

Nº 9  
2 EJ.



UNIVERSIDAD NACIONAL AUTONOMA  
DE MEXICO

FACULTAD DE CIENCIAS

SELECCION Y EVALUACION DE UN PAQUETE  
DE SOFTWARE MRP II PARA UNA EMPRESA  
MANUFACTURERA

**T E S I S**

QUE PARA OBTENER EL TITULO DE

A C T U A R I O

P R E S E N T A :

FRANCISCO J. CABRERA PEREZ



MEXICO, D. F.

1992

FALLA DE ORIGEN



Universidad Nacional  
Autónoma de México



## **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 está 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.

**SELECCION Y EVALUACION DE UN PAQUETE DE SOFTWARE  
MRP II PARA UNA EMPRESA MANUFACTURERA.**

**C O N T E N I D O**

Pag

<b>1. INTRODUCCION.</b>	
1.1 Sistema estándar de manufactura.....	5
1.2 El papel del actuario en la industria manufacturera.....	7
<b>2. LA MECANICA DE MRP II Y EL SISTEMA AUTOMATIZADO.</b>	
2.1 ¿Que es el sistema MRP II?.....	9
2.2 ¿Que hace que un sistema automatizado funcione?.....	20
<b>3. LA SELECCION DEL SOFTWARE.</b>	
3.1 Objetivos básicos para la selección del software.....	28
3.2 ¿Como obtener los objetivos de la selección?...	34
<b>4. EL PROCESO DE EVALUACION.</b>	
4.1 Reducir la lista de candidatos.....	46
4.2 Evaluación funcional del paquete.....	50
4.3 Aspectos técnicos a considerar.....	59

<b>5. UNA APLICACION A UNA INDUSTRIA DE PRODUCTOS</b>	
<b>FARMACEUTICOS.....</b>	<b>74</b>
<b>6. CONCLUSIONES.....</b>	<b>99</b>
<b>GLOSARIO.....</b>	<b>101</b>
<b>BILIOGRAFIA.....</b>	<b>104</b>

## 1. INTRODUCCION.

En la década de los 60's se tiene una de las etapas más decisivas cuando, en la industria de manufactura americana, surge un sistema para controlar los materiales por medio de un análisis desfasado en el tiempo. Este sistema de control se llamo MRP (Material Requirements Planning) ó Planeación de los Requerimientos de Materiales y vendria a ser con el tiempo el corazón de otro sistema administrativo con una cobertura más amplia.

Este otro sistema nace en la década de los 70's al entenderse la necesidad y conveniencia de contar con una planificación y un control, pero ya no sólo de los materiales que van a ser utilizados en la fabricación de productos, sino de todo el proceso de manufactura. Así nace el MRP II (Material Resource Planning), Planeación de los Recursos de Manufactura, un medio para poder planear los objetivos de una empresa, comunicarlos y medirlos mediante una continua retroalimentación de información entre todas las áreas del total de la empresa.

Con el tiempo, el MRP II se perfecciona, alcanzando reducir cada vez más las diferencias entre lo planeado y lo realmente obtenido, hasta que después de una década, se logra superar el 95% de acierto en estas actividades, con lo cual se creó la denominada Clase "A" de MRP II.

Hoy en día las empresas de manufactura están conscientes de que tener un sistema MRP II significa un incremento en la eficiencia, menos desperdicio y mayores utilidades. Desafortunadamente, mucha gente asume que para adquirir estos beneficios, todo lo que se tiene que hacer es comprar el paquete adecuado de software y esperar a que el sistema haga todo el trabajo. MRP II es una herramienta esencial, no una píldora mágica para eliminar los retos y el duro trabajo de las operaciones de manufactura. Más aun, manejar efectivamente las herramientas de un sistema MRP II no sucede por si solo. La excelencia en la calidad, estabilidad en el manejo de una compañía, el liderazgo positivo y los valores consistentes de una organización no se dan porque sí; todo es resultado de los esfuerzos de gente dedicada y preparada.

El proceso del MRP II se basa en la teoría de las RUTAS PARALELAS:

Por un lado se maneja la RUTA GERENCIAL. En ella se encuentran consideradas todas las actividades desarrolladas por el factor humano de una empresa y sin ésta vinculación no se considera factible una operación con un grado de eficiencia que corresponda a una empresa Clase "A".

La otra ruta es la denominada RUTA COMPUTARIZADA, es decir, todo lo relacionado al proceso y manejo de información por medios computarizados, tanto hardware como software. Es a través de esta ruta que se obtiene la información de apoyo a los componentes del

nivel operativo mediante información actualizada y precisa de lista de materiales, hojas de proceso, posición de inventario y rutas, entre otra.

#### 1.1 Sistema estándar de manufactura.

Así como en contabilidad los conceptos de entrada y salida son universales, existe una lógica para planear, controlar y coordinar las actividades de manufactura. Esta lógica es una simulación de la ecuación universal de manufactura. ¿Que vamos a hacer?. ¿Qué se necesita para hacerlo?. ¿Qué tenemos?. ¿Qué deseamos obtener?. Estas preguntas tienen que ser contestadas por cualquier empresa de manufactura y se les da respuesta mediante un sistema estándar:

- Elaboración del plan de comercialización.
- Elaboración del plan de operación y ventas.
- Elaboración del plan maestro de producción.
- Elaboración del plan de requerimiento de materiales.
- Elaboración del plan de requerimientos de capacidad.
- Ejecución del plan de capacidad.
- Ejecución del plan de materiales.
- Retroalimentación.

El plan de comercialización define la misión de la compañía, su mercado, ganancias y recursos financieros. Establece, en pesos, cuánto es lo que la compañía espera vender, qué planea producir, la

cantidad de dinero que será invertida en investigación y desarrollo y la cantidad de utilidades que espera tener.

El plan de operaciones y ventas establece (por familia de productos en unidad) la manera en que el plan de negocios se va a llevar a cabo. El plan de ventas establece cuotas de ventas; estas cuotas generalmente son por familia de productos importantes.

El plan maestro de producción es la cantidad de la producción en términos de productos terminados. La producción maestra para cada producto dentro de una familia está restringida por el plan de producción por familia. El total de la producción maestra debe de ser igual al plan de producción.

El plan de requerimientos de materiales descompone el plan maestro de producción en partes individuales para compra, fabricación y ensamble de todos los componentes.

El plan de requerimientos de capacidad traduce el plan de requerimientos de materiales en requerimientos de capacidad para cada centro de trabajo, fuentes de trabajo, herramientas, etc.

La retroalimentación comunica los problemas para ejecutar los planes y provee el método para cerrar el ciclo en el sistema. La retroalimentación es necesaria cuando los planes no se pueden ejecutar ó si se ejecutan deficientemente y deben de ser revisados.



## 1.2 El papel del actuario en la industria manufacturera.

El actuario asesora a la industria de manufactura con el manejo de las estadísticas, la evaluación de inversiones, así como en establecer procedimientos y sistemas automatizados para que la compañía tenga un mejor nivel de servicio respecto a los productos que fabrica, menos gastos de operación, más productividad y por lo tanto, alcanzar un nivel de calidad en sus actividades.

Existen 3 razones del por qué de este trabajo:

- 1) Aunque el software es sólo una parte de un sistema eficiente de MRP II, es un componente crítico. Muchos de los esfuerzos del MRP II pueden anularse por problemas de software. Un software confiable puede eliminar muchos de los obstáculos técnicos más comunes para alcanzar un alto rendimiento.
- 2) Ningún otro aspecto en la implantación del MRP II genera más expectativas y requiere más preparación que seleccionar y evaluar un software.
- 3) Para la mayoría de las compañías de manufactura el proceso de selección, evaluación y compra de software representa la parte más costosa en la implantación de un sistema MRP II.

Muchas compañías seleccionan y evalúan su software sin entender verdaderamente qué se debe de hacer para que el MRP II trabaje efectivamente. Como resultado, y así lo han demostrado varios estudios, el costo de la decisión en la compra de software casi siempre excede el costo real del software por un factor de 5.

El presente trabajo intenta dar una discusión comprensiva de los puntos que se necesitan considerar para determinar cuáles programas de software serán los más adecuados para satisfacer las necesidades de una compañía, evadir errores comunes que pueden llevar a retrasos costosos y determinar cuáles funciones en el software no se encuentran, necesitan ser modificadas o deben ser eliminadas.

## 2. LA MECANICA DE MRP II Y EL SISTEMA AUTOMATIZADO.

### 2.1. ¿Que es el sistema MRP II?

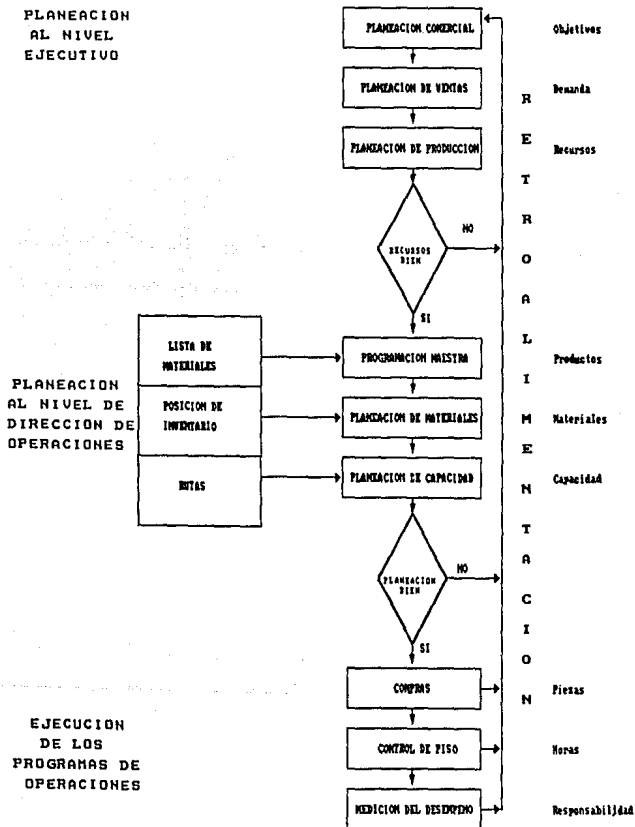
Para poder evaluar un software, lo primero que hay que entender es qué es el sistema MRP II. El ciclo cerrado de un sistema MRP II toma la programación maestra "¿Qué es lo que vamos a hacer?", explotándola a través de la lista de materiales "¿Qué necesito para hacerlo?" y lo compara con el inventario actual y ordenado "¿Qué es lo que tenemos?" para determinar los requerimientos de materiales "¿Qué es lo que tenemos que obtener?".

El cuadro del sistema cerrado del MRP II (fig 1-1) representa los procesos gerenciales para dirigir una empresa fabril con éxito.

Cada proceso en el cuadro se usa como base del siguiente, de tal forma que la información se va cimentando en cada paso del camino. Se tiene que completar cada proceso, a su vez, antes de que se pueda iniciar el que sigue. Una vez completo el proceso entero de la Planeación de los Recursos de Manufactura(MRP II) se comunican los objetivos a través del proceso del Sistema Cerrado; la retroalimentación proporciona un método de control para cada área. El sistema cerrado de retroalimentación garantiza que se comunican los resultados a la empresa entera y que están al día las operaciones y planes de la empresa.

# PLANEACION DE LOS RECURSOS DE MANUFACTURA (MRP II)

FIG 1-1



El camino gerencial consiste en tres grupos de procesos relacionados. El primer grupo es la Planeación al Nivel Ejecutivo. La Planeación al Nivel Ejecutivo consiste en el plan mensual comercial, el de ventas y el de producción.

El grupo intermedio de procesos del camino gerencial supone la Planeación al Nivel de la Dirección de Operaciones. Aquí se elaboran cada semana los planes detallados para el programa maestro, planeación de materiales y planeación de capacidad.

El tercio inferior describe la Ejecución de los Programas de Operaciones que consta de compras, control de piso y medición del desempeño. Estos procesos se programan y se manejan a diario.

El camino computarizado comprende la carga y mantenimiento de la información en la base de datos. Esta información consiste de la lista de materiales, posición de inventario y rutas.

La clave del Sistema Cerrado es la retroalimentación. Cualquier cambio o decisión puede afectar los otros procesos. Se tienen que comunicar estos cambios a cada función para determinar el impacto y tomar las medidas necesarias. La retroalimentación mantiene los planes, operaciones y ejecución íntegros, actualizados y precisos.

Cada uno de estos tres grupos entrañan un sistema cerrado. Dentro de cada grupo debe de llevarse a término cada función antes de

poder comenzar el grupo que sigue.

#### PLANEACION AL NIVEL EJECUTIVO

La planeación y revisión, desde el nivel ejecutivo hasta el nivel de ejecución de operaciones son esenciales para lograr la excelencia en la manufactura. La mayoría de los planes del nivel ejecutivo abarcan 12 meses, se revisan cada mes y se modifican cada trimestre. El proceso de planeación es una actividad dinámica. Los planes tienen un impacto recíproco y en consecuencia, los antiguos planes necesitan actualizarse para responder a los cambios del medio comercial.

#### Planeación comercial.

La planeación al nivel ejecutivo parte del plan comercial. El plan comercial comprende:

- . los productos a fabricar
- . los mercados a atender
- . ganancias y objetivos financieros necesarios para satisfacer los objetivos comerciales globales de la empresa

Se determina el plan comercial al nivel de línea de productos.

### Planeación de ventas

La planeación de ventas es un proceso dinámico que requiere revisión y modificación mensual. Ante todo se analiza cada nuevo plan de ventas en función de su capacidad para cumplir con el plan comercial. Se van modificando el plan de ventas o el plan comercial hasta que ambos planes cumplen con los planes y objetivos globales de la empresa.

La planeación de ventas utiliza las metas acordadas en el plan comercial para verificar el plan de ventas. El plan de ventas comprende:

- . La demanda de los clientes necesaria para cumplir el plan comercial
- . La cantidad de unidades a vender
- . Los planes de mercadeo de la empresa

También se determina el plan de ventas al nivel de líneas de productos. Como se ve, cada proceso se cimienta sobre los resultados del anterior.

### Planeación de producción

El último proceso en el sistema del nivel ejecutivo, la planeación de la producción, utiliza el plan de ventas, conjuntamente con el

plan de inventario de productos terminados o planes de pedidos pendientes, para elaborar el plan de producción. El plan de producción expresa la tasa de producción global por líneas de productos.

