  SKIP
  Lc_cont++
  ENDCASE
  ENDDO
  SELE 2
  SKIP
  Lc_nuva++
  ENDCASE
ENDDO
SET ESCAPE OFF
CPMNORMAL(A->NumMode)
CPMOPTIM(A->NumMode)
CPMBALANC(A->NumMode)
CN=1
DO WHILE CN < 3
  @ 03,01 clea to 23,78
  folder (03,02,23,75," Solución ")
  SETCOLOR(Gicolnor)
  SELE 1
  @ 07,06 SAY 'Proyecto'
  @ 07,39 SAY 'Duración Total'
  @ 07,62 SAY 'Costo Total'
  @ 08,06 SAY REPL('-',16)
  @ 08,24 SAY REPL('-',29)
  @ 08,55 SAY REPL('-',18)

```

```

@ 10.06 SAY 'Normal'
@ 12.06 SAY A->Dur'Norm Pct 999,999,999,999,99
@ 14.06 SAY A->Cos'Norm Pct 999,999,999,999,99
@ 12.06 SAY 'Optimo'
@ 12.35 SAY A->Dur'Opt Pct 999,999,999,999,99
@ 12.55 SAY A->Cos'Opt Pct 999,999,999,999,99
@ 14.06 SAY 'Rapura'
@ 14.35 SAY A->Dur'Rupt Pct 999,999,999,999,99
@ 14.55 SAY A->Cos'Rupt Pct 999,999,999,999,99
@ 19.20 TO 21.58
@ 20.22 PROMPT 'Consultar'
@ 20.36 PROMPT 'Imprimi'
@ 20.49 PROMPT 'Terminar'

```

SETCOLOR(Gleolmen)

MENU TO CN

SETCOLOR(Gleolnor)

DO CASE

CASE CN=1

OP=1

@ 10.06 PROMPT 'Normal'

@ 12.06 PROMPT 'Optimo'

@ 14.06 PROMPT 'Rapura'

SETCOLOR(Gleolmen)

MENU TO OP

SETCOLOR(Gleolnor)

TIME=SYS'norm'Optim'OP'4.4.

LA Param=SAVESCREEN=0.0.0.0.0.

4.06.00 CLEA TO 0.0.0.

4.06.00 S-A-1

Continuation: 1000 1000 1000 1000 1000 1000

4.06.00 S-A-1

```
INKEY(0)
IF !EMPTY(A->DesRecu)
  @ 06.03 CLEA TO 22,74
  SELE 4
  SET FILTER TO VAL(D->TipMode)=OP
  GO TOP
  RENG:=6
  LY2:=ALLTRIM(LOWER(A->UniPeri))
  LY3:=ALLTRIM(UPPER(SUBS(A->DesRecu,1,1))+LOWER(SUBS(A->DesRecu,2,15)))
  @ RENG,05 SAY 'Periodos: &LY2'
  @ RENG,39 SAY 'De!'
  @ RENG,49 SAY 'A!'
  @ RENG,68-LEN(LY3) SAY LY3
  RENG:=RENG+1
  @ RENG,35 SAY REPL('-',10)
  @ RENG,44 SAY REPL('-',10)
  @ RENG,53 SAY REPL('-',15)
  RENG:=RENG+1
  REGA:=0
  DO WHILE .NOT. EOF()
    CANT:=D->CanRecu
    DO WHILE .NOT. EOF() .AND. CANT==D->CanRecu
      SKIP
    ENDDO
    IF RENG >= 21
      @ 22.03 SAY 'Presione cualquier tecla para continuar...'
      INKEY(0)
      @ 08.03 CLEA TO 22,74
    ENDIF
    @ RENG,35 SAY REGA PICT '999,999'
    @ RENG,44 SAY IIF(EOF(),A->Dur&TIPO,D->NumPeri-1) PICT '999,999'
    @ RENG,54 SAY CANT PICT '999,999,999,99'
    REGA:=D->NumPeri-1
    RENG:=RENG+1
  ENDDO
  @ 22.03 SAY 'Presione cualquier tecla ...'
  INKEY(0)
ENDIF
SELE 1
RESTSCREEN(03,02,23,75,Lc_Pant2)
CASE CN=2
  SETCOLOR(GIcolnor)
ENDCASE
RETURN
```

El procedimiento anterior, es el programa que controla todas las llamadas de los algoritmos, para el análisis de redes, además realiza la captura de los datos y presenta la información obtenida en el análisis de redes, ya sea por pantalla o por impresora.

El procedimiento que coordina todas las actividades, para poder llevar a cabo el análisis por el método de la ruta crítica, comienza solicitando los datos que describen al proyecto, así como todas actividades que lo integran. La forma en que lo lleva a cabo es similar a todos los algoritmos del Sistema de Optimización de Recursos, solicitando los datos al usuario y almacenándolos directamente en los archivos que conforman la base de datos.

Una vez que se encuentran capturados todos los datos del proyecto ejecuta el procedimiento **CPMNORMAL**, el cual es el que se encarga de encontrar el modelo de ruta crítica normal, es decir se realiza el análisis de la red, utilizando los tiempos y costos en condiciones normales. Una vez que regresa el control, es ejecutado el procedimiento **CPMOPTIMI** el cual se encarga de comprimir la red hasta encontrar el modelo de ruptura, encontrando también el modelo óptimo, como se describió anteriormente.

Cuando se ha concluido el proceso de optimización y se cuenta con los tres modelos asociados al proyecto (normal, óptimo, ruptura) se procede a realizar un balanceo de los recursos disponibles, esto es se ejecuta el procedimiento **CPMBALANC** completando así las solución total del problema.

Después de haber encontrado la solución, continua presentando la información. Cuando la información es presentada en la pantalla de la computadora, se realiza un procedimiento de selección, para que el usuario pueda ver el detalle de los tiempos para cada modelo, esto se hace por la limitación de la pantalla para poder presentar mucha información. Para el caso de que la solución es presentada de forma impresa, la información es presentada de forma completa.

2.4 TECNICA DE EVALUACION Y REVISION DE PROGRAMAS.

La característica distintiva de la técnica de evaluación y revisión de programas (PERT) es su capacidad para abarcar la incertidumbre inherente a los tiempos calculados para la terminación de una actividad en ciertos tipos de proyectos. Aunque se puede prever con relativa seguridad, los requerimientos de tiempo para las actividades realizadas, a menudo, en el pasado y con poca variación del tiempo requerido, los cálculos de tiempo para las actividades necesarias para el desarrollo de nuevas tecnologías, o efectuar una actividad nueva y diferente son, en sí, menos exactos. Por ello, suele adoptarse el método PERT para proyectos de investigación y diseño, y que se utilicen modelos de red sin especificaciones para medir la incertidumbre en la planeación y el control de muchos proyectos de construcción, reconstrucción y ensamble.

La metodología de PERT permite analizar a proyectos y proporciona la información para la planeación y el control, proporciona la información que proporciona CPM, sólo que el modelo normal, es calculado como modelo medio esperado, como se menciona anteriormente, se puede obtener información como la siguiente:

1. Qué probabilidad existe de termina el proyecto a tiempo medio esperado, óptimo y/o de ruptura.
2. Cuál va ha ser el costo de terminar el proyecto a un tiempo determinado.
3. En cuánto tiempo se puede realizar el proyecto, corriendo el riesgo de no terminar a una fecha en un determinado porcentaje de probabilidad.
4. Qué probabilidad existe de terminar el proyecto a un tiempo cualquiera.

La metodología de desarrollo de PERT es muy parecida a la de CPM, salvo que se requieren realizar cálculos de probabilidades adicionales.

PERT logra un cálculo de probabilidades de la terminación de un proyecto a partir de tres cálculos para cada actividad, mediante la descripción de la distribución estadística de los tiempos posibles para cada actividad, y determinando la desviación estándar del tiempo de cada actividad, así como el tiempo para la terminación del proyecto. Los tres cálculos de tiempo PERT para cada actividad son:

1. Tiempo optimista (A), tiempo requerido para terminar la actividad si todo marcha bien.
2. Tiempo pesimista (B), tiempo requerido para terminar la actividad si todas las cosas salen mal.
3. Tiempo medio (M), tiempo requerido para completar la actividad en la mayor parte de los casos.

Los tiempos A y B se calculan sobre la base de que la probabilidad de que un tiempo real se encuentre fuera de sus límites es de una en cien. El tiempo esperado de la actividad y su cálculo de varianza se basan en la consideración de que la distribución de los tiempos de actividad se acerca a una distribución beta.

Los cálculos del tiempo esperado de la actividad (t_e) y su varianza (σ^2) son como sigue:

$$t_e = \frac{(A + 4M + B)}{6}$$

ESTIA TESIS FUE
 RECIBIDA EN LA BIBLIOTECA

$$\sigma^2 = \left[\frac{B - A}{6} \right]^2$$

Dado que el tiempo que se requiere para terminar cada actividad es una variable aleatoria, el tiempo esperado para terminar todo el proyecto (la suma de los tiempos esperados para las trayectorias críticas) también es una variable aleatoria. La varianza del tiempo que se espera es igual a la suma de las varianzas de las actividades en la ruta crítica.

La distribución de una suma de variables aleatorias sigue una distribución normal de forma de campana, sin importar cual sea la distribución de los componentes de la suma. Esto nos permite utilizar el cálculo de áreas bajo la curva normal para calcular la probabilidad de que el proyecto termine dentro de una estructura específica de tiempo.

Por lo tanto tenemos que, hay una probabilidad de 50% de que el proyecto se termine al tiempo esperado y un 50% de probabilidades de que tomé más tiempo. Ahora bien, si queremos saber la probabilidad de que se termine en un tiempo diferente al medio esperado, debemos calcular el número de desviaciones estándar que tiene un tiempo deseado con respecto al tiempo promedio de terminación. Se puede utilizar la siguiente fórmula:

$$z = \frac{T_D - T_E}{\sigma}$$

donde

T_D = tiempo deseado de terminación.
 T_E = tiempo medio esperado.
 z = número de desviaciones estándar que separan T_D y T_E

y a partir de la fórmula o tablas para encontrar el área bajo la curva (distribución normal) y la z, se puede calcular el porcentaje de probabilidades que existen en termina un proyecto.

```

PROCEDURE PERTVARI
PARA Pr_Mode
PRIV TotVari

SELE 3
SET ORDER TO 2
SELE 2
SEEK(Pr_Mode)           && Recorrido hacia adelante.
M->TotVari:=0
DO WHILE .NOT. EOF() .AND. B->NumMode=Pr_Mode
  Lc_Regi:=RECNO()
  IF B->ActiCrit
    PTRECADEL(B->NumMode,B->NumActi,B->Varlanz)
  ENDIF
SELE 2
GOTO Lc_Regi
SKIP
ENDDO
SELE 3
SET ORDER TO 1
SELE 1
REPL A->VarTota WITH M->TotVari
RETURN
    
```

Con el procedimiento **PERTVARI** se realiza un recorrido de la red, para poder obtener la suma de las varianzas de la ruta crítica, la cual será utilizada para calcular el número de desviaciones estándar que separa a los tiempos, el deseado y el medio esperado. Este procedimiento, también es recursivo, debido a que la estructura de la red, como se había mencionado anteriormente, también es recursiva, quedando el algoritmo genérico, para poder recorrer árboles con una gran cantidad de nodos y de ramificaciones.

Para poder usar los algoritmos de análisis de redes, descritos anteriormente, es necesario tener algunas consideraciones entre CPM y PERT:

- a) CPM tiene dos tiempos, el normal y el de ruptura. PERT utiliza tres, el optimista, medio y el pesimista. En el caso de PERT se utilizan los mismos procedimientos sobre la red, sólo que se considera que el tiempo óptimo equivale al tiempo de ruptura, es decir el menor tiempo que se puede realizar una actividad; el tiempo medio esperado se considera el tiempo normal, es decir, el tiempo que tiene un cincuenta por ciento de probabilidades de que se termine la actividad.
- b) El modelo de partida de CPM es el modelo normal, mientras que el modelo de partida de PERT es el modelo medio esperado.
- c) Los costos se manejan de la misma forma en ambos métodos.

```

FUNCTION NORMAL
** Integración trapezoidal de la curva de distribución normal.
PARA Z
LOCA Lc_Suma, Lc_Resu

Lc_Resu:=0.00
FOR I:=Z TO 5 STEP 0.1
Y0:=F(I)
Y1:=F(I+0.1)
DELTA:=-Y0-Y1
Lc_Resu:=Lc_Resu+0.1*(Y1+(DELTA/2))
NEXT
RETURN (Lc_Resu)
    
```

La función **NORMAL** realiza el cálculo del área bajo la curva que está descrita por la función **F**, que es la curva en forma de campana. La manera en que se calcula el área bajo la curva es por medio de un método numérico, denominado integración trapezoidal, el cual consiste en sumar el área de pequeños

trapezoides que se forman con la misma curva que se está analizando. De ésta forma se obtiene una aproximación aceptable para poder obtener la probabilidad de terminar en cierto tiempo una actividad. La acumulación del área de los trapecios que se van construyendo, se lleva a cabo por medio de un ciclo repetitivo que va desde el punto que se quiere calcular, el cual es mandado como parámetro, hasta el valor de 5, después de éste punto el área que se acumula es demasiado pequeña, debido a la estructura de la distribución, por lo cual se puede despreciar. De ésta forma se da la solución genérica, para el cálculo de probabilidades, sin tener que usar tablas.

```

FUNCTION Z
** Cálculo del punto porcentual para la distribución normal.
PARA Pr_Area
LOCA Lc_Resu, Lc_Area

Lc_Resu:=1
IF Pr_Area >= 0. AND Pr_Area <= 1
  Lc_Resu:=5
  Lc_Area:=0
  DO WHILE Lc_Area < Pr_Area
    Y0:=F(Lc_Resu-0.1)
    Y1:=F(Lc_Resu)
    DELTA:=Y0-Y1
    Lc_Area:=Lc_Area+0.1*(Y1+(DELTA/2))
    Lc_Resu:=Lc_Resu-0.1
  ENDDO
ENDIF
RETURN (Lc_Resu)

```

La función **Z** realiza el cálculo del punto porcentual de la distribución normal, para poder calcular el tiempo que se necesita para conseguir cierto porcentaje de probabilidad para terminar una actividad. La forma como trabaja es, al igual que la función **NORMAL**, sólo que en la función **Z** se efectúa un ciclo, en donde se va acumulando el área bajo la curva, mientras no sea mayor a la superficie que debiera ocupar la probabilidad que debemos encontrar, de ésta manera se va obteniendo el valor del punto porcentual; una vez que se llega a

la superficie que se busca, se regresa el valor del punto porcentual de la distribución en que se quedó, regresando, así, un valor con una precisión aceptable.

```
FUNCTION F
** Curva de la distribución normal.
PARA X
LOCA Lc_Resu
    Lc_Resu:=(1/(SQRT(2*3.1415927)))*EXP(-(X*X)/2)
RETURN (Lc_Resu)
```

La función **F** es la ecuación que describe la curva de distribución normal, ésta es utilizada para ser evaluada, tantas veces como sea necesario para poder ser encontrada el área bajo la curva, la cual proporciona el porcentaje de probabilidad de llevarse a cabo un evento. La función **F** consiste en una expresión matemática que está en función del parámetro **X** que se le pasa en el momento de la llamada, utiliza los operadores aritméticos y dos funciones definidas, la primera es **SQRT** la cual calcula la raíz cuadrada de su argumento y **EXP** que es el número exponencial de Euler.

Integrando los procedimientos anteriores junto con los algoritmos para el análisis de redes y el procedimiento de entrada y salida de datos, similar al de CPM, queda el programa completo para poder realizar el análisis de proyectos bajo condiciones de incertidumbre, la información que proporciona es suficiente para planear y controlar todas las actividades de un proyecto.

3. MANTENIMIENTO DE EQUIPO INDUSTRIAL

3.1 CONCEPTOS GENERALES.

Casi todo el equipo industrial se deteriora con la edad o con el uso, a menos que se tomen medidas para conservarlo o mantenerlo. En algunas ocasiones en lugar de conservarlo puede ser más económico reemplazarlo por otro. A menudo sucede que se sustituyen artículos no porque no cumplan con las normas de diseño, sino por que hay equipo más moderno que cumple con normas superiores.

En la práctica existe cierto traslape entre los problemas de mantenimiento y los de reemplazo, por ejemplo reemplazar una pieza de una máquina, se considera como reemplazo con respecto a la pieza y como mantenimiento con respecto a la máquina. En realidad, la mayoría de las actividades de mantenimiento y las reparaciones consisten en reemplazar los componentes o subsensibles; estos pueden desmontarse y desmantelarse o enviarse a un taller especializado para su reparación.

El mantenimiento es una actividad que conserva mejor el equipo u otros activos en condiciones que contribuyan mejor a las metas de la organización. Esto algunas veces se reduce al objetivo de minimizar los costos de mantenimiento a largo plazo. Sin embargo, concierne a la seguridad, confiabilidad, estabilidad del empleo y la supervivencia económica, por lo que las actividades de mantenimiento deben ser responsables de un amplio espectro de objetivos. Las decisiones de mantenimiento deben reflejar la viabilidad a largo plazo de todo el sistema.

A continuación se describen cuatro modelos para resolver problemas de reemplazo y mantenimiento. Estos modelos sirven para establecer programas óptimos de mantenimiento, desde el punto de vista de costos, cabe señalar que cuando está en riesgo la vida humana o se pone en peligro cualquier otro factor de importancia de una organización, estos modelos no son los adecuados.

3.2 DETERMINACION DE LA VIDA ECONOMICA DEL EQUIPO.

Un centro de trabajo, en la industria, siempre tiene asociado un cierto gasto de operación, éste incluye todos los desembolsos que tiene que realizar la organización para mantenerlo en operación y así contribuir a los objetivos de la organización. Estos desembolsos incluyen el valor de adquisición de equipo nuevo, la depreciación que sufre el equipo por usarse y los consumibles que se requieren para poderlo tener en operación, tales como reparaciones, combustibles, grasas, energía eléctrica, etc.

Debido al ciclo de vida de la mayoría del equipo industrial, los costos que se mencionaron anteriormente no son constantes, varían porque por un lado se tiene que, los primeros años el equipo sufrirá las depreciaciones más grandes mientras que los gastos para mantenerlo funcionando serán mínimos, ya que se encuentra en buenas condiciones, y conforme transcurre el tiempo los costos se invertirán.

El objetivo de los responsables de mantenimiento, es tener el centro de trabajo operando a los costos más bajos posibles. Esto implica saber hasta que punto es conveniente reemplazar el equipo, para que no se eleven los costos de operación, para ello existe un modelo matemático que sirve para determinar dicho punto, el modelo determina la Vida Económica del Equipo.

Este modelo sirve para analizar el equipo que se deteriora con el tiempo y saber cuando es el momento oportuno para reemplazarlo. Este modelo considera que los costos de operación de un activo se incrementan conforme transcurre el tiempo, así mismo considera el costo de capital o de depreciación, el cual puede ser expresado como costo promedio por periodo, y este costo por periodo promedio disminuirá mientras transcurre el tiempo. Por lo tanto, considerando los dos costos, llega un punto en el que la rapidez del aumento de los costos de operación ya no compensa los ahorros en el costo promedio de capital. En este punto es donde se justifica el reemplazo, si no se hace, los costos promedio se incrementarían gradualmente.

Para este análisis se requiere de la determinación de los costos totales de operación (**CTO**), estos son comúnmente obtenidos a partir de información histórica y consisten en los gastos por periodo que son necesarios para que el equipo funcione, tal como combustibles, grasas, aceites, energía eléctrica, mantenimiento, etc.

El costo de depreciación o costo total de capital (**CTC**) en la misma unidad de tiempo (periodo) que el **CTO**, es obtenido a partir del valor de adquisición (**VA**) y se le resta su valor de rescate en ese periodo.

$$CTC = VA - \text{Valor de Rescate}$$

Estos dos costos se van acumulando y para cada periodo se obtienen los costos totales:

$$CT_p = CTO_p + CTC_p$$

Al dividir los costos totales por el número de periodos se obtiene el costo promedio **CP**, el cual es el que determina el punto en que hay que reemplazar el equipo, es decir en el periodo en donde llega a su nivel más bajo el **CP** es el momento de reemplazar, si no se hace los costos se incrementarían indefinidamente.

En la implementación en computadora de este algoritmo, primero se tienen que capturar los datos del equipo que se quiere analizar, la forma en que se captura es similar a todos los algoritmos del Sistema de Optimización de Recursos, y los datos son: valor de adquisición, número de periodos de vida útil del equipo, valor de rescate en cada periodo y gastos de operación por periodo. A continuación se presenta el código del programa que se encarga de realizar la captura de los datos de entrada, estos datos se deben generar a partir de los datos históricos de los activos de la empresa:

