

**UNIVERSIDAD TECNOLÓGICA NACIONAL  
FACULTAD REGIONAL RESISTENCIA**

*Ingeniería en Sistemas de Información*

**Materia Integradora de 4° nivel  
Administración de Recursos**

**Unidad Temática II: La Administración de los SI/TI – Subsistema de  
Provisión – SW y HW**

**Sección 2: Provisión de SW**

**Material de Estudio**

**Elaborado sobre la base de:**

**Robson, Wendy. Decisiones Estratégicas en Sistemas de Información I.  
Tomo 4. Colección Management Estratégico de Sistemas de  
Información. MP Ediciones. 2ª edición.**

**Puigjaner, Ramón y otros. Evaluación y Explotación de sistemas  
informáticos. Editorial Síntesis S.A. 1995. España.**

**Borovits, Israel. Computer Systems Performance Evaluation: Criteria,  
Measurement, Techniques, and Costs - Lexington Books.**

**Equipo Docente:**

**Prof. Asoc. Int. a/c de Cátedra: Ing. Claudia A. SORIA OJEDA  
Jefe de Trabajos Prácticos: Ing. Rosina RAMIREZ  
Auxiliar docente de 1ª: Ing. Jorge ROA**

## 2. PROVISIÓN DE SOFTWARE

Es el elemento de un sistema con el que interactúan los usuarios, y como es lo que determina qué pueden lograr, su adquisición es muy importante. Las decisiones sobre “¿de dónde?” obtener el hardware y software principalmente giran en torno a la opción del *canal* de suministro. Mientras que para el hardware esto significa qué fuente externa usar, para el software representa la opción de varios enfoques, tanto externos como internos, del proceso de desarrollo para el management. Tales opciones comprenden un continuo de posibilidades que se muestra en la **Figura 6**, donde se observa que no existe una división fija entre los desarrollos externos e internos, y que sólo las opciones 1a, 1b, 4 y 5 pueden definirse claramente.

Desarrollo Interno				Desarrollo Externo			
<b>1a</b> Sistemas y software de aplicación interno	<b>1b</b> Desarrollo de aplicación del usuario interno	<b>2</b> Personal contratado que trabaja en la empresa	<b>3a</b> Sistema de un fabricante adaptado a medida por personal interno	<b>3b</b> Sistema de un fabricante adaptado a medida por personal contratado	<b>3c</b> Sistema de un fabricante adaptado a medida por el fabricante	<b>4</b> Solución a medida de un fabricante ya lista	<b>5</b> Paquete estándar

**Figura 6. Continuo de alternativas obtención de software**

Si el ritmo de selección ha determinado la elección del software antes de elegir el hardware (o al menos como *parte* de la elección), entonces debe estar disponible todo el espectro de desarrollo. Sin embargo, si se ha optado por un hardware determinado, o si debe utilizarse un conjunto determinado, esa limitación puede implicar que sólo son posibles las opciones 1, 2 y 4.

Independientemente de que el *desarrollo* definitivo sea interno o no, es necesaria la participación de personal experto interno para establecer los *requerimientos*. Lo que determina qué rutas de adquisición son las *más* apropiadas es la índole de esos requerimientos, que capturan los principios comprendidos en la estrategia de IS, probablemente en términos de necesidades de compatibilidad, integración, comunicación, etc. Es obvio que los requerimientos deben incluir limitaciones al *proceso* de selección, por ejemplo, decidir a favor de cierto tipo de proceso, como el uso de paquetes estándar o realizados por contrato.

Está fuera del ámbito de esta cátedra analizar los detalles específicos de los diferentes métodos mediante los cuales los desarrolladores internos o los equipos de desarrollo externos pueden crear un sistema. Existe abundante material excelente donde se describen los principios de un método específico y las técnicas empleadas, o donde se considera la elección del método para determinada situación. Aquí nos ocuparemos de la elección del proceso que implica una negociación entre el control y los costos; entre la flexibilidad y el control de equipos formados por personal de la empresa, y los plazos más breves, los precios más bajos y posiblemente una mayor certidumbre que ofrecen las agencias externas. En esta sección se tratarán someramente las dos categorías extremas: los **desarrollos internos** y los **paquetes de soluciones**. Ha habido un desplazamiento desde la primera hacia la segunda, a pesar de lo cual aún existe la posibilidad real de una opción. Como señala Vowler (1995): “hemos tenido el dilema de comprar o construir durante décadas, y nunca desaparece por completo”.

### 2.1. Desarrollo interno

Sería extraordinario que las unidades principales de sistemas de software (el sistema operativo y el software de management de red) se desarrollaran en forma interna. El trabajo de desarrollo interno puede generar software de sistemas adicional que mantiene el funcionamiento de una infraestructura particular, a pesar de que ese *middleware* también se puede adquirir en forma separada. Los enormes centros de datos, o las empresas que tienen necesidades muy específicas, pueden ser la excepción a esta regla, en cuyo caso generalmente presentarán las características de una empresa de software.

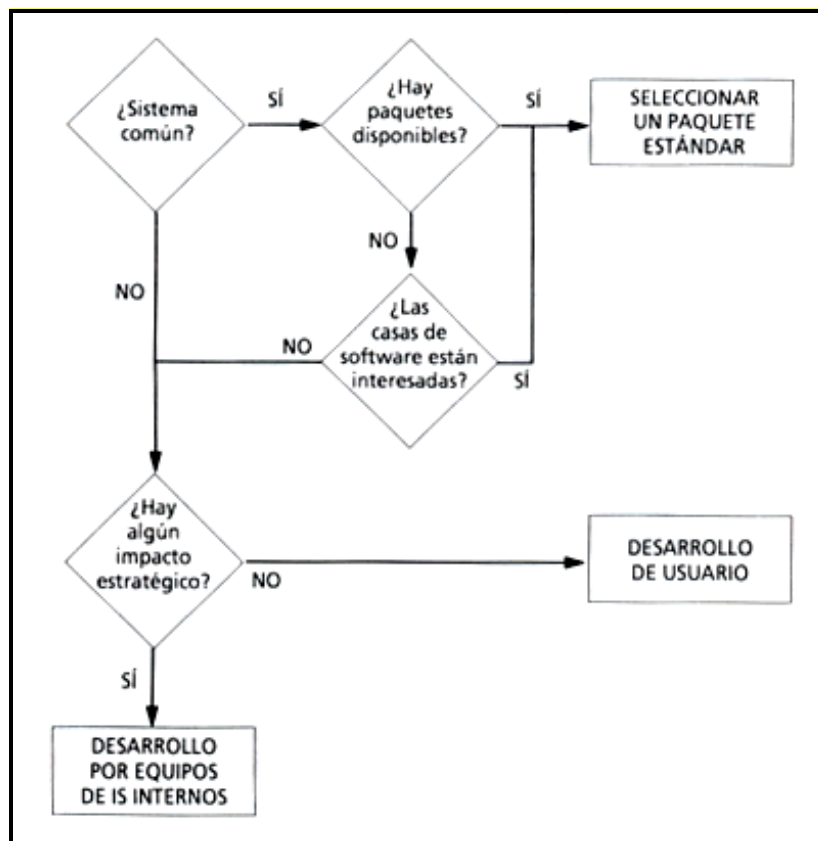
Sin embargo, cuando se considera el software de aplicaciones, la situación es muy diferente, y muchas empresas eligen desarrollar sus propios sistemas. Cuanto mayor es la sección de IS, y cuanto más centralizada está, mayores son las probabilidades de que desarrolle sistemas a gran escala. Las funciones de IS muy centralizadas toman muchos de los atributos de los organismos externos, y sus desarrollos comparten diversas características de los tipos (3) y (4) de la **Figura 6**, excepto que el margen de costos inevitable se retiene *dentro* de la corporación, en lugar de trasladarlo a un organismo externo.

Los enfoques de desarrollo de software han dejado de tener rigurosos requerimientos que ocasionaban largos plazos de entrega. Los proyectos de desarrollo de software actuales contienen incertidumbres inherentes debido a la necesidad de retener la flexibilidad en un entorno que cambia con rapidez. Esa incertidumbre inherente puede originar proyectos desmedidos que generan costos mucho mayores en relación con los beneficios potenciales y, lo que es muy importante para muchas empresas, las demoras equivalen a perder las oportunidades de competencia soportadas por los IS. Para evitar ese problema, el trabajo de desarrollo interno requiere:

- Control efectivo del proyecto por medio de enfoques actuales de management de proyecto.
- Predisposición y herramientas para la producción de software.

La necesidad de un control eficaz de proyecto es obvia, pero puede ser difícil lograrlo a menos que el software se realice mediante ingeniería de producción en lugar de ser creado individualmente. La ingeniería de producción crea la *fábrica* de software y reconoce los beneficios de realizar productos que, siempre que sea posible, contengan componentes estandarizados reutilizables. La orientación hacia los enfoques de desarrollos basados en objetos consiste precisamente en obtener componentes reutilizables. Dicha utilización de subensambles estandarizados permite que el esfuerzo se aproveche allí donde rinde beneficios, en lugar de que se gaste en elementos no esenciales. La construcción de componentes estándar y su ensamblaje en sistemas puede recibir buen apoyo de herramientas tecnológicas. Los componentes podrían obtenerse de fabricantes externos, y la reutilización interna o los desarrollos de propósito tendrían todos los elementos ensamblados rápidamente para adecuar las necesidades actuales. El desarrollo de software como proceso de producción y no como una tarea manual mejorará la productividad de todos los equipos de desarrollo de software; las casas de software fueron las primeras en reconocerlo, y también en emplear las herramientas y técnicas de producción. Los que dependen competitivamente de generar software para terceros han reconocido que el desarrollo de software crea sus productos, mientras que los grupos de trabajo internos se inclinaron a mantener una actitud de oficio “artesanal” hacia el desarrollo de software.