El plan de producción precisa la tasa de producción prevista, la línea de productos, con una perspectiva de 12 meses. Se revisa y actualiza cada mes.

Existen dos fórmulas simples que ayudan a llevar a cabo los planes de producción. Una fórmula para productos hechos por inventario, la cual controla la producción basada en el inventario actual y deseado. La otra fórmula para productos hechos por orden, la cual controla la producción basada en el "backlog" actual y deseado. Ambas formulas son similares y pueden ser representadas de la siguiente manera:

$$\text{Plan de producción} = \text{Plan de ventas} + \text{ajustes en backlog ó inventario}$$

De otra manera:

$$\text{Inventario} = \text{Inventario inicial} - \text{Plan de ventas} + \text{Plan de producción}$$

\*Backlog = Backlog inicial - Plan de ventas + Plan de producción.

\*ver glosario



Ya completo el sistema del nivel ejecutivo, el plan comercial proporciona información para elaborar el plan de ventas que, a su vez, sirve para crear el plan de producción.

#### PLANEACION A NIVEL DE LA DIRECCION DE OPERACIONES

Ya puede ponerse en marcha el sistema de la Dirección de Operaciones. A diferencia de los Planes del Nivel Ejecutivo, los planes del proceso de la Dirección de Operaciones se actualizan y revisan con una frecuencia mayor (una vez a la semana). El sistema de la Dirección de Operaciones controla:

- . la mezcla de productos a producir, es decir, los artículos específicos
- . los materiales necesarios para producir los productos
- . la capacidad que se requiere para producir los productos

#### Programación maestra

La dirección de operaciones parte del programa maestro. El programa maestro precisa qué habrá de producirse por artículo específico cada semana.

#### Planeación de materiales

El programa maestro proporciona la información para la siguiente

función, la planeación de materiales. El plan de requisitos de materiales determina qué materiales se requieren y cuándo se requieren, de modo que estén disponibles para atender el programa maestro.

#### Planeación de capacidad

La planeación de capacidad constituye el último paso en el sistema de la Dirección de Operaciones. La planeación de capacidad garantiza que los recursos de la planta y el personal estén disponible para cumplir el programa maestro.

#### LA BASE DE DATOS COMPUTARIZADA

La base de datos computarizada da apoyo al Proceso del Sistema Cerrado guardando en la computadora datos tales como:

- . Lista de materiales
- . Posición de inventario
- . Circulación de materiales (Rutas)

Se almacenan y mantienen en la computadora gran parte de la información necesaria para dirigir una empresa fabril. Esta información tiene que ser exacta y oportuna de modo que puedan tomarse las decisiones gerenciales más acertadas posibles. La actualización de la base de datos es una actividad importante.

## EJECUCION DE LOS PROGRAMAS DE OPERACIONES

Tras haberse revisado y verificado todos los aspectos del sistema de planeación al nivel de la Dirección de Operaciones, puede iniciarse la ejecución de los programas de operaciones. La función de la ejecución de los programas de operaciones es llevar a efecto los planes elaborados en la planeación a nivel ejecutivo y al nivel de la Dirección de Operaciones. La ejecución de los programas de operaciones consiste en:

- . obtener las piezas o materiales para fabricar los productos (Compras)
- . desempeñar la labor de fabricarlos (Control de piso)
- . medir el desempeño y verificar responsabilidades

Se actualizan y revisan a diario las funciones de la Ejecución de los Programas de Operaciones.

### Compras

La ejecución de los Programas de Operaciones empieza con las compras. Las compras consisten en:

- . controlar la capacidad y prioridades de los proveedores
- . informar sobre los estados de órdenes de compra y actualizarlos

- . cumplir con los objetivos de calidad, de entrega y de costo

Controlando las compras, se dispone de los materiales necesarios para cumplir con la proporción de productos del Programa Maestro semanal y del plan mensual de producción.

#### Control de piso

El control de piso es la ejecución de los programas de fabricación. En el control de piso convergen el plan de materiales, la proporción de productos del programa maestro, y el plan de capacidad, de tal forma que se cumplen los planes de producción del día.

Los tres factores críticos del control de piso son:

- . la disposición física de la planta industrial
- . el control de capacidad
- . las secuencias y prioridades

#### Medición del desempeño

Cuando se mide el desempeño, se evalúa el desempeño real y se compara con el nivel de desempeño previsto. La medición del desempeño se inicia estableciendo objetivos, planeando cómo cumplir

esos objetivos y asignando responsabilidades. Luego hay que acordar los métodos para medir el desempeño. Por último se mide el desempeño y se retroalimentan los resultados del sistema para tomar acción correctiva. El proceso de medición de desempeño requiere que haya quien responda del desempeño en cada segmento del MRP II. Esta responsabilidad constituye un elemento importante para lograr un desempeño de clase A.

La medición del desempeño cierra al proceso gerencial verificando el desempeño de la operación real de cada función y comparándolo con el objetivo de su función.

La medición del desempeño es de acuerdo a:

Medición de la exactitud del registro del inventario. La mínima exactitud del inventario necesaria para operar un sistema MRP II es del 95% . Esto significa que el 95% de los productos en inventario deben estar exactos dentro de su tolerancia de conteo.

Medición del desempeño de la producción maestra. El desempeño necesario para la producción maestra es del 95%. Es decir el 95% de todas las órdenes maestras producidas en un período de tiempo debe de ser entregadas a tiempo.

Medición de desempeño de la manufactura. El 95% de las ordenes de manufactura deben de estar a tiempo.

Medición del desempeño de los distribuidores. El desempeño de entrega de los distribuidores es similar al de manufactura. El objetivo de una compañía es adquirir por lo menos un 95% de todas las órdenes de compra entregadas a tiempo.

Estos parámetros de medición son establecidos por este sistema de trabajo, aunque es importante mencionar, que los parámetros más importantes para MRP-II, son la exactitud del inventario (95%), la exactitud de la lista de materiales (98%) y la exactitud de las rutas (95%). Los demás parámetros pueden ser establecidos por la compañía, de acuerdo a los objetivos establecidos.

## 2.2. ¿Qué hace que un sistema automatizado funcione?.

Antes de comenzar a seleccionar y ver distribuidores de software para MPR II, hay que entender por qué algunos sistemas automatizados funcionan y otros no. En su más simple expresión, los sistemas efectivos son aquéllos que apoyan los esfuerzos humanos. En el transporte ayudan a guiar aviones rápido y con seguridad a sus destinos. En las compañías fabriles, los sistemas efectivos ayudan a minimizar el inventario y a producir bienes y servicios tan eficientemente como sea posible.

Todo sistema efectivo, ya sea para contabilidad, control de tráfico aéreo o para MRP II, consta de varios elementos: la gente, los datos y las herramientas (hardware y software). La gente es lo más importante, después los datos y por último las herramientas. El poder de la computadora recae en la capacidad para apoyar las tareas básicas que la gente realiza con volúmenes muy grandes de información, particularmente cuando estos volúmenes están más allá de las capacidades de los sistemas manuales. Además de la gente, hay cuatro factores que determinan la utilidad de cualquier sistema automatizado: (1) la integridad de los datos, (2) la validez de la simulación, (3) la oportunidad de los resultados y 4) el grado en el cual el sistema le muestra al usuario cómo llegó a cierto resultado en un tiempo dado (esta propiedad se conoce como "transparencia").

## FACTORES HUMANOS

### Entendimiento.

En sistemas y en procesamiento de datos, el ciclo tradicional para desarrollar un sistema automatizado (Diseño lógico, Diseño físico, Programación, Pruebas, Implementación y Operación), se considera no muy efectivo. Mucha gente le echa la culpa a la cantidad de tiempo entre el Diseño lógico (entrevistas con el usuario) y la implantación, debido a que en ese lapso el usuario generalmente cambia los requerimientos. El problema con el ciclo tradicional para desarrollar un sistema, no es que los usuarios cambien constantemente sus requerimientos, sino que en la mayoría de los casos ellos mismos no entienden lo que realmente desean.

La solución a este problema no está en la gente de informática, la cuál no debe tomar decisiones sobre lo que debe o no tener un sistema. La solución es educar a los usuarios acerca de las herramientas que están disponibles para realizar sus actividades diarias. Los usuarios son los más adecuados para comprender los problemas relacionados con su trabajo. Si se encuentran apropiadamente informados de la lógica que hay detrás de estas herramientas, podrán evaluar e implantar soluciones efectivas con ellas.

La necesidad de que la gente comprenda la lógica de un sistema es un argumento poderoso para que el sistema sea sencillo de manejar.



Otro argumento es el que la gente encuentra difícil entender como todas las diferentes características, opciones y códigos trabajan juntos para producir resultados o recomendaciones.

#### Pertenencia

Además de comprender la lógica del sistema, la gente debe de tener un sentimiento de que el sistema les pertenece, esto hará que el sistema les ayude a hacer mejor su trabajo. Cualquiera que sienta que el sistema le pertenece, será responsable por los resultados que genere. Es el mismo sentimiento y satisfacción que un autor, artista o diseñador tiene de su trabajo.

Por otro lado si el usuario del sistema no tiene el sentimiento de pertenencia, será muy dado a pensar que es sólo un sistema de procesamiento de datos, o que es un sistema para el departamento de manufactura, y no tendrá una actitud para producir resultados, aunque el sistema tenga la lógica correcta o los datos sean confiables.

Al implantar un nuevo sistema, la mejor alternativa para elegir a la persona encargada del proyecto es el usuario final. Por ejemplo, cuando se implanta un sistema de contabilidad, una persona del departamento de contabilidad, no del departamento de ventas o mercadotecnia debe de estar a cargo. De la misma manera, cuando se implanta un sistema MRP II los miembros del departamento de sistemas no deben ser los responsables.

Muchas compañías, por ejemplo, cuando implantan un sistema automatizado para MRP II cometen uno ó mas de los siguientes errores: (1) contratan consultores para que implanten el sistema, (2) responsabilizan a la gente del departamento de informática para que el sistema se eche a volar y (3) asignan al departamento de Recursos Humanos, la tarea de educar a los usuarios. Cada uno de estos errores forzosamente causa mayores problemas.

Durante la implantación y operación de un sistema automatizado, hay maneras de promover la pertenencia del sistema por parte de los usuarios, así como definir responsabilidades. Una de estas maneras es tener programas de seguridad en el software, de tal modo que solamente le permita a cierta gente manejar y actualizar la información.

#### Responsabilidad

La responsabilidad en el manejo de un sistema automatizado es muy importante. No se le puede dejar el control total al sistema, sin verificar que los resultados que se producen sean los correctos. Por esta razón, no se debe de apartar a la gente de la etapa de aprobación en los sistemas tales como el control de plantas nucleares y del control del espacio aéreo.

Cuando se diseña un sistema automatizado, se plantean las siguientes preguntas:

1. ¿El software posee la capacidad de evaluar si el tomar o no esta acción tiene sentido en todas las circunstancias?

2. ¿Es el software capaz de determinar si la decisión debe de llevarse a cabo en todas las circunstancias? ( ¿será la gente la que es responsable para llevar a cabo la decisión? )

A continuación se listan ciertas características que se encuentran comúnmente en software para MRP II, que pueden causar problemas de responsabilidad :

1. Generación automática de la programación maestra.
2. Reprogramación automática.
3. Liberación automática de órdenes.
4. Reglas sofisticados de orden.
5. Lógica de carga finita\*.
6. Cálculos estadísticos de proyección que carecen de revisión y aprobación antes de actualizar la información.

La conclusión es que en cualquier sistema, los usuarios deben de tener el control directo sobre aquello que se debe de verificar.

#### Simulación de la realidad

Antes de automatizar cualquier proceso, es importante hacernos la siguiente pregunta: ¿Qué es lo que realmente debe suceder?. Uno de los errores más comunes que se llega a cometer cuando se

automatiza un proceso por primera vez, es asumir que el objetivo fundamental es producir resultados. Esta posición es equívoca, debido a que muchas veces el sistema manual actual no es correcto y se están produciendo errores.

#### FACTORES DE UN SISTEMA.

##### Integridad de los datos

En cualquier sistema hay datos que deben de ser exactos todo el tiempo y otros que son menos críticos. En un sistema MRP II, la planeación maestra debe de ser válida, las listas de materiales deben de estar correctas en un 98% y los registros en el inventario deben de estar por lo menos 95% correctos.

A pesar de que es obvio que los datos erróneos producen resultados incorrectos, la carencia de la integridad de los datos en áreas críticas es un gran problema que se presenta en varias compañías. Muy a menudo no están conscientes de que el problema existe y esto es verdaderamente desafortunado, ya que la integridad de los datos es lo más fácil de corregir de entre todos los factores de un sistema.

##### Tiempo de respuesta

Además de ser una simulación válida, un sistema puede ser efectivo solamente si realiza su simulación dentro de un tiempo razonable.

Un sistema de pronóstico de tiempo, por ejemplo, es de muy poco valor si tomara más de 24 horas para generar el pronóstico del día siguiente.

### Transparencia

Se puede tener un sistema que tiene la lógica correcta y produce resultados en tiempos razonables de un modo conceptual, pero si carece de "transparencia" ó "visibilidad" es de uso limitado. La transparencia le da al usuario la habilidad de ver el sistema y contestar la pregunta: ¿Por qué generó ese número esta vez?.

El poder comprender cómo cualquier sistema llega a cierto resultado es crítico, porque habrá situaciones en las cuáles no se hará lo que el sistema recomienda. Si esto sucede un sistema efectivo debe:

- (1) Proveer una manera para entender por qué la recomendación fue hecha.
- (2) Atenerse a una persona para evaluar la situación, tener una decisión e implantar una solución diferente y
- (3) Descubrir problemas en la solución e identificarlos.

A menos que un sistema sea transparente, podrá tener problemas de responsabilidad. Como un ejemplo, en MRP II, una recomendación común es un mensaje de excepción: "Reprogramar esta orden para una fecha más cercana". Si las personas a cargo de la planeación no pueden determinar por qué el sistema les está enviando este mensaje, no podrán cumplir con la recomendación. ¿Cómo entonces se les puede hacer responsables de programar desde MRP II?.

Una solución a este problema es el trazar el origen del requerimiento de regreso a la orden de manufactura, del cliente o del pronóstico que lo causó.

