



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

---

FACULTAD DE ESTUDIOS SUPERIORES  
A C A T L A N

"REDES DE MERCADEO DIRECTO: IMPLEMENTANDO UN  
MODELO N-ARIO DE ÁRBOLES"

T E S I S

QUE PARA OBTENER EL TÍTULO DE  
LICENCIADO EN MATEMÁTICAS APLICADAS Y  
COMPUTACIÓN

P R E S E N T A

**JOSÉ PABLO HORACIO TEJEDA ESPINOSA**

DIRECTOR DE TESIS  
M. en C. Sara Camacho Cancino



NAUCALPAN, EDO DE MEXICO

OCTUBRE, 2016



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

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

## CONTENIDO

<b>Introducción</b>	4
<b>Capítulo 1: Redes de mercadeo directo: un modelo n-ario de árboles</b>	8
1.1 <i>¿Qué es una red de mercadeo directo?</i>	8
1.2 <i>Principales definiciones</i>	10
1.3 <i>Estructura de una red de mercadeo directo</i>	12
1.4 <i>Elementos de una red de mercadeo directo</i>	15
1.5 <i>Tipos más comunes de redes de mercadeo</i>	23
1.6 <i>Problemática en las redes de mercadeo directo</i>	24
<b>Capítulo 2: Un caso de una organización n-aria</b>	26
2.1 <i>ECOMEX: Su estructura comercial</i>	26
2.2 <i>Conceptos previos</i>	27
2.3 <i>Proceso mensual de incidencias</i>	28
2.3.1 <i>Registro de la plantilla de miembros (ensamblado de la red)</i>	29
2.3.2 <i>Registro de las compras realizadas (VP, VG y VTG)</i>	30
2.3.3 <i>Cálculo de bonos y comisiones (recorrido de la red)</i>	31
2.4 <i>Niveles y posiciones (plan de pagos)</i>	33
2.4.2 <i>Comisiones del promotor</i>	34
2.4.3 <i>Comisiones del supervisor</i>	36
2.4.4 <i>Comisiones del gerente</i>	37
2.4.5 <i>Comisiones del director</i>	39
2.4.6 <i>Comisiones del ejecutivo</i>	40
2.5 <i>El conteo de generaciones.</i>	41
2.6 <i>Roll-up en miembros no calificados</i>	43
<b>Capítulo 3: Teoría de la estructura de árboles n-arios</b>	47
3.1 <i>Teoría de gráficas</i>	47
3.1.1 <i>Conceptos y definiciones</i>	47
3.1.2 <i>Arco dirigido</i>	48
3.1.3 <i>Gráfica dirigida</i>	49
3.1.4 <i>Grado externo de un nodo</i>	49
3.1.5 <i>Grado interno de un nodo</i>	49
3.1.6 <i>Arco no dirigido</i>	50
3.1.7 <i>Lazo o loop</i>	50
3.1.8 <i>Gráfica no dirigida</i>	51
3.1.9 <i>Gráfica mixta</i>	51
3.1.10 <i>Arcos pesados</i>	52