Ha habido una tendencia estable a prescindir de los equipos de desarrollo a gran escala formados por personal interno. Los factores que motivaron esa tendencia están dados por todo un historial de poca fiabilidad, falta de un management de proyectos capacitado y dificultad de retener los equipos de personal con aptitudes en un entorno de cambios veloces. Una de sus consecuencias fue el desarrollo de las aplicaciones orientadas al usuario, que constituyen la forma más común de trabajo interno para la mayoría de las empresas. Por lo general, los proyectos que no son adecuados para el desarrollo de usuarios simplemente se derivan a los proveedores externos. En la **Figura 7** se sugiere un posible proceso para seleccionar la ruta de desarrollo de software.



**Figura 7. Proceso de elección del enfoque de adquisición de software.**

En la actualidad, muchas empresas se ven obligadas a justificar sus decisiones de recurrir al desarrollo interno, cuando antes ésa era la alternativa usual. Este cambio puede ocurrir incluso durante grandes proyectos; por ejemplo, la municipalidad del distrito de Bournemouth (Reino Unido), que para su sistema de cobro a la comuna (que vendería a otras municipalidades) había optado por el desarrollo interno, en 1992 se decidió por un paquete de soluciones para reemplazar el sistema de impuestos. Se estimó que el precio de compra del paquete era más barato que el del desarrollo interno, y el plazo mucho más manejable. La **Figura 8** ilustra un ejemplo de la posible diferencia de costos entre un desarrollo interno y un paquete. Oponiéndose en cierta medida a esta tendencia, se encuentra el caso muy conocido de Hong Kong y el enfoque del Banco de Shanghai hacia la informática. Para ese banco, la informática es la banca y por lo tanto considera que la precisión que ofrecen los buenos desarrollos internos es esencial para el éxito de los negocios, y casi no utiliza paquetes a pesar del *aparente* ahorro de costos que representan.

Actividad de desarrollo	Desarrollo interno	Paquete estándar adquirido
Análisis de las posibilidades iniciales	160.000	160.000
Definición de los requerimientos	400.000	320.000
Diseño de sistema detallado	160.000	-N/A-
Codificación y testeo de módulos	320.000	-N/D
Testeo de sistema	210.000	160.000
Documentación y procedimientos manuales	190.000	40.000
Instalación, preparación de datos y conversión	160.000	240.000
Precio de compra	-N/D-	160.000
<b>Total</b>	<b>£ 1.600.000</b>	<b>1.080.000</b>

**Figura 8. Ejemplo de costos internos vs. costos de paquetes**

Como resultado de la tendencia hacia el desarrollo de aplicaciones por usuarios y de los paquetes adquiridos, gran parte del trabajo interno consiste en el mantenimiento y mejora de los sistemas existentes, por lo general en la reingeniería de dichos sistemas, y muy pocas actividades se dedican a los nuevos desarrollos. Los sistemas preexistentes suelen denominarse “sistemas heredados”, aludiendo a que son viejos elementos sin valor que deben reemplazarse tan rápido como sea posible. Cuanto mayor sea el volumen de desarrollos internos antiguos, mayor esfuerzo requerirá su mantenimiento. La tendencia actual hacia los paquetes de soluciones generalmente reduce el tiempo de mantenimiento, a pesar de que no se conocen los efectos a largo plazo que tienen las actividades extensivas de desarrollo de usuarios bajo presión de mantenimiento. Los paquetes, por supuesto, también tienen muchos de los aspectos negativos de los sistemas “heredados”; los paquetes de aplicaciones “viejos” también pueden provocar tanta inflexibilidad como el “antiguo” software a medida.

## 2.2. Uso de paquetes estándar.

Cada vez más empresas prefieren adquirir este tipo de software, que en esta sección estudiaremos con cierto detalle. En primer lugar, cabe señalar que esta sección no se refiere simplemente a los paquetes para PC. A pesar de que la plataforma PC ha incentivado el uso de las aplicaciones estándar, dichos paquetes están presentes en *todas* las arquitecturas de hardware; por ejemplo, Uniplex, un paquete popular de procesador de textos, corre en las arquitecturas Unix de mediano rango.

El número de empresas que dependen de los IS se incrementa en forma gradual, en especial en las áreas comerciales comunes, y aun así muchas carecen de especialistas internos en IS, de modo que para resolver este problema se volcaron a los paquetes de soluciones estandarizados. En 1985, aproximadamente un tercio de las empresas utilizaba paquetes de soluciones; en la actualidad, esa cifra alcanza el 95 por ciento, y abarca una amplia gama de aplicaciones. Los métodos comerciales no son tan variados como la mayoría de las empresas supone, y casi todos los paquetes estándar están bien equipados para satisfacer sus requerimientos. Por definición, los paquetes están predesarrollados, pero muchas firmas pueden ingresar datos en el proceso de desarrollo. Como están predesarrollados, deben tener como objetivo un mercado “amplio”. Para la mayoría de las empresas, adoptar dichas prácticas es una ventaja, sobre todo en las áreas de aplicación de baja importancia estratégica. Para las pequeñas empresas que se han informatizado, es impensable considerar otra cosa que no sean las soluciones en paquete. En la **Figura 9** se ilustran apenas algunos de los beneficios que ofrecen los paquetes comparados con los desarrollos a medida.

<ul style="list-style-type: none"><li>▪ Rápida disponibilidad</li><li>▪ Procedimientos comerciales sólidos</li><li>▪ Calidad conocida y comprobable</li><li>▪ Menores costos iniciales y generales</li></ul>	<ul style="list-style-type: none"><li>▪ Documentación verificable</li><li>▪ Mantenimiento disponible</li><li>▪ Investigación y actualización continuas</li><li>▪ Soporte y capacitación variados.</li></ul>
--	---

Figura 9. Ventajas de los paquetes estandarizados

Los rendimientos básicos en cuanto a tiempo y dinero suelen verse favorecidos por las soluciones en paquete. El costo de la compra o de la licencia del paquete, si bien puede ser alto, ofrece *certidumbre*, a diferencia del costo de un desarrollo interno de software. Mientras que los costos iniciales de la compra de un paquete son *relativamente* bajos, los costos de los desarrollos internos deberían compararse con su valor en cuanto a la incorporación de conocimientos a largo plazo. Cuando se adquiere un paquete, todo el aprendizaje de desarrollo queda retenido por el fabricante.

La etapa de análisis de requerimientos de la adquisición de un paquete es tan exigente como en el caso de los desarrollos internos, pero una vez determinada, la *entrega* de la solución puede ser muy rápida. Este hecho es muy atractivo para las empresas que luchan con los extensos plazos de los IS. La rapidez del proceso está sustentada por el hecho de que, en general, todos los paquetes incluyen diseños modulares, “señuelos” vinculados a otros paquetes, software de desarrollo interno y parámetros determinables, que definen las características detalladas del sistema. Por ejemplo, un paquete para realizar facturas permite hacer cambios en el diseño de la pantalla, en los porcentajes de descuento, en los plazos de pago y en el formato de los informes. Además, si bien la calidad de la documentación incluida en los paquetes puede *parecer* pobre, es probable que sea de un nivel mucho mayor que la generada por los equipos técnicos de desarrollo interno. Los paquetes estandarizados son el resultado del trabajo de las empresas de software que deben vender a los clientes, por lo que la buena presentación del producto es *competitivamente* fundamental para esas empresas. A diferencia de los grupos de trabajo internos, dichas empresas suelen emplear la autoría y la especialización técnica, y ellas mismas pueden ser consecuencia de la *comercialización* de las secciones internas de IS.

El soporte de un paquete puede variar en calidad, pero generalmente está incluido en el precio de compra. Sin embargo existe una tendencia de parte de las empresas más grandes a cobrar diferentes tarifas según los distintos niveles de soporte. La capacitación se incrementa al extenderse el uso del software, que si bien puede estar menos orientado a la empresa, debe ser más barato y estar más disponible. Existen paquetes de capacitación para la mayoría del software de más difusión.

En la adquisición de paquetes, el esfuerzo y las aptitudes dejan de centrarse en *el desarrollo* del software y pasan a enfocarse en la *selección* del paquete, que lleva un ciclo de desarrollo bastante diferente que el del software a medida. Además, como la mayoría de los paquetes se pueden adecuar, existen diferentes caminos a seguir, que también es preciso seleccionar. La identificación y la priorización de los requerimientos pueden compararse con algo ya existente, y el comprador tiene que localizar dónde están los desfases y decidir sobre un proceso para manejar la disparidad de capacidades. En la **Figura 10** se ilustran las distintas estrategias para tratar con los desajustes inevitables que habrá entre las necesidades de la empresa y las capacidades del paquete, aunque no deben adquirirse paquetes cuya adecuación sea menor que el 80 por ciento.