Como se podrá ver, la transparencia es un aspecto vital de un sistema efectivo. A menos que la responsabilidad para las decisiones y su ejecución pueda ser transferida completamente a la computadora, tendrá que ser llevada a cabo por la gente. Pero hasta que alcancemos tal utopía, los sistemas deben de proveer los medios para que la gente comprenda la lógica y señale la fuente de sus recomendaciones

### 3. LA SELECCION DEL SOFTWARE.

#### 3.1 Objetivos básicos para la selección del software.

Mientras que el objetivo de la búsqueda de un software es encontrar aquél que le permita a una compañía operar un sistema MRP II, hay algunos principios que deben de ser resueltos como parte de la decisión, por ejemplo, los de las modificaciones que se tendrán que llevar a cabo, las interfases que tendrán que ser construidas y determinar quién las llevará a cabo. Tales interfases son realmente significativas: Un paquete MRP cuya interfase con el sistema financiero de una compañía tuviera una vida útil de 5 años es de muy poco valor, aun si pareciera un sistema excelente desde otras perspectivas.

Hay también el cuestionamiento de la utilidad y valor recibido. ¿Cómo se puede medir la utilidad?. Si no se tiene un método para llevar a cabo tal determinación, desde un principio se tendrá dificultad para comparar dos paquetes, cuando ofrecen diferentes configuraciones, funciones, características, opciones, códigos y capacidades a diferentes precios.

Finalmente ¿cómo se puede estar seguro de que lo que se ve es realmente lo que se obtiene?. Si no se identifican funciones que pueden ser específicas y ligadas a un contrato, se tendrá una masa de código inútil y muy cara. Es esencial comprender los objetivos

básicos del proceso de evaluación para evitar retrasos. La experiencia sugiere que varios de los problemas en el proceso de evaluación y en la fase de instalación pueden ser evitados si cada una de las personas asociadas con el proyecto comprenden y están de acuerdo en 4 objetivos básicos:

1. Encontrar un software utilizable para MRP-II

Muy a menudo la gente se entusiasma en defender su paquete favorito. Se pueden crear bandos, olvidando que el objetivo es obtener un software funcional en estos momentos, en vez del mejor paquete después de 5 años de iniciar la búsqueda. Antes y después de la selección es mejor no elogiar las virtudes de un software. La gente tiene que estar de acuerdo en que el software servirá para las necesidades de la compañía y que será usado para llevar a cabo sus actividades.

El peligro de permitir que preferencias personales guíen el proceso de selección se ilustra con la experiencia de la compañía Arnold Company (U.S.A), una compañía manufacturera de equipo pesado, la cual había tenido la experiencia de operar MRP II clase A en tres de sus plantas y decidió implantarlo en una cuarta. Al mismo tiempo, tenía que decidir si debería de modificar el software existente o comprar un nuevo paquete. El análisis rápidamente se convirtió en un problema político y el proceso de selección se atrasó por más de un año. Las partes en disputa perdieron de vista



los objetivos, que eran encontrar un sistema funcional en el tiempo más corto. La parte más triste del asunto, es que a pesar de que un bando fue el ganador (el grupo buscando el nuevo software), la compañía fue al final la perdedora en términos de tiempo valioso y dinero.

## 2. Identificar modificaciones para funciones ausentes y los puntos de interfase para sistemas existentes

Se ha enfatizado el hecho de que no existe el paquete completo y que algunas modificaciones y grado de uso del sistema serán necesarios. El objetivo es determinar qué cambios y/o adiciones se tienen que llevar a cabo y quién las realizará. Esto también significa determinar si es factible que la compañía realice tales modificaciones, si se hará un contrato para que el proveedor las lleve a cabo o si se necesita encontrar recursos adicionales. En cualquier caso se tiene que averiguar cuánto costará y cuándo se puede realizar. Estas cuestiones deben de ser resueltas por un plan de programación de recursos.

La idea no es escribir especificaciones detalladas de cómo un problema en particular será resuelto, sino identificar sorpresas que de otra manera ocurrirán en el momento menos adecuado, a la mitad de la implantación.

A pesar de que las compañías sobreestiman el esfuerzo que las

modificaciones requieren (y como resultado retrasan la implementación varios meses), muy a menudo se sorprenden con la noticia de que se tienen que llevar a cabo varias modificaciones en el sistema.

Se debe conocer lo que se va a necesitar anticipadamente. La gente no puede dejar todo al último minuto, ya que se puede descubrir que el software no tiene algunas de las funciones que son indispensables.

### 3. Juzgar alternativas de software justamente y no en la base de costo contra utilidad

Este objetivo no es fácil de obtener, debido a que los paquetes pueden tener diferentes características, funciones, opciones y costos. ¿Es como comparar manzanas con naranjas?. No, si la comparación se basa en sus respectivos costos y atrasos. ¿Cuánto costará el tener cada uno con la funcionalidad de un sistema estándar MRP II?. El costo real de un software es:

Precio del paquete + Costo de las modificaciones + Costo del retraso

Comprender cómo analizar el costo del software puede ahorrarle mucho dinero a una compañía. Por ejemplo hace algunos años varias compañías compararon un paquete para MRP que costaba

aproximadamente \$250,000 dólares (\$150,000 por el software, y \$100,00 por el soporte) contra otro paquete que costaba \$60,000 (\$50,000 por el software y \$10,000 por el soporte). El software de menor costo era un sistema MRP regenerador\*, mientras que el más caro era un paquete MRP de cambio neto\*. Esta diferencia no justificaba la diferencia en el precio. Cuando las compañías realizaron evaluaciones detalladas , encontraron que el paquete menos caro ofrecía funciones más avanzadas.

De hecho en la mayoría de las situaciones donde dos paquetes fueron comparados contra la lógica estándar del MRP II y evaluados en términos de funcionalidades ausentes y en los costos asociados en el retraso para crear las características deseadas, las compañías concluyeron que los contendientes tenían casi el mismo grado de utilidad.

¿Qué era lo que causaba la enorme discrepancia en el precio? Algunos dijeron que el paquete más barato estaba por debajo de precio, mientras que otros apuntaban al paquete más caro como arriba de su precio. El desarrollador del paquete menos caro tenía una perspectiva interesante en la situación, de acuerdo a su vicepresidente del área técnica: "Los precios en el software son probablemente muy altos, subirán más por un período corto, pero eventualmente bajarán".

\*ver glosario

De hecho el precio en el paquete de \$250,000 dólares, eventualmente bajo a \$110,000 dólares. Sin embargo no se necesita esperar para obtener reducciones drásticas en el precio. Todo lo que una compañía necesita se puede encontrar en un paquete de menor precio que le de tanto ó más valor.

#### 4. Negociar un contrato mutuamente aceptable.

Una vez que se han identificado qué funcionalidad existe en el sistema y de que trabaja correctamente, lo que está ausente del paquete y como será corregido, el siguiente paso es pasar la información de los documentos de valuación y planes en un contrato. Por lo tanto una vez que se ha terminado la evaluación de una parte del software, se desea incorporar en el contrato las funciones que trabajan conforme a la lógica estándar para planear y programar, además de cuáles componentes específicos están ausentes y deberán ser añadidos.

Algunas personas objetarán diciendo que no es justo sorprender al distribuidor del software al final cambiando las reglas. Después de todo está ofreciendo un producto que concuerda con la documentación que se tuvo la oportunidad de revisar. El hecho es que no debe de sorprender al distribuidor del todo. Se dijo al principio que se desea la compra de un sistema para MRP II. El distribuidor propone algo y participa en la evaluación para determinar si iguala la lógica básica estándar para planear y programar. No debe de haber

ninguna sorpresa al final y el distribuidor debe de estar más que disponible para especificar qué herramientas el sistema actualmente incluye y cuáles no y escribirlo. El escribir qué herramientas son parte del sistema, da protección tanto al distribuidor como para el comprador, la compañía.

Finalmente en situaciones donde el distribuidor ha manifestado que una característica o función no es parte del software, pero que estará en algún futuro liberada, es importante que tales intenciones estén especificadas en el contrato.

### 3.2 ¿Como obtener los objetivos de la selección?

Para cumplir con los objetivos en la selección del software hay que verificar los siguientes puntos.

1. Nunca llevar a cabo una decisión de software hasta que la educación esté completa.

Uno de los grandes errores que una compañía que va operar MRP II ó cualquier otro sistema por primera vez puede cometer es buscar un software antes de completar una educación inicial. La gente que no ha participado en el proceso educativo tiene más preguntas acerca de cada una de las funcionalidades del software y por lo tanto se termina tomando decisiones de compra menos efectivas, menos informadas y que toman más tiempo en realizarse.

Al final, por supuesto, el resultado real no es que tan rápido cualquier decisión de compra se tomó, sino si una decisión informada fue realizada en el periodo de tiempo más corto posible.

Muchos de los problemas en evaluar software para MRP II radican en la carencia de educación por parte del usuario y en la manera en la cual el software es presentado. Se consume tiempo importante en revisar características y opciones en el software que no han sido probadas en la práctica o las cuales no son aplicables a las situaciones específicas de la compañía. En esta situación la carencia de conocimiento sobre MRP II genera confusión debido a que la gente comienza a debatir si estas opciones y características son esenciales para hacer que el sistema trabaje. Esta confusión lleva a un proceso largo de selección, la cual resulta en retrasos costosos en la implantación.

El proceso de educación no debe de durar más de lo necesario. A pesar de que el proceso de educación no hará más cortos los procesos de evaluación y selección, sí hará más corto el tiempo total y mejorará la calidad en la decisión, debido a que los usuarios están mejor informados.

## 2. Tener gente capaz en el equipo de selección/implantación.

Se han discutido con anterioridad las desventajas de llevar a cabo una selección rápida para elegir un software, así como la

importancia de dejar la responsabilidad para su implantación a aquéllos que lo seleccionaron. Asumiendo que las personas que seleccionaron el software serán también las encargadas de su implantación. ¿Qué nivel debe de tener el personal?. Varias compañías cometen el error de incluir gente que está debajo del nivel de supervisión o gerencial. Esto usualmente fracasa debido a que MRP es un cambio de comportamiento dentro de una organización y varios estudios han determinado que la única gente capaz de realizar cambios profundos y efectivos son aquéllos percibidos como líderes.

3. Dejar a la gente del departamento de sistemas coordinar los esfuerzos de la evaluación

Mientras que el departamento de procesamiento de datos no debe tener la responsabilidad de seleccionar el software, debe de ser, sin embargo, el responsable de contactar a los distribuidores, coordinar las presentaciones, visitas de distribuidores, etc, y debe de dar una recomendación basada en la tecnología usada en el paquete. Si la compañía cuenta con un equipo , sistema operativo y software de manejo de base de datos, el departamento de sistemas debe de obtener un software compatible con todo esto. Debido a que la mayoría de las compañías se sienten más seguras de que el departamento de sistemas haga el contacto inicial con los distribuidores de software, se le tiene que dar la responsabilidad de reducir el universo de posibles distribuidores.

#### 4. No pretender más allá de lo necesario

Es mejor evitar la tendencia de comenzar con el sistema más elegante. Se tiene que poner mayor atención a aquellas interfases y módulos que son esenciales para MRP II.

#### 5. No desperdiciar, si es posible, sistemas existentes

Varios de los desarrolladores de software asumen que las compañías están en posición de cambiar sus sistemas existentes (Lista de materiales, inventario, Pronósticos, etc), aún si se encuentran trabajando. En una compañía grande, el deshacerse de sistemas que trabajan es una experiencia desagradable que requiere volver a educar a todos los usuarios y que puede retrasar los beneficios de usar MRP. Además, toda compañía tiene "bolsillos de excelencia", subsistemas que funcionan extremadamente bien. Podría ser el subsistema de La Producción Maestra, un sistema de pronóstico, o un sistema de control de ingeniería. En cualquier caso, hay que ver qué es lo que se puede utilizar antes de invertir en todo un nuevo sistema.

Por otro lado hacer que el sistema que se está utilizando actualmente funcione puede traer más problemas. Si este sistema tiene la lógica errónea, la única alternativa es reemplazarlo.

No hay que olvidar el objetivo principal: obtener un sistema



efectivo MRP II en un mínimo de tiempo, ya sea modificando los sistemas actuales o trayendo un nuevo paquete al ambiente.

6. Evitar documentación extensa, apoyándose en las descripciones estandarizadas

Si existe un hoyo negro para la implantación de MRP II o de cualquier sistema, es la necesidad de documentar todo. Esto puede llevar a un retraso del proyecto y por lo tanto no se obtienen los beneficios del MRP en el menor tiempo posible. Se necesita evaluar el sistema existente para ver si existen módulos que se puedan utilizar. ¿Cómo se puede llevar a cabo sin realizar una documentación extensa?. La solución es usar un estándar de las funciones para MRP II como la base de cualquier análisis de software. La pregunta entonces se convierte en: ¿Cómo llevamos a cabo cada una de estas funciones?. ¿El sistema actual tiene dichas funciones?. Si no es así, ¿Cómo lo podemos modificar?. ¿Qué paquete externo sería el idóneo?. Tomando este enfoque al problema se tendrá la mejor oportunidad de determinar las necesidades con un mínimo de documentación.

7. Diferenciar las distintas fuentes para adquirir el software.

Un software para MRP II puede ser obtenido de tres fuentes: 1) Distribuidores de software, 2) Fabricantes de hardware y 3) del

Departamento de procesamiento de datos de la compañía. Las primeras dos categorías tienen ventajas y desventajas. La tercera tiene problemas significativos cuando el software está desarrollado de la nada, pero puede ser una alternativa excelente cuando hay sistemas trabajando y todo lo que hay que hacer es añadir módulos no existentes.

a) Fabricantes de Hardware

Los mayores fabricantes de hardware y también distribuidores de software son: IBM , Hewlett-Packard, Unisys, NCR, y Digital. La mayoría de ellos ofrecen software diseñado para correr en sus propios sistemas.

Ventajas:

- 1.- El software está garantizado que corra en el equipo del fabricante.
- 2.- El mismo distribuidor es responsable tanto por el software como por el hardware.
- 3.- El precio del software puede ser significativamente más bajo que el software de las otras fuentes.