3.2.	<i>Árboles</i>	53
3.2.1.	<i>Conceptos y definiciones</i>	53
3.2.2.	<i>Árboles ordenados, orientados y libres</i>	55
3.2.3.	<i>Representación de árboles en la computadora</i>	56
3.2.4.	<i>Recorrido en árboles binarios</i>	56
3.3.	<i>Propuesta de operación para los árboles n-arios</i>	59
3.3.1.	<i>Algoritmos de recorrido para árboles n-arios</i>	59
3.3.2.	<i>Representación de un árbol n-ario en la computadora</i>	61
3.3.3.	<i>Recorrido en un árbol n-ario</i>	62
3.3.4.	<i>Sugerencias de implementación</i>	65
	<b>Capítulo 4: Implementación del modelo de arbol n-ario</b>	66
4.1.	<i>Utilización del programa delphi (borland) para ejemplificar la implementación del modelo.</i>	66
4.2.	<i>Registro de la plantilla de miembros (ensamblado de la red)</i>	68
4.2.2.	<i>Clave de patrocinador.</i>	70
4.2.3.	<i>¿Quién es el "nodo raíz"?</i>	70
4.2.4.	<i>Cómo evitar los nodos "huérfanos"</i>	70
4.2.4.1.	<i>Ejemplo de procedimiento para adoptar descendencias de nodos eliminados.</i>	72
4.3.	<i>Registro de las compras realizadas (obtención del VP)</i>	73
4.4.	<i>Cálculo de bonos y comisiones (recorrido de la red)</i>	75
4.4.1.	<i>Implementación del algoritmo de recorrido del árbol.</i>	75
4.4.1.1.	<i>Elementos básicos utilizados durante el proceso de recorrido</i>	77
4.4.1.2.	<i>Programación no recursiva para el recorrido del árbol</i>	78
4.4.1.3.	<i>Cálculo del volumen de grupo (VG) y volumen total de grupo (VTG)</i>	82
4.4.1.4.	<i>Cálculo de comisiones del promotor (descuentos y diferencias por descuento).</i>	83
4.4.1.5.	<i>Cálculo de comisiones del supervisor.</i>	86
4.4.1.6.	<i>Cálculo de comisiones del gerente.</i>	88
4.4.1.7.	<i>Cálculo de comisiones del director.</i>	92
4.4.1.8.	<i>Cálculo de comisiones del ejecutivo.</i>	93
4.4.1.9.	<i>El conteo de generaciones y roll-up en miembros no calificados.</i>	96
	<b>Conclusiones y recomendaciones</b>	97
	<b>Bibliografía</b>	100

## Introducción

En las últimas décadas han surgido cierto tipo de empresas que se distinguen de las tradicionales por su forma de comercializar sus productos. Su fundación ha tenido origen con aquel sueño de las amas de casa de emprender un negocio que les permitiera apoyar en la economía familiar sin descuidar sus ocupaciones cotidianas. Lo que en un principio surgió como un negocio complementario familiar, se ha convertido pese al escepticismo de muchos, en un poderoso mercado económico, que no solamente contribuye a la economía familiar, sino que además es una fuente generadora de empleos y de captación fiscal.

Estas empresas son las Redes de Mercadeo Directo, Empresas Multinivel (MLM), Net Marketing, etc. las cuales están cambiando gradualmente la manera de hacer negocios y generar riqueza.

Tradicionalmente, existen diversas maneras en que las personas generan riqueza; las más comunes son:

- a) siendo asalariados
- b) por medio de un negocio propio
- c) comisión sobre ventas de productos

Sin embargo, las crisis mundiales recurrentes, el desempleo, la falta de oportunidades en países tercermundistas, etc. causan que muchos de ellos busquen otra manera de generar riqueza. Las redes de mercadeo directo les han brindado esa opción. No es de extrañar que aún en países desarrollados como Estados Unidos, los “ratings” de ventas en este tipo de empresas sean obtenidos por latinoamericanos o inmigrantes de un país tercermundista.

Este fenómeno se debe principalmente a que estas personas por lo general tienen la necesidad de:

- a) generar mayor riqueza en beneficio de ellos y de los suyos
- b) sentirse más partícipes en el crecimiento de una empresa y que esto reditúe en su economía y status social
- c) fundar un negocio lucrativo con pocos recursos y a corto plazo en donde sientan que ellos son dueños de su propio trabajo

Lo que en un principio es visto como un complemento a los ingresos, poco a poco, y conforme se comprende la manera como funciona el esquema multinivel, va cobrando mayor importancia en una economía familiar de tal manera que se convierte en la primera fuente generadora de ingresos.

El crecimiento mundial de las redes de mercadeo obedece a un sencillo pero universal paradigma de ventas: la “recomendación”.