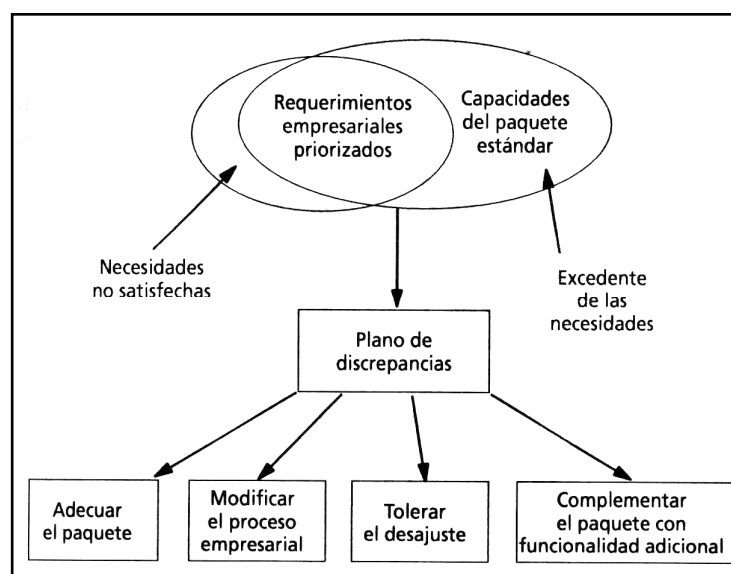


Figura 10. Estrategias para el desajuste entre las capacidades y las necesidades.

Modificar <sup>1</sup> el paquete no siempre es lo mejor. Como el copyright del software lo tiene el fabricante, el retiro de una licencia puede anular un costoso trabajo de adecuación. La mejora y la actualización de los paquetes, si bien suele ser conveniente, puede equivaler a que las actualizaciones también impidan el trabajo de adecuación ya realizado, tal vez hasta llegar al punto de que todo un sistema, conformado en base a una vieja versión y dependiente de características que ya no existen, se torne obsoleto. Según lo informa Green-Armytage (1995), un estudio de OTR demuestra que muchas empresas consideran que la actualización (con el consiguiente retiro de soporte para las viejas versiones) constituye una “trampa”, y que el 75 por ciento opina que no genera beneficios económicos.

El sentido de la adecuación (donde la adecuación no equivale a la adaptabilidad incorporada del paquete en sí) no sólo está cuestionado por el potencial de perder sistemas; un buen paquete utiliza buenos métodos, y la empresa se beneficia al cambiar las prácticas de trabajo consideradas no usuales en lugar de contemplar el agregado de procedimientos. Puede ser que el argumento de los costos favorezca tolerar un desajuste menor entre el paquete y las prácticas laborales, o por el contrario, tal vez la empresa pueda vincular el paquete o complementarlo con aplicaciones adicionales de desarrollo interno para evitar su modificación.

Antes de embarcarse en el proceso de adquirir un paquete estándar, o de definir una política de “paquetes preferidos” para la adquisición de software, existe una serie de aspectos relacionados con el management que se deben considerar. Ya hemos mencionado el tema de adaptar la empresa al concepto que tiene una persona ajena acerca de cómo debe llevarse a cabo el proceso empresarial, a pesar de que esto es menos polémico en la actualidad que en la época inicial de los paquetes. En primer lugar, es probable que exista una cantidad mucho mayor de soluciones que ofrecen más opciones, y así haya mayores posibilidades de adecuación. En segundo lugar, la orientación más precisa hacia las operaciones empresariales y la diferencia entre procesos fundamentales y actividades de apoyo han reducido la mentalidad del tipo “siempre se hizo así”. Se reconoce que la disciplina de imponer el estándar, los bajos costos y los métodos en las actividades de apoyo constituyen una ventaja.

Los que puedan elegir libremente entre los desarrollos internos y los paquetes de soluciones generalmente harán la selección de un paquete en base a la conveniencia de los costos y del tiempo. No obstante, cabe destacar que esto cada vez más está dejando de ser una cuestión de libre elección, pues sería imposible no comprar un paquete si previamente se decidió eliminar los equipos de desarrollo interno. El *downsizing*, mencionado en la Sección 3.2, es otra tendencia que favorece el uso de los paquetes. Para implementar la política de trasladar plataformas por lo general se requiere de plazos menores, característica inherente al uso de paquetes. Probablemente la política de *downsizing* se haya visto motivada por el ahorro potencial de costos; de hecho, el uso de paquetes está caracterizado por costos fijos y por una mayor flexibilidad. La adopción de los sistemas abiertos, señalada en la Sección 1.1, también favorece la utilización de paquetes, dado que la mayor independencia del fabricante genera una mayor competencia, y así se reducen los costos de los paquetes, haciéndolos aún más atractivos.

Para algunos elementos del portfolio de sistemas, el ahorro de costos es la principal consideración para hacer la selección. Es probable que una estrategia de IS genérica de recursos escasos se corresponda con esos sistemas de importancia empresarial relativamente baja, tal vez los categorizados como soporte de los segmentos industriales de la matriz de importancia estratégica de McFarlan y McKenney (1983). Para evaluar el ahorro de costos, se deben comparar los costos con precisión, aunque el ahorro generalmente surge del elemento codificador del ciclo de desarrollo, y no de las actividades de análisis empresarial. La tabla de la **Figura 8** (mostrada anteriormente) es un ejemplo de ese ahorro. Para la mayoría de las empresas, cuanto más lejos está el área de aplicación de la empresa “central”, más probabilidad habrá de que los costos sean los que motiven la selección de un paquete; a la inversa, cuanto más cerca esté la aplicación del núcleo de la empresa, mayor probabilidad habrá de que se incline por un desarrollo interno.

Los paquetes permiten menores costos y un rendimiento de inversión más rápido que los desarrollos internos o adaptados a medida por un fabricante, tal vez un 15 por ciento más rápidos y un 45 por ciento más baratos, pero estas ventajas deben evaluarse de manera realista. Schlender (1989) cita un caso extremo de ahorro potencial: la Lotus

---

<sup>1</sup> Hay 3 tipos básicos de modificaciones:

1. CORRECTIVAS. Un cambio correctivo implica el arreglo de un error preexistente. Este puede ser un error en las especificaciones originales o en el código en sí mismo. Esto es parte de un proceso llamado “depuración” (debugging), en el cual se examina un programa y se lo prueba para asegurar que ejecuta las tareas para las cuales fue diseñado. Estos cambios generalmente son la responsabilidad de los individuos que originalmente escribieron el programa.

2. ADAPTATIVAS. Implican la modificación del SW para adecuarlo al entorno en el cual se ejecuta la aplicación. Ejemplos de este tipo pueden ser la modificación de un programa para agregar otra opción, calcular proporciones adicionales o reestructurar archivos para que contengan más datos. Estos cambios se producen como resultado del aumento del uso de un programa, y la necesidad de que satisfaga dichas demandas. La responsabilidad de estos cambios recae sobre el usuario del producto de SW y se pueden hacer dentro de la organización o contratarlo externamente.

3. PERFECTIVAS. Los cambios perfectivos se refieren a la mejora o aumento de un producto de SW para incrementar las capacidades de performance. Al igual que para el caso de los cambios adaptativos, la responsabilidad de implementación de la modificación recae sobre el usuario del producto de SW.

Para los fines de esta discusión, los cambios adaptativos y los perfectivos se agrupan juntos y se llaman simplemente modificaciones o mejoras, para destacarlos de los cambios correctivos.

Development Corporation gastó 7 millones de dólares para desarrollar la versión 3 de Lotus 1-2-3, y 15 millones adicionales en testeo y control de calidad; actualmente se vende en el Reino Unido a unos \$ 480; claramente el enorme volumen de ventas puede resultar ventajoso para el comprador. Adecuar el paquete para que incluya rutinas del usuario, independientemente de quién haga el verdadero trabajo de adaptación a medida, suma costos adicionales al precio de compra del paquete. Dicha alteración siempre es más lenta y más cara que lo que se suponía, en especial si se realiza internamente, pues el equipo interno estará trabajando con material que no le es propio. Las cuestiones tales como convertir archivos en el formato requerido pueden ser técnicamente difíciles y muy lentas, pues los desarrollos internos suelen trabajar con los formatos existentes. Los problemas de conversión de archivos se reducen cuando los datos ya son mantenidos por un paquete y a medida que emergen los formatos estándar de datos y mejoran las rutinas de exportación e importación incluidas en los paquetes. Los desarrollos internos casi no las utilizan.

Los menores costos de desarrollo pueden estar compensados por los mayores costos operativos. Es posible que el enfoque de "menor denominador común" adoptado por un paquete genere más ineficiencias que los desarrollos internos, que pueden mejorarse para un entorno dado. Sin embargo, también es posible que un paquete sea más eficiente en todo excepto en el entorno más reservado, pues una de las características de un paquete es la capacidad del fabricante para hacer testeo y realizar modificaciones a gran escala. Utilizar paquetes como herramientas de desarrollo puede generar costos altos y no tan necesarios. Vowler (1994) advierte que el empleo de entornos visuales de bajo costo orientados al desarrollo de usuario, por ejemplo las capacidades de las planillas de Windows, oculta la lógica del código, que finalmente ocasiona mayores costos generales de IS a pesar del bajo costo inicial del paquete.