```
@ 05,02 TO 21.77
SETCOLOR(GColor)
@ 05,16 SAY ' Descripción del Modelo [Ctrl-W para Salir] '
SETCOLOR(GColor)
REPL A->DesMemo WITH MEMOEDIT(A->DesMemo,06.03.20,76,.T...,120,3)
@ 05,02 CLEA TO 21.77
```

MANTENIMIENTO DE EQUIPO INDUSTRIAL

```

FOLDER(04,04,18,64,"      Modelo      ")
@ 11,05 SAY 'Descripción      '
@ 13,05 SAY 'Valor de adquisición:'
@ 15,05 SAY 'Número de periodos:'
@ 11,25 GET A->DesGene Pict '@S39' VALID !EMPTY(A->DesGene)
@ 13,25 GET A->ValAdq
@ 15,25 GET A->NumPeri
READ
IF A->NumPeri>0
  folder06,09,20,69,"      Datos por periodo      ")
  @ 12,11 SAY 'No. de Periodo      '
  @ 14,11 SAY 'Gasto de operación:'
  @ 16,11 SAY 'Valor de rescate      '
  Lc_nuva:=1
  SELECT 2
  SET ESCAPE ON
  DO WHILE Lc_nuva <= A->NumPeri .AND. LASTKEY() <> 27
    Lc_llav:=Lc_nuva+StrZero(Lc_nuva,3,0)
    SEEK(Lc_llav)
    IF EOF()
      APPEND BLANK
      REPL B->NumMode WITH Lc_llav.:
      B->NumPeri WITH StrZero(Lc_nuva,3,0)
    ENDIF
    SETCOLOR(Glcobri)
    @ 12,30 SAY LTRIM(STR(VAL(B->NumPeri),3,0))
    SETCOLOR(GlcInnr)
    @ 14,30 GET B->GasOper Pict '999999999.99'
    @ 16,30 GET B->ValResc Pict '999999999.99'
    READ
    Lc_tect:=Lastkey()
    DO CASE
    CASE Lc_tect=18
      IF Lc_nuva > 1
        SKIP -1
        Lc_nuva--
      ENDIF
    OTHERWISE
      SKIP
      Lc_nuva++
    ENDCASE
  ENDDO
  SET ESCAPE OFF

```

El procedimiento comienza con la captura de la descripción del problema y sus datos generales, como son el valor de adquisición del equipo y el número de periodos que se van a analizar, los periodos pueden ser días, semanas, meses, años, etc., según del equipo que se trate.

Una vez que se tienen los datos generales del problema, se lleva a cabo la captura de los datos de cada uno de los periodos en que se va a analizar el problema, estos datos son, los gastos de operación y el valor de rescate del activo en ese periodo.

Una vez que se tienen los datos que describen al problema, sólo es necesario que se ejecute el siguiente procedimiento, el cual realiza el análisis:

```

PROCEDURE Vidaecono
PARA nModelo
LOCA LcCtop, LcCtmm, LcNupe

SELECT 1
SEEK(nModelo)
SELECT 2
SEEK(nModelo)
LcCtop = 00
LcCtmm = Cran_M
LcNupe = "
DO WHILE NOT EOF() AND B->NumMode = nModelo
LcCtop = LcCtop + B->GasOper
REPL B->CosOper WITH LcCtop,;
      B->CosDerr WITH A->ValAdq - B->ValResc,;
      B->CosTota WITH B->CosOper - B->CosDepr,;
      B->CosProm WITH B->CosTota / VAL(B->NumPer),;
      B->PerOpt WITH "
IF LcCtmm > B->CosProm
LcCtmm = B->CosProm
LcNupe = B->NumPer
ENDIF
SKIP
ENDDO
    
```

MANTENIMIENTO DE EQUIPO INDUSTRIAL

SEEK(nModelo+LeNupe)
REPL B->PerOpti WITH 'S'
SELE 1
RETURN

El procedimiento **Vidaecon** consiste de un ciclo iterativo que se realiza un número de veces igual al número de periodos que se está analizando al equipo, para cada iteración calcula el costo de depreciación, restando del valor de adquisición el valor de rescate y acumula el costo de operación en cada periodo. El costo total lo calcula sumando el costo de depreciación más el costo de operación acumulado. Finalmente el costo promedio, lo calcula dividiendo el costo total del periodo entre el número de periodo que se está analizando.

El procedimiento **Vidaecon** también se encarga de marcar el periodo en donde el costo promedio es el más bajo.

Con esto, se tienen calculados los costos por periodo así como se identifica aquel periodo con los costos más bajos, y pueden ser presentados ya sea por pantalla o por impresora.

La información es presentada en pantalla con el siguiente procedimiento:

@ 03,01 cles to 23,78 folder (03,02,23.75." SETCOLOR(Gicolnor)	Solución)
@ 06,03 SAY 'Periodo'		
@ 06,18 SAY 'Costo de'		
@ 06,34 SAY 'Costo de'		
@ 06,53 SAY 'Costo'		
@ 06,69 SAY 'Costo'		
@ 07,17 SAY 'Operación'		
@ 07,30 SAY 'Depreciación'		
@ 07,53 SAY 'Total'		
@ 07,66 SAY 'Promedio'		
@ 08,03 SAY REPL('-',7)		
@ 08,12 SAY REPL('-',14)		
@ 08,28 SAY REPL('-',14)		

MANTENIMIENTO DE EQUIPO INDUSTRIAL

```

@ 08.44 SAY REPL('-',14)
@ 08.60 SAY REPL('-',14)
REN1:=9
SELE 2
SELE(Lc_Llav)
DO WHILE .NOT. EOF() .AND. B->NumMode=Lc_Llav
  IF REN1 >= 21
    @ 22.08 SAY SPACE(68)
    @ 22.08 SAY 'Presione cualquier tecla para continuar...'
    INKEY(0)
    @ 07.05 CLEA TO 21,75
    REN1:=9
  ENDIF
  IF B->PerOptu='S'
    SETCOLOR(gicolbri)
  ENDIF
  @ REN1,05 SAY B->NumPeri
  @ REN1,12 SAY B->CosOper PICT '999,999,999.99'
  @ REN1,28 SAY B->CosDepr PICT '999,999,999.99'
  @ REN1,44 SAY B->CosTota PICT '999,999,999.99'
  @ REN1,60 SAY B->CosProm PICT '999,999,999.99'
  IF B->PerOptu='S'
    SETCOLOR(gicolnor)
  ENDIF
  REN1:=REN1-1
  SKIP
ENDDO
@ 22.05 SAY SPACE(68)
@ 22.05 SAY 'Presione cualquier tecla para continuar...'
INKEY(0)
@ 22.05 SAY SPACE(68)
SETCOLOR(Gicolnor)

```

El procedimiento anterior presenta, para cada periodo, el desglose de todos sus costo, esto incluye el costo de operación, el costo de capital o depreciación, el costo total y el costo promedio. Para el caso en el que el periodo es el indicado para realizar el reemplazo, es decir que se ha llegado al final de la vida económica del equipo, este renglón se presenta de forma resaltada en la pantalla, para indicar así cual es dicho periodo.

3.3 ANALISIS DE REEMPLAZO.

Las dos razones más importantes para reemplazar la maquinaria son incrementar la capacidad productiva y reducir los costos de operación. Una tercera razón es verse libremente de la maquinaria inadecuada o de las máquinas que se hayan averiado o desgastado. Otra causa es la insuficiencia de las existencias para realizar trabajos de mayores dimensiones o ejecutar labores de tolerancias más rigurosas. Una quinta razón es que los problemas de mano de obra se reducen obteniendo máquinas que pueden instalarse en grupos que permitan a un operario atender a dos o más de ellas. Una sexta razón es simplificar las operaciones, consiguiendo máquinas, como las semiautomáticas y automáticas, en las cuales pueden combinarse una serie de operaciones sucesivas en lugar de hacer el trabajo en varias individualmente. Es evidente que muchas de estas razones incluyen además un aumento en la capacidad o una reducción en el costo o ambas cosas a la vez.

El modelo de análisis de reemplazo sirve para determinar cual es la mejor alternativa para reemplazar el equipo con que se cuenta, considerando como criterio el menor costo total a través del tiempo en el que incurren todas las alternativas que se tienen.

Este tipo de modelo sirve para decidir, cuando existe más de una alternativa, acerca de lo que más conviene en el momento de reemplazar al equipo de una empresa industrial. Para este tipo de análisis es indispensable contar con el valor de adquisición del equipo, así como el costo de operación y el tiempo de vida útil para cada una de las alternativas. Así mismo es necesario saber la tasa de interés, es decir el porcentaje de inflación que sufre la moneda con que se está analizando el equipo.

Si C es el valor de adquisición, y R_n el costo de operación en el año n , i la tasa de interés y suponiendo que el equipo se va a reemplazar después de r años y considerando que el gasto tiene lugar a principio de cada año, el valor presente del gastos, esta determinado por:

$$P(r) = C + R_1 \cdot (1+i)^{-1} + R_2 \cdot (1+i)^{-2} + \dots + R_r \cdot (1+i)^{-r}$$

Ahora bien $P(r)$ aumenta conforme r aumenta, de manera que el valor presente si reemplazamos después de $(r+1)$ años, es mayor que si reemplazamos después de r años.

Al igual que en el algoritmo para determinar la vida económica del equipo, es necesario primero capturar los datos, con lo cual se realizará el análisis para determinar la mejor alternativa de reemplazo de equipo industrial, desde el punto de vista de reducción de costos. El procedimiento que realiza la captura de estos datos se encuentra codificado a continuación:

```
@ 05,02 TO 21,77
SETCOLOR(GlcColbr)
@ 05,16 SAY ' Descripción del Modelo [Ctrl-W para Salir] '
SETCOLOR(GlcInor)
REPL A->DesMemo WITH MEMOEDIT(A->DesMemo,06,03,20,76..T.,120,3)
FOLDER(04,04,18,64," Modelo ")
@ 11,05 SAY 'Descripción '
@ 13,05 SAY 'Número de alternativas:'
@ 15,05 SAY 'Tasa de interés '
Lc_llav:=StrZero(Lc_numo,6,0)
SELE 1
@ 11,28 GET A->DesGene Pict '@S36' VALID !EMPTY(A->DesGene)
@ 13,28 GET A->NumAlte
@ 15,28 GET A->TasInte
READ
IF A->NumAlte > 0
  folder(06,09,20,69," Alternativas ")
  @ 10,11 SAY 'No. de alternativa '
  @ 12,11 SAY 'Valor de adquisición:'
  @ 14,11 SAY 'Vida útil '
  @ 16,11 SAY 'Valor de rescate '
  Lc_nuva:=1
  SELECT 2
  SET ESCAPE ON
  DO WHILE Lc_nuva <= A->NumAlte .AND. LASTKEY() <> 27
    Lc_llal:=Lc_llav+StrZero(Lc_nuva,3,0)
```

MANTENIMIENTO DE EQUIPO INDUSTRIAL

```

SEEK(Lc_llav)
IF EOF()
  APPEND BLANK
  REPL B->NumMode WITH Lc_llav.;
  B->NumAlte WITH StrZero(Lc_nuva,3,0)
ENDIF
SETCOLOR(Glcolhri)
@ 10,34 SAY LTRIM(STR(VAL(B->NumAlte),3,0))
SETCOLOR(Glcolnor)
@ 12,34 GET B->ValAdq;
@ 14,34 GET B->VidUtil
@ 16,34 GET B->ValResc
READ
Lc_tec1=Lastkey()
DO CASE
CASE Lc_tec1=18
  IF Lc_nuva > 1
    SKIP -1
    Lc_nuva--
  ENDIF
OTHERWISE
SELECT 3
  Lc_cont:=1
  DO WHILE Lc_cont <= B->VidUtil .AND. LASTKEY() <> 27
    Lc_colu:=Lc_llav+B->NumAlte+StrZero(Lc_cont,3,0)
    SEEK(Lc_colu)
    IF EOF()
      APPEND BLANK
      REPL C->NumMode WITH A->NumMode.;
      C->NumAlte WITH B->NumAlte.;
      C->NumPeri WITH StrZero(Lc_cont,3,0)
    ENDIF
    Lc_pan3:=SAVESCREEN(18,26,22,74)
    @ 18,26 CLEA TO 22,74
    @ 18,26 TO 22,74
    @ 19,28 SAY 'No. de Periodo : '
    SETCOLOR(Glcolhri)
    @ 19,49 SAY LTRIM(STR(VAL(C->NumPeri),3,0))
    SETCOLOR(Glcolnor)
    @ 21,28 SAY 'Gastos de operación: ' GET C->CosPeri PICT '999999999,99'
  READ
  RESTSCREEN(18,26,22,74,Lc_pan3)
  Lc_tec1=LASTKEY()
  DO CASE
  CASE Lc_tec1=18
    IF Lc_cont > 1
      Skip -1
      Lc_cont--
    ENDIF
  OTHERWISE
    SKIP
    Lc_cont++
  ENDCASE
ENDDO
SELE 2

```

```

SKIP
Lc_nuva++
ENDCASE
ENDDO
SET ESCAPE OFF
MINCOM:=0.00

```

En el procedimiento anterior primero se realiza la captura de los datos generales del problema, estos datos son la descripción del problema, el número de alternativas que se tienen y la tasa de interés. Una vez que se ha realizado esta captura, se introducen los datos que describen a cada una de las alternativas que se tienen, como son, el valor de adquisición, el número de periodos de vida útil y el valor de rescate, así mismo, para cada periodo de vida útil se captura el gasto de operación que es necesario realizar. Al igual que en el modelo anterior, los periodos de vida útil pueden ser días, semanas, meses, años, etc.

A continuación se presentan los programas en donde son implementados los algoritmos para realizar el análisis de reemplazo:

```

FUNCTION MCM
** Cálculo del M.C.M.
PARA Lc_mode
LOCA Lc_mcm

Lc_mcm:=0
SELE 2
SEEK(Lc_mode)
DO WHILE !EOF() .AND. Lc_mode=B->NumMode
IF B->VidUtil!>0
IF Lc_mcm==0
Lc_mcm:=B->VidUtil
ELSE
Minimo:=MIN(Lc_mcm,B->VidUtil)
Maximo:=Lc_mcm*B->VidUtil
DO WHILE Minimo <= Maximo .AND.;
(Minimo/B->VidUtil != INT(Minimo/B->VidUtil)) .OR.;
Minimo/Lc_mcm != INT(Minimo/Lc_mcm)
Minimo++
ENDDO
Lc_mcm:=minimo
ENDIF
ENDIF
ENDIF

```

```

SKIP
ENDDO
RETURN (Lc_mcm)

```

Lo primero que se requiere para poder comparar las diferentes alternativas que se tienen en un problema de reemplazo, es determinar el mínimo común múltiplo de los periodos que integran todas las alternativas posibles, es decir es necesario normalizar a todas las alternativas a un determinado número de periodos común a todas, para poder compararlas con la misma unidad de costos (misma moneda) y de tiempo (mismo número de periodos). El procedimiento **MCM** se encarga de esto; consiste en un ciclo iterativo en donde se evalúa cada una de las alternativas y se va obteniendo el mínimo común múltiplo de todas las alternativas de manera gradual.

```

PROCEDURE AnaReemplazo
** Procedimiento de analisis de reemplazo.
PARA Pr_mode

  SELE 4
  ZAP
  SELE 1
  SEEK(Pr_mode)
  micomu:=MCM(A->NumMode)
  MinCom:=micomu
  FOR I:=1 TO A->NumAlte
  IS:=STRZERO(I,2,0)
  Lc_llav:=A->NumMode+STRZERO(I,3,0)
  SELE 2
  FIND &Lc_llav
  ValPre&IS:=0.00
  IF B->VidUtil!=0
  FOR J:=1 TO (micomu/B->VidUtil)
  ValPre&IS:=ValPre&IS+PF(B->ValAdq,A->TasInte,(J-1)*B->VidUtil)
  FOR K:=1 TO B->VidUtil
  Lc_llav2:=Lc_llav+STRZERO(K,3,0)
  SELE 3
  SEEK(Lc_llav2)
  SELE 2
  ValPre&IS:=ValPre&IS+PF(C->CosPeri,A->TasInte,K*B->VidUtil*(J-1))
  NEXT
  ValPre&IS:=ValPre&IS+PF(B->ValResc,A->TasInte,B->VidUtil*J)
  NEXT
  ENDIF
  SELE 4

```

MANTENIMIENTO DE EQUIPO INDUSTRIAL

```
APPE BLANK
REPL D->NumAle WITH STR(1,3,0);
  D->DesAle WITH ";
  D->ValPres WITH ValPre&IS
NEXT
SELE 4
GO TOP
MINIM:=GRAN_M
REGIS:=0
DO WHILE .NOT. EOF()
  IF MINIM > D->ValPres
    MINIM := D->ValPres
    REGIS := RECN()
  ENDIF
SKIP
ENDDO
GOTO REGIS
REPL D->SelOpti WITH 'S'
SELE 1
RETURN
```

El procedimiento **AnaReemplazo** comienza con llamar a la función **MCM** para obtener el mínimo común múltiplo; después se realiza un ciclo iterativo que se repite un número de veces igual al número de alternativas que existen. Para cada iteración, es decir para cada alternativa, se trasladan a valor presente, todos los desembolos en que incurrirían si se optara por seleccionarla y mantenerla el número de periodos igual minimo común múltiplo obtenido con anterioridad. Una vez que se tienen todos los valores presentes de las diferentes alternativas que se tiene para realizar el reemplazo, se procede a determinar cual de ellas es la de menor costo, ésto se lleva a cabo de forma directa por medio del segundo ciclo iterativo que se muestra en el código del programa.

A continuación se muestra la función **PF** que es la que se encarga de trasladar cierta cantidad a valor presente, para ello se le tiene que mandar como parámetro el valor la tasa de interés y el número de periodos que se quieren trasladar, regresa al lugar donde fue llamada, el valor presente de la cantidad.

```

FUNCTION PF
** Valor presente.
PARA VALOR, TASA, PERIODOS

RETURN ((1+(TASA/100))^(PERIODOS))*VALOR
    
```

El resultados de este modelo es presentado en la pantalla de la computadora por medio del siguiente programa:

```

@ 03,01 clea to 23,78
folder(03,02,23,75,"
SETCOLOR(Gcolnor)      Solución      ")
LY:="Análisis de reemplazo a '+LTRIM(TRANS(MINCOM,'999,999'))+' periodos.'
@ 07,INT((80-LEN(LY))/2) SAY LY
@ 08,INT((80-LEN(LY))/2) SAY REPL('-',LEN(LY))
REN1:=10
SELE 4
GO TOP
DO WHILE .NOT. EOF()
IF REN1 >= 21
    @ 22,05 SAY SPACE(64)
    @ 22,05 SAY 'Presione cualquier tecla para continuar...'
    INKEY(0)
    @ 22,05 SAY SPACE(65)
    REN1:=10
    @ 10,05 CLEA TO 21,75
ENDIF
IF D->SelOpti='S'
    SETCOLOR(Gcolbri)
ENDIF
@ REN1,05 SAY D->NumAlte
@ REN1,10 SAY D->DesAlte
@ REN1,54 SAY D->ValPres PICT '999,999,999,999'
SETCOLOR(Gcolnor)
SKIP
REN1:=REN1+1
ENDDO
@ 22,05 SAY SPACE(64)
@ 22,05 SAY 'Presione cualquier tecla para continuar...'
    