Toda red de mercadeo directo funciona bajo este sencillo principio: *cuando un producto es bueno, se recomienda consumirlo*. Además del producto, si su venta ofrece atractivos rendimientos, *se recomienda venderlo*. Tan funcional es el esquema aún si cada miembro compra para consumir sin comprar para vender. Cada compra para consumo es una venta para la empresa y como casi el 50% de su valor es utilidad, esta se distribuye entre los consumidores. Es por ello que son redes de mercadeo “directo” porque el vínculo entre el productor y el consumidor es más directo sin pasar por tantos intermediarios.

Existen redes de mercadeo directo 100% mexicanas, con productos mexicanos de la rama del calzado, perfumería, complementos alimenticios, productos naturistas, etc. Una de las de mayor éxito es Excelencia en Mercadeo, S.A. de C.V. (ECOMEX). Fundada en 1996 por el Sr. Rafael G. Rojas con recursos propios, su crecimiento ha sido casi exponencial logrando que al poco tiempo se ubicase como la red mexicana de mercadeo directo mejor estructurada.

Sin embargo, como se verá en esta tesis y por cuestiones particularmente específicas a las redes de mercadeo directo, resulta prácticamente imposible administrar las ventas de la empresa sin una herramienta efectiva de software, y lo que es peor: difícilmente los empresarios de redes de mercadeo directo comprenden que la estructura de su empresa sigue un modelo matemático completamente definido y estructurado aunque no muy entendido.

Todas las redes de mercadeo directo obedecen a un modelo matemático de árboles generales o n-arios. Por sugerencia de especialistas, la mayoría de estas empresas adoptan el caso particular de árboles binarios debido a que existe mayor divulgación literaria sobre las técnicas computacionales de operación para este tipo de árboles. Este hecho no debe restringir el modelo empresarial. Como veremos, es necesario adaptar estas técnicas computacionales para que el modelo del árbol general o n-ario sea ampliamente implementado en una empresa realmente multinivel.

Para el año en que ECOMEX se fundó (1996), no existía en México un software completo orientado a las redes de mercadeo. La inconveniencia de importarlo de Estados Unidos dio como resultado que la empresa **Ingeniería, Métodos y Sistemas Aplicados (IMESIS)** fundada por el Sr. **José Pablo Tejada Espinosa** fuera contratada para su total desarrollo.

Toda la metodología teórica y práctica es presentada en este documento. En éste se explicará la estructura organizacional de las redes de mercadeo directo y su relación con los árboles n-arios describiendo la metodología de implementación del modelo en forma automatizada. Cabe mencionar que existe poca información acerca de la relación entre las empresas de mercadeo directo con el modelo general de árbol y menos aún de la metodología utilizada para su implementación.

Por una parte, la poca literatura disponible sobre el tema y por otra el hermetismo del sector informático a revelar algoritmos de representación y recorrido de árboles generales, ya que son muy caros, hace que los actuales desarrolladores de software

“reinventen” los mecanismos utilizados para tal propósito cada vez que se desarrolla una aplicación para estas empresas.

No entra en los alcances de esta tesis exponer el software, programas fuente, lenguajes, etc. ni las técnicas de programación computacional utilizadas para su desarrollo.

Sin embargo, al finalizar su análisis, el lector podrá ser capaz de comprender el método sugerido que le permita desarrollar y programar un software con la tecnología de su elección.

En el Capítulo 1: **Redes de mercadeo directo: un modelo n-ario de árboles** se analiza el significado de las "redes de mercadeo directo", su evolución en las últimas décadas, el éxito en su permanencia y crecimiento. Se darán algunas definiciones de lo que son las redes de mercadeo directo y cómo están estructuradas a fin de comprender su funcionamiento; así como también el por qué de su problemática en su administración.

En el Capítulo 2: **Un caso de una organización n-aria** se presenta la estructura básica de organización de una red de mercadeo con el propósito de comprender el esquema de participación de bonos y compensaciones a los promotores.