Un costo adicional final es el de la capacitación, que por supuesto no es exclusivo de los paquetes, sino también un costo extra subestimado en el trabajo interno. La capacitación requiere agregar una suma de dinero al precio de compra, pero puede ser menor que si se utilizara software de desarrollo interno, pues los paquetes estándar brindan cursos de capacitación estandarizados. Además de la capacitación, también se estandariza el soporte, aunque en algunos lugares, como en el Reino Unido, se carece de soporte a distancia. Sin embargo, lo importante del software en paquetes es que si su estilo, documentación o soporte parecen inadecuados, se puede elegir otro. No todos los paquetes son iguales; la confiabilidad del proveedor y las expectativas a largo plazo pueden ser factores importantes. En la Sección 4 se consideran las cuestiones relacionadas con la selección y el management del fabricante.

Ya hemos mencionado que adquirir un paquete puede no ser producto de una libre elección. Si la empresa opta por una práctica en la que se prefieren los paquetes, está estableciendo que no habrá especialización interna, lo que a su vez predetermina las opciones futuras. Los proyectos de desarrollo interno son el terreno de capacitación para que exista especialización en el desarrollo. A menos que esta adquisición represente una parte menor y trivial del portafolio de desarrollo, la opción de utilizar paquetes constituye una decisión a largo plazo; sin embargo, para equilibrar ese lapso, es posible que se permita la expansión a opciones posteriores de internase, migración e integración. En muchos casos, se puede adquirir el mismo paquete para una amplia gama de arquitecturas de hardware y sistemas de software, y cuando es probable que se alteren, como en el caso de las empresas que se embarcan en iniciativas de reducción y migración hacia sistemas abiertos, seleccionar un paquete adecuado en el presente puede facilitar los cambios en el futuro. Un argumento en contra de adoptar una perspectiva a largo plazo cuando se seleccionan paquetes consiste en que una empresa no debería emplear procesos de selección prolongados, complejos y costosos orientados a necesidades de largo plazo cuando adquiere paquetes para las áreas que cambian rápidamente; en cambio, la empresa debería utilizar un proceso de selección rápido y barato para la adquisición y concentrarse sólo en las necesidades actuales más fácilmente identificables. Ese proceso de selección más económico permite que el paquete se amortice rápidamente, tal vez al cabo de tres años, y que se comience otra vez con un paquete más nuevo, con más capacidades, más adecuado para los requerimientos de ese momento.

Una característica importante de la adquisición de soluciones de software en paquete es que la pericia fundamental de adquisición deja de estar basada en el desarrollo y se concentra en la especificación de los requerimientos, es decir, pasa del área de computación, de diseño y programación de sistemas, al área empresarial, de priorización y selección. En consecuencia, surge un aspecto importante relativo al management, dado que no necesariamente se requiere que el especialista en IS desempeñe una función principal en la selección del paquete. Aun cuando los IS estén involucrados, en lugar de controlar la mayor parte del proceso, pueden desempeñar una serie de funciones, tal como lo hacen con los desarrollos internos que no tengan que ver con las aplicaciones controladas por el usuario. La computación departamental distribuida y las soluciones de paquetes estandarizados se alimentan mutuamente; en la **Figura 11** se muestra una posible división de tareas.

<b>Tarea</b>	<b>Grupo responsable</b>
Adquisición de recursos	Unidad de negocios
Definición de requerimientos	Unidad de negocios
Implementación del sistema	Unidad de negocios
Management del proceso de adquisición	Unidad corporativa de IS
Resguardo de la calidad técnica	Unidad corporativa de IS
Negociación del contrato	Unidad corporativa de IS
Modificación del paquete	Unidad corporativa de IS y/o fabricante del paquete
Mantenimiento del sistema	Unidad corporativa de IS y/o fabricante del paquete
Creación del equipo de personal	En conjunto
Selección del paquete	En conjunto

**Figura 11. Posible división de tareas para la adquisición de paquetes de software.**

Como todo software requiere mantenimiento, deberá establecerse quién será responsable por el mantenimiento del paquete y cómo se adecuan las modificaciones en el contrato de mantenimiento. Si la empresa va a realizar modificaciones, debe quedar claro quién tiene los derechos del código fuente. Como el costo de mantenimiento fácilmente puede igualar el precio de compra, es posible que se justifique una solución aún más generalizada, como un generador de aplicaciones en lugar de un paquete de aplicaciones. El código fuente raramente es proporcionado por los fabricantes, y es posible que se encuentre depositado en una agencia de custodia (que lo posea un tercero, con acuerdo de ambas partes), de modo que la empresa compradora tenga la garantía de contar con un soporte alternativo en caso de que el fabricante original cese sus funciones.

Muchos de los factores relacionados con el management que se describen en esta sección pesan claramente a favor de los paquetes de soluciones. El factor en contra más importante está dado por el lapso inherente que tiene lugar antes de que un proceso empresarial determinado reciba el apoyo de un paquete. Obviamente, la empresa que realiza algo por primera vez no tendrá acceso a un paquete. Las fuerzas operantes en el mercado deben dar tiempo a que los desarrolladores de software identifiquen las necesidades del mercado para luego diseñar y crear el paquete, que eventualmente será una mercadería “masiva”.

El desarrollo interno seguirá relacionado con la iniciativa de obtener ventaja competitiva en un área del núcleo empresarial. Por definición, dicha iniciativa estará basada en la innovación, pero, también por definición, un paquete debería estar disponible para todos. El proceso de elegir el método para la adquisición del paquete se ilustró anteriormente en la **Figura 7**, y la esencia de ese esquema de decisión implica que existen tres situaciones ideales de uso que recomiendan la opción de paquetes de soluciones:

- Un conjunto de aplicaciones bien integradas para una empresa relativamente pequeña que desea adoptar un paquete sin realizar cambios. Esto también podría aplicarse a una división pequeña de un grupo mayor, y se recomienda especialmente cuando dicho conjunto de aplicaciones es provisto por un solo fabricante.
- Una aplicación no fundamental para la actividad principal de la empresa, que tenga una internase bien definida y relativamente simple para otras secciones del portfolio de aplicaciones de la empresa.
- Una aplicación compleja que necesite de especialización técnica en un área donde la empresa no considere que puede obtener una ventaja competitiva importante. Si la especialización técnica es de bajo valor competitivo, el costo de desarrollar dicha especialización no se justifica, al menos solo en esta área.

Los paquetes pueden elegirse no sólo como elementos individuales; existen dos situaciones que se deben considerar. La primera consiste en emplear enfoques estandarizados. Una empresa puede optar por comprarle todos sus paquetes al mismo proveedor a fin de tener una internase común que incremente la productividad del usuario, y tal vez reduzca los costos de compra, capacitación y soporte. Utilizar paquetes estandarizados debiera ofrecer economías de escala en las copias con licencia, brindar más soporte y capacitación, e incrementar la movilidad del personal en la empresa. Sin embargo, los individuos, las cuestiones políticas y otros factores más concretos, como la diferencia de lenguajes y las leyes de los grupos internacionales, pueden requerir compromisos. El problema inherente al preferir un paquete (es decir, el de no estar hecho a medida) en lugar de un desarrollo interno es doble cuando una sección de la empresa debe utilizar un paquete estándar que ni siquiera ha elegido.



Existen cuatro enfoques diferentes para definir el grado de libertad que pueden tener las áreas empresariales al seleccionar los paquetes:

1. **Libertad total:** este enfoque permite que todas las áreas empresariales elijan su paquete preferido y puede provocar el caos, donde nadie sabe qué comprar, cuáles son los costos de instalación, capacitación y soporte, ni tampoco los costos futuros, posiblemente enormes, que surgen de las oportunidades perdidas de integración. Sin embargo, cada área puede optar por un paquete que se adecue bien a sus requerimientos sin compromiso.
2. **Intercambio de datos:** este enfoque permite que los sectores individuales adquieran el paquete que desean siempre que soporte el intercambio de datos con otros paquetes y sistemas, según sea necesario.
3. **Lista:** en este enfoque, algún área de la empresa dedicada a la definición de normas confecciona una lista breve de los paquetes aceptables; los sectores individuales sólo pueden optar sobre la base de esa lista.
4. **Normas fijas.** el área dedicada a la definición de normas establece el paquete permitido, y de ese modo los sectores individuales pueden optar por “cualquier color que deseen, siempre que sea negro”.

Todos estos enfoques, excepto el primero, traen a colación el tema de quién define y controla las normas a seguir. El último enfoque, al especificar un solo paquete para toda la empresa, tiene muchas ventajas en cuanto a su costo, pero puede incrementar los costos de hardware, pues puede tratarse de un paquete poderoso que “consume” mucha capacidad del procesador, más de la necesaria para los sectores individuales, o bien puede reducir su eficacia si se toma un enfoque de mínimo denominador común para los requerimientos de la aplicación.

La segunda situación referida consiste en comprar un paquete integrado o bien un juego de paquetes, en lugar de seleccionar paquete por paquete. Los costos convenientes, la compatibilidad garantizada y la facilidad de uso asistida uniformemente son las ventajas que ofrecen ambas clases de paquetes. Como lo demuestra la **Figura 12**, las ventas de la mayoría de los paquetes integrados e individuales de PC están perdiendo terreno con respecto a las ventas de *suites* de paquetes, lo que indica la popularidad de esas selecciones agrupadas. Las *suites*, ya sean de un único proveedor o para un único entorno, ofrecen ventajas de compatibilidad y uniformidad al adoptar una internase de usuario única, como la *de* Microsoft o para Microsoft Windows, la popular internase gráfica de usuarios de PC.