```

MANTENIMIENTO DE EQUIPO INDUSTRIAL

INKEY(0)
@ 22,05 SAY SPACE(65)
SETCOLOR(Gicolor)

El procedimiento anterior muestra el costo total de cada una de las alternativa, trasladado a valor presente y evaluadas para un número de periodos común a todas. También indica, por medio de un color diferente en la pantalla de la computadora, la alternativa más económica.

3.4 MANTENIMIENTO PREVENTIVO.

La forma tradicional en que se establecen los programas de mantenimiento preventivo consiste en distribuir la capacidad del área encargada del mantenimiento, en el equipo al que hay que mantener de manera uniforme, considerando que el objetivo consiste en proporcionar mantenimiento preventivo a todo el equipo, sin importar el costo ni la probabilidad de que falle uno u otro equipo en condiciones críticas. El objetivo del mantenimiento preventivo es mantener operando el mayor tiempo, con las menores interrupciones, y al menor costo posible a todo el sistema, por lo cual no puede ser un buen programa de mantenimiento aquel que considera que todo el equipo con el que cuenta una planta tiene la misma posibilidad de descomponerse o que todo el equipo tiene la misma importancia de acuerdo a los objetivos generales de la organización. Existe equipo en la industria en el que el costo por hora equivale a el costos por hora de todo el sistema, esto es por que si deja de funcionar cierto tiempo, la organización entera deja de producir ese mismo tiempo, tal es el caso de calderas, subestaciones eléctricas, sistemas de aprovisionamiento de materiales, etc. Por otro lado existe equipo que si deja de funcionar hay alternativas para desarrollar las funciones que realizaba, que el tiempo de reposición es demasiado corto, que sólo se quedarían sin operar pequeñas partes sin importancia en la planta, que se puede diferir el trabajo y que incluso el costo del mantenimiento correctivo (costo de reparación más el impacto de dejar de operar) es menor que el costo mantenimiento preventivo. La forma de saber la importancia que tiene el equipo en la industria, se lleva a cabo a través de análisis ABC en donde se clasifican de acuerdo a su importancia, tiempo de reposición, costo total de dejar de operar, etc

Un programa efectivo de mantenimiento preventivo necesita un sistema de registro de información, personal adiestrado, inspecciones regulares y servicio adecuado. El costo se incrementa conforme lo hacen las actividades de mantenimiento. Por otro lado, cuando el equipo se descompone, los trabajadores y las máquinas están ociosos, resultando un tiempo de producción perdido, retraso de la programación y costos altos de reparación de emergencia. Estos costos se reducen al aumentar las cuadrillas y al aumentar las actividades de mantenimiento preventivo. Los costos de mantenimiento preventivo generalmente son menores

a los costos del mantenimiento correctivo hasta cierto punto en donde la integración de ambos costos da como resultado el costo más bajo, más allá el mantenimiento preventivo adicional no está económicamente justificado (aunque la seguridad y otros objetivos pueden garantizarlo).

Los modelos de probabilidad son especialmente útiles para el análisis de 1) la política que debe seguirse de mantenimiento preventivo, y 2) si se sigue una política de mantenimiento preventivo, qué tan frecuente debe ser realizado el servicio.

Los datos colectados son:

- 1) Costo de servicio de mantenimiento preventivo.
- 2) Costo de mantenimiento correctivo.
- 3) Probabilidad de reparación.

La probabilidad de reparación refleja el hecho de que la falla ocurra aun si el mantenimiento preventivo es realizado, pero la probabilidad de falla generalmente se incrementa con el tiempo, después de una actividad de mantenimiento.

El número esperado acumulado de fallas B en M meses es:

$$B_n = N \sum_1^n P_n + B_{n-1}P_1 + B_{n-2}P_2 + \dots + B_1P_{n-1}$$

donde N = número de unidades.

P = probabilidad de falla durante un mes dado después del mantenimiento.

n = periodos de mantenimiento.

El programa donde se capturan los datos de entrada se presenta a continuación:

```
@ 05.02 TO 21.77
SETCOLOR(GIcolbri)
@ 05.16 SAY ' Descripción del Modelo [Ctrl-W para Salir] '
SETCOLOR(GIcolnor)
REPL A->DesMemo WITH MEMOEDIT(A->DesMemo,06,03,20,76..T...120,3)
@ 05.02 CLEA TO 21.77
FOLDER(04,04,18,64," Modelo ")
@ 11.05 SAY 'Descripción : '
@ 12.05 SAY 'Número de elementos : '
@ 13.05 SAY 'Costo de falla : '
@ 14.05 SAY 'Costo de reemplazo : '
@ 15.05 SAY 'Número de periodos : '
@ 11.28 GET A->DesGene Pict '@Sj6' VALID !EMPTY(A->DesGene)
@ 12.28 GET A->NumElem
@ 13.28 GET A->CosFall
@ 14.28 GET A->CosReem
@ 15.28 GET A->NumPeri
READ
IF A->NumPeri>0
  folder(06,09,20,69," Probabilidad de Falla ")
  @ 13.11 SAY 'No. de Periodo : '
  @ 15.11 SAY 'Probabilidad [%] : '
  Lc_nuva:=1
  SELECT 2
  SET ESCAPE ON
  DO WHILE Lc_nuva <= A->NumPeri .AND. LASTKEY() <> 27
    Lc_llav:=Lc_llav+StrZero(lc_nuva,3,0)
    SEK(Lc_llav)
    IF EOF()
      APPEND BLANK
      REPL B->NumMode WITH Lc_llav.;
      B->NumPeri WITH StrZero(Lc_nuva,3,0)
    ENDIF
    SETCOLOR(GIcolbri)
    @ 13.29 SAY 'LTRIM(STR(VAL(B->NumPeri),3,0))
    SETCOLOR(GIcolnor)
    @ 13.29 GET B->ProFall
  READ
  Lc_tech:=Lastkey()
```

MANTENIMIENTO DE EQUIPO INDUSTRIAL

```
DO CASE
CASE Lc_tec1=18
IF Lc_nuva > 1
SKIP -1
Lc_nuva--
ENDIF
OTHERWISE
SKIP
Lc_nuva++
ENDCASE
ENDDO
SET ESCAPE OFF
@ 03,01 clea to 23,78
```

Los datos de entrada que se capturan son el número de elementos que se van a analizar, el costo de ocurrir una falla, el costo de reemplazo y el número de periodos que se van a evaluar. Después para cada periodo se captura el porcentaje de probabilidad de falla.

Como se puede observar la ecuación que describe el número esperado acumulado de fallas es de naturaleza recursiva, similar a los modelos de redes, por lo que es necesario utilizar un procedimiento recursivo para poder implementarla en computadora. El procedimiento **ProbFall** realiza el cálculo del número esperado acumulado de fallas, en el se realiza una llamada a la función **Probab**, la cual se encarga de tomar de los archivos de la base de datos, la probabilidad de un periodo determinado y la acumula.

```
FUNCTION ProbFall
** Cálculo de probabilidad de fallas.
PARA Pr_unid, Pr_peri
LOCA Lc_prob, l
Lc_prob:=0.00
FOR l:=1 TO Pr_peri
Lc_prob:=Lc_prob+Probab(l)
NEXT
Lc_prob:=Lc_prob*Pr_unid
FOR l:=1 TO Pr_peri-1
Lc_prob:=Lc_prob+ProbFall(Pr_unid,Pr_peri-l)*Probab(l)
NEXT
RETURN (Lc_prob)
```

MANTENIMIENTO DE EQUIPO INDUSTRIAL

```
FUNCTION Probab
PARA Pr_name

SELE 2
SEEK(A->NumMode->StrZero(Pr_name.3.0))
SELE 1
RETURN (B->ProFall)
```

Los resultados del análisis son presentados mediante una simple llamada a la función **ProbFall** como se muestra a continuación:

```
@ 03.01 CLEA TO 23.78
folder (03.02.23.75." Solución ")
SETCOLOR(Gicolnor)
REN1:=8
SELE 2
SEEK(A->NumMode)
J:=1
DO WHILE .NOT. EOF() .AND. A->NumMode=B->NumMode
  IF REN1 >= 21
    @ 22.05 SAY SPACE(64)
    @ 22.05 SAY "Presione cualquier tecla para continuar..."
    INKEY(0)
    @ 08.03 CLEA TO 22.70
    REN1:=8
  ENDIF
  REGIS:=REC(N)
  @ REN1.20 SAY ProbFall(A->NumElem,J) PICT '999,999,999.99'
  J:=J+1
  REN1:=REN1+1
SELE 2
GOTO REGIS
SKIP
ENDDO
SELE 1
@ 22.05 SAY SPACE(64)
@ 22.05 SAY "Presione cualquier tecla para continuar..."
INKEY(0)
@ 22.05 SAY SPACE(65)
SETCOLOR(Gicolnor)
```

En el procedimiento anterior se presenta para cada periodo el número de fallas esperado, al igual que en los modelos anteriores de mantenimiento, los periodos pueden ser días, semanas, meses, años, etc.

3.5 MANTENIMIENTO CORRECTIVO.

Para determinar los costos de mantenimiento correctivo es muy útil usar el modelo de valor esperado. Este modelo requiere datos de la frecuencia y costo de reparaciones anteriores.

El modelo del valor esperado, sirve para determinar los costos esperados por las actividades de mantenimiento correctivo, suponiendo una frecuencia de reparaciones la cual puede ser tomada a partir de la información histórica. Este modelo en si lo que realiza es calcular el promedio de fallas por periodo a partir de las fallas total es del equipo.

Convirtiendo las frecuencias en una distribución de probabilidad y determinando el costo esperado por periodo de reparación, podemos obtener lo siguiente:

- Número de fallas (X)
 - ◆ Frecuencia en meses $F(X)$
 - ◆ Frecuencia en porcentajes $P(X) = F(X) / \Sigma F(X) * 100$

- Valor esperado $X * P(X)/100$

Los costos de reparación están determinados por

$$\text{Costo esperado} = (\Sigma P(X)) * (\text{Costo promedio de reparación})$$

El procedimiento inicial de captura de datos para el modelo de análisis de mantenimiento correctivo, se presenta a continuación:

```

@ 05.02 TO 21.77
SETCOLOR(G|colbn)
@ 05.16 SAY " Descripción del Modelo [Ctrl-W para Salir] "
SETCOLOR(G|colnor)
REPL A->DesMemo WITH MEMOEDIT(A->DesMemo,06,03,20,76,.T.,,120,3)
@ 05.02 CLEA TO 21.77
FOLDER(04,04,18,64," Modelo ")
@ 11.05 SAY "Descripción "
@ 13.05 SAY "Costo de reparación "
@ 15.05 SAY "Número de ocurrencias "
Lc_llav:=StrZero(Lc_numc,6,0)
SELE 1
SEEK (Lc_llav)
IF EOF()
  SELE 1
  APPEND BLANK
  REPL A->NumMode WITH Lc_llav
ENDIF
@ 11.30 GET A->DesGene Pict '@S34' VALID !EMPTY(A->DesGene)
@ 13.30 GET A->CosEape
@ 15.30 GET A->NumPeri
READ
IF A->NumPeri>0
  folder(06,09,20,69," Ocurrencias de Fallas ")
  @ 13.11 SAY "Ocurrencias:"
  @ 14.11 SAY "Número de fallas:"
  @ 15.11 SAY "Periodos que ocurre:"
  Lc_nuva:=1
  SELECT 2
  SET ESCAPE ON
  DO WHILE Lc_nuva <= A->NumPeri .AND. LASTKEY() <> 27
    Lc_ll1:=Lc_llav+StrZero(lc_nuva,3,0)
    SEEK(Lc_ll1)
    IF EOF()
      APPEND BLANK
      REPL B->NumMode WITH Lc_llav.;
      B->NumPeri WITH StrZero(Lc_nuva,3,0)
    ENDIF
    SETCOLOR(G|colbrn)
    @ 13.29 SAY LTRIM(STR(VAL(B->NumPeri),3,0))
    SETCOLOR(G|colnor)
    @ 14.29 GET B->NumFall
    @ 15.29 GET B->NumMese
  READ

```

```

Lc_tec1:=Lastkey()
DO CASE
CASE Lc_tec1=18
IF Lc_nuva > 1
SKIP -1
Lc_nuva--
ENDIF
OTHERWISE
SKIP
Lc_nuva++
ENDCASE
ENDDO
SET ESCAPE OFF
    
```

En el procedimiento inicial se capturan los datos generales del problema como son su descripción, el costo de reparación y el número de ocurrencias que se van a analizar; después para cada ocurrencia, se captura el número de fallas probables y el mes en que ocurren.

Una vez que se tienen los datos capturados es posible efectuar el análisis del valor esperado, por medio del siguiente procedimiento:

```

PROCEDURE VALESPE
PARA Pr_Mode

SELE 2
SEEK(Pr_Mode)
FreTota:=0.00
DO WHILE .NOT. EOF() .AND. Pr_Mode=B->NumMode
FreTota:=FreTota+B->NumMese
SKIP
ENDDO
SELE 2
SEEK(Pr_Mode)
IF FreTota<=0.00
DO WHILE .NOT. EOF() .AND. Pr_Mode=B->NumMode
REPL B->FrePorc WITH B->NumMese/FreTota.;
B->ValEspe WITH B->FrePorc*B->NumFall
SKIP
ENDDO
    
```

ENDIF
SELE 1
RETURN

El procedimiento anterior consiste de dos ciclos iterativos, que se repiten un número de veces igual al número de ocurrencias que tiene el problema. El primer ciclo calcula la frecuencia acumulada total del modelo. El segundo ciclo calcula, para cada ocurrencia, la frecuencia en porcentaje y el valor esperado.

Los resultados son presentados por medio del siguiente procedimiento:

```
@ 03.01 clea to 23,78
folder (03,02,23,75,"
SETCOLOR(Glcolnor)
@ 06.08 SAY 'Número de'
@ 06.25 SAY 'Frecuencia'
@ 06.45 SAY 'Frecuencia'
@ 06.67 SAY 'Valor'
@ 07.10 SAY 'fallas'
@ 07.24 SAY 'por periodo'
@ 07.43 SAY 'en porciento'
@ 07.64 SAY 'esperado'
@ 08.08 SAY REPL('-',10)
@ 08.24 SAY REPL('-',12)
@ 08.43 SAY REPL('-',12)
@ 08.62 SAY REPL('-',10)
VALOR:=0.00
REN1:=9
SELE 2
SEEK(Lc_llav)
DO WHILE .NOT. EOF() .AND. Lc_llav=B->NumMode
  IF REN1 >= 20
    @ 22.05 SAY SPACE(64)
    @ 22.05 SAY 'Presione cualquier tecla para continuar...'
    @ 22.05 SAY SPACE(64)
    INKEY(0)
    @ 09.05 CLEA TO 21,73
    REN1:=9
  ENDIF
  @ REN1,08 SAY B->NumFall
  @ REN1,26 SAY B->NumMese
  @ REN1,42 SAY ROUND(B->FrePorc*100,4) PICT '99999999.9999'
  @ REN1,59 SAY B->ValEspe PICT '99999999.9999'
  VALOR:=VALOR+B->ValEspe
  REN1:=REN1+1
```