### Desventajas:

- 1.- Los fabricantes de hardware están en el negocio de desarrollar y vender computadoras. Debido a que el software nunca va a ser una parte importante de sus utilidades pueden poner menos esfuerzo en mejorarlo.
- 2.- Un fabricante de hardware en general tiene menos interés en reducir los recursos necesarios para correr su software.
- 3.- El software puede estar atado a un tipo particular de computadora o sistema de manejo de base de datos. Como resultado de moverse de una computadora o base de datos a otra, puede requerir modificaciones importantes al sistema MRP II.

#### b) Desarrolladores de Software.

El número de desarrolladores de software para MRP II ha crecido a más de 150 en los últimos 15 años. Además de enfocarse a desarrollar software de excelencia, pueden ofrecer consultoría, soporte y entrenamiento en conjunto con sus programas. En algunos casos estos servicios pueden ser de gran valor:

## Ventajas:

1. El software es generalmente el producto más importante de los desarrolladores de software, de tal manera que el paquete puede tener más características deseables, más funcionalidad, mejor ingeniería de software que los paquetes ofrecidos por los fabricantes de hardware.
2. El desarrollador tiene interés en reducir los requerimientos de hardware, debido a que su mercado aumenta si más computadoras pueden correr su software.
3. Se puede tener un grado alto de independencia.  
En algunos casos, se podrá correr el software en más de un tipo de hardware, lo cual da la flexibilidad de cambiar las computadoras en el futuro, sin incurrir en el trauma de cambiar los sistemas operativos. Los sistemas desarrollados con el sistema operativo UNIX, son un ejemplo del software de este tipo el cual puede correr en más de 40 diferentes marcas y tamaños de computadoras.
4. Los desarrolladores de software generalmente dan mantenimiento en períodos largos para sus productos. Generalmente el soporte está disponible el mismo

tiempo en que los fabricantes de hardware ofrecen soporte.

#### Desventajas

1. El tamaño de los distribuidores de software como compañía es significativamente más pequeño que el de los fabricantes de hardware y su capacidad para ofrecer soporte puede disminuir considerablemente si el paquete se vuelve muy popular.
2. En ocasiones la documentación del sistema no es clara ó es escasa.
3. Debido a que el sistema es desarrollado para ser utilizado en cualquier empresa, la adaptabilidad de éste a las características específicas de una compañía en particular, muchas veces no coinciden con las desarrolladas en el paquete, por lo que se tienen que realizar una gran cantidad de cambios al sistema.

#### c) Desarrollo dentro de la compañía.

Después de examinar varios paquetes MRP de fabricantes de hardware y desarrolladores de software y de determinar los esfuerzos requeridos para realizar la interfase con los sistemas existentes, algunas compañías podrían tomar la decisión de desarrollar el

software ellos mismos. El desarrollar el software puede ser una idea más acertada que comprar un paquete, si se tiene una de las siguientes situaciones:

1. Varios o casi todos los subsistemas que hacen un sistema MRP II se encuentran actualmente en el software y en donde el esfuerzo del desarrollo se centra en poner en funcionamiento estos subsistemas.
2. Hay unos módulos ausentes en el sistema, pero todos los demás módulos se encuentran instalados.

Por otro lado, si no se tiene un software y se desarrolla un sistema MRP II con una lógica errónea puede traer más problemas. A continuación se listan las desventajas asociadas con el desarrollo de software para MRP II.

**Desventajas:**

1. El tiempo involucrado. Una parte del problema es el hecho de que un software para MRP es demasiado grande debido a los estándares de desarrollo de software. Muchos paquetes comerciales tienen aproximadamente 1,000,000 de líneas de código y el equivalente a 50 ó más años de esfuerzos en el desarrollo antes de que fueran liberados.

2. El síndrome del desarrollo. Otro problema con el desarrollo del software es lo que se ha dado en llamar el "Síndrome del desarrollo", en el cual una compañía invierte una gran cantidad de recursos en crear un sistema, sólo para descubrir que se está aún desarrollando después del tiempo en el cual pudo haber sido implantado un producto comercial. Las compañías que diseñan su propio software descubren y cometen todos los errores que los distribuidores de software y usuarios han hecho durante los últimos 25 años.
  
3. Por último el software desarrollado en la compañía tiende a ser diseñado alrededor de las decisiones actuales de la misma. Por ejemplo, una compañía puede no tener centros de distribución ó varios almacenes cuando el software es desarrollado. Como resultado, el software será diseñado sin la capacidad de soportar varios centros de distribución. En los siguientes años, si la compañía desea introducir otros centros de distribución, tendrá que volver a diseñar el sistema y mientras el software es rediseñado, estos centros de distribución funcionarán sin las ventajas del MRP.

#### 4. EL PROCESO DE EVALUACION.

Muchas compañías ven el proceso de evaluación como una tarea fácil. Después de todo, con 150 a 200 paquetes en el mercado, todo lo que se tiene que hacer es reducir la lista de paquetes, contactando a todos los proveedores, revisar su documentación y evaluar cada paquete. Pero ¿NO se va a llevar varios meses solamente el obtener una lista de candidatos?.

Afortunadamente la respuesta es "NO". Una compañía que planea evaluar y seleccionar el software en tres meses, no podrá contactar a 150 distribuidores y conducir una evaluación de cada paquete, acaso tendría tiempo para evaluar 4 ó 6 sistemas. Pero éste no es el problema, ya que 4 ó 6 paquetes son suficientes para tomar una decisión muy acertada. El hecho es que una compañía puede disminuir su lista de distribuidores dentro de las primeras semanas. Mientras está revisando estos paquetes, puede examinar los sistemas existentes para determinar qué subsistemas pueden sobrevivir ó ser adaptados.

Muchos de los puntos tratados en este capítulo, están relacionados con la selección del software para compañías que van a usar MRP por primera vez, o que se encuentran operando un sistema MRP en niveles bajos y desean mejorar su eficiencia o rendimiento.

Las acciones a realizar para este fin se detallan en las secciones que siguen.



#### 4.1 Reducir la lista de candidatos.

Hay que pensar por un momento en el proceso de comprar un auto. Un método sería investigar todas las posibles alternativas y entonces visitar a cada una de las agencias. Otro método sería investigar un número pequeño de alternativas aceptables, definiendo el conjunto de criterios, que reduciría de inmediato la lista de candidatos. Estos criterios podrían incluir el tamaño del auto, rango de precios, tipo de manufactura, etc.

De la misma manera se definen criterios de selección, características, costos, etc, para disminuir el número de software. Entre éstos se encuentran:

1. Hardware y software existentes
2. Equipo requerido.
3. La presencia del distribuidor dentro del mercado.
4. El rango de precios del software
5. Innovaciones en las técnicas de procesamiento de base de datos
6. Recomendaciones de expertos.

#### 1. Hardware y software existente

Las compañías con hardware y software están en una posición envidiable cuando se seleccionan nuevas aplicaciones. El tipo de

computadora y sistema de manejo de bases de datos eliminarán a un 80% ó más de los paquetes de la lista. Por ejemplo una compañía con una IBM 4361, con un sistema operativo DOS/VSE y con un manejador de base de datos IDMS puede eliminar de 5 a 6 paquetes, mientras una compañía que utiliza una minicomputadora como una IBM AS/400 ó una Hewlett-Packard HP-3000 necesitará revisar menos de 10 sistemas.

## 2. Equipo requerido.

Para aquellas compañías que no cuentan con un hardware existente, parte del proceso de disminuir la lista de alternativas probablemente se basará en un análisis del hardware y software en conjunto. Este análisis necesita considerar el estado actual del negocio, sus pronósticos de crecimiento, así como las alternativas de crecimiento de hardware. El proceso de análisis de requerimientos de hardware está más allá del objetivo de este trabajo. Sin embargo es posible decir que una búsqueda cuidadosa en una compañía puede determinar un número de restricciones, como la necesidad de que una computadora soporte un número determinado de terminales, que apoye a un negocio de un tamaño en particular (número de artículos producidos, lista de materiales, etc), que maneje un volumen de transacciones, además de permitir el crecimiento de la compañía sin una conversión de equipo. Tales restricciones probablemente eliminarán algunos candidatos de software y hardware. Por ejemplo, sabiendo que eventualmente se

tendrán de 30 a 50 estaciones de trabajo, probablemente no se tomará en cuenta software que corre en redes para PC's. En algunas ocasiones se tendrá cierto tamaño de computadora, tal como el de una IBM AS-400 ó equipo central la IBM 4361. Finalmente la compañía podrá requerir un tipo específico de sistema operativo, tal vez UNIX y sus derivados.

### 3. La presencia del distribuidor dentro del mercado.

Es importante saber si el distribuidor está atendiendo a compañías con experiencia en el manejo correcto del software que ofrece. Serán tomados muy en cuenta aquellos distribuidores que tengan como usuarios compañías que se encuentren operando MRP II en un buen nivel, pues de esta manera también se reduce el riesgo al implantar MRP II.

### 4. El rango de precios del software.

El costo de un paquete para MRP varía de \$50,000 a \$150,000 dólares para minicomputadoras y puede llegar hasta \$1000,000 de dólares para equipos centrales.

Cualquier compañía debe de entender que el comprar un software representa una inversión. Sin embargo debe también reconocer que implantar MRP II correctamente hará posible la recuperación del costo de cualquier paquete. De esta manera, el reducir la lista de

software basándose en el costo del paquete puede no ser la mejor decisión. Eliminar el software que se acomoda mejor a las actividades técnicas de la compañía por su precio puede ocasionar costos mayores a la larga. Hay que tener presente que el costo final se debe de calcular tomando en cuenta la recuperación de la inversión una vez instalado el paquete.

#### 5. Innovaciones en las técnicas de procesamiento de base de datos

Durante los próximos años el procesamiento de datos va a cambiar radicalmente. Nueva tecnología, herramientas e innovaciones (como el manejo de sistemas de base de datos, lenguajes de cuarta generación, prototipos, etc) prometen reducir el esfuerzo requerido para desarrollar y modificar sistemas.

#### 6. Recomendaciones de gente experta.

Se puede usar fuente externa para implantar MRP. Las consultorías y expertos que han trabajado en este campo durante algún tiempo es probable que hayan manejado diferentes paquetes de software ó tengan conocimientos directos de los paquetes disponibles en la industria. Tales expertos pueden sugerir varios paquetes que valdrá la pena revisar. Es importante que no recomienden un paquete en particular.

Una vez que se tiene una lista de distribuidores, el siguiente paso

es evaluar cada software en término de funcionalidad.

#### 4.2 Evaluación funcional del paquete.

Se puede usar un sistema estándar establecido (funciones importantes que el paquete debe de contemplar) para evaluar la funcionalidad de un software. Asumiendo que la compañía está dispuesta a invertir en el esfuerzo de buscar sistemas, el método más efectivo para revisar un software será el que esta basado en un análisis funcional de la documentación e incluye los siguientes pasos:

- 1) Leer la documentación
- 2) Reunirse con el distribuidor del software y usar el estándar establecido
- 3) Revisar ejemplos de salidas o pruebas
- 4) Obtener comentarios de otros usuarios
- 5) Ligar los resultados de la comparación con el estándar del contrato

Revisar la documentación.

Lo primero que hay que hacer en un análisis funcional del software es obtener una documentación completa. Esta documentación debe incluir tanto documentación para el usuario como técnica, particularmente aquellos manuales que tienen que ver con la lógica,

procesamiento, ediciones y otros aspectos claves del programa. El objetivo de obtener una documentación completa es el de revisarla toda saltando aquellas partes que no se necesitan y no tener solo un resumen que no tenga la información requerida para determinar la presencia o carencia de características esenciales y su lógica.

Alguien necesita comprender y analizar la documentación para hacer la evaluación inicial del software. En la mayoría de los casos las persona asignadas a este trabajo serán del departamento de sistemas, ya que sus miembros están acostumbrados a leer documentación técnica de software. Este trabajo, sin embargo podrá ser hecho por una persona del departamento de manufactura con experiencia.

Antes de leer la documentación se tiene que verificar lo siguiente:

- a) Si se cuenta con un buen sistema estándar como base parralevar a cabo la comparación.
- b) Si se comprende correctamente cómo debe de trabajar la lógica para cada función especificada en el sistema estándar. Además se debe de tener un buen conocimiento de las diferentes alternativas aceptables dentro de estas funciones. (No se necesita identificar aquellas características que no operan normalmente, sino las que funcionan actualmente)
- c) Se preparan algunas hojas de evaluación listando los

puntos más importantes en la evaluación del sistema establecido. Estas hojas de trabajo son usadas para organizar el esfuerzo y como recordatorio para cubrir todas las funciones necesarias en el sistema.

La mayoría de la gente que evalúa un software usando este método debe planear dos lecturas de la documentación. Una de estas lecturas revisará toda la documentación para un entendimiento general del sistema, la segunda lectura será para descubrir los detalles que existen detrás de cada una de las funciones esenciales del sistema.

En la segunda lectura es recomendable hacer notas de cómo trabaja el software. Estas notas deben de incluir una descripción de la lógica específica usada en cada opción. Se puede también describir las relaciones entre archivos, describir reglas de edición en transacciones y listar los elementos más importantes en el sistema. Además, las funciones que son parte del sistema pero que no trabajan correctamente deben de ser identificadas para discutir las con el distribuidor. Las opciones globales en el sistema (por ejemplo, opciones de control de archivos) deben también ser identificadas.

Mucha gente encuentra muy útil hacer notas y seleccionar páginas de la documentación, ejemplos de transacciones y formatos de reportes y de pantallas. El hacer esto hace más fácil reconstruir como

funciona la lógica del sistema.

El resultado final de esta lectura y de este proceso de tomar notas es un entendimiento detallado de cómo un software está diseñado para operar, de la complejidad potencial del software y los posibles problemas que tendrán que ser resueltos. Cuando las hojas de trabajo se encuentran en blanco, esto es evidencia de que o bien las funciones más importantes no son parte del sistema o van a ser difíciles de implantar debido a la carencia de documentación. Una buena regla es: "Si no hay documentación, no hay software".

La reunión con el distribuidor

Después de completar la revisión de la documentación, la mayoría de las compañías encuentran útil compartir lo que se encontró en el software con los miembros del equipo de selección. Esto hace que cada miembro este consciente de los problemas potenciales en el software.

Se utiliza un sistema estándar y las notas que se obtuvieron de la revisión de la documentación para dirigir la conversación con el distribuidor a aquellas áreas en las que se tienen problemas potenciales.