<i>Áreas de aplicación con un incremento constante en el volumen de ventas:</i>	<i>Áreas de aplicación con una reducción constante en el volumen de ventas:</i>
<ul style="list-style-type: none"> <li>• Suites de paquetes (ej.: Office 95)</li> <li>• Software de management de proyectos (ej.: EasyProject)</li> <li>• Motores de bases de datos (ej.: Access)</li> <li>• Aplicaciones integradas para oficina (ej.: Works)</li> </ul>	<ul style="list-style-type: none"> <li>• Planillas de cálculo (ej.: Excel)</li> <li>• Procesadores de textos (ej.: Word)</li> <li>• Paquetes contables (ej.: Sage)</li> <li>• Gráficos para presentaciones (ej.: PowerPoint)</li> </ul>
<p><b>Nota:</b> durante la década de 1990, el incremento en el volumen de ventas de las <i>suites</i> de paquetes ha sido de varios cientos por ciento, mientras que el incremento de ventas de las aplicaciones de management de proyectos ha sido menor que el 100 por ciento; el aumento en las ventas de las aplicaciones integradas puede haber sido de un solo dígito. El bajo incremento y la reducción neta en los paquetes de aplicaciones <i>stand-alone</i> reflejan tanto la madurez del mercado (y por consiguiente, una limitación a la capacidad de crecimiento) como la pérdida de ventas con respecto a las <i>suites</i>.</p>	

**Figura 12. Patrón cambiante del volumen de ventas de paquetes de sw durante la década de 1990**

## 2.3 UNA TÉCNICA PARA LA DETERMINACIÓN DEL CANAL DE OBTENCIÓN DE SW.

En el proceso de determinación del canal de obtención del SW, el punto de decisión esencial es el de la definición del canal más adecuado. A continuación se propone una técnica, consistente en una lista de criterios y un análisis de riesgos, que puede aplicarse para definir la conveniencia de alguna de las alternativas.

### 2.3.1. Criterios para seleccionar canales de provisión de SW

Se listan a continuación una serie de criterios (NO SON EXCLUSIVOS NI EXCLUYENTES respecto de otros que puedan surgir dadas las características particulares de la situación/organización en estudio) que pueden utilizarse en el proceso de decisión sobre la provisión de SW. Los criterios listados en el siguiente punto deben considerarse en el proceso de decisión del canal de obtención de SW. Esta consideración consiste en **determinar, para cada alternativa, en qué grado los criterios listados se cumplen, afectan, benefician, constituyen una ventaja o una desventaja, según la situación de la organización y la estrategia**. Es decir, según el impacto estratégico que representan. Los requerimientos varían según la organización y la situación, y no hay una lista de criterios igual a otra.

- **Costo:** ¿La organización puede justificar el gasto desde una perspectiva costo/beneficio?.
- **Disponibilidad:** ¿Cuánto tiempo después de tomada la decisión estará el SW listo para usar?. ¿Cuán fácil será ponerlo en operación?.
- **Mantenimiento:** ¿Quién será responsable de mantener el sistema efectuando modificaciones correctivas?.
- **Especificidad de la Aplicación:** ¿El SW satisficará necesidades específicas o será demasiado general?.
- **Modificaciones:** ¿Cuán adaptable es el SW?. ¿Se lo puede modificar y mejorar fácilmente (modificaciones adaptativas y perfectivas)?.
- **Documentación:** ¿El SW está bien documentado?. ¿El código del programa es legible?. ¿Se lo puede entender fácilmente, de manera de poderse efectuar las modificaciones?.
- **Utilización de recursos:** ¿Cómo se utiliza a los analistas y programadores actuales de la organización usuaria?. ¿Hay amplitud de tiempo de máquina para el desarrollo, sin afectar las operaciones diarias?.

### 2.3.2. Análisis de alternativas de obtención de SW

Una vez que se han determinado todas las alternativas, se procede a:

1. Determinar un subconjunto de los criterios que serán la base de la evaluación.
2. Luego, se asignan pesos según la importancia relativa de cada criterio base.
3. Después se asignan puntajes para cada combinación Alternativa/Criterio, sobre la base de la escala definida  
Es decir, el equipo técnico deberá establecer una **Escala de Puntuación**, por ejemplo de 1 a 10, con la cual calificará a cada alternativa en función al nivel de cumplimiento de cada criterio determinado.”
4. Finalmente, se multiplican los puntajes por los pesos y se suman los resultados para cada alternativa. La alternativa que tenga el puntaje más alto satisfará mejor los requerimientos del usuario.

**Ejemplo.**

Se definen tres criterios: costo, documentación y disponibilidad. Las tres alternativas a considerar son adaptación del fabricante, desarrollo interno y desarrollo por personal contratado.

La **Tabla 1** muestra la aplicación del procedimiento descrito.

CRITERIOS	PESO	ALTERNATIVAS					
		ADAPTACIÓN DEL FABRICANTE		DESARROLLO INTERNO		DESARROLLO PERSONAL CONTRATADO	
		P	PxP	P	PxP	P	PxP
Costo	0,3	6	1,2	9	2,7	8	2,4
Especificidad de la Aplicación	0,5	6	3,0	10	5,0	10	5,0
Disponibilidad	0,2	5	1,0	3	0,6	4	0,8
TOTAL	1,0		5,2		8,3		8,2

donde: P = Puntaje ; PxP = Puntaje x peso

Se plantea un mínimo de calificación (o punto o nivel de corte) que debe estar previamente definido.

Ejemplo: Supongamos que se define una calificación de 7 puntos porque esto refleja que la alternativa calificada está asegurando entre un 70% a 75% de cumplimiento de los requisitos (traducidos a los criterios usados). Entonces, la opción "Adaptación del fabricante" es la menos calificada y consecuentemente se descartará.

Si se impone un mayor análisis por alguna razón (en el caso del ejemplo, las dos alternativas poseen puntaje muy próximo); se puede recurrir a una segunda técnica: **Análisis de las probabilidades de éxito y riesgo.**

### 2.3.3. Análisis de las probabilidades de éxito y riesgo.

El objetivo de este análisis es descartar aquellas opciones que tengan alta probabilidad de ocurrencia de un factor de alto impacto estratégico. El procedimiento es el siguiente:

1. Se determinan los factores de riesgo.
2. A cada riesgo se le da una probabilidad de ocurrencia (alta, mediana o baja) y un puntaje similar por el impacto que tendrá en la organización.
3. Se obtienen los totales.
4. Se selecciona la alternativa de menor impacto o riesgo.
5. Se deben descartar aquellas opciones que tengan alta probabilidad de ocurrencia de un factor de riesgo con alto impacto.

Continuando con el ejemplo, se considerarán 2 factores: utilización de recursos y mantenimiento. Los riesgos asociados son sobreutilización de recursos y escaso apoyo para mantenimiento.

La Tabla 2 muestra el análisis.

RIESGOS	ALTERNATIVAS			
	DESARROLLO INTERNO		DESARROLLO PERSONAL CONTRATADO	
	Probab.Impacto		Probab. Impacto	
Sobreutilización de recursos	A	A	B	A
Escaso apoyo para mantenimiento	M	M	A	M

De la Tabla 2 se deriva que para la situación ejemplificada la opción "Desarrollo interno" tiene una alta probabilidad de sobreutilización de recursos cuyo riesgo es alto. El escaso apoyo para mantenimiento es un riesgo de mediano impacto para la organización, con una mediana probabilidad de ocurrencia en la alternativa "Desarrollo interno" y una alta probabilidad de ocurrencia en la alternativa "Desarrollo personal contratado".

Puesto que la alternativa desarrollo interno en el ejemplo tiene una alta probabilidad de sobreutilizar los recursos del sistema y es un factor de alto impacto, debe descartarse. La alternativa favorable será contratar personal para el desarrollo del SW.

## **2.4. MÉTODO DE EVALUACIÓN DE PAQUETES ESTANDARES.**

Si del proceso de decisión sobre el canal de provisión de SW, se define que la alternativa es la adquisición de un paquete estándar, el proceso tiene una segunda etapa que es la decisión sobre cuál de las ofertas de paquetes disponibles en mercado se adecue mejor a las necesidades de la organización. Entonces la cuestión central para esta segunda etapa es la de cuantificar y/o medir los requisitos que el paquete deberá cumplir. Por ello es esencial definir cuidadosamente el conjunto de criterios para evaluar a las ofertas.

### **2.4.1. CRITERIOS DE SELECCIÓN DE PAQUETES ESTANDARES DE SW**

El proceso de provisión de Software por paquetes estándar requiere obligatoriamente considerar la carga de trabajo representativa, de modo tal de determinar los potenciales particulares de/los paquetes, respecto a las expectativas particulares de los destinatarios del mismo. Esto significa que existirán parámetros cualitativos y cuantitativos necesarios de ser reflejados dentro de las características y sub características presentadas en la lista de evaluación, pero orientadas a aspectos particulares de los Usuarios. En consecuencia, primeramente no se evaluarán aspectos técnicos de los paquetes, sino ASPECTOS RELACIONADOS CON LOS USUARIOS.

Se hace necesario entonces establecer un conjunto de Criterios que permitan ser ajustados en función a las expectativas deseadas de los paquetes a evaluar, que pudieran reflejar potenciales particulares, y que sirvan al mismo tiempo de elementos de evaluación.