```
SKIP
ENDDO
@ 20.05 CLEA TO 22.74
SETCOLOR(GIcolbr1)
LY:="Costo esperado $ "+LTRIM(TRANS(VALOR*A->CosEspe,'999,999,999,999'))+" por periodo."
@ 21.INT((80-LEN(LY))/2) SAY LY
SETCOLOR(GIcolnor)
SELE 1
@ 22.05 SAY SPACE(64)
@ 22.05 SAY "Presione cualquier tecla para continuar..."
INKEY(0)
@ 22.05 SAY SPACE(65)
SETCOLOR(GIcolnor)
```

El procedimiento de resultado presenta una tabla en la pantalla con los datos que a continuación se presentan, así como el costo esperado por periodo.

- Número de fallas.
- Frecuencia en meses de fallas.
- Frecuencia en porcentaje.
- Valor esperado

4. ANALISIS DE LINEAS DE ESPERA.

4.1 CONCEPTOS GENERALES.

El tener que esperar en una cola es una experiencia cotidiana que normalmente se considera desagradable. Esperar un elevador, ser servido en un restaurante o en la cola de un banco es una confrontación con la pérdida de tiempo. Si la espera es demasiado larga, las personas se vuelven irritables e inquietas: los temperamentos se ofuscan. Por supuesto, "demasiado larga" es relativo. Por ejemplo, la espera puede ser más larga si se está sentado (como en un restaurante) que si se está parado (como en una tienda de abarrotes o supermercado). Aun así, la paciencia tiene un límite.

Por otro lado, en la industria, el esperar significa estar inútil. Es desperdicio. Significa que algún recurso está inactivo cuando podría usarse en forma más productiva en otra parte. De hecho representa un costo de oportunidad. Cuando un centro de trabajo, que sea un recurso limitado del sistema, se encuentra esperando la liberación de material para poder empezar a trabajar, se pierde productividad; es dinero que se pierde y nunca podrá ser recuperado.

Es fácil observar que el proporcionar suficiente capacidad de servicio para eliminar la espera sería muy costoso. Cuántos cajeros serían necesarios en un banco o cuántas máquinas en una nave industrial para eliminar todas las colas. (Aún si esto fuera posible, todavía se tendría que esperar mientras se proporciona el servicio). Es claro que se necesita algún tipo de balance o compromiso para que el tiempo de espera no sea muy largo y el costo de servicio no sea muy alto.

El problema de los gerentes consiste en determinar qué capacidad o tasa de servicio proporciona el balance apropiado. Este sería un problema sencillo, si cada entrada al sistema llegara de acuerdo a un horario fijo y si el tiempo de servicio también fuera fijo. Como en una línea de ensamble, se podría balancear con exactitud la capacidad de servicio con las llegadas. Cualquier capacidad extra sería un desperdicio, menos

capacidad significaría que algunas llegadas no se atenderían. Sin embargo, en muchas situaciones ni el tiempo de llegada ni el tiempo de servicio son predecibles. Los sistemas de líneas de espera son sistemas probabilistas o aleatorios.

Con experiencia y sentido común, muchos gerentes encuentran un balance aproximado entre los costos de espera y de servicio sin elaborar ningún cálculo. No obstante, hay ocasiones en las que la intuición necesita ayuda, como cuando va de por medio una inversión substancial de capital o cuando el balance no es evidente. El análisis cuantitativo con frecuencia es útil en estas situaciones.

En el análisis de líneas de espera se aplica la teoría de colas. Una cola es una línea de espera y la teoría de colas es una colección de modelos matemáticos que describen sistemas de líneas de espera particulares o sistemas de colas. Los modelos sirven para encontrar el comportamiento en estado estable, como la longitud promedio de la línea y el tiempo de espera promedio para un sistema dado. Esta información, junto con los costos pertinentes, se usa entonces, para determinar la capacidad de servicio apropiado.

En la sección que sigue se describe la terminología de la teoría de colas y los tipos de sistemas de líneas de espera que se pueden describir con los modelos de análisis para después, en las dos secciones restantes del capítulo, describir los programas de computadora en donde se implementan los dos modelos más importantes en el estudio de líneas de espera.

4.2 TEORÍA DE COLAS.

Un sistema de líneas de espera puede dividirse en sus dos componentes de mayor importancia, la línea de espera y la instalación de servicio. Las *llegadas* son las unidades que entran en el sistema para recibir el servicio. Siempre se unen primero a la línea de espera; si no hay línea de espera se dice que la línea está vacía. De la línea, las llegadas van a la instalación de servicio de acuerdo con la *disciplina* de la línea. Una vez que se completa el servicio, las llegadas se convierten en *salidas*.

Las líneas de espera (o colas) ocurren siempre que los clientes que llegan buscan servicio en una o más estaciones (servidores), cada una de las cuales realiza una o más tareas. La teoría de colas nos brinda un conjunto de modelos, que sirven para describir una gran cantidad de situaciones donde ocurren líneas de espera. Varias suposiciones pueden hacerse en relación con la tasa de llegada de los clientes, el orden de servicio y la distribución de servicio para su análisis. Así mismo se requiere suponer que se ha alcanzado su *estado estable* en el que las características de la línea ya no cambian con el tiempo.

Los resultados que se buscan al aplicar la teoría de colas en el análisis de líneas de espera son principalmente: 1) la probabilidad de ser demorados, 2) el número promedio de clientes que estarán esperando y 3) el tiempo promedio de espera. Algunas veces disponemos de resultados adicionales que son útiles. Todos estos nos puede entonces servir como insumos para análisis económicos, permitiéndonos preparar las instalaciones de servicio óptimas.

A continuación se describe detalladamente cada uno de los elementos que forman parte de un sistema de líneas de espera, así como su notación:

Proceso de llegada. El proceso de entrada se conoce, por lo general, como el proceso de llegada. La llegada son los clientes. En todos los modelos que así se analizan se supone que sólo hay una llegada en un instante dado. En general suponemos que el proceso de llegada no es afectado por el número de clientes

presentes en el sistema, así el proceso de llegada lo expresamos normalmente especificando una distribución de probabilidad que gobierna el tiempo de llegadas sucesivas.

Una variable aleatoria X tiene **distribución exponencial** con parámetro λ si la densidad de X está definida por

$$f(t) = \lambda e^{-\lambda t} \quad (t \geq 0)$$

Entonces

$$E(X) = \frac{1}{\lambda} \quad \text{y} \quad \text{var } X = \frac{1}{\lambda^2}$$

La distribución exponencial tiene la propiedad de amnesia. Esto quiere decir, por ejemplo, que si los tiempos entre llegadas tienen distribución exponencial con rapidez, frecuencia o parámetro λ entonces sin importar cuánto tiempo ha pasado desde la última llegada, hay una probabilidad $\lambda \Delta t$ de que se tenga una llegada durante los siguientes Δt unidades de tiempo.

Los tiempos entre llegadas son exponenciales con parámetro λ si y sólo si el número de llegadas que se tienen en un intervalo de duración t sigue una **distribución de Poisson** con parámetros λt . La función para una distribución de Poisson con parámetro λ está representado por

$$P(N=n) = \frac{e^{-\lambda} \lambda^n}{n!} \quad (n = 0, 1, 2, \dots)$$

Proceso de salida o de servicio. Para describir el proceso de salida, que con frecuencia se llama proceso de servicio, de un sistema de líneas de espera, en general especificamos una distribución de probabilidad: la distribución del tiempo de servicio, que gobierna el tiempo de servicio a un cliente. En la

mayor parte de los casos suponemos que la distribución del tiempo de servicio, es independiente del número de clientes presentes en el sistema.

Número de servidores en paralelo. Es el número de estaciones que proporcionan servicio a los clientes de manera simultánea.

Disciplina de la línea de espera. La disciplina de la línea de espera es el método que se usa para determinar el orden en el que se sirve a los clientes. La disciplina más común es la disciplina PLPS (Primero en Llegar es el Primero en ser Servido), en el que los clientes son servidos en el orden de su llegada. Bajo la disciplina ULPS (Último en Llegar, Primero en ser Servido), las llegadas más recientes son la primeras en recibir el servicio. A veces, el orden en el que llegan los clientes no tiene efecto alguno sobre el orden en que se les sirve. Ese sería el caso en el que el siguiente cliente en ser atendido se selecciona al azar entre los que están esperando para ser atendidos. A este caso se le llama SEOA (Servicio En Orden Aleatorio).

Tamaño de la línea de espera. Es el número máximo de clientes que puede esperar en el sistema, incluyendo en espera y en servicio.

Población. Es la cantidad de clientes posibles que pueden ingresar al sistema de líneas de espera.

Para representar los sistemas de líneas de espera existe una notación, denominada notación de Kendall-Lee (en honor a su autor), en la cual un sistema de líneas de espera se representa por seis características:

1/2/3/4/5/6

La primera característica especifica la naturaleza del proceso de llegada. Se usan las siguientes abreviaturas normales:

- M.- Los tiempos de llegada son independientes, distribuidos idénticamente, y las variables aleatorias tienen distribución exponencial.
- D.- Los tiempos de llegada son idénticamente distribuidos y deterministas.

E_k - Los tiempos de llegada son idénticamente distribuidos con distribución de Erlang con parámetro de forma k .

GI - Los tiempos de llegada son idénticamente distribuidos y están gobernados por alguna distribución general.

La segunda característica especifica la naturaleza de los tiempos de servicio:

M - Los tiempos de servicio son idénticamente distribuidos y tienen distribución exponencial.

D - Los tiempos de servicio son idénticamente distribuidos y deterministas.

E_k - Los tiempos de servicio son idénticamente distribuidos y con distribución de Erlang con parámetro de forma k .

GI - Los tiempos de servicio son idénticamente distribuidos y siguen alguna distribución general.

La tercera característica es el número de servidores en paralelo. La cuarta característica describe la disciplina de la línea de espera:

PLPS.- Primero en llegar, primero en ser servido.

ULPS.- Ultimo en llegar, primero en ser servido.

SEOA.- Servicio en orden aleatorio.

DG.- Disciplina general en la línea.

La quinta característica especifica el número máximo permisible de clientes en el sistema, incluyendo los que esperan y los que están siendo atendidos. La sexta característica da el tamaño de la población de la cual se toman los clientes. A menos que el número de clientes potenciales sea del mismo orden de magnitud que el número de servidores, se considera que es infinito el tamaño de la población. En muchos modelos importantes, las características 4/5/6 son $DG/\infty/\infty$. Si éste es el caso, entonces con frecuencia se omite 4/5/6.

En general, existen varios arreglos de sistemas de líneas de espera tal como, un sistema de un servidor y una línea que puede describir un lavado de carros automático o un muelle de descarga de un sólo lugar; un sistema con una línea y múltiples servidores, es típico en industrias donde las operaciones se realizan en varios centros de trabajo similares y los clientes toman un número al entrar y se le sirve cuando les llega el turno; otro servidor es aquél en el que cada servidor tiene una línea separada, es característico de los bancos y tiendas de autoservicio. Para este tipo de servicio pueden separarse los servidores y tratarlos como sistemas independientes de un servidor y una línea. Esto sería válido sólo si hubiera muy pocos intercambios en la línea; otro sistema es el de servidores en serie, que puede describir entre otras cosas la línea de ensamble de una fábrica. De nuevo, para propósitos de análisis, es posible separar los subsistemas y usar el modelo de un servidor y una línea.

Si se quiere y si se pueden estimar ciertos costos, es posible de construir modelos de líneas de espera con costos esperados, los costos que se requieren son:

Cs - costo por hora de tener un servidor disponible.

Cw - costos por hora de tener un cliente esperando en el sistema.

Cuando el costo unitario de espera es medible, como el caso de camiones en el muelle de carga y descarga, los cálculos son directos. Como el costo casi siempre es proporcional al tiempo de espera, el costo total puede expresarse como el costo de espera por hora multiplicado por la longitud promedio de la línea (L):

$$\text{Costo total de espera} = Cw L$$

4.3 MODELO DE UNA FILA Y UN SERVIDOR: M/M/1/PLPS/∞/∞

Este modelo puede aplicarse a personas esperando en una línea para comprar boletos, a mecánicos que esperan obtener herramientas de un expendio o a trabajos de computadora que esperan tiempo de procesador. Es uno de los modelos más antiguos, más sencillos y más comunes en la teoría de colas. A continuación se analizarán las suposiciones necesarias de este modelo y después se presentarán los programas que implementan en computadora al modelo.

Llegadas. Se supone que las llegadas entran al sistema de manera completamente aleatoria. No tienen horario, es impredecible en que momento llegarán. De una manera más formal, esto significa que la probabilidad de una llegada en cualquier instante de tiempo es la misma que en cualquier otro momento. Esto equivale a afirmar que el número de llegadas por unidad de tiempo tiene una distribución Poisson. La suposición de llegadas aleatorias es válido para una infinidad de sistemas reales.

El modelo también supone que las llegadas vienen de una población infinita y llegan una a la vez. Siempre que no falten llegadas, es decir, se acaben, puede considerarse que su fuente es infinita. No se permiten llegadas simultáneas, ya que causarían múltiples líneas y este es un modelo de una sola línea.

Cola. En este modelo se considera que el tamaño de la cola es infinito. Es cierto que todas las colas tienen límites en el tamaño, pero si este límite no desanima o evita las llegadas, puede ignorarse. La disciplina de la cola es primero en llegar, primero en ser servido sin prioridades especiales. También se supone que las llegadas no pueden cambiar de lugares en la línea o dejar la cola antes de ser servidas.

Instalación de servicio. Se supone que un solo servidor proporciona el servicio que varía aleatoriamente. En particular, el tiempo de servicio sigue una distribución exponencial. De hecho, esto se deriva de la suposición que las salidas son completamente aleatorias, la misma suposición que se usó para las llegadas.

Salidas. No se permite que las unidades que salen vuelvan a entrar de inmediato al sistema. Si bien esto sucede en ocasiones en los sistemas reales, es muy raro. Si sucediera con frecuencia, afectaría la distribución de las llegadas.

Características de operación. Las características de operación son las medidas de lo bien que funciona el sistema. En la mayoría de las aplicaciones de líneas de espera, el estado estable es de primera importancia. Los estados transitorios, como el de echar a andar y apagar el sistema, no se analizan. De las suposiciones anteriores, la característica de operación en estado estable pueden derivarse llegando a los siguientes resultados:

COLA

$$\text{Longitud promedio de la línea (Lq)} = \frac{\lambda^2}{\mu(\mu - \lambda)}$$

$$\text{Tiempo de espera promedio (Wq)} = \frac{Lq}{\lambda} = \frac{\lambda}{\mu(\mu - \lambda)}$$

SISTEMA

$$\text{Longitud promedio de la línea (Ls)} = Lq + \frac{\lambda}{\mu} = \frac{\lambda}{\mu - \lambda}$$

$$\text{Tiempo de espera promedio (Ws)} = \frac{Ls}{\lambda} = \frac{1}{\mu - \lambda}$$

$$\text{Intensidad de tráfico } \rho = \frac{\lambda}{\mu}$$

$$\text{Probabilidad de que la línea excede a } n: P(L_s > n) = \left[\frac{\lambda}{\mu} \right]^{n+1}$$

donde λ es la tasa promedio de llegadas (llegadas por unidad de tiempo) y μ es la tasa promedio de servicio (llegadas por unidad de tiempo).

Para la implementación en computadora se debe comenzar con el procedimiento de entrada de datos en los que se describe cada uno de los modelos para poder ser analizados; a continuación se muestra el programa en donde se capturan estos datos:

```
LC_OTRO='S'
DO WHILE LC_OTRO='S'
  CLS
  @ 00.00 TO 24.79 DOUBLE
  SETCOLOR(glcobri)
  @ 01.20 SAY 'LINEAS DE ESPERA (UNA FILA-UN SERVIDOR)'
  SETCOLOR(glcobnor)
  @ 02.01 SAY REPL('-',78)
  SELECT 1
  MODELOS:=ARRAY(RECC()+1)
  MODELOS[1]='NUEVO'
  ULTMODE:=""
  CONT:=2
  GO TOP
  DO WHILE .NOT. EOF()
    MODELOS[CONT]:=A->DesGene
    ULTMODE:=A->NumMode
    CONT:=CONT+1
    SKIP
  ENDDO
  RENPAN := IIF(CONT > 15, 15, CONT)
  RENINIC := INT((24 - RENPAN) / 2) + 1
  RENFINA := RENINIC + RENPAN
  @ RENINIC, 19 TO RENFINA, 59
```

ANALISIS DE LINEAS DE ESPERA

```

SETCOLOR(Glcobfbi)
@ RENINIC.28 Say 'Selección del Modelo '
SETCOLOR(Glcoblnor)
SET KEY 19 TO NADA()
SET KEY 04 TO NADA()
MODEL:=ACHOICE(RENINIC+1,20,RENFINA-1,58,MODELOS)
SET KEY 04 TO
SET KEY 19 TO
GO TOP
@ RENINIC.19 CLEA TO RENFINA,59
IF MODEL==1
  Lc_num:=VAL(ULTMODE)-1
ELSEIF MODEL==0
  CLOSE DATA
  RESTSCREEN(00,00,24,79,lc_pant)
  RETURN
ELSE
  SKIP MODEL-2
  Lc_num:=VAL(A->NumMode)
ENDIF
Lc_llav:=StrZero(Lc_num,6,0)
SELE 1
SEEK(Lc_llav)
IF EOF()
  SELE 1
  APPE BLANK
  REPL A->NumMode WITH Lc_llav;
  A->TipLine WITH '1'
ENDIF
@ 05.02 TO 21.77
SETCOLOR(Glcobfbi)
@ 05.16 SAY ' Descripción del Modelo [Ctrl-W para Salir] '
SETCOLOR(Glcoblnor)
REPL A->DesMemo WITH MEMOEDIT(A->DesMemo.06.03.20,76,.,T...,120,3)
@ 05.02 CLEA TO 21.77
@ 04.04,18.64," Modelo "
@ 10.05 SAY 'Descripción : '
@ 12.05 SAY 'Tasa de llegada : '
@ 14.05 SAY 'Tasa de servicio : '
@ 10.23 GET A->DesGene PICT '@S40' VALID !Empty(A->DesGene)
@ 12.23 GET A->TasLleg
@ 14.23 GET A->TasServ
READ

```

El programa inicia con la captura de los datos que describen al modelo, así como también permite seleccionar el modelo en caso de que ya exista almacenado en la base de datos. Los datos generales del modelo son la descripción general, la tasa de llegada en porcentaje y la tasa de servicio también en

porcentaje. Este mismo procedimiento crea los registros necesarios en la base de datos, cuando el modelo es introducido por primera vez y almacena los datos capturados que describen al problema.

El procedimiento que realiza el análisis del modelo de la línea de espera para una fila y un servidor es el siguiente:

```

FUNCTION Linespus
PARA nModelo
LOCA LcResul

LcResul:=F.
SELE 1
SEEK(nModelo)
IF !EOF()
REPL A->IntTraf WITH IIF(A->TasServ==0,0,A->TasLleg/A->TasServ)
IF A->IntTraf<=0 .OR. A->IntTraf>= 1
RETURN (LcResul)
ELSE
LcResul= T
REPL A->ProDeso WITH 1-A->IntTraf.;
A->EleSist WITH (A->IntTraf)/(1-A->IntTraf).;
A->EleLine WITH (A->IntTraf^2)/(1-A->IntTraf).;
A->EleServ WITH IntTraf.;
A->TieSist WITH A->EleSist / A->TasLleg.;
A->TieLine WITH A->EleLine / A->TasLleg.;
A->TieServ WITH A->EleServ / A->TasLleg
ENDIF
ENDIF
RETURN (LcResul)

```

El procedimiento comienza por calcular la *intensidad de tráfico* en la línea, este se calcula dividiendo la tasa de llegada entre la tasa de servicio:

$$\text{Intensidad de tráfico} = \text{Tasa de llegada} + \text{Tasa de servicio}$$

Con la intensidad de tráfico es posible determinar si el sistema se encuentra en estado estable, esto es si la intensidad de tráfico es mayor o igual que uno el sistema "explota"; para el caso contrario el sistema se

comporta de manera estable a lo largo del tiempo y el análisis se puede realizar utilizando las ecuaciones anteriormente mostradas.

La información que proporciona es la siguiente:

- Número de elementos en el sistema.
- Número de elementos en la línea de espera.
- Número de elementos en servicio.
- Tiempo promedio en el sistema.
- Tiempo promedio en espera.
- Tiempo promedio en servicio.

El procedimiento que a continuación se presenta sirve para mostrar los resultados, tanto en pantalla como en impresora, según se requiera:

```

folder(06.09.20.69," Solución      ")
@ 12.12 SAY 'Servidor desocupado:'
@ 13.12 SAY 'Intensidad de tráfico:'
@ 14.12 SAY 'Elementos en el sistema:'
@ 15.12 SAY 'Elementos en espera:'
@ 16.12 SAY 'Elementos en servicio:'
@ 17.12 SAY 'Tiempo promedio en el sistema:'
@ 18.12 SAY 'Tiempo promedio en espera:'
@ 19.12 SAY 'Tiempo promedio en servicio:'
SETCOLOR(glcobri)
LEYE:=ALLTRIM(A->DesGene)
COLL:=9+INT((60-LEN(LEYE))/2)
@ 10,COLL SAY LEYE
@ 12,33 SAY LTRIM(TRANS(A->ProDeso*100,'999,999,999.99'))
@ 13,35 SAY LTRIM(TRANS(A->IntTra*100,'999,999,999.99'))
@ 14,37 SAY LTRIM(TRANS(A->EleSist,'999,999,999.999'))
@ 15,33 SAY LTRIM(TRANS(A->EleLine,'999,999,999.999'))
@ 16,35 SAY LTRIM(TRANS(A->EleServ,'999,999,999.999'))
@ 17,43 SAY LTRIM(TRANS(A->TieSist,'999,999,999.999'))
@ 18,39 SAY LTRIM(TRANS(A->TieLine,'999,999,999.999'))
@ 19,41 SAY LTRIM(TRANS(A->TieServ,'999,999,999.999'))
SETCOLOR(glcobri)

```

ANALISIS DE LINEAS DE ESPERA

```

@ 12,33+LEN(LTRIM(TRANS(A->ProDeso*100,999,999,999,99)))=1 SAY '%'
@ 13,35+LEN(LTRIM(TRANS(A->IntTra*100,999,999,999,99)))=1 SAY '%'
@ 14,37+LEN(LTRIM(TRANS(A->EleSist,999,999,999,999)))=1 SAY A->EleSist
@ 15,33+LEN(LTRIM(TRANS(A->EleLine,999,999,999,999)))=1 SAY A->EleLine
@ 16,35+LEN(LTRIM(TRANS(A->EleServ,999,999,999,999)))=1 SAY A->EleServ
@ 17,43+LEN(LTRIM(TRANS(A->TieSist,999,999,999,999)))=1 SAY A->TieSist
@ 18,39+LEN(LTRIM(TRANS(A->TieLine,999,999,999,999)))=1 SAY A->TieLine
@ 19,41+LEN(LTRIM(TRANS(A->TieServ,999,999,999,999)))=1 SAY A->TieServ
inkey(0)
IF ISPRINTER()
  LcPant:=SaveScreen(12,20,16,59)
  @ 12,20 CLEA TO 16,59
  @ 12,20 TO 16,59 DOUBLE
  @ 14,21 SAY '¿Desea imprimir los resultados (S/N)?'
  LcResp:=''
  DO WHILE AT(LcResp,'SN')=0
    LcResp:=UPPER(CHR(INKEY(0)))
  ENDDO
  IF LcResp='S'
    @ 13,21 CLEA TO 14,58
    SETCOLOR(glcobln)
    @ 14,27 SAY 'Imprimiendo resultados ...'
    SETCOLOR(glcoblnor)
    SET DEVI TO PRINT
    SET CONSOL OFF
    SET PRINT ON
    @ 05,01 SAY 'SISTEMA DE OPTIMIZACION DE RECURSOS'
    @ 05,19 SAY 'VERSION 1.0'
    @ 06,01 SAY 'TRANFECHA(DATE())'
    @ 07,01 SAY 'LINEAS DE ESPERA (UNA FILA - UN SERVIDOR)'
    @ 09,01 SAY REPL(' ',130)
    LEVE:=ALLTRIM(A->DesGene)
    @ 12,12 SAY LEVE
    @ 15,12 SAY 'Tasa de llegada:'
    @ 15,29 SAY LTRIM(TRANS(A->TasLeg,999,999,999,99))
    @ 15,29 SAY LTRIM(TRANS(A->TasLeg,999,999,999,99))
    @ 16,12 SAY 'Tasa de servicio:'
    @ 16,30 SAY LTRIM(TRANS(A->TasServ,999,999,999,99))
    @ 16,30 SAY LTRIM(TRANS(A->TasServ,999,999,999,99))
    @ 17,12 SAY 'Servidor desocupado:'
    @ 17,33 SAY LTRIM(TRANS(A->ProDeso*100,999,999,999,99))
    @ 17,33 SAY LTRIM(TRANS(A->ProDeso*100,999,999,999,99))
    @ 17,33+LEN(LTRIM(TRANS(A->ProDeso*100,999,999,999,99)))=1 SAY '%'
    @ 18,12 SAY 'Intensidad de tráfico:'
    @ 18,35 SAY LTRIM(TRANS(A->IntTra*100,999,999,999,99))
    @ 18,35 SAY LTRIM(TRANS(A->IntTra*100,999,999,999,99))
    @ 18,35+LEN(LTRIM(TRANS(A->IntTra*100,999,999,999,99)))=1 SAY '%'
    @ 19,12 SAY 'Elementos en el sistema:'
    @ 19,37 SAY LTRIM(TRANS(A->EleSist,999,999,999,999))
    @ 19,37 SAY LTRIM(TRANS(A->EleSist,999,999,999,999))
    @ 20,12 SAY 'Elementos en espera:'
    @ 20,33 SAY LTRIM(TRANS(A->EleLine,999,999,999,999))
    @ 20,33 SAY LTRIM(TRANS(A->EleLine,999,999,999,999))
    @ 21,12 SAY 'Elementos en servicio:'
    @ 21,35 SAY LTRIM(TRANS(A->EleServ,999,999,999,999))

```

ANALISIS DE LINEAS DE ESPERA