Debe de reservarse alguna parte del día para una demostración del sistema. Una demostración breve evade fácilmente problemas y no da



un conocimiento profundo de la lógica del sistema. Si se va a hacer una demostración, se debe de verificar que sea de un buen nivel y que el software exista actualmente. Debe de revelar la funcionalidad del sistema y producir un nivel de confianza en el software. Un aspecto importante de la reunión con el distribuidor es el nivel de experiencia de sus representantes técnicos. Dada la naturaleza de las preguntas que probablemente surgirán y la necesidad de tratar con la lógica específica del sistema, el distribuidor debe de estar preparado en traer gente del equipo técnico a esta reunión.

La reunión con el distribuidor es la prueba principal para verificar que el software cubre las necesidades de la compañía y el distribuidor dará el servicio necesario. Será también la oportunidad de identificar características y funciones que podrán surgir y las cuáles el distribuidor no está dispuesto a identificarlas públicamente. Hay que recordar aquellas funciones que podrán estar en el sistema y que se identificaron al revisar la documentación.

Finalmente, en consideración a las funciones que no trabajan adecuadamente y que se identificaron, debe de haber una discusión sobre el esfuerzo necesario para realizar el cambio y qué efecto tendrá en la garantía del software. Si el equipo de selección se sale de la reunión sin comprender los principales cambios de los cuáles la compañía será responsable y sin un acuerdo en el tiempo

programado para que el distribuidor lleve a cabo las modificaciones, habrá fracasado en sus responsabilidades.

Un paso importante después de la reunión con el distribuidor para evaluaciones del software basadas en una revisión de la documentación, es la revisión de las pruebas o ejemplos de salidas de información. No se tiene que probar toda función existente en el software, sino tratar de verificar alguna información básica y comprender qué reportes serán utilizados. Por ejemplo cuantos reportes despliegan la siguiente información a un planeador: Información descriptiva básica, el tiempo de desfase, "pegging"\*, especificaciones para programar entradas y órdenes planeadas y mensajes de excepción. La experiencia indica que los paquetes de software con más problemas casi siempre exhiben ejemplos de reportes o de pruebas en pequeñas cantidades para no mostrar estos problemas.

La reacción de usuarios de otras compañías al paquete.

En la fase final para evaluar la funcionalidad del software, un comprador responsable debe de contactar algunos usuarios del sistema como un medio para certificar que el software trabaja sin problemas. La mayoría de los compradores bien informados visitan por lo menos a uno de los usuarios del software y hablan por teléfono con los demás como referencia. Generalmente el distribuidor del software debe de proveer los nombres de las

compañías que son usuarios del sistema.

El método de escoger referencias de usuarios es también importante. Las referencias ideales son aquellas compañías con un buen funcionamiento de operación de MRP II, debido a que estas compañías estarán utilizando la mayoría de las funciones del sistema de una manera normal. Las compañías que se encuentran operando el sistema con un nivel más bajo de ejecución, pueden aún así ser referencias valiosas, pero debe de entenderse que probablemente no están usando las funciones del sistema de la manera correcta. En otras palabras, hay problemas en el software que son aparentes a esta compañía. De la misma manera pueden creer que hay problemas o diferencias en el sistema cuando, de hecho, no existen.

Las referencias menos valiosas son de aquellas compañías que han comprado el software recientemente y no se encuentran operando por completo, si es que lo hacen en alguna medida usando las nuevas herramientas proporcionadas por el sistema.

Algunas preguntas surgidas del sistema estándar dan una guía para entrevistar compañías que se encuentran usando un software en particular:

Progreso en la implantación. ¿Qué módulos del sistema han sido usados y por cuánto tiempo? El objetivo es determinar si las funciones del sistema han sido manejadas.

Funcionalidad. ¿Las funciones explicadas en la documentación del sistema y por el distribuidor trabajan? El objetivo es determinar si la documentación del sistema, así como el distribuidor, comprenden y han explicado correctamente las funciones del sistema.

Errores. ¿Los usuarios del sistema han tenido problemas importantes al trabajar con el sistema? Hay que entender lo que son problemas importantes. Todos los sistemas tienen errores. Un error pequeño no desvía las operaciones, sin embargo, es un problema diferente de uno que suspende el progreso y retarda la implantación.

Soporte del distribuidor. ¿Como evalúan los usuarios del sistema el soporte proporcionado por el distribuidor?.

Usuario. ¿Le gusta el sistema a la gente que lo usa?. ¿Qué características o cambios harían que se sintieran mejor con el sistema?.

Otros. ¿Hay algunos comentarios acerca del sistema que serían de interés, tales como el mantenimiento de la documentación, tiempos de proceso u otro aspecto importante?.

Para las compañías que tienen un software existente que fue desarrollado en varios años y el cuál está siendo evaluado como un

software alternativo, los métodos presentados conllevan algunas dificultades aparentes. Primero hay una carencia relativa de documentación para usarla como base de una revisión, sin embargo puede ser usada para examinarla al igual que la reunión con el distribuidor y las preguntas con los usuarios del sistema, se puede juntar a la gente del departamento de sistemas que desarrolló y dio soporte al sistema, así como a los usuarios de los diferentes departamentos que han estado operando el sistema. Como en el caso del distribuidor, la intención de la reunión es obtener una comprensión del sistema, identificar los principales problemas y planear cómo las limitaciones pueden ser eliminadas.

Mientras el método que se describió para evaluar la funcionalidad de un paquete para MRP II está basado en una revisión de la documentación, otro método para probar las funciones del sistema es:

- Paso 1. Reunirse con el distribuidor del software y utilizar un estándar como agenda.
- Paso 2. Instalar la versión de la demostración del software en el equipo de la compañía o en otro equipo accesible.
- Paso 3. Probar las funciones del software con el sistema estándar. Revisar las salidas.
- Paso 4. Buscar comentarios de otros usuarios.
- Paso 5. Igualar los resultados de la comparación con el

estándar en un contrato.

Este método puede ser uno que la compañía podría escoger si el software existe, pero no hay documentación o es escasa. En la práctica este método no funcionaría muy bien para software que se ejecute en equipos centrales, por el tamaño del equipo que se requiere para ejecutar pruebas.

#### 4.3 Aspectos técnicos a considerar.

Cualquier evaluación debe tener como punto central una revisión de la funcionalidad del software. Pero la evaluación estará incompleta si no se exploran algunos resultados técnicos:

1. ¿Las modificaciones requeridas se pueden llevar a cabo en un tiempo aceptable?.
2. ¿Está el paquete basado en una filosofía de diseño que aliente a los usuarios a usar su ingenio en vez de esperar todo del sistema?
3. ¿Vale la pena invertir en el paquete comparado con otras alternativas? ¿Cuál es el costo real en términos de retrasos?
4. ¿Se pueden crear interfases con módulos existentes o el software se cierra sólo a productos del distribuidor?
5. ¿Que nivel de soporte prometió dar el distribuidor ?6. ¿Es adecuada la documentación? ¿Puede ser usada ycomprendida

por usuarios no técnicos?

Determinando esfuerzos de modificaciones.

No es de sorprender que un aspecto importante al investigar cualquier paquete para MRP II, es el identificar las modificaciones que son esenciales para que la compañía use eficientemente el software. Poder identificar estas modificaciones y calcular el esfuerzo requerido para llevarlas a cabo. El resultado de usar un sistema estándar para revisar la funcionalidad del sistema es que cuando se compara el software prospecto con el sistema estándar propuesto, se encontrará un número de puntos que bien no se encuentran en el sistema o que están presentes pero no trabajan correctamente( para cada función que aparentemente no funciona correctamente, se necesitará tener una discusión con el distribuidor del software, así como del propio grupo de sistemas).

El objetivo es desarrollar un conjunto de especificaciones que podrán ser utilizadas después como un plan para llevar a cabo los cambios. Las especificaciones deben de incluir la magnitud estimada de los cambios necesarios, así como mencionar quién los va a llevar a cabo.

Otro conjunto de modificaciones son las concernientes a las interfases principales. Para planear las interfases, el distribuidor debe primero proveer un entendimiento básico de la

arquitectura del sistema. Una vez que se ha establecido un nivel de conformidad, la gente de sistemas puede comenzar con los planes para las interfases temporales y permanentes.

Después de haber determinado quién es el responsable de llevar a cabo las modificaciones e interfases, el siguiente paso es desarrollar un plan para llevar a cabo estas modificaciones. Al hacer este plan, la mayoría de las compañías asumen que el distribuidor será el responsable de la mayoría, si no es que de todas, las modificaciones, mientras que la compañía en cuestión hará las interfases temporales y permanentes a los sistemas existentes. Desafortunadamente, las compañías aceptan las fechas que el distribuidor le da para liberar los cambios, sin cuestionar las consecuencias de los tropiezos en lo programado.

El plan para llevar a cabo las modificaciones tiene dos partes. Primero verificar lo razonable: ¿El distribuidor ó la compañía han participado antes en un proyecto de esta magnitud? Si es así, ¿La compañía o el distribuidor pudieron llevar a cabo los cambios en un tiempo razonable? Si la compañía nunca antes ha tenido un proyecto de este tipo, ¿Qué es lo que otras compañías han encontrado necesario para llevar a cabo un proyecto de este tipo?.

La segunda parte del plan para las modificaciones es tener una posición de respaldo, algunas veces llamado "Plan B". En una situación donde el distribuidor estará realizando los cambios



requeridos, es esencial idear un plan de respaldo en el caso en el que el distribuidor fracase en tener el software a tiempo para la implantación. El desarrollo de este tipo de planes también generan nuevo interés y una perspectiva diferente en el plan del distribuidor. Supongamos que se ha encontrado un paquete que carece de Plan de Capacidad, pero que en todos los demás aspectos está completo. Una estimación revela que desarrollar un módulo de Plan de Capacidad, usando recursos de la compañía, tomará 4 meses de programación. Esto significa que para tener el módulo de plan de capacidad disponible en 12 meses, la compañía debe de comenzar su desarrollo en el mes 8 ó más tarde.

La pregunta clave es si la fecha de liberación del distribuidor es antes que la fecha necesaria para llevar a cabo las modificaciones. La fecha necesaria no es la fecha cuando se planea comenzar la implantación, sino la fecha en la se comenzará a programar si el distribuidor falla en la liberación de los cambios. En el ejemplo del módulo de planeación de capacidad, la fecha clave comienza en el mes ocho. Si el distribuidor no libera los cambios para esta fecha, la programación debe de comenzar inmediatamente.

Lamentablemente la entrega del software para nuevas aplicaciones no es mejor que la entrega de nuevos productos de cualquier otro tipo. Hay que pensar en los productos que una compañía ha introducido en los últimos 5 años. ¿Cuántos de ellos llegan a tiempo y trabajan según lo especificado? El acierto en el tiempo de entrega es

probablemente menos del 50 %. Y este porcentaje es lo que probablemente se puede esperar de una compañía de software con respecto a las principales mejoras de un sistema. A pesar de que hay distribuidores de software con una experiencia respetable para liberar nuevos productos a tiempo, la mayoría probablemente tendrá problemas anticipados con las mejoras y que causarán uno de los siguientes:

- a) El usuario revisa el problema y decide realizar los cambios necesarios. En la mayoría de los casos, esto significa un retraso importante en la liberación del sistema.
- b) El distribuidor revisa el problema y decide liberar el sistema con deficiencias conocidas. En este caso el sistema es implantado a tiempo, pero puede no ser utilizable.

#### Filosofía de Diseño.

La filosofía de diseño del distribuidor tiene un mayor impacto en el uso de un paquete. Hay un número de características que son parte de un buen paquete, pero las cuáles no son funciones específicas. Estas son las filosofías de diseño que han sido incorporadas en el software. Un diseñador que no comprende estas filosofías, producirá un paquete con limitaciones y defectos. Las filosofías más importantes son las siguientes:

Simulación de la realidad. Un sistema MRP II tiene sólo un propósito: simular correctamente las realidades de un ambiente de una empresa manufacturera. Un software que no simule correctamente la realidad, perderá el objetivo principal del sistema.

Simplicidad. Todas las grandes cosas son simples y un sistema MRP II no es la excepción. Un software necesita un conjunto de funciones. Cualquier característica adicional en el sistema muchas veces es innecesaria y no deseable. Hacen generalmente la operación del sistema más complicada y por lo tanto destruyen su simplicidad natural e inherente.

Responsabilidad. Los sistemas no hacen que las cosas sucedan, la gente lo hace. La mayoría de las cosas se llevan a cabo porque alguien es directamente responsable de un trabajo o una decisión. Un software debe de estar diseñado para apoyar los requerimientos de responsabilidad de las operaciones diarias de un sistema. No debe de obscurecer, impedir o tratar de asumir estas responsabilidades. Un buen sistema debe de reconocer la necesidad de presentar a los usuarios la información para explicar por qué lo que el usuario realizó tiene sentido.

Un buen sistema presenta los problemas sin tratar de dar la solución automáticamente. En vez de esto, el sistema le debe de permitir a la gente encontrar una solución para poder implantarla dentro del sistema.

Trazar el costo real del paquete.

Diez años atrás, un distribuidor con un software MRP II relativamente completo y funcionando en un equipo central podía pedir cualquier precio. En ese tiempo el costo del software estaba relacionado con la funcionalidad del paquete. Si el paquete estaba más completo, el costo era más alto. Ahora un número de paquetes completos están entrando al mercado con precios más bajos. El poder de las minicomputadoras ha aumentado, así como el de las PC'S. Varios paquetes para minis y micros están lo suficientemente completos comparados en relación a sistemas que se ejecutan en máquinas grandes. En el análisis final para la compra de un paquete, cada compañía debe de tomar su propia decisión estimando el costo verdadero de cada alternativa.

El primer paso en determinar si un paquete en particular vale su precio, sin importar el tamaño de la computadora para el cual fue diseñado, es completar la justificación del costo de implantación del sistema. El número clave, por supuesto, serán los años o meses de beneficio una vez que el sistema sea implantado correctamente. También este número será el costo por el retraso de poner al sistema a tono.

El segundo paso es estimar el número de meses de retraso que resultarán de tener que modificar el sistema. Al hacer esto, se tiene que tomar cada función que tendrá que ser desarrollada o

modificada. ¿Quién lo hará? y ¿Qué se tendrá que hacer en términos de instalar interfases temporales o permanentes?.