Casi todos los objetivos de los productos de SW se ubican en una o más de las 11 categorías siguientes. Para evaluar cada criterio debe contestarse un conjunto de preguntas, basado en el cuestionario del usuario, cuestionario del vendedor y/o los resultados del benchmarking de paquetes.

**Funcionalidad.** (Se refiere a aspectos operativos: si el paquete cubre todos los procesos que debería realizar)

¿Las transacciones de entrada, archivos e informes, contienen los elementos de datos necesarios?. ¿Se efectúan todos los cálculos y pasos de procesamiento como Ud. quiere que se realicen?. ¿Los tipos de consultas ad-hoc son los que Ud. quería?.

**Capacidad.** (Si el paquete cubre aspectos de administración de datos, en cuanto a tamaño y cantidades, transacciones)

¿Será el producto capaz de manejar sus requerimientos en cuanto a tamaño de archivos, número de elementos de datos, número de entradas de tabla, volumen de transacciones, volumen de informes y número de ocurrencia de ciertos elementos de datos?.

**Flexibilidad**

¿Se pueden cambiar fácilmente el formato de las transacciones y de los informes?. ¿La disposición de las pantallas puede cambiarse fácilmente?. ¿Es difícil agregar nuevos cálculos o pasos de procesamiento?. ¿Pueden adaptarse los programas a nuevas aplicaciones?.

**Uso** (Se refiere a aspectos de facilidad de uso, interfaz amigable, nivel de conocimientos técnicos, etc.)

¿El nivel de conocimiento técnico necesario para usar este producto correctamente, armoniza con el nivel de conocimiento de los que lo van a usar?. ¿Los informes y despliegues por pantalla, son fáciles de leer e interpretar, e informativos?. ¿El vendedor ofrece entrenamiento?. ¿Los usuarios estarán entusiasmados con el producto?.

**Confiabilidad** (Que el paquete falle lo menos posible (MTBF) y que se pueda recuperar rápidamente de “caídas”)

El producto tiene diseño modular?. ¿Se ha usado lo suficiente como para asegurarse que la mayoría de sus errores se han depurado?. ¿Qué partes del sistema no se podrán usar cuando otra parte falle?. ¿El producto se apoya en HW proclive a errores, o en enlaces de comunicación poco aptos?. ¿El producto incorpora capacidades para detectar y auto-corregir errores?. ¿El usuario puede cometer errores que produzcan la caída del sistema?. ¿Cuáles son las capacidades de recuperación?.

**Seguridad** (Se refiere a aspectos de restricciones de ACCESO Y RESGUARDO de la información)

¿El producto incorpora tipos standard de controles?. ¿Permite procedimientos de respaldo correctos?. ¿Tiene maneras de prevenir la extracción de datos importantes de archivos o de transacciones?. ¿Ayuda a prevenir que los empleados entren transacciones sin autorizar o corran programas sin estar autorizados?. ¿Suministra detección y diagnóstico adecuado de errores de entrada de datos, y de otro tipo de errores?. ¿Se lleva un diario de transacciones?. ¿Se provee un procedimiento standard para verificar que el sistema funciona correctamente?.

**Performance** (Se refiere al NIVEL DE RESPUESTA del paquete, dependiendo de una carga de trabajo representativa. Siempre está “en conflicto” con el criterio de Capacidad)

¿En qué proporción el producto usará ciclos de máquina?. ¿En qué proporción requerirá acceso a discos?. ¿Cuánta memoria principal requerirá?. Si se corre en un sistema virtual, ¿cuánta actividad de paginado causará?. ¿Cuántos usuarios pueden actuar sobre el sistema antes de que degrade?.

**Mantenimiento**

¿Se suministran los programas fuente?. (A menudo, no). Si el vendedor realiza el mantenimiento, ¿cuán confiable y accesible es su empresa?. Algunos vendedores operan a través de casillas de correo: ¿qué nivel y calidad de mantenimiento suministrará el vendedor?. ¿La garantía es por escrito?. ¿Las modificaciones al sistema, cancelarán la garantía?. ¿El producto se puede usar fácilmente en otro entorno operativo?. Si su personal se ocupa del mantenimiento, ¿los programas están escritos en un lenguaje que les es familiar?. ¿Usan técnicas que les son desconocidas?. ¿Se suministran los conjuntos de datos de prueba con documentación adecuada de cómo usarlos y qué resultados esperar?. Obtenga las opiniones de usuarios pasados y actuales.

**Propiedad** (Se refiere a los TIPOS DE LICENCIA Y DERECHOS que tendrá sobre el producto: Licencia de Uso, Lic. De Propiedad, Copyright del Producto y/o de la Documentación, etc.)

¿Qué clase de derechos sobre el producto Ud. compra?. ¿Puede venderlo o alquilarlo a otro?. Si el vendedor hace este producto especialmente para Ud., ¿lo comercializará a otros? Si es así, ¿le pagará regalías por cada venta?. ¿Hay restricciones en los propósitos para los cuales se puede usar el producto?. ¿Hay restricciones de copiado del producto o su documentación?. ¿Podrá Ud. obtener derechos completos de propiedad y copias de los programas fuente si el vendedor abandona el negocio?.

**Minimización de Costos operativos y de mantenimiento**

¿Cuánto cobra el vendedor por mantener y mejorar el producto?. ¿Con cuánta frecuencia se requerirá mantenimiento?. ¿Cuál es el costo, por transacción, de uso de este producto? ¿Cuál será el costo de almacenamiento?. La compra de este producto, ¿requerirá la adquisición de HW adicional, o personal?. ¿Cuánto costará esto?. ¿Cuál será el periodo de vida útil de este producto?.

**Minimización de Costos de compra e Instalación**

¿Qué costos iniciales existen además de los costos básicos del producto?. ¿Deberá Ud. pagar gastos de envío y transporte y alojamiento del staff del vendedor que instale el sistema y entrene a su personal?. ¿Habrá demoras entre la colocación de la orden de compra y la entrega real del producto listo para usar?. ¿Cuánto costará esto?. ¿Cuáles serán los costos de conversión?.

### **2.4.2. TÉCNICA DE EVALUACIÓN.**

Una vez establecidos los criterios, se puede efectuar una comparación de los paquetes de SW propuestos. La Tabla 3 presenta un ejemplo de asignación de pesos y los puntajes. Como se estableció la escala de calificación 1 a 10, la puntuación debe mantenerse, entonces

Tabla 3.- Puntajes con asignación de pesos.

CRITERIOS DE SELECCIÓN	PAQUETE 1			PAQUETE 2	
	PESO	PUNTOS	P x P	PUNTOS	P x P
Funcionalidad	0,10	5	0,5	7	0,7
Capacidad	0,10	6	0,6	8	0,8
Flexibilidad	0,10	6	0,6	6	0,6
Uso	0,20	8	1,6	4	0,8
Confiabilidad	0,05	8	0,4	8	0,4
Seguridad	0,05	6	0,3	8	0,4
Performance	0,20	4	0,8	10	2,0
Mantenimiento	0,05	6	0,3	2	0,1
Propiedad	0,05	4	0,2	6	0,3
Costos de Operac./Mant.	0,05	10	0,5	8	0,4
Costos de Compra/Insta.	0,05	8	0,4	8	0,4
	-----		----		----
Puntaje total c/peso	1,00		6,2		6,9

Nota: El “Puntaje total c/peso” de cada alternativa debe ser un valor dentro del rango definido para la escala.

Debido a que no todos los criterios presentados pueden ser representativos, es importante establecer solamente un mínimo de criterios que respondan a necesidades reales, a través de una “pre-ponderación” de los mismos en función al nivel de necesidad de la empresa. Esta pre-ponderación permitirá eliminar algunos de los criterios que carezcan de representatividad para el escenario particular, disminuyendo de esta manera la complejidad del proceso de evaluación.

Bajo esta idea, la tabla 3.3 debería responder a un número de Criterios menor que los presentados, dejando de lados aquellos que posean una valoración por debajo de un NIVEL DE CORTE preestablecido con anterioridad.

Finalizada esta primera selección, es posible pasar a un segundo proceso llamado de Evaluación Técnica, solo en el caso en que los valores obtenidos por oferentes se encuentren muy cercanos entre sí. En este punto se establece una nueva lista de componentes, elementos, características y subcaracterísticas, referidos a aspectos técnicos del paquete a seleccionar.

La finalidad de establecer una nueva evaluación técnica sobre el conjunto de oferentes, refleja la premisa de que “debido a que los paquetes presentados cubren en similar medida todos los criterios de evaluación establecidos para la evaluación planteada a través de los Criterios de Selección, se pasa ahora a una nuevo análisis para detectar cuál de ellos es el más óptimo, tomando como referencia sus capacidades técnicas”.

A continuación se explica una técnica para evaluar las ofertas, denominada “Método de Evaluación Manual”.

### **2.4.3 Método de Evaluación Manual .**

El **Método de Evaluación Manual (MEM)** se aplica en forma común para evaluar **técnicamente** ofertas de HW y de paquetes de SW.

Los elementos que se deben generar a fin de poder llegar a materializar el sistema de la aplicación del método de evaluación manual son:

- 1- Lista de Requerimientos (definitiva)
- 2- Tabla de Ponderación
- 3- Cuestionario
- 4- Planilla Comparativa
- 5- Planilla de Evaluación
- 6- Planilla de Resultado
- 7- Informe Final

Estos siete elementos enunciados son los que se van generando a medida que se va avanzando en el método de evaluación manual. Cada uno de dichos pasos tiene una justificación. A continuación se explica el MEM para SW. En la sección siguiente (Adquisición de HW) se explica el MEM para HW.