```
@ 21,35 SAY LTRIM(TRANS(A->EleServ,999,999,999,999'))
@ 22,12 SAY 'Tiempo promedio en el sistema:'
@ 22,43 SAY LTRIM(TRANS(A->TieSist,999,999,999,999'))
@ 22,43 SAY LTRIM(TRANS(A->TieSist,999,999,999,999'))
@ 23,12 SAY 'Tiempo promedio en espera:'
@ 23,39 SAY LTRIM(TRANS(A->TieLine,999,999,999,999'))
@ 23,39 SAY LTRIM(TRANS(A->TieLine,999,999,999,999'))
@ 24,12 SAY 'Tiempo promedio en servicio:'
@ 24,41 SAY LTRIM(TRANS(A->TieServ,999,999,999,999'))
@ 24,41 SAY LTRIM(TRANS(A->TieServ,999,999,999,999'))
SET PRINT OFF
SET CONSOL OFF
SET DEVI TO SCREEN
ENDIF
ENDIF
ELSE
  Mensaje('Sistema inestable.')
ENDIF
Lc_Pan2:=SAVESCREEN(10,19,14,61)
@ 10,19 CLEA TO 14,59
@ 10,19 TO 14,59 DOUBLE
@ 12,22 SAY '¿Desea resolver otro modelo (S/N)?'
Lc_Otro:''
DO WHILE AT(Lc_Otro,'SN')==0
  Lc_Otro:=UPPER(CHR(INKEY(0)))
ENDDO
RESTSCREEN(10,19,14,61,Lc_Pan2)
ENDDO
RESTSCREEN(00,00,24,79,lc_pant)
RETURN
```

El procedimiento anterior muestra en pantalla los resultados obtenidos por el procedimiento de análisis de líneas de espera de un servidor y una fila, y en caso de que se encuentra encendida la impresora se le pregunta al usuario si se requiere que los resultados sean impresos. En casos de que se acepte que los resultados sean impresos, se configura la impresora y se envían los resultados para poder ser impresos en un formato en donde se describan las características del modelo. En caso de que la función de análisis detectara que el modelo pertenece a un sistema inestable, el procedimiento anterior, también se encarga de avisarle al usuario.

De esta manera queda implementado el modelo de líneas de espera para una fila y un servidor.

4.4 MODELO DE UNA FILA Y SERVIDORES EN PARALELO: $M/M/S/PLPS/\infty/\infty$

En muchas situaciones reales habrá más de un servidor disponible para atender las llegadas. Para este caso existe un modelo general para un sistema de múltiples servidores que tienen una sola línea.

A continuación se analizarán las suposiciones necesarias de este modelo:

Llegadas. Se supone que las llegadas entran al sistema de manera completamente aleatoria. Al igual que en el modelo anterior, esto equivale a afirmar que el número de llegadas por unidad de tiempo tiene una distribución Poisson. El modelo también supone que las llegadas vienen de una población infinita y llegan una a la vez. Siempre que no falten llegadas, es decir, se acaben, puede considerarse que su fuente es infinita.

Cola. En este modelo se considera que hay una sola cola y su tamaño es infinito. La disciplina de la cola es primero en llegar, primero en ser servido sin prioridades especiales.

Instalación de servicio. En este modelo se considera que existen varios servidores que proporcionan el servicio que varía aleatoriamente. En particular, el tiempo de servicio sigue una distribución exponencial. De hecho, esto se deriva de la suposición que las salidas son completamente aleatorias, la misma suposición que se usó para las llegadas.

Salidas. No se permite que las unidades que salen vuelvan a entrar de inmediato al sistema. Si bien esto sucede en ocasiones en los sistemas reales, es muy raro. Si sucediera con frecuencia, afectaría la distribución de las llegadas.

Características de operación. Las ecuaciones de las características de operación se vuelven un poco más complicadas:

Sean

s .- Número de servidores.

λ .- Tasa promedio de llegada (llegada por unidad de tiempo).

μ .- Tasa promedio de servicio para cada servidor.

Entonces se tienen las siguientes ecuaciones:

Intensidad de tráfico:

$$\rho = \frac{\lambda}{s\mu}$$

Probabilidad de estado estable ($\rho < 1$):

$$\pi_0 = \frac{1}{\sum_{i=0}^{s-1} \frac{(s\lambda)^i}{i!} + \frac{(s\lambda)^s}{s! (1-\rho)}}$$

La probabilidad de estado estable de que todas los servidores estén ocupados:

$$p(j \geq s) = \frac{(s\lambda)^s \pi_0}{s! (1-\rho)}$$

La longitud de la línea de espera:

$$Lq = \frac{\rho(j \geq s) \rho}{1 - \rho}$$

Tiempo promedio que pasa un elemento en la línea:

$$Wq = \frac{Lq}{\lambda}$$

Número de elementos presentes en el sistema:

$$L = Lq + \frac{\lambda}{\mu}$$

Tiempo promedio que pasa un elemento en el sistema:

$$W = \frac{L}{\lambda}$$

Tiempo y elementos en servicio:

$$Ls = L - Lq$$

$$Ws = W - Wq$$

A partir de las fórmulas que describen las características de operación de un sistema de líneas de espera con una fila y múltiples servidores en paralelo, es posible implementarlo en computadora a través de un programa que efectúe dichos cálculos.

El programa debe iniciar con un procedimiento de captura de datos semejante al sistema de una fila y un servidor, sólo que debe solicitar además de la tasa de llegada y la tasa de servicio, el número de servidores presentes en el sistema.

A continuación se muestran los procedimientos que se utilizan en la implementación de las fórmulas que describen las características de operación del modelo de una fila y servidores en paralelo:

```

FUNCTION Lespvaser
PARA nModelo
LOCAL LcResul

LcResul:=F.
SELE I
SEEK(nModelo)
IF !EOF()
  REPL A->IntTraf WITH A->TasLleg.(A->TasServ*A->NumServ)
  IF A->IntTraf > 0 .AND. A->IntTraf < 1
    LcResul:=T.
    REPL A->TieLine WITH (Probabilidad(A->IntTraf,A->NumServ))/(A->NumServ*A->TasServ-A->TasLleg.);
    A->EleLine WITH A->TieLine * A->TasLleg.;
    A->EleSist WITH A->EleLine + (A->TasLleg/A->TasServ);
    A->TieSist WITH A->EleSist / A->TasLleg.;
    A->EleServ WITH A->EleSist - A->EleLine.;
    A->TieServ WITH A->TieSist - A->TieLine
  ELSE
    Mensaje(' Sistema inestable. ')
  ENDF
ENDIF
RETURN (LcResul)

```

El procedimiento tipo función **Lespvaser** tiene como objetivo tomar las tasas que describen al modelo y el número de servidores, de la base de datos, y como primer paso calcular la intensidad de tráfico, para determinar si el sistema se encuentra en estado estable.

Una vez que se determinó que el sistema se encuentra en estado estable, se continúa con el cálculo del tiempo de un elemento en la línea de espera, para ello es necesario ejecutar el procedimiento **Probabilidad**, el cual se encarga de calcular la probabilidad de que todos los servidores estén ocupados. Y una vez calculado, se procede a calcular los tiempos de servicio y total del sistema, así como los elementos que permanecen en espera, en servicio y en todo el sistema.

```
FUNCTION Probabilidad
PARA RoNumb, SeNumb
LOCA LcResu

  LcResu:=0.00
  IF RoNumb < 1 .AND. SeNumb > 0
    LcResu:=((RoNumb*SeNumb)^SeNumb * P10(RoNumb,SeNumb)) / ;
      (Factorial(SeNumb) * (1 - RoNumb))
  ENDIF
RETURN (LcResu)
```

El procedimiento tipo función **Probabilidad**, se encarga de calcular la probabilidad de que todos los servidores estén ocupados, para ello es necesario calcular primero la probabilidad de que el sistema se encuentre en estado estable, para lo que usamos otro procedimiento tipo función, llamado **P10**, así mismo es necesario contar con una función que calcule el factorial de un número cualquiera, ya que se requiere en el cálculo, para ello se utiliza la función **Factorial**.

```
FUNCTION P10
PARA RoNumb, SeNumb
LOCA LcResu, LcResp, I

  LcResp:=0.00
  FOR I:=0 TO (SeNumb-1)
    LcResp:=LcResp + ((RoNumb*SeNumb)^I) / Factorial(I)
  NEXT
  LcResu:=1/(LcResp+(((SeNumb*RoNumb)^SeNumb)/(Factorial(SeNumb)*(1-RoNumb))))
RETURN LcResu
```

El procedimiento tipo función **P10** consiste de un ciclo iterativo que se ejecuta una cantidad de veces igual al número de servidores presentes en el sistema menos uno. El ciclo acumula el producto de la intensidad de tráfico por el número de servidores elevado a una potencia igual al número de iteración en el que va, dividiendo todo esto por el factorial del número de la iteración en donde se encuentra. Una vez que se tiene esta sumatoria se procede a realizar el cociente que determina la probabilidad de que el sistema se

encuentre en estado estable. Como se puede observar, aquí también es necesario usar la función que determina el factorial de un número.