El tercer paso es calcular el costo adicional necesario para obtener un sistema trabajando con lo básico.

Finalmente, el costo verdadero del software es el precio del paquete (éste puede incluir el costo del hardware, si se cambia el equipo, o se obtiene uno nuevo), más los gastos adicionales, más el costo de los retrasos. Usando este método se puede descubrir que lo que parece ser un paquete relativamente barato, puede terminar costando más que el paquete más caro, si se requieren modificaciones extensivas para hacerlo funcionar de acuerdo a las necesidades de la compañía. Por esta razón la fórmula: "Precio base + Modificaciones + Costos por retrasos" es crítica en la evaluación.

Requerimientos técnicos.

Hay generalmente dos tipos de requerimientos técnicos para un software MRP II: fijos y flexibles. Los requerimientos fijos se refieren a la compatibilidad de hardware y software que actualmente existe, tales como la marca de la computadora, el sistema operativo ó el sistema manejador de base de datos. Si una compañía tiene una NCR Tower con UNIX, probablemente necesitará comprar software si desea seguir usando esta máquina. Si se decide a mantener la

máquina, entonces la compatibilidad del software se volverá un requerimiento fijo.

En contraste, los requerimientos técnicos flexibles son más una cuestión de preferencia e incluyen cosas tales como el lenguaje de programación. Por ejemplo se puede tener una preferencia por COBOL, pero se pueden aceptar programas desarrollados en PL/1. De la misma manera, se desean programas desarrollados en RPG para un equipo AS/400, pero se aceptan programas en COBOL.

Desde un punto de vista de usuario, también hay requerimientos fijos y flexibles. Por ejemplo un requerimiento fijo puede ser que el software tenga cantidades con tres dígitos de precisión para la cantidad en mano de un artículo. Si un software soporta solamente campos enteros para este campo, la implantación apropiada del nuevo sistema no es redondear todos los números arriba (o abajo), para obtener esta precisión. La única situación en la cuál se continuará revisando un paquete que no tiene el tamaño adecuado para ciertos campos clave será cuando el software es fácil de corregir. Y actualmente sólo algunos paquetes tienen tales capacidades. Algunos ejemplos son:

1. Paquetes que toman todas las ventajas de la arquitectura del equipo IBM AS/400.
2. El Manejador de materiales/3000 y el Manejador de producción/3000 de Hewlett-Packard.

3. Computadoras que corren en el sistema PICK.
4. Sistemas basados en lenguajes de cuarta generación como MITROL (para IBM) y sistemas QAD (para UNIX)

Tiempos de respuesta y ejecución.

Anteriormente cuando los sistemas estaban más orientados a los procesos por lote, no era muy importante hacer estimados comprensivos y precisos del tiempo de ejecución. Debido a que actualmente una gran parte de los sistemas se utiliza para procesos en línea, es de suma importancia tener mejores tiempos de respuesta. En otras palabras, no tiene sentido contratar gente que puede ser no productiva debido a que deben esperar mucho tiempo que les responda el sistema. Si no se tiene una idea de cuánto tiempo requerirá el sistema para procesar cierta información, se pueden tener sorpresas desagradables.

Los distribuidores deben de proporcionar un estimado de la ejecución del sistema basado en casos de prueba que le permitan especificar tiempos de ejecución para varias compañías. El otro método es en el que el distribuidor proporciona estadísticas de usuarios que utilizan equipos con tamaños de base de datos similares (esto también nos regresa al punto acerca de considerar softwares que tienen un número suficiente de usuarios, de los cuáles se puede obtener experiencia un paquete nuevo no podrá proporcionar este tipo de información). Si se obtienen estimados de

la ejecución del sistema por parte del distribuidor, hay una serie de preguntas a realizar. En general, se deseará conocer estimados para el tiempo de respuesta promedio y para los principales procesos por lote. Parte de tener un conocimiento de si estos tiempos son aceptables, es para conocer qué tan frecuente estos procesos por lote se harán. Por ejemplo un tiempo de ejecución de nueve horas sería inaceptable si se planea correr la planeación de los requerimientos de materiales (MRP) cada noche; sin embargo sería más que adecuado si se planea correrlo cada semana.

Finalmente, se le debe de pedir al distribuidor información por tipo de transacción, así como por el tiempo de respuesta promedio y tiempos de ejecución en aquello que se procesa frecuentemente, tal como: el reporte de producción maestra, planeación de los requerimientos de materiales, planeación de los requerimientos de capacidad, reportes de control de entradas y salidas, etc.

Explorar otras aplicaciones de interfase ofrecidas por el distribuidor.

Algunos distribuidores tienen aplicaciones de interfase que son parte de los sistemas ofrecidos por ellos. Se podrían usar estas aplicaciones actualmente o en un futuro. Algunos paquetes pueden crear problemas de interfase entre otros sistemas y pueden forzar a una compañía a desarrollar sus propias interfases.



### Soporte del distribuidor.

Como en cualquier aplicación, el soporte del software por parte del distribuidor es crucial para el uso correcto de un paquete. El soporte incluye las siguientes áreas: La gente que será responsable de ayudar a implantar el sistema, los servicios de soporte técnico y los programas de entrenamiento del sistema, y el soporte ofrecido después de la implantación.

Mientras que es esencial que la gente experta ofrecida por el distribuidor ayude a implantar el sistema, es aún más importante que la compañía participe en la implantación para obtener un sentimiento de propiedad del sistema. Es aceptable que el distribuidor instale el hardware y software y deje el control a la compañía. Pero si además de instalar el hardware y software dirige totalmente la implantación, al final se tendrán problemas de control del sistema por parte de la compañía.

Otra consideración importante en el área de soporte es si un distribuidor justificará la cantidad de consulta y entrenamiento que propone dar. Se espera requerir menos soporte para un software costoso que para uno más barato, sin embargo, la experiencia sugiere lo contrario.

Revisar la calidad de la documentación.

La documentación es el veneno de la industria del software y el software para MRP II no es la excepción. Dada la enorme cantidad de material que acompaña a los paquetes para MRP II, la calidad de la documentación se vuelve crucial. Asumiendo que las explicaciones son claras, lo más importante es que también la documentación este organizada.

Estudios detallados indican que la calidad de la documentación tiene un efecto importante en el tiempo involucrado para conocer un nuevo sistema. Algunos estudios (basados en software para micros) muestra que la calidad de la documentación puede afectar el conocer un sistema hasta en un 50 %.

Revisar sólo el software de código objeto.

El software que contiene sólo código objeto se ha vuelto recientemente muy popular y acarrea problemas por las funciones ausentes. El software de código objeto ofrecido por la mayoría de los fabricantes de máquinas no es mejor ni peor que el de los distribuidores de software con programas fuente. Desde el punto de vista de un distribuidor, sin embargo, el software de código objeto ofrece dos distintas ventajas:

Primero, protege la inversión del distribuidor en el software.

Limitando el número de personas que tienen acceso al código fuente, el distribuidor mantiene sus secretos de diseño, técnicas especiales de programación y funciones especiales de los ojos de los competidores. Esto tiene sentido, ya que, a pesar de que no hay una importante razón para proteger la lógica de un sistema MRP II, un distribuidor con un enfoque innovador habrá invertido una cantidad significativa de tiempo y dinero en el esfuerzo para desarrollar el sistema y deberá asegurar cualquier nivel de protección posible.

Segundo, el código objeto simplificará el soporte, debido a que todos los clientes están usando una versión probada del sistema. Como resultado los usuarios no pueden hacer cambios en el sistema básico y causar problemas.

Pero lo que es bueno para el distribuidor no necesariamente es bueno para el usuario final. Cada usuario necesitará realizar un cierto número de modificaciones al software para satisfacer necesidades de su empresa. Además de las modificaciones funcionales y las modificaciones para satisfacer los requerimientos regulares, los usuarios desean hacer modificaciones a los campos, reportes, pantallas y transacciones.

Cuando se contemple comprar software de sólo código objeto, se deben en tomar en cuenta las siguientes preguntas: ¿Qué tipo de protección debe de ser establecida en el contrato? ¿Qué situaciones

deben de ser evitadas? ¿Quién es el responsable para hacer que el software funcione? Como una consideración técnica, sin embargo, los usuarios de software de código objeto deben de revisar los paquetes más funcionales, así como poder hacer cambios en algunos reportes, pantallas y transacciones. Deben también poder agregar software para satisfacer algunos requerimientos. Si cualquiera de estas características o capacidades no se encuentran o no pueden ser aportadas por el software, entonces es mejor pensar en otra opción

## 5. UNA APLICACION A UNA INDUSTRIA DE PRODUCTOS FARMACEUTICOS.

Los laboratorios Columbia S.A de C.V es una empresa mexicana que pertenece a la industria farmacéutica. Elabora alrededor de 149 productos terminados, los cuales son desarrollados por 3 divisiones:

(1) Humana, (2) Veterinaria, y (3) Consumo. Cuenta con una área administrativa, en la cual se administran todas las operaciones de la compañía y con una área de operaciones industriales, en la cual se realizan todas las operaciones necesarias para elaborar sus productos.

La compañía estaba muy interesada en implantar MRP II, por lo que su personal directivo y gerencial tomó cursos de MRP II. El personal operativo participó en sesiones de video sobre MRP II, para poder tener un claro panorama de lo que la compañía trataba de realizar.

Por este motivo se le asignó al departamento de sistemas la tarea de elaborar un software ó contactar distribuidores de software para poder implantar MRP II. El departamento de sistemas contaba con un director, un jefe de soporte técnico, tres líderes de proyecto, tres analistas y dos operadores. Tomando en cuenta su tamaño y experiencia, el departamento de sistemas optó por la opción de que la compañía comprara un paquete para MRP II. El sistema anterior (GLOBAL) que se encontraba instalado en un equipo IBM Sistema-36,

no contaba con las funciones necesarias para soportar la filosofía de MRP II, por lo que el paquete sería totalmente nuevo. La compañía se encontraba en negociaciones con IBM, para adquirir un equipo AS/400 modelo B35. El departamento de sistemas estudió tres candidatos cuyo software corre en este tipo de equipo. Estos paquetes fueron: (1) MAPICS, desarrollado por IBM, (2) BPCS, desarrollado por System Software Associates, Inc y (3) PANSOPHIC, desarrollado por Applications Software. La figura 5-1 muestra un resumen de las funciones que se estudiaron en general para cada uno de estos paquetes.

El proceso de obtener esta información fue un ejercicio muy interesante. Debe de notarse que esta pequeña guía no contiene toda la información que los proveedores dieron acerca de sus productos. Para cada una de las categorías en la tabla, por ejemplo, se le permitió a cada proveedor comentar las características y procedimientos relevantes para esa categoría. Comentarios detallados fueron hechos en cada punto. Estos comentarios cubrieron procedimientos, como el estado de desarrollo de ciertas características que estaban ausentes o características únicas de cómo cierto aspecto del paquete MRP-II fue implantado. Además a cada proveedor se le pidió identificar y discutir brevemente los puntos más importantes del paquete , así como un panorama general del mismo.

Al parecer, y así lo demuestra el estudio realizado, los tres

Atributo	MAPICS/DB	BPCS	PANSOPHIC
Disponible en mini	X	X	X
Disponible en micro			X
Entrenamiento	X	X	X
Lista de materiales	X	X	X
Transacciones de inventario	X	X	X
Entrada de ordenes	X	X	X
Planeación de capacidad finita		X	X
Planeación de capacidad infinita	X	X	X
Compras	X	X	X
Control de piso	X	X	X
Contabilidad	X	X	X
Procesamiento de ordenes	X	X	X
Nomina	X		X
Tutorial disponible	X	X	
Adaptable para trabajo en planta	X	X	X
Adaptable para procesos industriales	X	X	X

FIG 5-1

MAPICS/DB  
 IBM Corporation  
 Precio basado en el tipo de procesador

BPCS (Business Planning and Control System)  
 System Software Associates, Inc  
 \$24,000-\$800,000 Dlls

PANSOPHIC  
 Applications Software  
 Comunicarse a la empresa para mayor información

paquetes parecían tener las funciones y características que la compañía buscaba. Sin embargo además de que los paquetes tuvieran las funciones necesarias para que la compañía pudiera implantar MRP II, se buscaron las siguientes características:

1. Que el paquete fuera usado por un numero aceptable de compañías.
2. Que el paquete tuviera la lógica de lote, pues se trata de una industria farmacéutica.
3. Que la interacción con el usuario fuera a través de procesos en línea.
4. Que los programas fuente estuvieran disponibles para poder ser modificados, además de que el paquete no manejara funciones complicadas, pues la gente del departamento de sistemas no contaba con bastante experiencia en el manejo del equipo AS/400 .
5. Que el paquete aprovechara la arquitectura del sistema AS/400.

En el estudio realizado se encontró con lo siguiente:

1. El paquete con el mayor número de usuarios fue el desarrollado por System Software Associates. Entre sus usuarios se encontraron:

- . Jugos del Valle
- . CIBA GEIGI.



- . Pinturas ICY
- . Televisa
- . MeadJhonson
- . 3M de México

2. Entre los usuarios del sistema BPCS se encontraban industrias farmacéuticas, las cuales estaban satisfechas con la lógica de lote que manejaba el sistema.

3. El paquete BPCS además de ser un paquete integral, maneja una gran cantidad de procesos en línea, los cuales ayudan a ser más flexible la interacción con el usuario. La información puede ser obtenida sin necesidad de generar reportes, ya que hay una gran cantidad de consultas en todos los módulos que contiene el paquete. MAPICS carece de este tipo de funcionalidad, debido a que maneja una gran cantidad de reportes para consultar la información y por lo tanto es un poco más complicado para el usuario.

4. Los programas fuente del paquete BPCS pueden estar disponibles en cualquier momento. El sistema PAMSOPHIC cuenta con una herramienta de tecnología avanzada, como es el uso de herramientas CASE. Sin embargo tomando en cuenta el tamaño y experiencia del departamento de sistemas, esto puede ser un problema, debido a que el manejo de CASE es un poco rígido porque la lógica no es tan fácil de modificar y puede tomar más del tiempo necesario modificar un programa.

5. El sistema BPCS era un sistema disponible en el equipo IBM AS/400. BPCS/400 es una migración de BPCS/38, es decir no es un sistema nativo del AS/400, sin embargo aprovecha la gran mayoría de los recursos y funcionalidades de la arquitectura AS/400.