Se analizará el método habitual del cálculo de los bonos y compensaciones utilizado por las Redes de Mercadeo. Se mostrará la esencia de las políticas de participación que generalmente utilizan estas empresas analizando las de Excelencia en Mercadeo S.A. de C.V. (ECOMEX) como un caso práctico.

En este capítulo no se pretende exponer a detalle las políticas definidas para el pago de bonos y compensaciones por cada red de mercadeo debido a que varían de una empresa a otra; más bien, tratamos de subrayar la metodología seguida por estas en el proceso del pago de bonos.

En el Capítulo 3: **Teoría de la estructura de árboles n-arios**; partiendo de la teoría general de gráficas, se analizará el concepto de árbol n-ario dentro de las estructuras de datos compuestas; qué elementos contiene así como sus características formales desde el punto de vista teórico.

Se mostrará de qué manera son representados en la computadora y específicamente en una base de datos. Se verán metodologías seguidas para el recorrido de árboles y algunas de las operaciones más importantes que se practican sobre ellos así como sus respectivos algoritmos. Se propondrá una adaptación a los algoritmos tradicionales de recorrido para que el recorrido en un árbol n-ario sea posible sobretodo al implementarlo en una computadora.

En el Capítulo 4: **Implementación del modelo de árbol n-ario**, se verán los mecanismos recomendados para la implementación de los modelos de árboles en las empresas con estructura multinivel. Se considerarán los aspectos básicos para que el lector pueda desarrollar un sistema automatizado en el lenguaje de su preferencia. El objeto de nuestro estudio se centra principalmente en la representación del árbol de la organización

en la computadora y los mecanismos de cálculo mediante el algoritmo de recorrido sugerido en el capítulo 3.

## Capítulo 1

### REDES DE MERCADEO DIRECTO: UN MODELO N-ARIO DE ÁRBOLES

Cuando hablamos de redes de mercadeo directo o empresas multinivel debemos separarnos un tanto del concepto tradicional de las empresas comercializadoras de productos.

Las redes de mercadeo, aunque comercializadoras también, difieren en su estructura que las compone debido a la manera como sus productos son comercializados y la forma en que las ganancias son distribuidas entre quienes realizan la labor de venta.

Aunque no existe una definición formal que las identifique, tocaremos los conceptos y definiciones más utilizados para comprender la manera como funcionan.

#### *1.1 ¿Qué es una Red de Mercadeo Directo?*

Imagínese que usted come en un restaurante un exquisito platillo. El servicio que le brindan es de muy buena calidad. El ambiente es agradable y muy higiénico. Sus precios además, son bastante aceptables. Días después, se entera de que un familiar anda buscando un buen restaurante para llevar a su familia. Casi sin que le pregunten, usted se anticipa a recomendar aquel buen restaurante donde usted estuvo hace unos días. Cuando su familiar acude a él, el dueño le pregunta cómo eligió su restaurante y éste le expresa que fue por recomendación de usted. Así que, como muestra de gratitud, el dueño le hace un descuento a usted cada vez que consuma en su restaurante y cada vez que acuda su familiar; o bien, le bonifica en efectivo un porcentaje sobre su consumo. Usted ganó porque hizo ganar al restaurantero.

Usted y el dueño acuerdan entonces que por cada persona que acuda por su recomendación, usted recibirá un bono en efectivo sobre el consumo que estos hagan. Poco a poco y con el paso del tiempo usted se vuelve “patrocinador” de una red de recomendados y un buen vendedor del restaurante aunque no trabaje directamente ahí.

Al restaurantero no le resulta nada gravoso bonificarle esos porcentajes de consumo porque el es el fabricante directo de los platillos. De hecho, dichos bonos son obtenidos de la misma ganancia que usted le brindó al restaurantero.

Este es el principio sobre el cual se basan las redes de mercadeo directo o multinivel (MLM).