```
FUNCTION Factorial
PARA NumFact
LOCA LcResul, I

  LcResul:=1
  IF NumFact > 1
    FOR I:=1 TO NumFact
      LcResul:=LcResul*I
    NEXT
  ENDEF
RETURN (LcResul)
```

El procedimiento tipo función **Factorial** consiste en un ciclo iterativo que se realiza un número de veces igual al número al que se le quiere calcular su factorial. Para cada iteración, acumula el resultado de calcular el producto del valor que se tiene acumulado hasta ese momento por el número de iteración en la que se encuentra, dando como resultado, al terminó del ciclo, el factorial del número que se necesita.

Integrando los procedimientos, que se describieron con anterioridad, a los procedimientos de entrada y salida, similar al modelo de líneas de espera de una fila y un servidor, se obtiene la implementación completa del modelo de líneas de espera para poder analizar los sistemas de una fila y servidores en paralelo.

5. APLICACIÓN DEL SISTEMA EN CASOS PRACTICOS.

5.1 CONCEPTOS GENERALES.

En este capítulo se muestra un conjunto de problemas y sus soluciones obtenidas de manera impresa con el Sistema de Optimización de Recursos, el objetivo es mostrar de forma general, algunas de las situaciones en que se puede utilizar, así como también en mostrar como proporciona los resultados. Estos problemas forman sólo un pequeño grupo de una gran cantidad de situaciones en donde puede ser utilizado obteniendo grandes beneficios al optimizar las operaciones de la industria.

La forma en que se presentan los ejemplos es la siguiente: primero se presenta un problema de programación lineal para obtener la mezcla óptima de productos, después se presenta un problema que consiste en determinar un programa maestro de producción para optimizar el inventario de producto terminado, para resolverlo se utiliza el modelo de transporte, esto con la finalidad de mostrar que no sólo es utilizado este modelo en la transportación física de materiales, sino en todas aquellas situaciones en donde existan orígenes, destinos y cantidades a 'enviar'. A continuación se presenta un problema típico de asignaciones óptimas de máquinas a trabajos, utilizando para esto el modelo de asignación, concluyendo con esto los ejemplos de asignación y distribución de recursos.

Después se presentan dos problemas de planeación y control de proyectos, primero se presenta un ejemplo de manufactura sobre diseño, usando para esto el modelo de CPM y luego un ejemplo muy sencillo de la construcción de una planta industrial usando para esto el modelo de PERT.

Se presentan cuatro ejemplos de mantenimiento de equipo industrial, para mostrar como se pueden establecer programas de mantenimiento óptimos, en el equipo que comúnmente se encuentra en las plantas industriales, se presentan modelos de determinación de la vida económica del equipo, análisis de reemplazo, mantenimiento preventivo y mantenimiento correctivo.

Finalmente, se presentan tres ejemplos donde se pueden usar los modelos para el análisis de líneas de espera tanto de una fila y un servidor y como de una fila y servidores en paralelo..

5.2 MEZCLAS DE PRODUCTOS.

Las situaciones en las cuales se tienen que mezclar varias entradas (sustancias o componentes) en una proporción deseada para producir un artículo para su venta, son frecuentemente sujetos al análisis de la programación lineal: tales problemas se llaman *problemas de mezclas*. Este tipo de problemas se presenta en una gran cantidad de industrias tales como en las manufactureras, las químicas, las alimenticias, etc. A continuación se presenta un caso de este tipo de problemas y su solución obtenida con el Sistema de Optimización de Recursos. El problema consiste en establecer la mezcla óptima en la producción de gasolinas, sin embargo representa un modelo típico que puede generalizarse para poderlo aplicar a una gran gama de situaciones, pudiendo incrementar fácilmente el tipo y la cantidad de artículos que se producen, los componentes y las condiciones de producción.

La producción de tres tipos de gasolina, con código de producción GAS001, GAS002 y GAS003, se realiza mezclando tres tipos de petróleo crudo, con código de compra PTR001, PTR002 y PTR003. Los precios de venta por barril de las gasolinas son los siguientes:

GAS001	\$ 70.00	(dólares / barril)
GAS002	\$ 60.00	(dólares / barril)
GAS003	\$ 50.00	(dólares / barril)

Los precios de compra de petróleo crudo por barril son los siguientes:

PTR001	\$ 45.00	(dólares / barril)
PTR002	\$ 35.00	(dólares / barril)
PTR003	\$ 25.00	(dólares / barril)

Se pueden comprar hasta 5000 barriles de cada tipo de petróleo crudo diariamente.

Los tres tipos de gasolina difieren en su índice de octano y su contenido de azufre. La mezcla de petróleo que se utiliza para obtener la gasolina GAS001 tiene que tener un índice de octano promedio de por lo menos 10 y en lo más del 1% de azufre. La mezcla de petróleo crudo que se utiliza para obtener la GAS002 tiene que tener un índice de octano promedio de por lo menos 8 y a lo más 2 % de azufre. La mezcla de petróleo crudo para obtener la gasolina GAS003 tiene que tener un índice de octano de por lo menos 6 y a lo más 1 % de azufre. El índice de octano y el contenido de azufre de los tres tipo de petróleo se dan a continuación.

	INDICE DE OCTANO	CONTENIDO DE AZUFRE
PTR001	12	0.5 %
PTR002	6	2.0 %
PTR003	8	3.0 %

La demanda comprometida diaria de gasolina es la siguiente: GAS001 3000 barriles, GAS002 2000 barriles y GAS003 1000 barriles. Existe la posibilidad de estimular la demanda de los productos mediante uso de publicidad. Cada dólar invertido diariamente para cierto tipo de gasolina aumenta la demanda diaria de este tipo de gasolina en 10 barriles. La transformación de un barril de petróleo en un barril de gasolina cuesta 4 dólares y con la capacidad instalada se pueden producir diariamente, hasta 14,000 barriles de gasolina. El objetivo del problema consiste en maximizar la utilidad neta diaria.

Para resolver el problema con el Sistema de Optimización de Recursos es necesario expresar el problema en la forma general del modelo de programación lineal.

Para expresar el modelo de programación lineal, lo primero que se realiza es determinar las variables de decisión:

X_{ij} .- barriles del petróleo PTR00i que se usa diariamente para producir gasolina GAS00j.

a_i .- dólares gastados en la publicidad para la gasolina GAS00i.

De esta forma tenemos diariamente:

$$\text{Ingresos por venta} = 70 (X_{11} + X_{21} + X_{31}) + 60 (X_{12} + X_{22} + X_{32}) + 50 (X_{13} + X_{23} + X_{33})$$

$$\text{Costo de petróleo} = 45 (X_{11} + X_{12} + X_{13}) + 35 (X_{21} + X_{22} + X_{23}) + 25 (X_{31} + X_{32} + X_{33})$$

$$\text{Costo de producción} = 4 (X_{11} + X_{12} + X_{13} + X_{21} + X_{22} + X_{23} + X_{31} + X_{32} + X_{33})$$

$$\text{Costo de publicidad} = a_1 + a_2 + a_3$$

El objetivo es maximizar las utilidad diaria.

Utilidad diaria = Ingreso por venta - Costo de petróleo - Costo de producción - Costo de publicidad

$$Z = 21X_{11} + 11X_{12} + X_{13} + 31X_{21} + 21X_{22} + 11X_{23} + 41X_{31} + 31X_{32} + 21X_{33} - a_1 - a_2 - a_3$$

Las restricciones de la demanda las podemos expresar como:

$$X_{11} + X_{21} + X_{31} - 10 a_1 = 3000$$

$$X_{12} + X_{22} + X_{32} - 10 a_2 = 2000$$

$$X_{13} + X_{23} + X_{33} - 10 a_3 = 1000$$

La restricciones de compra las podemos expresar como:

$$X_{11} + X_{12} + X_{13} \leq 5000$$

$$X_{21} + X_{22} + X_{23} \leq 5000$$

$$X_{31} + X_{32} + X_{33} \leq 5000$$

La restricción de la cantidad máxima de producción quedaría así:

$$X_{11} + X_{21} + X_{31} + X_{12} + X_{22} + X_{32} + X_{13} + X_{23} + X_{33} \leq 14000$$

Para expresar las restricciones sobre el índice de octano, tenemos que determinar el índice de octano promedio de una mezcla de diferentes tipos de petróleo crudo. Suponemos que los índices de octano de diferentes crudos se mezclan linealmente por lo que podemos establecer:

$$\frac{\text{INDICE DE OCTANO TOTAL DE LA GASOLINA}}{\text{BARRILES DE GAS001 EN LA MEZCLA}} = \frac{12 X_{11} + 6X_{21} + 8X_{31}}{X_{11} + X_{21} + X_{31}} \geq 10$$

Por lo que podemos expresar la restricción de octano de GAS001 como:

$$2 X_{11} - 4 X_{21} - 2X_{31} \geq 0$$

De manera similar las restricciones de octano de GAS002 y GAS003 quedarían:

$$4 X_{12} - 2X_{22} \geq 0$$

$$6 X_{13} + 2X_{33} \geq 0$$

Esta última restricción no es necesario incluirla en el modelo, ya que por la definición de la programación lineal, se satisface que $X_{13} \geq 0$ y que $X_{33} \geq 0$, por los que la ecuación $6 X_{13} + 2 X_{33} \geq 0$ siempre se satisface.

De forma semejante podemos obtener las restricciones del contenido de azufre teniendo como resultado:

$$\begin{aligned} -0.005 X_{11} + 0.010 X_{21} + 0.020 X_{31} &\leq 0 \\ -0.015 X_{12} &+ 0.010 X_{32} \leq 0 \\ -0.005 X_{13} + 0.010 X_{23} + 0.020 X_{33} &\leq 0 \end{aligned}$$

Con esto se tiene un modelo de programación lineal de 12 variables y 12 restricciones de los tres tipos diferentes de restricciones que existen (mayor o igual, menor o igual e igual). El problema aún representa un modelo de modestas dimensiones de acuerdo con la gran cantidad de variables que hay que controlar en diferentes situaciones, sin embargo implica un poco más del trabajo de invertir una matriz de orden doce, el cual es verdaderamente difícil resolverlo y sin cometer errores de manera manual. El Sistema de Optimización de Recursos encuentra la solución en menos de un minuto (en una computadora a 100 Mhz).

Los datos capturados en la computadora consisten en el modelo que se desarrolló anteriormente, el cual consiste en la función objetivo y sus doce restricciones, con la posibilidad de incluir descripción por variable y tipo de unidad de medida, así como también para las restricciones.

El resultado impreso que se obtiene a través del Sistema es presentado a continuación:

SISTEMA DE OPTIMIZACION DE RECURSOS
01-ENE-97
PROGRAMACION LINEAL (ALGORITMO SIMPLEX)

VERSION 1.0
MAXIMIZACION

UTILIDAD NETA EN EL PROCESO DE PRODUCCION DE GASOLINAS

Producción de las gasolinas con código de producción GAS001, GAS002 y GAS003, por medio de la mezcla de petróleo crudo con código de compra PTR001, PTR002 y PTR003.

Características de las gasolinas:

Tipo	Indice de octano	Porcentaje de azufre	Precio de venta	Producción mínima
GAS001	10	1 %	\$ 70.00 barril	3000 barriles
GAS002	8	2 %	\$ 60.00 barril	2000 barriles
GAS003	6	1 %	\$ 50.00 barril	1000 barriles

Características del petróleo crudo:

Tipo	Indice de octano	Porcentaje de azufre	Precio de compra	Compra máxima al día
PTR001	12	0.5 %	\$ 45.00 barril	5000 barriles
PTR002	6	2.0 %	\$ 35.00 barril	5000 barriles
PTR002	8	3.0 %	\$ 25.00 barril	5000 barriles

Características del proceso de producción:

- Costo de producción por barril : \$ 4.00
- Producción diaria máxima: 14,000 barriles de gasolina.
- Publicidad: Por cada dólar se incrementan las ventas de un tipo de gasolina en 10 barriles.

Descripción	Cantidad	Unidad	Valor
SOLUCION			
PETROLEO PTR001 USADO EN GASOLINA GAS001	2,222.222	BARRILES	46,666.662
PETROLEO PTR001 USADO EN GASOLINA GAS002	2,111.111	BARRILES	23,222.221
PETROLEO PTR001 USADO EN GASOLINA GAS003		BARRILES	666.667
PETROLEO PTR002 USADO EN GASOLINA GAS001	444.444	BARRILES	13,777.764
PETROLEO PTR002 USADO EN GASOLINA GAS002	4,222.222	BARRILES	88,666.662
PETROLEO PTR002 USADO EN GASOLINA GAS003	333.333	BARRILES	3,666.663
PETROLEO PTR003 USADO EN GASOLINA GAS001	333.333	BARRILES	13,666.653
PETROLEO PTR003 USADO EN GASOLINA GAS002	3,166.667	BARRILES	98,166.677
GASTOS DE PUBLICIDAD EN GASOLINA GAS002	750.000	DOLARES	-750.000
			287,749.969

APLICACION DEL SISTEMA EN CASOS PRACTICOS

SISTEMA DE OPTIMIZACION DE RECURSOS
01-ENE-97
PROGRAMACION LINEAL (ALGORITMO SIMPLEX)

VERSION 1.0
MAXIMIZACION

Descripcion	Cantidad	Unidad	Valor
EXCEDENTES			
MINIMO OCTANAJE DE LA GASOLINA GAS001		INDICE	
CAPACIDAD MAXIMA DE COMPARA DE PETROLEO PTR03	1,500.000	BARRILES	
CAPACIDAD MAXIMA DE PRODUCCION	500.000	BARRILES	
MAXIMO CONTENIDO DE AZUFRE EN GASOLINA GAS003	0.000	PORCENTAJE	
PRECIOS SOMBRA			
PETROLEO PTR003 USADO EN GASOLINA GAS003	1	BARRILES (-)	0.000
GASTOS DE PUBLICIDAD EN GASOLINA GAS001	1	DOLARES (-)	209.000
GASTOS DE PUBLICIDAD EN GASOLINA GAS003	1	DOLARES (-)	409.000
MINIMO OCTANAJE DE GASOLINA GAS002	1	INDICE (-)	0.000
CAPACIDAD MAXIMA DE COMPRA DE PETROLEO P R001	1	BARRILES (-)	57.250
CAPACIDAD MAXIMA DE COMPRA DE PETROLEO PTR002	1	BARRILES (-)	20.900
MAXIMO CONTENIDO DE AZUFRE EN GASOLINA GAS001	1	PORCENTAJE (-)	3,090.000
MAXIMO CONTENIDO DE AZUFRE EN GASOLINA GAS002	1	PORCENTAJE (-)	3,090.000

Con los resultados obtenidos se concluye que se tendrían que producir 3000 barriles de gasolina GAS001 usando 2,222.222 barriles de crudo PTR001, 444.444 barriles de crudo PTR002, y 333.333 barriles de crudo PTR003; 9500 barriles de gasolina GAS002 usando 2,111.111 barriles de PTR001, 4,222.222 barriles de PTR002, y 3,166.667 de PTR003; 1000 barriles de gasolina GAS003 usando 666.667 barriles de PTR001 y 333.333 de PTR002. Se tendrían que gastar \$750.00 dólares en publicidad de la gasolina GAS002 y se tendría una utilidad neta de \$ 287,749.969.

5.3 PROBLEMA DE INVENTARIOS.

El problema de inventario que aquí se describe, consiste en determinar cuántos calentadores de agua modelo MGM-I hay que producir en cada uno de los cuatro trimestres del año, para integrar el Programa Maestro de Producción. La demanda durante cada uno de los trimestres es: primer trimestre, 85,000 calentadores; segundo trimestre 73,500 calentadores, tercer trimestre, 55,000 calentadores; cuarto trimestre 112,000 calentadores, se tiene que cumplir con la demandas programadas. Al principio del primer trimestre se tiene un inventario de 10,000 calentadores. Se tiene que decidir, al principio de cada trimestre, cuántos calentadores hay que fabricar en el trimestre. Durante cada trimestre, se pueden producir hasta 65,000 calentadores, en el tiempo regular de trabajo, a un costo total de producción (estándar) de \$405.23 pesos por calentador. Sin embargo se pueden producir calentadores hasta 40,000 calentadores adicionales al hacer que trabajen tiempo extra los empleados durante un trimestre, a un costo total de producción (estándar) de \$512.36 pesos por calentador. Al final de cada trimestre (después de terminar la producción y después de satisfacer la demanda del trimestre actual), se presentan costos de mantenimiento del inventario, de \$ 63.00 pesos por calentador. El objetivo consiste en integrar un programa de producción en donde se minimicen los costos de producción y del inventario durante los cuatro trimestres del año.

Para que este problema pueda ser resuelto por el Sistema de Optimización de Recursos es necesario plantearlo como un modelo de transporte, como a continuación se describe.

Los puntos de oferta los definimos como:

Punto 1: Inventario inicial: 10,000 calentadores

Punto 2: Producción en tiempo regular en el primer trimestre: 65,000 calentadores.

Punto 3: Producción en tiempo extra en el primer trimestre: 40,000 calentadores.

Punto 4: Producción en tiempo regular en el segundo trimestre: 65,000 calentadores.

Punto 5: Producción en tiempo extra en el segundo trimestre: 40,000 calentadores.

Punto 6: Producción en tiempo regular en el tercer trimestre: 65,000 calentadores.

Punto 7: Producción en tiempo extra en el tercer trimestre: 40,000 calentadores.

Punto 8: Producción en tiempo regular en el cuarto trimestre: 65,000 calentadores.

Punto 9: Producción en tiempo extra en el cuarto trimestre: 40,000 calentadores.

Los puntos de demanda se definen como:

Punto 1: Demanda del trimestre 1: 85,000 calentadores.

Punto 2: Demanda del trimestre 2: 73,500 calentadores.

Punto 3: Demanda del trimestre 3: 55,000 calentadores.

Punto 4: Demanda del trimestre 4: 112,000 calentadores.

Así tenemos que un envío del trimestre 1 en tiempo regular hacia la demanda del trimestre 3, significa producir 1 unidad en tiempo regular durante el trimestre 1 que se utilizará para satisfacer 1 unidad de la demanda del trimestre 3.

Los costos de envío (desde un trimestre regular o extra hacia una demanda de cualquier trimestre) se determinan sumando el costo de total producción (estándar) en el trimestre (extra o regular) en que se haya efectuado más el costo de mantener la existencia en inventario hasta que se llegue a su punto de demanda. Para asegurar que no se asignen calentadores para satisfacer la demanda de un trimestre anterior a su producción, se asigna a este tipo de envíos un costo muy grande M (M es un número muy grande).

A continuación se muestra la matriz de costos del modelo, la cual se tendría que capturar en el Sistema de Optimización de Recursos para resolver el problema:

APLICACION DEL SISTEMA EN CASOS PRACTICOS

T R I M E S T R E

	01	02	03	04	Oferta
EXISTENCIA INICIAL	0	63	126	189	10,000
PRODUCCION REGULAR PRIMER TRIMESTRE	405.23	468.23	531.23	594.23	65,000
PRODUCCION EXTRA PRIMER TRIMESTRE	512.36	575.36	638.36	701.36	40,000
PRODUCCION REGULAR SEGUNDO TRIMESTRE	M	405.23	468.23	531.23	65,000
PRODUCCION EXTRA SEGUNDO TRIMESTRE	M	512.36	575.36	638.36	40,000
PRODUCCION REGULAR TERCER TRIMESTRE	M	M	405.23	468.23	65,000
PRODUCCION EXTRA TERCER TRIMESTRE	M	M	512.36	575.36	40,000
PRODUCCION REGULAR CUARTO TRIMESTRE	M	M	M	405.23	65,000
PRODUCCION EXTRA CUARTO TRIMESTRE	M	M	M	512.36	40,000
Demanda	85,000	73,500	55,000	112,000	

SISTEMA DE OPTIMIZACION DE RECURSOS
01-ENE-97
TRANSPORTE

VERSION 1.0
MINIMIZACION

PROGRAMA MAESTRO DE PRODUCCION

Establecimiento del Programa Maestro de Producción Anual para calentadores de agua MGM-1, que se tienen las siguientes características de operación:

- Costo estándar de producción normal: \$ 405.23
- Costo estándar de producción extra: \$ 512.36
- Costo de mantenimiento de inventario: \$ 63.00

Capacidad de producción:	Cantidad
Existencia inicial	10,000
1er. trimestre tiempo regular	65,000
1do. trimestre tiempo extra	40,000
2do. trimestre tiempo regular	65,000
2do. trimestre tiempo extra	40,000
3er. trimestre tiempo regular	65,000
3er. trimestre tiempo extra	40,000
4to. trimestre tiempo regular	65,000
4to. trimestre tiempo extra	40,000
Demanda por trimestre:	Cantidad
Trimestre 01	85,000
Trimestre 02	73,500
Trimestre 03	55,000
Trimestre 04	112,000

Origen	Destino	Cantidad	Valor
EXISTENCIA ORIGINAL	DEMANDA 1er. TRIMESTRE	10,000.00	0.00
PRODUCCION REGULAR 1er. TRIMESTRE	DEMANDA 1er. TRIMESTRE	65,000.00	26,339,950.00
PRODUCCION EXTRA 1er. TRIMESTRE	DEMANDA 1er. TRIMESTRE	10,000.00	5,123,600.00
PRODUCCION REGULAR 2do. TRIMESTRE	DEMANDA 2do. TRIMESTRE	65,000.00	26,339,950.00
PRODUCCION EXTRA 2do. TRIMESTRE	DEMANDA 2do. TRIMESTRE	8,500.00	4,355,060.00
PRODUCCION REGULAR 3er. TRIMESTRE	DEMANDA 3er. TRIMESTRE	55,000.00	22,287,650.00
PRODUCCION REGULAR 3er. TRIMESTRE	DEMANDA 4to. TRIMESTRE	10,000.00	4,682,300.00
PRODUCCION REGULAR 4to. TRIMESTRE	DEMANDA 4to. TRIMESTRE	65,000.00	26,339,950.00
PRODUCCION EXTRA 4to. TRIMESTRE	DEMANDA 4to. TRIMESTRE	37,000.00	18,957,320.00
			134,425,780.00

De acuerdo con los resultados obtenidos se determina que la forma en que se debe producir es la siguiente: el primer trimestre hay que producir 65,000 calentadores, utilizar los 10,000 calentadores que se tienen en ese momento y producir 10,000 más en tiempo extra, esto tendría un gasto total de \$31,463,650.00; el segundo trimestre 65,000 calentadores en tiempo normal y 8,500 calentadores en tiempo extra, con un costo total de \$30,695,000.00; el tercer trimestre sólo hay que producir 65,000 en tiempo regular con costo de \$26,969,950.00; el cuarto trimestre hay que producir 102,000 calentadores con un costo de \$45,297,270.00.

Este programa de producción es el programa óptimo, consiste en la mejor forma de producir para poder conseguir los costos más bajos, es la mejor combinación y no existe otra que la mejore con las condiciones de operación que se tienen, cualquier cambio en las cantidades de producción significaría un incremento en el costo total.

Este tipo de modelos puede presentar situaciones que son difíciles de observar, tal como en algunas ocasiones en que se presenta que es más conveniente producir en tiempo extra aun que parezca que es más caro, pero se compensa con los ahorros en el manejo de materiales y control de inventarios asociados, o como que es más barato dejar de producir en tiempo regular en determinado momento aunque se encuentren ociosos los empleados y las máquinas, cosa que va de acuerdo a los nuevos enfoques de producción (Justo a Tiempo, Administración Total de la Calidad, Manufactura Sincronizada).