#### **1. Lista de Requerimientos**

Una vez realizada la detección de las necesidades, y la determinación de los requerimientos específicos (parámetros cualitativos y cuantitativos) se hace necesario formalizar en cuanto a la elaboración de una lista que refleje el conjunto de dispositivos de hardware y software necesarios para la implementación. Si el canal de obtención de SW es la adquisición de un paquete estándar, debido a la gran cantidad y diversidad de ofertas que pueden llegar a presentarse en el proceso de adquisición, se hace necesario establecer una estructura que permita agrupar los requisitos del SW en distintos niveles de la siguiente manera:

##### **1 – Componente 1 – SW....**

- 1.1 – Facilidad 1
- 1.2 – Facilidad 2
- 1.2.1 – Característica 1
- 1.2.2 – Característica 2
- 1.2.2.1 – Subcaracterística 1
- 1.2.2.2 – Subcaracterística 2
- 1.2.2.3 – Subcaracterística 3

##### **2 – Componente 2 – HW**

- 2.1 – Elemento 1
- 2.2.1 – Característica 1
- 2.2.1.1 – Subcaracterística 1
- 2.2.1.2 – Subcaracterística 2
- 2.2.1.3 – Subcaracterística 3
- 2.2.2 – Característica 2
- 2.2 – Elemento 2

.....

#### **2. Tabla de Ponderación**

Una vez determinados los parámetros se debe efectuar la respectiva ponderación de los mismos. Sobre la base de la lista de requerimientos, se debe determinar la importancia relativa que tiene cada requerimiento con respecto a los demás, y en función de esta diferencia se los pondera. La ponderación entonces deberá ser establecida solo al último nivel de la lista de requerimiento, sea esta una característica o subcaracterística. La suma de los pesos ponderados de cada nivel debe ser igual al total de pesos repartidos (10, 100, etc.).

El peso de cada facilidad (o requisito del SW) permitirá obtener en el paso final del proceso, una planilla de resultados que surge de la sumatoria de los productos integrados de cada nivel, junto a la calificación asignada a cada propuesta de los oferentes.

El envío de la tabla de ponderación al proveedor, le permite esmerarse en la definición del paquete a ofertar teniendo en cuenta las facilidades de software que tengan mayor peso. Otra ventaja es que, como todas las ofertas van a estar lo mejor ajustadas a necesidades establecidas, se podrá seleccionar la más conveniente. En algunos casos, el desconocimiento de los pesos por parte de los oferentes, puede provocar que el proceso de obtención de SW por adquisición quede reducida a un número muy pequeño de oferentes, debido a que los demás no supieron formular sus propuestas por desconocimiento de los niveles de importancia de cada facilidad solicitada por la organización.

Es necesario efectuar el proceso de ponderación para poder obtener un resultado que no induzca a errores en la elección de la oferta, ya que el resultado final de la evaluación de cada propuesta recibida tendrá integrada no sólo el grado y calidad de cumplimiento, sino también la importancia relativa que posee dicha oferta respecto a la característica evaluada.



### 3. Cuestionario

Consiste en una planilla que contiene en su primer columna la descripción de cada requisito de SW solicitado en la lista de requerimientos,. En esta, se procede a asentar a cada característica y subcaracterística de la lista de requerimientos indicando las unidades y/o valores posibles para cada característica y subcaracterística? Por ej.: “Mbps” o “Sí/No” o “GHz”, etc. En una segunda columna el nivel de obligatoriedad de cada característica y subcaracterística. Los posibles niveles de obligatoriedad son:

- a) sistema de prioridades Obligatorio (O), Altamente Deseable (AD), Deseables (D)
- b) sistema de prioridades Imprescindibles (I), Recomendables (R), Convenientes (C)

Cada una de ellas con una connotación distinta dentro del proceso de evaluación.

La confección de Cuestionarios obliga a que el proveedor plantee su oferta lo mas objetiva y acotada posible, en función a las necesidades planteadas. De manera contraria es muy posible que se requiera de un tiempo adicional para HOMOGENEIZAR todas las respuestas de cada proveedor en lo que se refiera a una característica o subcaracterística solicitada.

### 4. Planilla Comparativa

Sobre la base del cuestionario, y luego de haber recibido las respuestas de los oferentes (ofertas), se elabora una planilla comparativa por cada oferta y para cada facilidad. Es decir, la primera columna es la lista de requerimientos, la segunda el cuestionario, y cada una de las columnas siguientes representan a las ofertas. Esta planilla se completa volcando lo que cada oferta propone para cada característica y subcaracterística. La utilidad de esta planilla es sumamente importante pues se usa para descartar aquellas ofertas que no cumplen con los requisitos fundamentales, y evitar que prosigan en el etapa de evaluación.

### 5. Planilla de evaluación

Se debe confeccionar una planilla de evaluación por cada facilidad de SW que se identifique en la lista de requerimientos que se confecciona. Es necesario confeccionar esta Planillas de Evaluación tratando de lograr OBJETIVIDAD en las calificaciones.

La calificación de componentes y facilidades de carácter complejo que no admitan una calificación directa o la falta de capacidad para evaluar la incidencia de cada una de las partes respecto al todo, obliga a desagregar el todo en partes y analizar si se las puede calificar en forma directa, muchas veces por la complejidad de los componentes.

Cada componente o facilidad no puede ser calificados en forma directa. Para que la calificación que reciban tenga cierta objetividad, se deben dividir a estos en partes que sí se puedan calificar en forma directa y con objetividad.

Luego de calificar lo mas objetivamente a cada característica (en función al Cuestionario contestado por el Proveedor), será necesario aplicar un algoritmo que pudiera integrar dichas calificaciones respecto al peso de cada dispositivo.

**Valor Ponderado:** Es el valor obtenido de multiplicar el peso de una subcaracterística por el peso de la característica respectiva. Si no hay subcaracterísticas, el valor ponderado es igual al cuadrado del peso de la característica.

**Calificación:** Nota asignada a una característica /subcaracterística, en función al análisis de las ofertas del Cuestionario.

**Calificación Ponderada:** Valor que surge de multiplicar la nota de una subcaracterística por su correspondiente valor ponderado.

La diferencia entre multiplicar la calificación de una subcaracterística por su valor ponderado y multiplicar la calificación por el peso sólo de la subcaracterística es que en la primera se tendrá integrada el nivel de importancia TOTAL de la subcaracterística, mientras que en la segunda solo identificará la calificación ponderada de la subcaracterística sin tener en cuenta a la ponderación de los niveles de componentes, elementos, y características.

Por otra parte, la calificación ponderada a nivel de característica será:

- Si la característica se encuentra desagregada en subcaracterísticas, es la suma de las calificaciones ponderadas de las subcaracterísticas, o bien
- Si la característica no se encuentra desagregada en subcaracterísticas, es el producto del valor ponderado de la característica por la nota respectiva.

## 6. Planilla de Resultados

La planilla de resultados deberá reflejar la integración de todas las planillas de evaluación, al nivel de facilidad. Se puede confeccionar una sola que integre a toda la lista de requerimientos ya evaluada por oferta, o bien considerar una por cada facilidad. El que tenga mejor puntaje, será la oferta más viable.

Sumando en la planilla de evaluación el puntaje de cada facilidad, se obtiene un subtotal. Estos subtotales se suman en otra planilla obteniéndose el resultado final.

Cuando se agotan todos las facilidades de SW se hace un subtotal para ese componente. Es decir, se tendrá una planilla de resultado que integre los resultados parciales de las facilidades del componente SW.

En un proceso completo, donde se evalúan HW, SW y HW y SW de comunicaciones, se procede a realizar esta integración en la planilla de resultados de la forma descripta para todos los componentes definidos. Por ultimo se obtiene el total final. Puede ser físicamente una sola planilla o se pueden tener tres planillas, una por componente, y luego realizar una planilla de resultados finales en la que cada componente participa con su peso.

## 7. Informe Final

Permite reflejar la recomendación técnica final que surja del análisis de todo el proceso de evaluación y sus correspondientes resultados, ordenándolos por Puntaje final obtenido y el nombre del proveedor. A esto último se lo conoce como Orden de Mérito.

Es importante su confección ya que seguramente estos resultados serán tomados por otros equipos que continuarán aplicando evaluaciones orientadas a otros aspectos, como por ejemplo el Económico.

Algunos equipos de evaluación técnica, suelen establecer antes del proceso un valor denominado nivel de corte, que permita excluir a aquellos proveedores que hubieran obtenido puntuaciones finales por debajo de esta franja. Si bien es ventajoso poseer un indicador que permita excluir posibles ofertas por debajo de los niveles deseados, el nivel de corte es muy difícil de predeterminar, sobre todo considerando que debe ser establecido a priori del proceso, de manera que los oferentes lo conozcan, y esto haga mas transparente a toda la operatoria. Si el nivel de corte estuviera mal establecido, se podría correr el riesgo de quedarse sin ninguna oferta, o bien de que todas estén por arriba de esta franja.

Se presenta a continuación un ejemplo de MEM aplicado a una facilidad de SW como es el Administrador de Base de Datos.