Existen diversos nombres por los cuales se reconoce una empresa con esta estructura:

- a) MLM Multi Level Marketing
- b) Mercadeo Múltiple
- c) Marketing Multinivel
- d) Multilevel Marketing
- e) Venta Directa Multinivel
- f) Network
- g) Marketing
- h) Multinivel
- i) Network Marketing
- j) Direct Network Marketing

El nombre que el autor considera más apropiado es Direct Network Marketing, el cual se puede traducir como: Mercadeo Directo por Contactos o Red de Mercadeo Directo.

Durante el análisis de este documento podrá observarse que este término se apega más a las características del negocio.

Las redes de mercadeo directo tienen sus orígenes en los Estados Unidos. En 1906, una familia que fabricaba cepillos de cochera comenzó a comercializarlos ofreciéndolos de casa en casa. Con el paso del tiempo fundó la empresa The House of Fuller, la cual se transformó en The Fuller Brush Co. y que llegó a dar orígenes a Sara Lee. Esta empresa puede considerarse formalmente como la precursora del mercadeo directo porque la familia que fabricaba los cepillos era la misma quien los vendía.

Para la década de los 40's la empresa Tupperware había inventado un plástico flexible, resistente e impermeable. Aunque comenzó a comercializarlos de manera tradicional, uno de sus agentes de mayoreo ideó la manera de vender los productos en demostraciones realizadas en casas particulares por representantes.

Pero es quizás desde la década de los 30's cuando Carl Rehnborg quien funda una empresa elaboradora de suplementos alimenticios que surge lo que en verdad caracteriza a una empresa multinivel. Esta empresa, California Vitamins es la primera en incorporar programas de ventas y planes de compensación que son distribuidos en diferentes niveles en línea descendente en función a rangos jerárquicos que ocupaban sus vendedores. Además, esta empresa fue la primera en implementar un programa de capacitación a sus vendedores o distribuidores para que estos mismos reclutaran a su vez más consumidores-vendedores ganando comisiones por las ventas generadas por estos últimos. Esto permitía la creación de organizaciones comerciales independientes unas de otras porque cada una de estas desarrollaba su propio grupo de ventas.

Fue sólo hasta los años 80's cuando en México se creó por primera vez un pequeño grupo de empresas multinivel con productos 100% mexicanos. Antes de eso sí existieron empresas multinivel pero únicamente como ramificaciones de alguna empresa estadounidense.

## 1.2 Principales Definiciones

Existen diversas definiciones para una red de mercadeo directo, las más comunes son:

- “Es el movimiento de productos y servicios del fabricante al último consumidor a través de un sistema de redes independientes asociadas”. (Ken Leonard, 1994)
- “Es la creación de una red mediante la cual se venden productos o servicios. Es el proceso en que unas personas le hablan a otras de un producto o un servicio, y de llevar a estos productos o servicios directamente al consumidor”. (Averill y Corkin, 1994)
- “Forma innovadora de comercialización de productos y servicios utilizando la venta directa a través de una red de distribuidores independientes”. (Luis Monroy, 1995)

Las definiciones *b* y *c* señalan más a sus métodos de distribución que al concepto en sí; mientras que la definición *a* se presta un tanto a ambigüedad.

La definición que el autor propone es:

*Una Red de Mercadeo Directo es una organización ramificada de personas comercializadoras de un producto o servicio proporcionado directamente por el fabricante.*

En un esquema tradicional, los productos son comercializados mediante un proceso con su respectivo costo hasta llegar al consumidor final (Ver Fig. 1.1)