Este problema representa un problema pequeño (de 9 renglones por 4 columnas) pero los resultados obtenidos son significativos, los beneficios que se obtienen al aplicar este tipo de modelos son: se reduce el inventario, se mejora el flujo de efectivo ya que se compra sólo lo que hace falta en ese momento, y sobre todo se tiende a sincronizar el proceso de producción con el ritmo de la demanda.

5.4 ASIGNACION DE MAQUINAS.

El problema de asignación que a continuación se muestra, representa el modelo típico de estos problemas, el cual puede ser adaptado en una gran variedad de situaciones, sólo cambiando, agregando o eliminando orígenes y destinos, así como también cambiando la matriz de costos.

Se tienen que realizar siete trabajos y se cuenta con ocho máquinas que los pueden llevar a cabo. Hay que asignar cada máquina para que termine un trabajo completo. El tiempo requerido para preparación de cada máquina para que termine cada trabajo se muestra a continuación. El problema consiste en determinar la asignación óptima que minimice el tiempo total de preparación que se requiere para los siete trabajos.

TIEMPOS DE PREPARACION EN HORAS

	TRA01	TRA02	TRA03	TRA04	TRA05	TRA06	TRA07
MAQ01	3.16	9.21	6.36	9.81	7.21	8.72	10.27
MAQ02	10.21	3.16	4.15	6.27	4.32	5.21	7.35
MAQ03	8.16	4.97	3.52	7.56	9.21	6.93	6.55
MAQ04	3.15	9.37	5.19	6.15	7.27	2.15	8.96
MAQ05	8.21	9.15	5.28	9.17	4.21	3.02	8.73
MAQ06	3.21	6.15	7.23	9.18	6.21	6.21	3.27
MAQ07	8.27	9.21	9.55	6.16	6.39	6.17	2.16
MAQ08	7.27	11.00	9.55	6.25	3.21	6.82	8.32

APLICACION DEL SISTEMA EN CASOS PRACTICOS

Al Sistema de Optimización de Recursos se le tiene que capturar la matriz de tiempos y las descripciones de los orígenes y los destinos, para que pueda resolver el problema.

La asignación óptima la realiza en unos cuantos segundos, proporciona la mejor asignación con los tiempos marcados en esta matriz de tiempos, de tal forma que no existe otra asignación que mejore el tiempo total de preparación para este caso. El resultado (en reporte) que se obtienen es presentado a continuación:

SISTEMA DE OPTIMIZACION DE RECURSOS
01-ENE-97
ASIGNACION

VERSION 1.0
MINIMIZACION

ASIGNACION DE MAQUINAS

Asignación de máquinas para que realicen siete trabajos.

El problema consiste en determinar la asignación óptima que minimice el tiempo total de preparación que se requiere para terminar los siete trabajos.

Tiempos de preparación (horas):

		T R A B A J O						
MAQUINA		01	02	03	04	05	06	07
01		3.16	9.21	6.36	9.81	7.21	8.72	10.27
02		10.21	3.16	4.15	6.27	4.32	5.21	7.35
03		8.16	4.97	3.52	7.56	9.21	6.93	6.55
04		3.15	9.37	5.19	6.15	7.27	2.15	8.96
05		8.21	9.15	5.28	9.17	4.21	3.02	8.73
06		3.21	6.15	7.23	9.18	6.21	6.21	3.27
07		8.27	9.21	9.55	6.16	6.39	6.17	2.16
08		7.27	11.00	9.55	6.25	3.21	6.82	8.32

De	A	Valor
MAQUINA 01	TRABAJO 01	3.16
MAQUINA 02	TRABAJO 02	3.16
MAQUINA 03	TRABAJO 03	3.52
MAQUINA 08	TRABAJO 05	3.21
MAQUINA 04	TRABAJO 06	2.15
MAQUINA 07	TRABAJO 04	6.16
		21.36

5.5 FABRICACION SOBRE DISEÑO.

En algunas ocasiones la producción de algún bien se realiza por unidad, tal como el software personalizado, las casas, los edificios, las grandes embarcaciones, etc. En estas situaciones, las actividades de administración de la producción se complican por el hecho de que el trabajo es único, las actividades tienen alguna secuencia, algunas duran más que otras, así como también el costo entre cada una de ellas varía, para ello se cuenta con herramientas que ayudan a administrar eficientemente toda el proceso de producción, tal es el caso de el Método de la Ruta Crítica (CPM) y la Técnica de Evaluación y Revisión de Programas (PERT). En esta sección se presenta un pequeño ejemplo de fabricación sobre diseño y sus soluciones obtenidas con el Sistema de Optimización de Recursos; se resolverá con el Método de la Ruta Crítica, pudiéndose utilizar este método cuando los datos que se le proporcionan al sistema se conocen con bastante exactitud, esto es común cuando se ha realizado algún trabajo similar a otro y ya se cuenta con información precisa o con la experiencia suficiente para determinar los tiempos de duración y los costos asociados de cada actividad.

A continuación se describen las actividades que conforman el proceso de fabricación de una empresa que se dedica a fabricar sistemas para el manejo de materiales. Algunas operaciones son ensambles estándar fabricados con especificaciones de catálogo, en tanto que otros son diseños del cliente fabricados para satisfacer sus necesidades específicas. Estas actividades que se describen se requieren para terminar la fabricación de una orden específica diseñada por un cliente:

APLICACION DEL SISTEMA EN CASOS PRACTICOS

Código	Descripción	Precedencia	Duración (Semanas)		Costo (\$)	
			Normal	Ruptura	Normal	Ruptura
A	Diseño mecánico parte I	Ninguna	2.0	1.6	4,800	5,600
B	Diseño mecánico parte II	A	3.0	2.0	7,680	8,500
C	Diseño eléctrico parte I	A	1.0	0.9	3,100	3,600
D	Diseño eléctrico parte II	B,C	5.0	3.0	13,500	18,000
E	Fabricación mecánica parte I	B	2.0	0.6	4,940	6,000
F	Fabricación mecánica parte II	D	6.0	4.6	15,600	20,000
G	Fabricación eléctrica parte I	D	3.0	2.5	4,200	5,000
H	Fabricación eléctrica parte II	E,F,G	2.0	2.0	3,025	3,025
I	Subensamble mecánico parte I	E,F	3.0	2.0	4,100	4,400
J	Subensamble mecánico parte II	H	4.0	2.0	5,200	5,600
K	Instalación eléctrica parte I	H,I	5.0	4.0	3,730	4,500
L	Instalación eléctrica parte II	M	1.0	1.0	700	700
M	Instalación de tubería parte I	J,K	1.0	0.8	790	1,000
N	Instalación de tubería parte II	L	3.0	2.8	2,015	2,400
O	Inicio, prueba y embarque	N	1.0	1.0	2,100	2,100

El personal que se requiere para desarrollar cada una de las actividades que conforman al proyecto, es el siguiente:

Diseño mecánico parte I	7
Diseño mecánico parte II	5
Diseño eléctrico parte I	6
Diseño eléctrico parte II	8
Fabricación mecánica parte I	25
Fabricación mecánica parte II	32

Fabricación eléctrica parte I	15
Fabricación eléctrica parte II	12
Subensamble mecánico parte I	7
Subensamble mecánico parte II	7
Instalación eléctrica parte I	4
Instalación eléctrica parte II	2
Instalación de tubería parte I	5
Instalación de tubería parte II	5
Inicio, prueba y embarque	12

El objetivo consiste en determinar la duración total del proyecto en sus modelos normal, óptimo y de ruptura, los costos asociados, las actividades que determinan la duración de cada modelo y un análisis del personal requerido en cada etapa del proyecto.

La información de cada uno de los proyecto debe incluir la fecha en que puede iniciar cada una de las actividades, el máximo retraso posible en cada actividad, la fecha de terminación más cercana en caso de que se haya iniciado la actividad en el tiempo más próximo y la fecha de terminación más lejana en caso de que el inicio de la actividad se haya retrasado.

Para que el Sistema de Optimización de Recursos proporcione la información que se requiere es necesario capturar la tabla de actividades en la opción donde se encuentra el método de la Ruta Crítica, como se trata de un modelo determinista.

A continuación se muestran los resultados obtenidos de forma impresa con lo cuales queda descrito todo el proyecto, así como también, proporcionan toda la información que se requiere para la planeación y el control de las actividades.

APLICACION DEL SISTEMA EN CASOS PRACTICOS

SISTEMA DE OPTIMIZACION DE RECURSOS
01-ENE-97
PLANEACION Y CONTROL DE PROYECTOS (CPM)

VERSION 1.0

FABRICACION SOBRE DISEÑO

Fabricación sobre diseño de un sistema para el manejo de materiales.

Las actividades que se requieran para la fabricación de una orden específica diseñada por un cliente, son las siguientes:

Actividad	Precedencia	(Duración (semanas))		Costo (pesos)	
		Normal	Ruptura	Normal	Ruptura
Diseño mecánico I	Ninguna	2.0	1.0	4,800.00	5,600.00
Diseño mecánico II	A	3.0	2.0	7,680.00	8,500.00
Diseño eléctrico I	A	1.0	1.0	3,100.00	3,100.00
Diseño eléctrico II	B,C	5.0	3.0	13,500.00	18,000.00
Fabricación mecánica I	B	2.0	1.0	4,940.00	6,000.00
Fabricación mecánica II	D	6.0	5.0	15,600.00	20,000.00
Fabricación eléctrica I	D	3.0	2.0	4,200.00	5,000.00
Fabricación eléctrica II	E,F,G	2.0	2.0	3,025.00	3,025.00
Subensamble mecánico I	E,F	3.0	2.0	4,100.00	4,400.00
Subensamble mecánico II	H	4.0	2.0	5,200.00	5,600.00
Instalación eléctrica I	H,I	5.0	4.0	3,730.00	4,500.00
Instalación eléctrica II	M	1.0	1.0	700.00	700.00
Instalación tubería I	J,K	1.0	1.0	790.00	790.00
Instalación tubería II	L	3.0	2.0	2,015.00	2,400.00
Inicio, prueba y embarque	N	1.0	1.0	2,100.00	2,100.00

PROYECTO NORMAL

Distribución de Tiempo-Costo (semanas-5):

Costo fijo: \$ 15,900.00

Actividad	Duración	Costo	Inic.		Term.		
			Próxima	Próxima	Tardia	Tardia	
DISEÑO MECANICO PARTE I	(*)	2	4,800.00	0	2	0	2
DISEÑO MECANICO PARTE II	(*)	3	7,680.00	2	5	2	5
DISEÑO ELECTRICO PARTE I	(*)	1	3,100.00	2	3	4	5
DISEÑO ELECTRICO PARTE II	(*)	5	13,500.00	5	10	5	10
FABRICACION MECANICA PARTE I	(*)	2	4,940.00	5	7	14	16
FABRICACION MECANICA PARTE II	(*)	6	15,600.00	10	16	10	16
FABRICACION ELECTRICA PARTE I	(*)	3	4,200.00	10	13	14	17
FABRICACION ELECTRICA PARTE II	(*)	2	3,025.00	16	18	17	19
SUBENSAMBLADO MECANICO PARTE I	(*)	3	4,100.00	16	19	16	19

APLICACION DEL SISTEMA EN CASOS PRACTICOS

SISTEMA DE OPTIMIZACION DE RECURSOS
01-ENE-97
PLANEACION Y CONTROL DE PROYECTOS (CPM)

VERSION 1.0

FABRICACION SOBRE DISEÑO

PROYECTO NORMAL

Distribucion de Tiempo-Costo (semanas-\$):

Costo fijo: \$ 15,900.00

Actividad		Duración	Costo	Inic. Próxima	Term. Próxima	Inic. Tardia	Term. Tardia
SUBENSAMBLADO MECANICO PARTE II		4	5,200.00	18	22	20	24
INSTALACION ELECTRICA PARTE I	(*)	5	3,730.00	19	24	19	24
INSTALACION ELECTRICA PARTE II	(*)	1	700.00	25	26	25	26
INSTALACION DE TUBERIA PARTE I	(*)	1	790.00	24	25	24	25
INSTALACION TUBERIA PARTE II	(*)	3	2,015.00	26	29	26	29
INICIO, PRUEBA Y EMBARQUE	(*)	1	2,100.00	29	30	29	30

Duración total = 30 semanas

Costo Total = \$ 91,380.00

Balanco de recursos:

Periodos: semanas	Del	Al	Personal requerido
	0	2	7.00
	2	3	11.00
	3	5	5.00
	5	7	33.00
	7	10	8.00
	10	13	47.00
	13	16	32.00
	16	18	19.00
	18	19	14.00
	19	22	11.00
	22	24	4.00
	24	25	5.00
	25	26	2.00
	26	29	5.00
	29	30	12.00

SISTEMA DE OPTIMIZACION DE RECURSOS
01-ENE-97
PLANEACION Y CONTROL DE PROYECTOS (CPM)

VERSION 1.0

FABRICACION SOBRE DISEÑO

PROYECTO OPTIMO

Distribución de Tiempo-Costo (semanas- $\$$):

Costo fijo: $\$$ 15,370.00

Actividad		Duración	Costo	Inic. Próxima	Term. Próxima	Inic. Tardía	Term. Tardía
DISEÑO MECANICO PARTE I	(*)	2	4,800.00	0	2	0	2
DISEÑO MECANICO PARTE II	(*)	3	7,680.00	2	5	2	5
DISEÑO ELECTRICO PARTE I		1	3,100.00	2	3	4	5
DISEÑO ELECTRICO PARTE II	(*)	5	13,500.00	5	10	5	10
FABRICACION MECANICA PARTE I		2	4,940.00	5	7	14	16
FABRICACION MECANICA PARTE II	(*)	6	15,600.00	10	16	10	16
FABRICACION ELECTRICA PARTE I		3	4,200.00	10	13	13	16
FABRICACION ELECTRICA PARTE II	(*)	2	3,025.00	16	18	16	18
SUBENSAMBLADO MECANICO PARTE I	(*)	2	4,400.00	16	18	16	18
SUBENSAMBLADO MECANICO PARTE II		4	5,200.00	18	22	19	23
INSTALACION ELECTRICA PARTE I	(*)	5	3,730.00	18	23	18	23
INSTALACION ELECTRICA PARTE II	(*)	1	700.00	24	25	24	25
INSTALACION DE TUBERIA PARTE I	(*)	1	790.00	23	24	23	24
INSTALACION TUBERIA PARTE II	(*)	3	2,015.00	25	28	25	28
INICIO, PRUEBA Y EMBARQUE	(*)	1	2,100.00	28	29	28	29

Duración total = 29 semanas

Costo Total = $\$$ 91,150.00

Balanceo de recursos:

Periodos: semanas	Del	Al	Personal requerido
	0	2	7.00
	2	3	11.00
	3	5	5.00
	5	7	33.00
	7	10	8.00
	10	13	47.00
	13	16	32.00
	16	18	19.00
	18	22	11.00
	22	23	4.00
	23	24	5.00
	24	25	2.00
	25	28	5.00
	28	29	12.00

APLICACION DEL SISTEMA EN CASOS PRACTICOS

SISTEMA DE OPTIMIZACION DE RECURSOS
01-ENE-97
PLANEACION Y CONTROL DE PROYECTOS (CPM)

VERSION 1.0

FABRICACION SOBRE DISEÑO

PROYECTO DE RUPTURA

Distribución de Tiempo-Costo (semanas-\$):

Costo fijo: \$ 14,310.00

Actividad		Duración	Costo	Inic. Próxima	Term. Próxima	Inic. Tardía	Term. Tardía
DISEÑO MECANICO PARTE I	(*)	1	5,600.00	0	1	0	1
DISEÑO MECANICO PARTE II	(*)	3	7,680.00	1	4	1	4
DISEÑO ELECTRICO PARTE I		1	3,100.00	1	2	3	4
DISEÑO ELECTRICO PARTE II	(*)	5	13,500.00	4	9	4	9
FABRICACION MECANICA PARTE I		2	4,940.00	4	6	13	15
FABRICACION MECANICA PARTE II	(*)	6	15,600.00	9	15	9	15
FABRICACION ELECTRICA PARTE I		3	4,200.00	9	12	12	15
FABRICACION ELECTRICA PARTE II	(*)	2	3,025.00	15	17	15	17
SUBENSAMBLADO MECANICO PARTE I	(*)	2	4,400.00	15	17	15	17
SUBENSAMBLADO MECANICO PARTE II	(*)	4	5,200.00	17	21	17	21
INSTALACION ELECTRICA PARTE I	(*)	4	4,500.00	17	21	17	21
INSTALACION ELECTRICA PARTE II	(*)	1	700.00	22	23	22	23
INSTALACION DE TUBERIA PARTE I	(*)	1	790.00	21	22	21	22
INSTALACION TUBERIA PARTE II	(*)	3	2,015.00	23	26	23	26
INICIO, PRUEBA Y EMBARQUE	(*)	1	2,100.00	26	27	26	27

Duración total = 27 semanas

Costo Total = \$ 91,660.0

Balanceo de recursos:

Periodos: semanas	Del	Al	Personal requerido
	0	1	7.00
	1	2	11.00
	2	4	5.00
	4	6	33.00
	6	9	8.00
	9	12	47.00
	12	15	32.00
	15	17	19.00
	17	21	11.00
	21	22	5.00
	22	23	2.00
	23	26	5.00
	26	27	12.00

Nota: Las actividades marcadas con (*) son las que controlan la duración del proyecto.

5.6 CONSTRUCCION DE UNA PLANTA ELECTRICA.

Los planeadores de proyecto han empleado el juicio de varios reconocidos ingenieros, capataces y proveedores y han desarrollado las estimaciones de tiempo mostradas en la siguiente tabla, para la construcción de una planta eléctrica:

Actividades	Tiempos Estimados (en meses)		
	Optimista	Probable	Pesimista
Diseño de la planta	10	12	16
Selección del lugar	2	8	36
Selección de proveedores	1	4	5
Selección de personal	2	3	4
Preparación del lugar	8	12	20
Fabricación del generador	15	18	30
Preparación del material	3	5	8
Instalación del generador	2	4	8
Adiestramiento de los operadores	6	9	12
Licencia de la planta	4	6	14

Los costos y el personal requerido en cada una de las actividades que conforman al proyecto se muestra a continuación:

APLICACION DEL SISTEMA EN CASOS PRACTICOS

Actividades	Costos en miles de dólares.		
	Normal	Ruptura	Personal Requerido
Diseño de la planta	7,500	9,200	2
Selección del lugar	600	800	2
Selección de proveedores	400	635	6
Selección de personal	450	623	3
Preparación del lugar	600	900	10
Fabricación del generador	27,500	33,500	35
Preparación del material	1,200	1,620	7
Instalación del generador	6,350	10,320	16
Adiestramiento de los operadores	1,500	2,100	7
Licencia de la planta	730	915	2

Los costos fijos por periodo son de 75,000 dólares.

Se desea obtener la información necesaria para administrar eficientemente el proyecto, tal como las actividades que determinan la duración del proyecto, los modelos óptimo, normal y de ruptura, los costos asociados a cada modelo, el balance del personal requerido en cada etapa para cada modelo y la probabilidad que determinado proyecto se termine en cierto tiempo.

Al capturar los datos en el Sistema de Optimización de Recursos, de manera similar que en el modelo de Ruta Crítica, obtenemos los siguientes resultados de manera impresa

APLICACION DEL SISTEMA EN CASOS PRACTICOS

Actividades	Costos en miles de dólares.		
	Normal	Ruptura	Personal Requerido
Diseño de la planta	7,500	9,200	2
Selección del lugar	600	800	2
Selección de proveedores	400	635	6
Selección de personal	450	623	3
Preparación del lugar	600	900	10
Fabricación del generador	27,500	33,500	35
Preparación del material	1,200	1,620	7
Instalación del generador	6,350	10,320	16
Adiestramiento de los operadores	1,500	2,100	7
Licencia de la planta	730	915	2

Los costos fijos por periodo son de 75,000 dólares.

Se desea obtener la información necesaria para administrar eficientemente el proyecto, tal como las actividades que determinan la duración del proyecto, los modelos óptimo, normal y de ruptura, los costos asociados a cada modelo, el balance del personal requerido en cada etapa para cada modelo y la probabilidad que determinado proyecto se termine en cierto tiempo.

Al capturar los datos en el Sistema de Optimización de Recursos, de manera similar que en el modelo de Ruta Crítica, obtenemos los siguientes resultados de manera impresa:

SISTEMA DE OPTIMIZACION DE RECURSOS
01-ENE-97
PLANEACION Y CONTROL DE PROYECTOS (PERT)

VERSION 1.0

CONSTRUCCION DE PLANTA DE ENERGIA ELECTRICA

Diseño de una planta de energía eléctrica

Actividades	Precedencia	Duración (meses)			Costos (miles dls)		Personal
		Optimo	Normal	Pésimo	Normal	Ruptura	
1. Diseño de la planta	---	10	12	16	7,500	9,200	2
2. Selección del lugar	1	2	8	36	600	800	2
3. Selección de proveedor	1	1	4	5	400	635	6
4. Selección del personal	1	2	3	4	450	623	3
5. Preparación del lugar	2	8	12	20	600	900	10
6. Fabricación del generador	3	15	18	30	27,500	33,500	35
7. Preparación de manual operación	3	3	5	8	1,200	1,620	7
8. Instalación del generador	5,6	2	4	8	6,350	10,320	16
9. Adiestramiento de los operadores	7,4	6	9	12	1,500	2,100	7
10. Licencia de la planta	8,9	4	6	14	730	915	2

PROYECTO NORMAL

Distribución de Tiempo-Costo (meses-\$):

Costo fijo: \$ 3,600.00

Actividad	Duración	Costo	Inic. Termin.		Inici. Termin.	
			Próxima	Próxima	Tardía	Tardía
DISEÑO DE LA PLANTA	(*) 12	7,500.00	0	12	0	12
SELECCION DEL LUGAR	(*) 12	600.00	12	24	12	24
SELECCION DE PROVEEDORES	4	400.00	12	16	14	17
SELECCION DEL PERSONAL	3	450.00	12	15	29	32
PREPARACION DEL LUGAR	(*) 13	600.00	24	37	24	37
FABRICACION DEL GENERADOR	20	27,500.00	16	36	17	37
PREPARACION DE MANUAL	5	1,200.00	16	21	27	32
INSTALACION DEL GENERADOR	(*) 4	6,350.00	37	41	37	41
ADIESTRAMIENTO DE LOS OPERADORES	9	1,500.00	21	30	32	41
LICENCIA DE LA PLANTA	(*) 7	730.00	41	48	41	48

Duración total = 48 meses Costo Total = \$ 50,430.00

SISTEMA DE OPTIMIZACION DE RECURSOS
 01-ENE-97
 PLANEACION Y CONTROL DE PROYECTOS (PERT)

VERSION 1.0

CONSTRUCCION DE PLANTA DE ENERGIA ELECTRICA

PROYECTO NORMAL

Balanceo de recursos:

Periodos: meses	Del	Al	Personal requerido
	0	12	2.00
	12	15	11.00
	15	16	2.00
	16	23	44.00
	23	24	42.00
	24	30	52.00
	30	35	45.00
	35	37	10.00
	37	41	16.00
	41	48	2.00

APLICACION DEL SISTEMA EN CASOS PRACTICOS

SISTEMA DE OPTIMIZACION DE RECURSOS
01-ENE-97
PLANEACION Y CONTROL DE PROYECTOS (PERT)

VERSION 1.0

CONSTRUCCION DE PLANTA DE ENERGIA ELECTRICA

PROYECTO OPTIMO

Distribución de Tiempo-Costo (meses-\$):

Costo fijo: \$ 3,512.25

Actividad	Duracion	Costo	Inic.	Termin.	Inici.	Termin.
			Próxima	Próxima	Tardia	Tardia
DISEÑO DE LA PLANTA	(*) 12	7,500.00	0	12	0	12
SELECCION DEL LUGAR	(*) 11	624.20	12	23	12	23
SELECCION DE PROVEEDORES	(*) 4	400.00	12	16	12	16
SELECCION DEL PERSONAL	(*) 3	450.00	12	15	28	31
PREPARACION DEL LUGAR	(*) 13	600.00	23	36	23	36
FABRICACION DEL GENERADOR	(*) 20	27,500.00	16	36	16	36
PREPARACION DE MANUAL	(*) 5	1,200.00	16	21	26	31
INSTALACION DEL GENERADOR	(*) 4	6,350.00	36	40	36	40
ADIESTRAMIENTO DE LOS OPERADORES	(*) 9	1,500.00	21	30	31	40
LICENCIA DE LA PLANTA	(*) 7	730.00	40	47	40	47

Duración total = 47 meses

Costo Total = \$ 50,366.45

Balanco de recursos:

Periodos: meses	Del	Al	Personal requerido
	0	12	2.00
	12	15	11.00
	15	16	2.00
	16	22	44.00
	22	23	42.00
	23	30	52.00
	30	36	45.00
	36	40	16.00
	40	47	2.00

APLICACION DEL SISTEMA EN CASOS PRACTICOS

SISTEMA DE OPTIMIZACION DE RECURSOS
01-ENE-97
PLANEACION Y CONTROL DE PROYECTOS (PERT)

VERSION 1.0

CONSTRUCCION DE PLANTA DE ENERGIA ELECTRICA

PROYECTO DE RUPTURA

Distribución de Tiempo-Costo (meses-\$):

Costo fijo: \$ 3,312.00

Actividad	Duración	Costo	Inic. Próxima	Termin. Próxima	Inici. Tardía	Termin. Tardía
DISEÑO DE LA PLANTA	(*) 12	7,500.00	0	12	0	12
SELECCION DEL LUGAR	(*) 8	679.42	12	20	12	20
SELECCION DE PROVEEDORES	(*) 1	635.00	12	13	12	13
SELECCION DEL PERSONAL	3	450.00	12	15	25	28
PREPARACION DEL LUGAR	(*) 13	600.00	20	33	20	33
FABRICACION DEL GENERADOR	(*) 20	27,500.00	13	33	13	33
PREPARACION DE MANUAL	5	1,200.00	13	19	23	28
INSTALACION DEL GENERADOR	(*) 4	6,350.00	33	37	33	37
ADIESTRAMIENTO DE LOS OPERADORES	9	1,500.00	19	28	28	37
LICENCIA DE LA PLANTA	(*) 7	730.00	37	44	37	44

Duración total = 44 meses

Costo Total = \$ 50,456.42

Balanco de recursos:

Periodos: meses	Del	Al	Personal requerido
	0	12	2.00
	12	13	11.00
	13	15	47.00
	15	18	44.00
	18	19	37.00
	19	20	42.00
	20	28	52.00
	28	33	45.00
	33	37	16.00
	37	44	2.00

Nota: Las actividades marcadas con (*) son las que controlan la duración del proyecto.

SISTEMA DE OPTIMIZACION DE RECURSOS
 01-ENE-97
 PLANEACION Y CONTROL DE PROYECTOS (PERT)

VERSION 1.0

CONSTRUCCION DE PLANTA DE ENERGIA ELECTRICA

TABLA DE PROBABILIDADES DE TERMINACION

Probabilidad	Duración (meses)
10 %	39.687
20 %	42.245
30 %	44.163
40 %	46.082
50 %	48.000
60 %	49.279
70 %	51.197
80 %	53.116
90 %	55.673
100 %	79.972

5.7 ANALISIS DE LA VIDA ECONOMICA DE EQUIPO PARA TRANSPORTACION.