### a) PONDERACION DE LOS CRITERIOS DE SELECCIÓN.

<i>Criterios</i>	<i>Peso</i>
Funcionalidad	25
Capacidad	20
Confiabilidad	25
Uso	10
Seguridad	20

### b) REDISTRIBUCION DE LOS PESOS PONDERADOS.

Consideramos un punto de corte en el 90%, quedando así los criterios más relevantes a nuestro entender que son: Funcionalidad, Capacidad, Confiabilidad y Seguridad. Redistribuyendo los pesos, obtenemos los siguientes valores:

<i>Criterios</i>	<i>Peso</i>
Funcionalidad	27,7
Capacidad	22,2
Confiabilidad	27,7
Seguridad	22,2

**c)-EVALUACION DE LOS CRITERIOS DE SELECCIÓN.**

<i><b>Criterios</b></i>	<i><b>Peso</b></i>	<i><b>Oferente A</b></i>		<i><b>Oferente B</b></i>		<i><b>Oferente C</b></i>	
		Cal.	Cal.x Peso	Cal.	Cal.x Peso	Cal.	Cal.x Peso
Funcionalidad	<b>27,7</b>	3	83.1	2	55.4	2	55.4
Capacidad	<b>22,2</b>	2	44.4	1	22.2	2	44.4
Confiabilidad	<b>27,7</b>	3	83.1	2	55.4	2	54.4
Seguridad	<b>22,2</b>	2	44.4	2	44.4	3	66.6
<b>Total:</b>			<b>255 = 2.55</b>		<b>177.4 = 1.77</b>		<b>221.8 = 2.21</b>

Debido a que dos (las ofertas A y C) de los tres paquetes presentados cubren en similar medida todos los criterios de selección (han obtenido un puntaje superior a 2), se pasará a un nuevo análisis para detectar cual de ellos es el más óptimo, tomando como referencia las capacidades técnicas y los requerimientos presentados por el usuario.

**d)- EVALUACIÓN TÉCNICA**

**d1. LISTA DE REQUERIMIENTOS.+ TABLA DE PONDERACIÓN + CUESTIONARIO**

<b><u>2.- SOFTWARE</u></b>	Prioridad	Ponderación	
<b>2.1 ADMINISTRADOR DE B.D.</b>		Peso Característica	Peso Subcaracterística
2.1.1 Cliente Servidor	I	8	
2.1.2 Cache SQL compartido	I	8	
2.1.3 Backup en línea	R	7	
2.1.4 Recuperación en línea	C	6	
2.1.5 Consultas en paralelo	I	8	
2.1.6 Múltiples tipos de índices (árboles B, hash)	I	8	
2.1.7 Capacidad		10	
2.1.7.1 Cantidad de tablas >= 1000	I		4
2.1.7.2 Cantidad de registros por tabla >= 10000	I		6
2.1.8 Conectividad ODBC	I	9	9
2.1.9 Seguridad		9	
2.1.9.1 Soporte para passwords	I		9
2.1.10 Documentación		10	
2.1.10.1 Impresa	I		6
2.1.10.2 En línea	R		4
2.1.11 Lenguajes de programación soportados		8	
2.1.11.1 C	C		2
2.1.11.2 C++	I		3
2.1.11.3 Java	R		1
2.1.11.4 Perl	C		2
2.1.12 Sistemas operativos soportados		9	
2.1.12.1 Windows XP	I		3
2.1.12.2 OS/2	I		3
2.1.12.3 Linux	I		3

**d2. PLANILLA COMPARATIVA.**

<b>2.- SOFTWARE</b>	<b>Prioridad</b>	<b>OF. A</b>	<b>OF.C</b>
<b>2.1 ADMINISTRADOR DE B.D.</b>			
2.1.1 Cliente Servidor	I	Sí	Sí
2.1.2 Cache SQL compartido	I	Sí	Sí
2.1.3 Backup en línea	R	Sí	Sí
2.1.4 Recuperación en línea	C	NO	Sí
2.1.5 Consultas en paralelo	I	Sí	Sí
2.1.6 Múltiples tipos de índices	I	Sí	Sí
2.1.7 Capacidad			
2.1.7.1 Cantidad de tablas >= X	I	Sí	Sí
2.1.7.2 Cant. de registros por tabla >= Y	I	Sí	Sí
2.1.8 Conectividad ODBC	I	Sí	Sí
2.1.9 Seguridad			
2.1.9.1 Soporte para passwords	I	Sí	SÍ
2.1.10 Documentación			
2.1.10.1 Impresa	I	SÍ	SÍ
2.1.10.2 En línea	R	SÍ	NO
2.1.11 Lenguajes de programación soportados			
2.1.11.1 C	C	SÍ	SÍ
2.1.11.2 C++	I	SÍ	SÍ
2.1.11.3 Java	R	SÍ	NO
2.1.11.4 Perl	C	NO	Sí
2.1.12 Sistemas operativos soportados			
2.1.12.1 Windows XP	I	Sí	Sí
2.1.12.2 OS/2	I	Sí	Sí
2.1.12.3 Linux	I	Sí	Sí

**En caso de que alguna de las ofertas no cumpliera con tan sólo una (1) condición imprescindible (o bien, obligatoria) debe ser eliminada.**

**d3.1 PLANILLA DE EVALUACION PARA EL OFERENTE A.**

**2.- SOFTWARE**

**2.1 ADMINISTRADOR DE B.D.**

- 2.1.1 Cliente Servidor
- 2.1.2 Cache SQL compartido
- 2.1.3 Backup en línea
- 2.1.4 Recuperación en línea
- 2.1.5 Consultas en paralelo
- 2.1.6 Múltiples tipos de índices
- 2.1.7 Capacidad
- 2.1.7.1 Cantidad de tablas  $\geq X$
- 2.1.7.2 Cant. de registros por tabla  $\geq Y$
- 2.1.8 Conectividad ODBC
- 2.1.9 Seguridad
  - 2.1.9.1 Soporte para passwords
- 2.1.10 Documentación
- 2.1.10.1 Impresa
- 2.1.10.2 En línea
- 2.1.11 Lenguajes de programación soportados
- 2.1.11.1 C
- 2.1.11.2 C++
- 2.1.11.3 Java
- 2.1.11.4 Perl
- 2.1.12 Sistemas operativos soportados
  - 2.1.12.1 Windows XP
  - 2.1.12.2 OS/2
  - 2.1.12.3 Linux

PONDERACIÓN			OF. A		
PC	PS	VP (PC*PS)	CAL	CPS	CPC
8	8	64	3		192
8	8	64	3		192
7	7	49	2		98
6	6	36	1		36
8	8	64	3		192
8	8	64	3		192
10					300
	4	40	3	120	
	6	60	3	180	
9	9	81	3		243
9					243
	9	81	3	243	
10					180
	6	60	3	180	
	4	40	0	0	
8					120
	2	16	2	32	
	3	24	3	72	
	1	8	0	0	
	2	16	1	16	
9					243
	3	27	3	81	
	3	27	3	81	
	3	27	3	81	
100	100				2,231

**d3.2 PLANILLA DE EVALUACION PARA EL OFERENTE C.**

**2.- SOFTWARE**

**2.1 ADMINISTRADOR DE B.D.**

- 2.1.1 Cliente Servidor
- 2.1.2 Cache SQL compartido
- 2.1.3 Backup en línea
- 2.1.4 Recuperación en línea
- 2.1.5 Consultas en paralelo
- 2.1.6 Múltiples tipos de índices
- 2.1.7 Capacidad
  - 2.1.7.1 Cantidad de tablas  $\geq X$
  - 2.1.7.2 Cant. de registros por tabla  $\geq Y$
- 2.1.8 Conectividad ODBC
- 2.1.9 Seguridad
  - 2.1.9.1 Soporte para passwords
- 2.1.10 Documentación
  - 2.1.10.1 Impresa
  - 2.1.10.2 En línea
- 2.1.11 Lenguajes de programación soportados
  - 2.1.11.1 C
  - 2.1.11.2 C++
  - 2.1.11.3 Java
  - 2.1.11.4 Perl
- 2.1.12 Sistemas operativos soportados
  - 2.1.12.1 Windows XP
  - 2.1.12.2 OS/2
  - 2.1.12.3 Linux

PONDERACIÓN			OF. C		
PC	PS	VP (PC*PS)	CAL	CPS	CPC
8	8	64	3		192
8	8	64	3		192
7	7	49	2		98
6	6	36	3		108
8	8	64	3		192
8	8	64	3		192
10					300
	4	40	3	120	
	6	60	3	180	
9	9	81	3		243
9					243
	9	81	3	243	
10					260
	6	60	3	180	
	4	40	2	80	
8					176
	2	16	2	32	
	3	24	3	72	
	1	8	3	24	
	2	16	3	48	
9					243
	3	27	3	81	
	3	27	3	81	
	3	27	3	81	
100	100				2,439

**d4. PLANILLA DE RESULTADOS.**

	<b><u>PESO</u></b>	<b>OF. A</b>	<b>OF. C</b>
		<b>Cal pond/peso</b>	<b>Cal pond/peso</b>
<b>2.1 ADMINISTRADOR DE B.D.</b>	100	2,231	2,439

**e) INFORME TECNICO.**

a) El orden de mérito obtenido es:

1) Oferta A: 2,439

2) Oferta C: 2.231

b) Recomendaciones (\*):

...

---

\* Punto a desarrollar en clase