BPCS mostraba tener las características y funciones que la compañía buscaba para poder implantar MRP II. Por tal motivo se decidió evaluarlo más detalladamente para poder comprobar que era un buen simulador del sistema MRP II y de las actividades de la compañía.

A continuación se presenta la evaluación del paquete utilizando el sistema estándar propuesto (las compañías Oliver Wight publicaron un reporte conteniendo las funciones mínimas que debe de contener un sistema para software MPR II).

El sistema estándar es un conjunto de especificaciones funcionales simples, así como la explicación de las suposiciones y la experiencia que lleva a la necesidad de estas funciones.

El sistema estándar tiene grupos lógicos de funciones que incluyen las siguientes áreas:

1. Manejo de la demanda.
2. Programación maestra.
3. Planeación de los requisitos de materiales.
4. El subsistema de la lista de materiales.

6. El subsistema de transacciones de inventario.
7. El subsistema de control de piso y rutas.
8. Planeación de los recursos de distribución.
9. Planeación de los requerimientos de capacidad
10. Compras.

A pesar de que el sistema estándar es un conjunto completo de funciones, no es un sistema ideal con todas las posibles funciones que pueden existir. En vez de esto, el sistema estándar describe el conjunto de funciones más simples, necesarias para hacer que MRP II funcione para este caso. A continuación se presenta las funciones del sistema estándar evaluadas para BPCS.

#### Manejo de la demanda.

El manejo de la demanda está integrado por los módulos de proyección de ventas y el de entrada de órdenes. El sistema BPCS contiene un módulo de proyección de ventas, el cual tiene las siguientes funcionalidades.

1. Cuenta con varias técnicas de proyección de ventas, así como con un método para evaluarlas.
2. Cuenta con un factor de proyección de ventas por temporada.
3. Proyección de ventas por familia de productos.
4. Se puede revisar y aprobar las proyecciones antes de actualizar el sistema.

El módulo de proyección de ventas maneja dos factores para proyectar las ventas: (1) Intrínseco y (2) Extrínseco. La proyección intrínseca está basada en información histórica. La información extrínseca está basada en información externa, como la que puede proporcionar el departamento de mercadotecnia.

El sistema de entrada de órdenes es usado para añadir, borrar y cambiar órdenes de clientes en línea. Se pueden crear varias líneas de productos en una sola orden.

Se encuentran cinco tipos de información en la orden del producto: (1) número de producto, (2) fecha prometida, (3) cantidad, (4) número de cliente y (5) fecha requerida por el cliente.

El sistema también incluye un método para reducir proyecciones de órdenes de venta. El objetivo de esta lógica es presentar un panorama más exacto de las necesidades del mercado basado en proyecciones en curso y órdenes actuales.

#### Planeación de órdenes y explosión.

La lógica de planeación de órdenes en BPCS provee un método para satisfacer las demandas no cubiertas por el balance actual del inventario o de las entradas programadas. Las órdenes planeadas son creadas para satisfacer estas demandas.

La cantidad de la orden planeada es calculada usando una política

de orden para un producto. Estas políticas caen dentro de los siguientes grupos.

1. Discreto (lote por lote). Genera órdenes planeadas en cantidad igual a los requerimientos netos en cada día.
2. Costo mínimo. Usa el algoritmo Boe-Yilmaz para aproximarse al algoritmo de Wargner (Production and Inventory Management Journal of APICS, Second Quarter, 1983). Es muy útil para obtener la demanda inconsistente. Un producto con este código de orden debe de tener un costo especificado. Este parámetro indica el porcentaje promedio del costo estándar que tiene el producto en inventario para el año. El algoritmo calcula el costo de retención para un tiempo especificado:

$$U = \frac{\$ \text{ costo de retención } \times \text{ costo std}}{\text{req prox ord } 100 \% \times 365 \text{ días}} \quad M = \text{Cantidad}$$

$$\text{Costo de retención} = U \times (\# \text{ días hasta prox orden} - 1) \times M$$

El sistema compara el costo de retención con el costo por ordenar, para ver si es mejor combinar la demanda futura con la demanda actual en una sola orden o esperar a ordenar la demanda futura cuando sea necesaria.

Consideremos un ejemplo. Supongamos que se está listo para ordenar un producto y observar futuras demandas para ver si se

debe de incluir la siguiente demanda con la orden actual. Dada la siguiente información:

½ costo de retención = 50      costo std = 2

numero de días a retener (hasta la sig demanda) = 8

numero requerido para la sig demanda = 100

costo de orden por producto = 1.50

Costo de retención =  $\frac{50 \times 2}{100 \times 365} \times (8 - 1) \times 100 = \frac{700}{365} = 1.918$

100 x 365

365

Debido a que el cálculo muestra que el costo de retener un producto en inventario (1.918) es mayor que el costo para reordenar cuando la demanda existe (1.50), el sistema producirá una orden planeada sólo para la demanda actual. La orden para los 100 productos requeridos en 8 días será considerada en el momento de la explosión de los requerimientos de materiales.

3. Discreto arriba del tamaño del lote estándar. Producirá una orden planeada de un lote si los requerimientos son menores o iguales al tamaño del lote. Si los requerimientos exceden el tamaño del lote, el tamaño de la orden planeada será incrementada por el exceso.

El proceso de explosión en el sistema usa las órdenes planeadas para crear requerimientos por componentes. La explosión toma la orden planeada y lee la lista de materiales para determinar

requerimientos por componentes.

La lógica de explosión provee un método para manejar el factor de desecho para la relación padre-componente. El objetivo de este tipo de factor de desecho, es incrementar los requerimientos para un componente cuando ese componente tiene un desperdicio o proporción de pérdida de los otros componentes.

#### Programación Maestra.

El modulo de programación maestra contiene las siguientes opciones que ayudan a desarrollar la producción.

1. Proyección de la producción.
2. Generación de la producción maestra.
3. Generación de ordenes de producción.

El sistema permite mantener proyecciones de demanda de producción.

El sistema contempla varios métodos para utilizar la proyección.

Estos métodos son:

1. Mayor proyección a órdenes de clientes.  
(consumo de la proyección por demanda)
2. Suma de órdenes (de clientes) y proyecciones.  
(aumento de la demanda)
3. Sólo órdenes proyectadas.  
(demanda anticipada)

La generación de la producción maestra crea órdenes planeadas y requerimientos. La generación de la producción puede hacerse utilizando 2 métodos de MRP II: (1) cambio neto\* ó (2) regenerativo\*.

La programación maestra y la planeación de los recursos de materiales proveen una planeación efectiva y eficiente de las prioridades relacionadas con las necesidades de los materiales. Estos dos sistemas dan control a la producción, compras, y manejo de la información necesaria para el control de las operaciones de compra.

El modulo de programación maestra intenta ser realista con respecto a la producción de los productos. En BPCS la programación maestra es manejada por los requerimientos que vienen de la proyección. Se pueden llevar a cabo correcciones a la programación maestra o al plan de capacidad. La simulación de la programación maestra se puede ejecutar varias veces hasta que la programación maestra sea lo que se intenta producir y cuando se intenta producirlo.

Una vez que la programación maestra se establece, los requerimientos de materiales son ejecutados para calcular todos los requerimientos nivel por nivel, usando:

- 1) La programación maestra para saber lo que se fabrica.

\*ver glosario



- 2) Lista de materiales para determinar requerimientos para la programación maestra.
- 3) Inventario para conocer lo que está disponible.
  - 4) Tamaño de lotes para modificar lo que se ordenará.
  - 5) Tiempo de seguimiento para determinar cuando ordenarlo.

El sistema no actualiza automáticamente ninguna orden de producción desarrollada por la empresa. Una vez generada la producción maestra, se puede verificar lo programado mediante un reporte, el cual indica algún cambio o sugerencia (mensajes de excepción).

El sistema contempla seis mensajes de excepción que son:

- 1). Reprogramar la orden de producción para una fecha anterior.
- 2). Reprogramar la orden de producción para una fecha posterior.
- 3). No existe suficiente producción maestra para cubrir las demandas.
- 4). Orden de producción vencida para liberación.
- 5). Orden de producción vencida.
- 6). Orden de producción cancelada.

#### Planeación de los requisitos de materiales.

La planeación de los requisitos de materiales sugiere o recomienda modificaciones a introducir en las ordenes de producción y las ordenes programadas. La compañía sigue siendo responsable de

seleccionar y ejecutar las recomendaciones dadas por el sistema. El planificador de materiales, no el sistema, toma las decisiones y mantiene los programas.

Las acciones o recomendaciones del sistema de planeación de requisitos de materiales incluyen:

- Ordenar. Liberar la orden para la fabricación ó compra de materiales. Incluye cantidad, fecha de entrega y de inicio.
- Acelerar. Reprogramar el recibo de un artículo adquirido ó fabricado.
- Aplazar. Reprogramar la salida de un artículo adquirido ó fabricado.
- Cancelar. Suprimir o retirar una orden de compra ó fabricación del proceso de planeación.

Las dos alternativas básicas del sistemas de planeación de los requisitos de materiales son la regeneración de lo programado ó el cambio neto.

#### El subsistema de lista de materiales.

El subsistema de lista de materiales, como el subsistema de órdenes planeadas y el sistema de transacciones de inventario, proveen la información necesaria para calcular y determinar prioridades. Es un subsistema de soporte porque provee información, pero no forma

parte de los cálculos.

El subsistema de la lista de materiales define el como uno ó más componentes son traídos juntos para formar un producto padre. Estos productos pueden ser manufacturados, comprados ó materiales nuevos.

El sistema provee las capacidades para añadir, cambiar y borrar una lista de material. Le permite a un número ilimitado de componentes pertenecer a cualquier lista de materiales. BPCS incluye una cantidad de desperdicio para un componente. Esta cantidad de desperdicio representa cualquier pérdida para ese componente durante el proceso de fabricación.

El sistema de lista de materiales incluye los siguientes reportes:

- 1) Lista de material
- 2) Donde es usado un material
- 3) Varios niveles de lista de materiales

En un sistema de cambio neto, se requiere un mecanismo para mantener los requerimientos, siempre que las listas de materiales cambien. El sistema provee dos maneras de llevar a acabo esto: (1) Actualizando todo el tiempo las listas de material que son cambiadas y (2) Incluye una lógica que destruye completamente todos los requerimientos y órdenes planeadas cuando un cambio es realizado a lista de material.

### Subsistema de transacciones de inventario.

Las transacciones de inventario pueden dividirse en dos categorías básicas dentro del sistema: (1) planeadas y (2) no planeadas. Las transacciones planeadas son aquellas en donde algún tipo de asignación, recibo programado ó información de inspección debe de llevarse a cabo y actualizarse. El sistema soporta, por ejemplo, mover material directamente a inventario ó recibir material al área de inspección y subsecuentemente mover el material a inventario. El sistema incluye transacciones adicionales para rechazo, retorno, desperdicio y retrabajo de material.

Las transacciones no planeadas actualizan solamente el balance actual del inventario, como la entrada y salida no planeada de material y los ajustes de inventario.

El sistema maneja unidades fraccionarias ó decimales. El módulo de inventario cuenta con una opción de conteo cíclico de inventario. El conteo cíclico ayuda a identificar errores de registro en el inventario, además de mejorar la información de los registros de inventario. Por lo tanto ayuda a obtener una de las medidas importantes para MRP-II, como es el de tener un inventario del 95 por ciento de exactitud. Provee 3 métodos de conteo: (1) Ejemplo al azar de conteo cíclico, (2) conteo cíclico ABC y (3) conteo cíclico en grupo.

### Subsistema de control de piso y rutas.

En BPCS el control de piso permite:

- . La planeación de la disposición y flujo de fabricación
- . Control de la capacidad y las prioridades de fabricación
- . Cumplimiento de los objetivos de desempeño en cuanto a la calidad, entregas y producción.

El subsistema de rutas provee la información necesaria para calcular requerimientos de capacidad en el futuro. Define el proceso necesario para hacer ó comprar todos los productos.

El sistema provee las capacidades para mantener los recursos (centro de trabajo) y los registros de operación de ruta que relacionan los recursos a un producto. Existen transacciones para añadir, cambiar o borrar centros de trabajo. Además existen transacciones para añadir, cambiar o borrar una operación en la ruta para un producto. El sistema permite definir una ruta para componentes comprados. El sistema provee dos reportes importantes:

- 1) Consulta de rutas.
- 2) Lista de centros de trabajo.

El sistema permite desviaciones de la ruta normal para una orden, de la misma manera que permite desviaciones en la lista de material. Desviaciones en la ruta normal son aquellas situaciones en donde una operación es añadida, borrada ó cambiada para una

orden planeada.

#### Planeación de los requerimientos de capacidad.

La planeación de los requerimientos de capacidad en BPCS es una herramienta que muestra la capacidad para un centro de trabajo del mismo modo que los reportes de requerimientos de materiales muestran el panorama para el material.

El sistema muestra los requerimientos de capacidad utilizando dos métodos: (1) Carga finita\* y (2) Carga infinita\*, sin embargo, se recomienda siempre usar el método de carga infinita, ya que no se ha demostrado que los sistemas de carga finita funcionen.

Los requerimientos de capacidad son calculados, usando tanto las ordenes planeadas, como las que han sido liberadas. Los requerimientos de capacidad son calculados en horas estándar. La cantidad de la orden multiplicada por las horas estándar requeridas para una operación dan el total de horas estándar. La ventaja de trabajar con horas estándar es debido a que es la unidad de capacidad más simple y fundamental.

El sistema genera requerimientos de capacidad de la explosión del MRP. El sistema carga la capacidad como se encuentre. Después de generar los requerimientos, algún centro de trabajo puede estar con una sobrecapacidad. BPCS no puede hacer decisiones sobre las

\*ver glosario

necesidades de la compañía para ajustar lo programado para evitar la sobrecapacidad en los centros de trabajo. Estas desiciones pueden llevarse a cabo sólo por una persona dentro de la compañía. BPCS genera cinco reportes de capacidad.

1. Requerimientos de recursos de trabajo.
2. Requerimientos de recursos de maquinaria.
3. Actualización de los requerimientos de los recursos.
4. Actualización de los requerimientos de trabajo.
5. Actualización de los requerimientos de maquinaria.

Estos reportes contienen información similar en varios formatos, los cuales permiten tener un panorama de los datos en diferentes perspectivas.