Para la distribución de productos terminados, una empresa, opera un camión que cuando se adquirió nuevo, costo \$ 3,000.00 dólares. Se desea determinar la frecuencia de reemplazo, y para ello se cuenta con las siguientes estimaciones:

Año	Valor de Rescate (dólares)	Costo de Operación (dólares)
1	2,000	600
2	1,333	700
3	1,000	800
4	750	900
5	500	1,000
6	300	1,200
7	300	1,500

El valor de rescate es la cantidad a la que el camión podría venderse al fin del año, y los costos de operación son debidos a la gasolina, impuestos, mantenimiento y reparaciones durante el año.

A continuación se muestran los resultados obtenidos con el Sistema de Optimización de Recursos, con lo que se puede ver que el costo más bajo por año se alcanza reemplazando al camión al final del quinto año.

SISTEMA DE OPTIMIZACION DE RECURSOS
01-ENE-97
VIDA ECONOMICA DEL EQUIPO

VERSION 1.0

VIDA ECONOMICA DE EQUIPO DE TRANSPORTE

Determinación de la vida económica de equipo de transportación, se tienen las siguientes estimaciones:

Año	Valor de Rescate (dólares)	Costo de Operación (dólares)
1	2,000.00	600.00
2	1,333.00	700.00
3	1,000.00	800.00
4	750.00	900.00
5	500.00	1,000.00
6	300.00	1,200.00
7	300.00	1,500.00

Periodo	Costo de Operación	Costo de Depreciación	Costo Total	Costo Promedio
001	600.00	1,000.00	1,600.00	1,600.00
002	1,300.00	1,667.00	2,967.00	1,483.50
003	2,100.00	2,000.00	4,100.00	1,366.67
004	3,000.00	2,250.00	5,250.00	1,312.50
005	4,000.00	2,500.00	6,500.00	1,300.00
006	5,200.00	2,700.00	7,900.00	1,316.67
007	6,700.00	2,700.00	9,400.00	1,342.86

5.8 ANALISIS DE REEMPLAZO DE FRESADORA PARA TROQUELES.

Una compañía de herramientas y troqueles esta considerando la compra de una fresadora adicional. La compañía tiene la oportunidad de comprar una máquina ligeramente usada por \$ 15,000.00 dólares o comprar una nueva por \$ 21,000.00. Como la máquina nueva es un modelo más sofisticado con algunos avances automatizados, se espera que su costo operativo anual sea de \$ 2,000.00 mientras que los mismos costos para la máquina usada se estima de \$ 3,200 anuales. Se espera que la máquina nueva tenga una vida útil de 10 años y la máquina usada tenga una vida útil de 8 años. El valor de salvamento se espera que sea del 5 % de su precio original para cada una. Se estima una tasa de interés del 25 % anual. Se desea determinar cual es la opción que le conviene a la empresa elegir, desde un punto de vista económico.

SISTEMA DE OPTIMIZACION DE RECURSOS
01-ENE-97
ANALISIS DE REEMPLAZO

VERSION 1.0

Análisis de reemplazo para la selección de la mejor opción de compra de una fresadora.

Opción 1:

Máquina usada.	
Precio de compra:	\$ 15,000.00 dólares.
Costo de operación anual:	\$ 3,200.00 dólares.
Vida útil:	8 años.
Valor de rescate:	\$ 750.00 dólares.

Opción 2:

Máquina nueva.	
Precio de compra:	\$ 21,000.00 dólares.
Costo de operación anual:	\$ 2,000.00 dólares.
Vida útil:	10 años.
Valor de rescate:	\$ 1,050.00 dólares.

Tasa de interés: 25 % anual

Análisis de reemplazo a 40 periodos.

1	FRESADORA USADA	25,869.28
2	FRESADORA NUEVA	36,194.98

5.9 POLITICA DE MANTENIMIENTO PREVENTIVO PARA EQUIPO DE COMPUTO.

El centro de cómputo de una organización, cuenta con 75 impresoras de matriz de puntos para las tareas de impresión de documentos de gran volumen, estas impresoras han estado funcionando durante varios años. De acuerdo con datos histórico - estadísticos se obtuvo la probabilidad de falla de las impresoras después del mantenimiento realizado y a continuación se muestra:

Año Después del Mantenimiento	Probabilidad de Falla
1	0.2
2	0.4
3	0.2
4	0.1
5	0.1

Se desea saber el número esperado de fallas en los cinco años siguientes, para establecer cuando es conveniente proporcionar el mantenimiento preventivo, para reducir los costos de mantenimiento. El costo unitario de mantenimiento preventivo es de \$ 26.66 mientras que el costo de mantenimiento correctivo (incluyendo todo el impacto que sufre la empresa al fallar una impresora) es \$ 40.00.

Una vez capturados los datos en el Sistema de Optimización de Recursos, se obtienen los siguientes resultados, de forma impresa, con lo que se puede establecer la política óptima de mantenimiento de acuerdo al costo esperado.

SISTEMA DE OPTIMIZACION DE RECURSOS
01-ENERO-97
MANTENIMIENTO PREVENTIVO

VERSION 1.0

MANTENIMIENTO PREVENTIVO DE UN SISTEMA DE COMPUTO

Establecimiento de un programa de mantenimiento preventivo en un centro de computo.

Probabilidades de falla después del mantenimiento para una línea de impresoras que han estado funcionando durante varios años.

Años después del Mantenimiento	Probabilidad de falla
1	0.2
2	0.4
3	0.2
4	0.1
5	0.1

Número de Periodo	Número de Fallas Esperados	Costo de M. Correctivo	Costo total por periodo
1	15.00	600.00	2,600.00
2	48.00	1,920.00	1,960.00
3	75.60	3,024.00	1,674.66
4	104.82	4,192.80	1,548.20
5	137.30	5,492.16	1,498.80

5.10 MANTENIMIENTO DE UN SISTEMA DE PROCESAMIENTO AUTOMATICO.

Una empresa dedicada a prestar servicios de reservaciones de viajes ha experimentado los números de reparaciones por mes en su sistema de procesamiento automático de reservaciones durante los dos últimos años, como se indican a continuación:

Número de Fallas	Número de Meses en que ocurre
0	2
1	8
2	10
3	3
4	1

Cada reparación cuesta en promedio de \$ 280.00 pesos. Por un costo de \$ 150.00 por mes, se puede contratar a una empresa de procesamiento de datos para realizar el mantenimiento preventivo, la cual garantiza que limita el número de reparaciones a un promedio de uno por mes. (Si las reparaciones exceden este número, la empresa de procesamiento de datos los procesará sin cargo). ¿Cuál arreglo de mantenimiento es preferible desde un punto de vista de costos, la política actual de reparaciones o un contrato de mantenimiento preventivo?

SISTEMA DE OPTIMIZACION DE RECURSOS
01-ENE-97
MANTENIMIENTO CORRECTIVO

VERSION 1.0

PROCESO AUTOMATICO DE RESERVACIONES

Números de reparaciones por mes en su sistema de procesamiento automático de reservaciones durante los dos últimos años, que se indican a continuación:

Número de Fallas: 10 | 11 | 2 | 3 | 4 |
Número de meses que esto ocurre: 12 | 8 | 10 | 3 | 1 |

Cada reparación cuesta a la empresa un promedio de \$ 280.00. Por un costo de \$ 150.00 por mes. Se puede contratar a una empresa de procesamiento de datos para realizar el mantenimiento preventivo, la cual garantiza que limita al número de reparaciones promedio de uno por mes. (Si las reparaciones exceden este número, la empresa procesará datos sin cargo.)

¿Cuál arreglo de mantenimiento es preferible desde un punto de vista de costo, la política actual de reparaciones o un contrato de mantenimiento preventivo.

Costo promedio de reparación / mes: \$ 280.00
Costo del contrato de mantenimiento / mes: \$ 150.00
Total: \$ 430.00

Número de fallas	Frecuencia por periodo	Frecuencia en porcentaje	Valor esperado
0	2	8.3333	0.0000
1	8	33.3333	0.3333
2	10	41.6667	0.8333
3	3	12.5000	0.3750
4	1	4.1667	0.1667

Costo esperado \$ 478.33 por periodo.

El costo de mantenimiento preventivo por mes es de una reparación más el costo del contrato de mantenimiento preventivo: \$ 280.00 + \$ 150.00 = \$430.00.

Ventaja del mantenimiento preventivo = \$ 478.33 - \$ 430.00 = \$48.33 por mes

5.11 OPTIMIZACION DE SISTEMAS DE COLAS EN PLANTA TROQUELADORA.

Los mecánicos que trabajan en una planta de troquelado deben sacar herramientas de un almacén. Llega un promedio de diez mecánicos por hora buscando partes. En la actualidad el almacén está a cargo de un empleado a quien se le paga 6 pesos/hora y gasta un promedio de 5 minutos para entregar las herramientas de cada solicitud. Como a los mecánicos se les paga 10 pesos/hora, cada hora que pasa un mecánico en el almacén de herramientas le cuesta 10 pesos a la empresa. Esta ha de decidir si vale la pena contratar, a 4 pesos/hora, un ayudante del almacenista. Si se contrata al ayudante, el almacenista sólo tardará un promedio de 4 minutos para atender las solicitudes de herramientas. Se supone que son exponenciales, tanto los tiempos de servicio como los tiempos entre llegadas. El problema consiste en determinar si se debe contratar al ayudante del almacenista o no.

En este problema, la meta de la empresa es minimizar la suma del costo horario de servicio y el costo horario esperado debido a los tiempos de inactividad de los mecánicos. En los problemas de optimización de colas, el componente de costos debido a clientes que esperan en la cola se llama costo de demora. Así la empresa desea minimizar

$$\text{Costo esperado/hora} = \text{Costo de servicio/hora} + \text{Costo de demora/hora}$$

En general, el cálculo del costo horario de servicio es sencillo. El modo más fácil de calcular el costo horario de demora es tomando nota de que

$$\text{Costo horario de demora/hora} = (\text{Costo esperado de demora/hora}) (\text{Clientes esperados/hora})$$

De esta manera se tienen dos sistemas de colas que los analizamos en el Sistema de Optimización de Recursos y se obtuvieron los siguientes resultados:

SISTEMA DE OPTIMIZACION DE RECURSOS

VERSION 1.0

01-ENE-97

LINEAS DE ESPERA (UNA FILA - UN SERVIDOR)

DESPACHO DE HERRAMIENTAS SIN AYUDANTE

Despacho de herramienta para una planta de troquelado.

Número de servidores: 1
 Tasa de llegada: 10 mecánicos / hora
 Tasa de servicio: 5 minutos / mecánico
 12 mecánicos/hora

Tasa de llegada: 10.00
 Tasa de servicio: 12.00
 Servidor desocupado: 16.67 %
 Intensidad de tráfico: 83.33 %
 Elementos en el sistema: 5.000
 Elementos en espera: 4.167
 Elementos en servicio: 0.833
 Tiempo promedio en el sistema: 0.500
 Tiempo promedio en espera: 0.417
 Tiempo promedio en servicio: 0.083

SISTEMA DE OPTIMIZACION DE RECURSOS

01-ENE-97

LINEAS DE ESPERA (UNA FILA - UN SERVIDOR)

VERSION 1.0

DESPACHO DE HERRAMIENTAS CON AYUDANTE

Despacho de herramientas en una planta de troquelado.

Número de servidores:	1
Tasa de entrada:	10 mecánicos / hora.
Tasa de servicio:	15 mecánicos / hora.

Tasa de llegada:	10.00
Tasa de servicio:	15.00
Servidor desocupado:	33.33 %
Intensidad de tráfico:	66.67 %
Elementos en el sistema:	2.000
Elementos en espera:	1.333
Elementos en servicio:	0.667
Tiempo promedio en el sistema:	0.200
Tiempo promedio en espera:	0.133
Tiempo promedio en servicio:	0.067

Ahora podemos comparar el costo esperado por hora, si no se contratara al ayudante con el correspondiente si se le contrata. Si no se contrata la tasa de llegada es de 10 mecánicos por hora y la tasa de servicio es de 12 mecánicos por hora. El tiempo promedio en el sistema es de 0.500 horas. Como al despachador se le pagan 6 pesos por hora, tenemos que

$$\text{Costo de servicio/Hora} = 6 \text{ pesos}$$

$$\text{Costo esperado de demora/Hora} = (10) (0.500) (10) = 50 \text{ pesos}$$

$$\text{Costo esperado/Hora} = 6 + 50 = 56 \text{ pesos}$$

Con el ayudante, tenemos que una tasa de llegada de 10 mecánicos por hora y la tasa de servicio es de 15 clientes por hora. Por lo tanto el tiempo promedio en el sistema es de 0.200 horas, y tenemos que

$$\text{Costo de servicio/Horas} = 6 + 4 = 10 \text{ pesos}$$

$$\text{Costo esperado de demora/Hora} = (10) (0.200) (10) = 20 \text{ pesos}$$

$$\text{Costo esperado/Hora} = 10 + 20 = 30 \text{ pesos}$$

Por lo tanto, se debe de contratar al ayudante porque se ahorran $50 - 20 = 30$ pesos por hora en costos de demora, lo cual más que compensa su salario de 4 pesos por hora.

5.12 DESCRIPCION DEL SISTEMA DE COLAS DE UN BANCO.

Un banco tiene dos cajeros. Llegan al banco un promedio de 80 clientes por hora y esperan en una sola cola para que los atiendan. El tiempo promedio que se necesita para atender a un cliente es de 1.2 minutos. Se desea calcular

- 1.- Número esperado de clientes en el banco.
- 2.- Tiempo esperado que pasa un cliente en el banco.

Al capturar los datos en el Sistema de Optimización de Recursos se obtienen los resultados con lo que podemos resolver este problemas.

De acuerdo con el listado obtenido con el Sistema de Optimización de Recursos se obtiene el número esperado de clientes en el banco que es de 4.444 clientes, que es lo que se necesita en el punto 1.

El punto 2 solicita el tiempo esperado que pasa un cliente en el banco y de acuerdo con los resultados es de 0.056 horas, es decir de 3.36 minutos.

SISTEMA DE OPTIMIZACION DE RECURSOS
 01-ENE-97
 LINEAS DE ESPERA (UNA FILA - SERVIDORES EN PARALELO)

VERSION 1.0

CAJEROS DE UN BANCO

Número de servidores: 2 cajeros.
 Tasa de llegada: 80 clientes/hora.
 Tasa de servicio: 1 cliente cada 1.2 minutos.
 50 clientes/hora.

Determinar:

1. Número de clientes en el banco.
2. Tiempo esperado que pasa un cliente en el banco.

Tasa de llegada:	80.00
Tasa de servicio:	50.00
Número de servidores:	2
Intensidad de tráfico:	80.00 %
Elementos en el sistema:	4.444
Elementos en espera:	2.844
Elementos en servicio:	1.600
Tiempo promedio en el sistema:	0.056
Tiempo promedio en espera:	0.036
Tiempo promedio en servicio:	0.020

5.13 SISTEMAS DE LINEAS DE ESPERA EN SERIE (LINEA DE ENSAMBLE).

Las dos últimas cosas que hacen en un automóvil para completar su ensamble son instalar el motor y poner los neumáticos. Llega un promedio de 54 automóviles/h que necesita estas dos tareas. Un trabajador instala el motor y puede atender un promedio de 60 automóviles/h. Después de instalar el motor, el automóvil pasa a la estación de los neumáticos y espera para que le pongan los neumáticos. En esta estación trabajan tres personas. Cada una trabaja en un automóvil a la vez y puede colocar neumáticos en un automóvil en un promedio de 3 minutos por cada auto. Los tiempos entre llegadas y los de servicio son ambos exponenciales. El problema consiste en determinar:

1. Longitud promedio de la cola en cada estación de trabajo.
2. Tiempo total esperado que pasa un automóvil esperando su turno.

Para que el Sistema de Optimización de Recursos pueda resolver este problema, es necesario capturarlo como dos sistemas de líneas de espera, uno con un servidor y otro con tres servidores, dando como resultado, de manera impresa los resultados mostrados en las siguientes hojas, de donde se puede observar que:

La longitud promedio de la cola en la etapa 1 es de 8.1 elementos.

La longitud promedio de la cola en la etapa 2 es de 7.354 elementos.

Tiempo total esperado que pasa un automóvil esperando su turno es de $0.15 + 0.136 = 0.286$ horas

SISTEMA DE OPTIMIZACION DE RECURSOS
01-ENE-97
LINEAS DE ESPERA (UNA FILA - UN SERVIDOR)

VERSION 1.0

PRIMERA ETAPA DEL ENSAMBLE FINAL DE AUTOMOVILES

Sistema de líneas de espera en serie.

Tasa de llegada = 54 automóviles / hora.

Tasa de servicio = 60 automóviles / hora.

Número de servidores = 1

Tasa de llegada:	54.00
Tasa de servicio:	60.00
Servidor desocupado:	10.00 %
Intensidad de tráfico:	90.00 %
Elementos en el sistema:	9.000
Elementos en espera:	8.100
Elementos en servicio:	0.900
Tiempo promedio en el sistema:	0.167
Tiempo promedio en espera:	0.150
Tiempo promedio en servicio:	0.017

SISTEMA DE OPTIMIZACION DE RECURSOS
01-ENE-97
LINEAS DE ESPERA (UNA FILA - SERVIDORES EN PARALELO)

VERSION 1.0

SEGUNDA ETAPA DEL ENSAMBLE FINAL DE AUTOMOVILES

Sistema de líneas de espera en serie.

Tasa de llegada = 54 automóviles / hora.

Tasa de servicio = 20 automóviles / hora.

Número de servidores = 3

Tasa de llegada:	54.00
Tasa de servicio:	20.00
Número de servidores:	3
Intensidad de tráfico:	90.00 %
Elementos en el sistema:	10.054
Elementos en espera:	7.354
Elementos en servicio:	2.700
Tiempo promedio en el sistema:	0.186
Tiempo promedio en espera:	0.136
Tiempo promedio en servicio:	0.050

CONCLUSIONES

El resultado principal de este proyecto fue la obtención de una herramienta basada en computadora, en la cual se encuentran implementados un conjunto de algoritmos del área de Investigación de Operaciones, y que puede ser utilizada para ayudar a resolver algunos de los problemas que comúnmente se encuentran en la operación normal de las plantas industriales. Esta herramienta está diseñada para resolver problemas reales, de grandes dimensiones, y presenta la información de una forma profesional para su mejor interpretación.

Además, puede ser de gran utilidad en el campo académico, pudiéndose utilizar como herramienta de apoyo en cursos de Investigación de Operaciones, en donde el enfoque sea la aplicación de los modelos, y se requiera comprobar las técnicas y los métodos de los algoritmos.

Por otro lado, la documentación del proyecto, representa el puente entre dos áreas de gran interés en nuestros días, la Investigación de Operaciones y la Ciencias de la Computación. Aunque tradicionalmente estas dos áreas se han desarrollado de manera paralela y de forma conjunta existe muy poca información en como se pueden relacionar, como la computadora puede ser utilizada para resolver problemas como los de Investigación de Operaciones, ya que por un lado se encuentra una extensa bibliografía de Investigación de Operaciones pero incluyen muy poco de programación de este tipo de algoritmos; por el otro lado se encuentra otra gran cantidad de bibliografía de programación de computadoras en diferentes lenguajes de alto nivel, pero presentan ejemplos sencillos que no permiten apreciar ni comprender los métodos que se requieren para programar algoritmos complejos como son los de Investigación de Operaciones.

Finalmente, los programas desarrollados forman una librería de procedimientos que pueden ser utilizados en aplicaciones específicas, enriqueciendo de esta manera al mismo lenguaje de programación en donde fueron desarrollados ya que solo es necesario entender como son utilizados y cualquier programa puede utilizarlos si se compilan y enlazan de manera conjunta.

El costo de desarrollo del Sistema de Optimización de Recursos, equivale al costo de 1920 horas-hombre, de investigación, análisis, diseño, programación, pruebas y documentación. Los beneficios al utilizar este sistema en una sola empresa, de tamaño mediano, resolviendo un sólo problema, como por ejemplo el de la obtención de las mezclas de productos, compensa el costo de desarrollo y reporta grandes beneficios con el abatimiento de los costos de producción en hasta dos meses, considerando un buen volumen de producción y varias líneas de productos, tal como reducir el costo unitario de producción en \$ 1.00 (un peso) para un volumen de producción de 50,000 unidades mensuales, situación factible para empresas medianas de manufactura repetitiva. Este tipo de análisis debe hacerse, definitivamente, en cada empresa en particular, ya que la implementación en cada caso, tendrá un impacto diferente; sin embargo se puede apreciar que para una gran gama de industrias en México se justifica la implementación, sobre todo aquellas que cuentan con un conjunto de alternativas, tanto de producción, comercialización y/o distribución, y que no cuenten con la operaciones óptimas en cada una de sus áreas.

El uso de estos modelos en una organización incrementa enormemente su eficiencia para hacerla más competitiva, por medio de los siguientes beneficios: se incrementa la posibilidad de tomar mejores decisiones, se elaboran mejores planes de producción, se mejora la coordinación entre los múltiples componentes de la organización, se mejora el control del sistema, se logra un mejor sistema al hacer que éste opere con costos más bajos y se reducen los tiempos de respuesta al mercado, entre otros. Todos son puntos claves en la carrera por la ventaja competitiva y pueden ser la diferencia entre la desaparición y el crecimiento de una organización.

BIBLIOGRAFIA

- 1.- **Ackoff/ Sasieni.**
Fundamentos de Investigación de Operaciones.
Editorial Limusa.
Primera Edición.

- 2.- **Computers Associates, Inc.**
CA-Clipper versión 5.3
Reference Manual
Volumen I y II

- 3.- **Computers Associates, Inc.**
CA-Clipper versión 5.3
User's Guide

- 4.- **Davis / McKeown.**
Modelos Cuantitativos para Administración.
Grupo Editorial Iberoamérica.
Primera Edición.

- 5.- **Fogarty / Blackstone / Hoffman.**
Administración de la Producción e Inventarios.
APICS / CECSA
Segunda Edición.

6.- Monks, Joseph G.

Administración de Operaciones.

McGraw Hill / Hispanoamericana de España, S.A.

Primera Edición.

7.- Prawda Witenberg, Juan.

Métodos y Modelos de Investigación de Operaciones.

Editorial Limusa.

Primera Edición.

8.- Pressman, Rogger S.

Ingeniería de Software.

McGraw / Hispanoamericana de España, S.A.

Segunda Edición.

9.- Wirth, Nicklaus.

Algoritmos y Estructuras de Datos.

Prentice Hall Hispanoamericana.

Primera Edición.

10.- Winston, Wayne L.

Investigación de Operaciones.

Grupo Editorial Iberoamérica.

Segunda Edición.