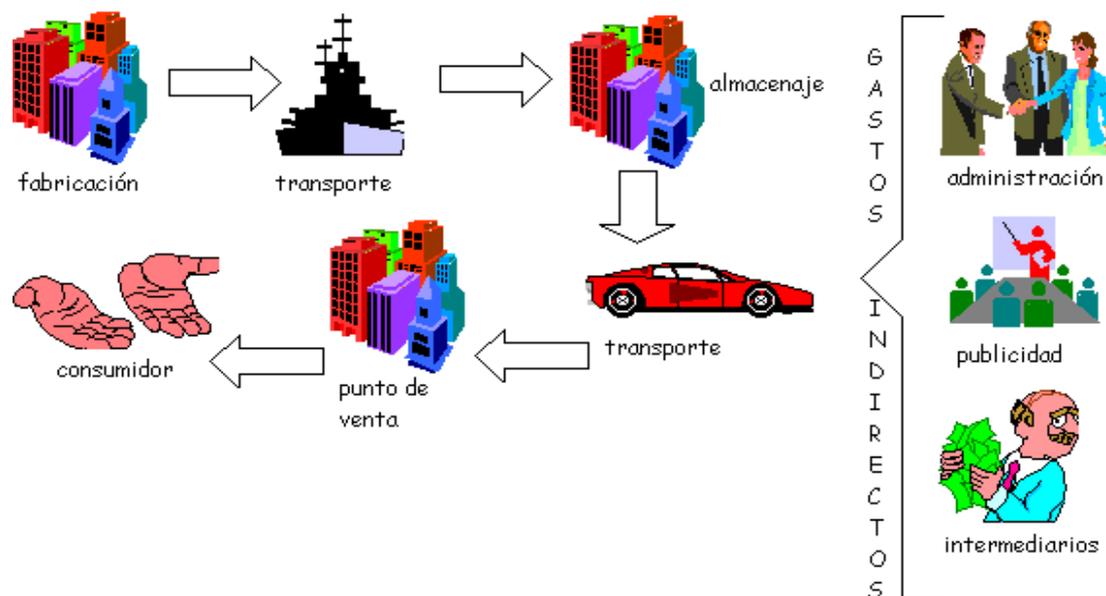


Fig. 1.1 Esquema tradicional de comercialización

En esta forma tradicional de comercialización existen diversos gastos indirectos que son reflejados en el precio final del producto. Como deja ver el señalamiento de cierto autor:

*“...los costos de manufactura han bajado tanto durante las dos décadas pasadas que la distribución actual representa el 85% del valor de los bienes en las ventas al menudeo, por lo tanto, las grandes oportunidades en la década de los 90's son para quienes reducen el costo de la distribución”* (Cisneros,1996).

Pero en el sistema de red de mercadeo el proceso de comercialización es más directo (Ver Fig. 1.2)

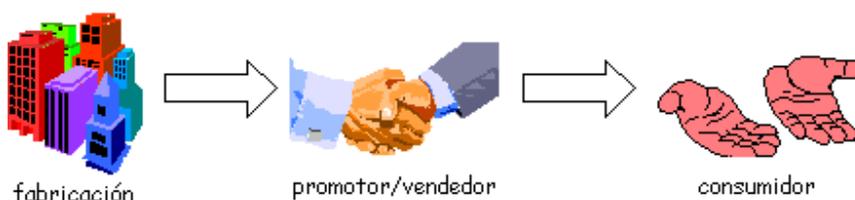


Fig. 1.2 Esquema directo de comercialización

Este método de mercadeo trae consigo diversos beneficios:

- a) no hay gasto por franquicia
- b) poco o ningún riesgo de capital por parte de los vendedores
- c) no hay limite territorial
- d) no existe el contrato por limite de tiempo
- e) reducción de costos indirectos
- f) publicidad ambulante
- g) grandes beneficios económicos para los promotores/vendedores

Cabe señalar que no todas las redes de mercadeo son directas, es decir, que los productos distribuidos al consumidor son proporcionados por el fabricante. Muchas redes de mercadeo adoptan el esquema de comercialización multinivel sin que sean organizadas directamente por el fabricante. Estas últimas son intermediarias entre el fabricante y el consumidor. Pero las más exitosas son aquellas patrocinadas directamente por el fabricante, porque esto trae consigo diversos beneficios:

- a) Los productos fabricados son de marca propia, lo cual implica que el consumidor final no encontrará el producto en ningún otro lado.
- b) Los costos de comercialización se reducen al mínimo
- c) Existe un margen de utilidad mayor, el cual es distribuido entre los vendedores de la organización.