#### Control de entrada/salida.

El sistema imprime un reporte que resume semanalmente las horas planeadas y actuales de entrada para cada centro de trabajo, así como una desviación acumulada. También incluye las horas planeadas y actuales de salida y una desviación acumulada. El sistema toma la información cuatro semanas hacia atrás y tres semanas hacia adelante.

#### Planeación de los recursos de distribución.

La planeación de los recursos de distribución en BPCS, permite una visibilidad completa de la red entera de distribución. Provee un

panorama exacto de la carga de tránsito y programación necesaria para soportar el programa de distribución.

La planeación del transporte es una parte interna de los recursos de distribución. La planeación del transporte es una manera de planear el peso y volumen a ser transportados, basado en el plan de los recursos de distribución.

El sistema BPCS despliega el total de los requerimientos de transportación por destino y método de envío, incluyendo:

- 1) Períodos de planeación de transporte.
- 2) Volumen y peso requerido.
- 3) Capacidad disponible en peso y volumen.
- 4) La capacidad bajo/sobre la capacidad disponible.

#### Compras.

En el módulo de compras de BPCS se pueden añadir, cambiar ó borrar requisitos de compra. El sistema consolida líneas individuales de requisitos, órdenes planeadas y/u órdenes planeadas en firme para crear órdenes de compra de productos. El sistema también permite registrar bienes en inventario, recibidos de los proveedores. El sistema trata a cada línea de compra como una transacción separada de inventario.

No se tomó la planeación de ventas como parte del sistema estándar,



debido a que la compañía elabora su planeación de ventas utilizando paquetes que se encuentran instalados en PC, por este motivo no era esencial que BPCS contara con un módulo de planeación de ventas. Se llevó a cabo una interfase con el modulo de nómina del sistema anterior (GOBLAL) por parte del departamento de sistemas, ya que BPCS no contaba con los parámetros necesarios para poder utilizar este módulo. El módulo de contabilidad fue implantado e integrado con los demás módulos del sistema estándar.

BPCS cuenta además con un reportador llamado BIR (BPCS Information Retrieval), el cual facilita el desarrollo de reportes que no son propios del sistema. La documentación de cada modulo de BPCS puede ser obtenida a través de los manuales y ayudas que se encuentran en cada módulo. El sistema tiene también una referencia cruzada de la base de datos para conocer cada uno de los archivos pertenecientes a BPCS. A continuación se describen los tiempos de respuesta que se evaluaron para el sistema:

Tiempo de respuesta	Día	Noche
Programación maestra.	1 hr	30 min
Planeación de los req. de material.	1 hr	30 min
Planeación de los req. de capacidad.	45 min	20 min
Reporte de control de entrada/salida.	45 min	30 min
Respaldos (Backups)		1 hr

Es también importante destacar que el sistema BPCS contiene,

además de los principios de diseño mencionados anteriormente, los principios del sistema AS/400 de diseño de interfase, los cuales son:

**Consistencia.** El primer paso para hacer que una interfase sea fácil de aprender es asegurando la consistencia. La consistencia le permite al usuario aprender técnicas y aplicarlas repetidamente a través de una aplicación.

**Orientación a objetos.** Uno de los mas importantes principios en diseñar una interfase con el usuario es orientar su atención en los objetos importantes para él. Los objetos son aquellas cosas que los usuarios conocen y trabajan con ellos en sus actividades diarias. En una aplicación de inventario, por ejemplo, los objetos pueden ser partes, clientes y almacenes.

**Flujo de acción de objetos.** Una consideración importante en la interfase con el usuario es la manera en la que el usuario requiere efectuar una acción. Un usuario puede requerir de una acción de dos maneras. La primera es identificando la acción deseada o trabajo y entonces designar al objeto en el cual realizar la acción. La segunda manera es identificar el objeto al cual se va ejecutar la acción y entonces designar la acción que se desea que se lleve a cabo.

**Reconocimiento y selección.** Reconocimiento y selección es también

una interfase fundamental. Debido a que el reconocer algo es más fácil que recordarlo y seleccionar es más fácil que teclear, una interfase con el usuario debe, cuando sea posible, presentarle al usuario opciones para poder seleccionar.

Rutas rápidas. Las rutas rápidas le permiten a un usuario con experiencia moverse directamente a una acción u objeto deseado.

Proveer al usuario de un texto. No importa como un usuario llega a una función, es importante darle suficiente información textual para hacerles saber que han llegado a la función que deseaba.

Agrupar. Otra manera de ayudar a un usuario es agrupando objetos: presentar piezas relacionadas de información en grupos.

En resumen, BPCS tiene la mayoría de las funcionalidades que la compañía buscaba. Por tal motivo se adquirió el paquete con el siguiente precio:

Precio base = \$ 12,000 dls por módulo

Modificaciones = 0

Costo de retrasos = 0

Por lo tanto el precio del sistema fue de :  $12,000 \times 6 = 72000$  dls

No se llevaron a cabo modificaciones importantes al sistema, debido a que primero se establecieron políticas y procedimientos para poder implantar MRP II, es decir se modificaron algunas funciones dentro de la compañía, en donde la gente participo con gran entusiasmo para cambiar su manera de trabajar. La mayoría de las modificaciones que se llevaron a cabo fueron en reportes, para cambiar títulos y leyendas. La figura 5-2 muestra el menú principal del sistema. Actualmente se encuentra instalado el paquete BPCS con los módulos del sistema estándar mencionados anteriormente.

Entre los beneficio que se han obtenido con el sistema se encuentran:

1. Existen programas validos para compra de productos y material nuevo. Esto permite un mejoramiento en la ejecución de entrega de pedidos.
2. Se pueden desplegar proyecciones seguras de compra futuras.
3. El uso de la lista de materiales ha tenido beneficios en distintos departamentos de la empresa, tales como:
  - a) Mercadotecnia. Para configurar los pedidos y requisitos del cliente
  - b) Planeación de materiales. Para planear los programas de materiales y fabricación.

LABORATORIOS COLUMBIA, S.A. de C.V. W4

Business Planning and Control System

Master Menu

Financial

1. Cost Accounting
2. Accounts Payable
3. Accounts Receivable
4. General Ledger
5. Multiple Currencies
6. Currency Translation
7. Payroll
8. Cash Management

- (Menu SSAC00)
- (Menu SSAP00)
- (Menu SSAR00)
- (Menu SSAG00)
- (Menu SSAM00)
- (Menu SSAT00)
- (Menu SSAU00)
- (Menu SSAD00)

Manufacturing

16. Mfg Data Management
17. Shop Floor Control
18. Capacity Planning
19. Master Scheduling
20. Material Req Planning
21. J1/Repetitive Mfg
22. C/Path
23. Advanced Process Ind.
24. Performance Measurement

- (Menu SSM00)
- (Menu SSM00)
- (Menu SSM00)
- (Menu SSM00)
- (Menu SSM00)
- (Menu SSM00)
- (Menu SSM00)
- (Menu SSM00)
- (Menu SSM00)

Distribution

9. Inventory
10. Purchasing
11. Order Processing
12. Billing
13. Sales Analysis
14. Dist Resource Planning
15. Forecasting

- (Menu SSIM00)
- (Menu SSIP00)
- (Menu SSOP00)
- (Menu SSOB00)
- (Menu SSAS00)
- (Menu SSAR00)
- (Menu SSFO00)

Systems Applications

25. BPCS Information Retrieval BPCDD
26. BPCS Budgets and Models
27. System Parameters
28. Documentation
29. Data Base X-Reference

- (Menu SSM00)
- (Menu SSM00)
- (Menu SSM00)
- (Menu SSM00)
- (Menu SSM00)

BPCS (c) 1990 by System Software Associates, Inc. Chicago, IL 60606 USA

Enter Option: 18. Signoff User: 2.0 stf  
 CMD/1-End CMD/3-Hsq CMD/4-Outq CMD/5-SndHsq CMD/6-Hsq CMD/8-Jobs HELP

FIG 5-2

c) Fabricación. Para describir la forma en que se han de producir los productos.

d) Contabilidad de costos. Para determinar los costos planeados y reales de los productos.

4) El uso del conteo cíclico ha permitido tener un registro de inventario casi exacto.

BPCS ha permitido además de implantar un sistema de trabajo como lo es MRP II, poder utilizar la información para poder medir el desempeño de las actividades de la empresa.

Como experiencia en este proyecto, se observaron algunos puntos que serían importantes mencionar en el caso de volver a participar en la selección del software y su implantación.

1. Involucrar más a los usuarios finales. Es de suma importancia involucrar al usuario en la implantación del sistema, ya que algunos usuarios se encontraron con un sistema instalado, y les llevo más tiempo adaptarse a él.
2. Documentar cada una de las actividades realizadas, así como los cambios realizados a los programas y archivos.
3. Definir un plan de trabajo por períodos de tiempo fijos y revisarlo en juntas programadas.

4. Contar con más de un asesor para dudas surgidas en la implantación.
5. Dar una explicación general del sistema a los usuarios de la planta y administrativos con el objeto de que comprendan el flujo de las operaciones de la compañía aplicadas al sistema automatizado.
6. Definir correctamente la información que será cargada al sistema por parte de las áreas involucradas en la compañía.

## 6. CONCLUSIONES

Se ha presentado una visión general de cómo seleccionar y evaluar un paquete de manufactura para MRP II. Sin embargo esta guía puede ser usada para evaluar un paquete de manufactura para cualquier sistema de elaboración de productos, llámese MRP-II, justo a tiempo (Just-In-Time), KAPAN, etc.

El desarrollo de este tipo de sistemas de manufactura permiten obtener objetivos y beneficios tangibles para cualquier compañía fabril. Es importante enfatizar que los sistemas automatizados solo son un soporte importante para la implantación de estas filosofías de trabajo, es decir, la gente es el principal componente para poder llevar a cabo cada una de las funciones involucradas en estos sistemas.

Tratando de participar en este proceso, se realizó el presente trabajo, que no pretende otra cosa, que contribuir modestamente a dar una idea de cómo obtener un paquete de software para poder implantar este tipo de sistemas de trabajo.

Para concretar esto, se presentó un caso práctico con una empresa que actualmente se encuentra operando uno de estos sistemas, es decir un sistema MRP II. Se utilizó un sistema estándar para poder verificar que el paquete contaba con las funcionalidades necesarias para poder implantar el sistema MRP II, además de verificar que no



contenía programas que pudieran afectar sus rendimiento. El paquete seleccionado y evaluado no necesariamente es el mejor dentro del mercado. En este caso contaba con las características deseadas, además de ser un buen simulador del sistema MRP II y de las actividades de la compañía.

## GLOSARIO

**BACKLOG.** Pedidos pendientes de surtir. Algunas veces referido como ordenes abiertas.

**CAMBIO NETO MRP.** Un método para procesar los requerimientos de materiales. La planeación de producción es hecha, tomando en cuenta los productos que han tenido un cambio que pudiera haber afectado el plan existente.

**CARGA FINITA.** Significa no asignar más trabajo en un centro de trabajo de lo que se espera puede ejecutarse. El término usualmente se utiliza cuando los centros de trabajo son cargados en el punto en el cual, el sistema comienza a tomar acción para resolver las condiciones de sobrecarga proyectada.

**CENTRO DE DISTRIBUCION.** Un almacén con bienes y/o productos terminados.

**CONTEO CICLICO.** Una técnica de conteo de inventario físico en donde el inventario es contado en un período programado en vez de una vez al año.

**DEMANDA.** Necesidad de un producto en particular.

**ENTRADA DE ORDENES.** El proceso de aceptar y trasladar lo que un

cliente desea en términos usados por la compañía.

**KABAN.** Sistema de manufactura desarrollado en Japón. Kaban significa "carta".

**MANEJO DE LA DEMANDA.** La función de reconocer y manejar todas las demandas de los productos para asegurar que la programación maestra esta informada.

**ORDEN PLANEADA.** Una orden con cantidad sugerida y fecha de terminación creada por el proceso de la planeación de los requerimientos (MRP). Las órdenes planeadas son creadas por el sistema, existen sólo dentro del sistema y pueden ser cambiadas o borradas por el sistema durante los procesos subsecuentes del MRP.

**PEGGING.** Desplegar, para un producto dado, los detalles de las fuentes de requerimientos. Contiene la información sobre donde es utilizado el producto.

**PLANEACION DE LOS REQUERIMIENTOS DE CAPACIDAD.** El proceso de determinar cuanto trabajo y/o recursos de maquinaria son requeridos para realizar los trabajos de producción y hacer los planes para proveer estos recursos. Las órdenes abiertas, así como ordenes planeadas en el sistema MRP II, son las entradas de los requerimientos de capacidad, las cuales las traduce en horas de trabajo para cada centro de trabajo por período de tiempo. La

porción de la computadora de los requerimientos de capacidad fue llamada carga infinita.

RECIBO PROGRAMADO. Dentro del MRP, las órdenes abiertas de producción y órdenes abiertas de compra son considerados como "recibos programados" en sus fechas de vencimiento y serán tratadas como parte disponible del inventario durante los procesos netos para el período de tiempo en cuestión.

REGENERACION MRP. Un método para procesar los requisitos de materiales. La regeneración destruye el antiguo plan de producción y recalcula completamente uno nuevo.

REPROGRAMACION AUTOMATICA. Permitir al sistema cambiar automáticamente las fechas de vencimiento en ordenes programadas cuando detecta que las fechas de vencimiento y de requerimiento se encuentran desfasadas.

RUTA. Un documento que detalla la manera en que se manufactura un producto. Incluye las operaciones a hacer realizadas, sus secuencias, los diferentes centros de trabajo que están involucrados y los estándares para que se lleven a cabo las operaciones.

**BIBLIOGRAFIA.**

1. MRP II: Unlocking America's Productivity Potencial. Oliver W. Wigth. 1984.
2. Production and Inventory Management into Computer age. Van Nostrand Reinhold.
3. Material Requirements Planning. Joseph Orlicky, Mac Graw-Hill Book Company NY. 1974.
4. MPR II. Making it happen. The implementer's guide to success with MRP II. Thomas F Wallace; Oliver Wigth Limited Publications, Inc, 1985.
5. APICS Dictionary. American Production and Inventory Control Society. Washington, DC, 1980.
6. Master Production Sheduling. Principles and Practice. WilliamL Berry. American Production and Inventory Control Society, Washington, DC, 1974.