### 1.3 Estructura de una Red de Mercadeo Directo

Se dice que es una red porque esta organización presenta una estructura de red de personas vinculadas unas con otras en el consumo y comercialización de un producto. El vínculo que une a una persona con la otra es la “recomendación” del producto o del negocio. Cuando una persona A que consume un producto X, le recomienda este producto a una persona B o le invita a participar en la comercialización de él (Ver Fig. 1.3), se dice que se crea un *vínculo de recomendación A-B*:

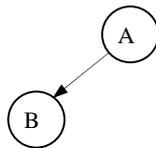


Fig. 1.3 Vínculo de recomendación A-B

Posteriormente, si la persona B hace lo mismo con la persona C (Ver Fig. 1.4) se agrega además un vínculo B-C a la organización:

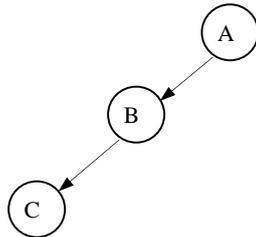


Fig. 1.4 Vínculo de recomendación B-C

Si la persona A sigue el mismo proceso para la persona D (Ver Fig. 1.5) se agrega también un vínculo A-D:

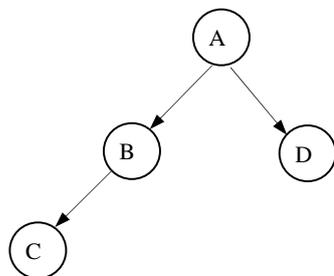


Fig. 1.5 Vínculo de recomendación A-D

Con el paso del tiempo se llega a formar una red de vínculos de personas (Ver Fig. 1.6):

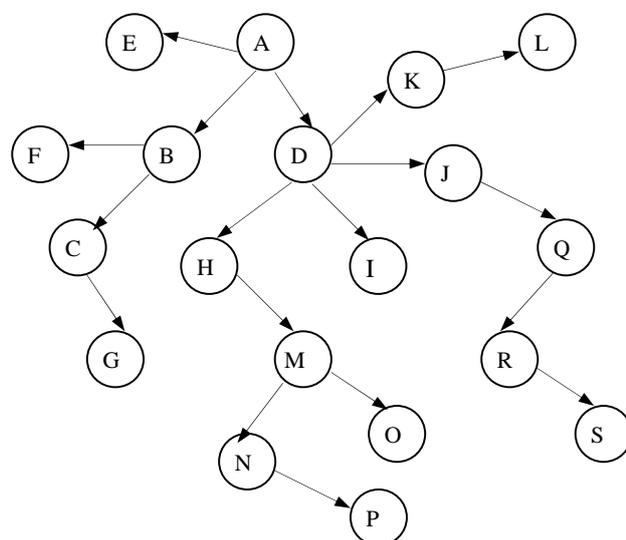


Fig. 1.6 Red de vínculos entre personas

Este esquema obedece a una estructura tipo red en donde los nodos son personas y los vínculos son las “recomendaciones”.

Es importante establecer un identificador en la gráfica para saber quién recomienda a quién. En el vínculo A-B (Fig. 1.6) note que la línea sale de A y termina en B; el identificador es la flecha al final de la línea. Con esto se entiende que A patrocina a B, es decir, que A recomendó a B. Si la flecha apuntara de B hacia A se entendería que B es patrocinador de A.

Por lo regular, este esquema ofrece al promotor la independencia en el manejo de su organización. Por ejemplo, la organización de A son todas las ramificaciones incluyendo a las personas B...S.

La organización de J son las personas Q, R, S. Con el paso del tiempo J puede crecer tanto como A o más, causando que a su vez que A y D crezcan por encontrarse en su ramificación.

Este crecimiento es independiente al crecimiento obtenido por ejemplo por K. Aunque J y K se encuentren en el mismo nivel respecto a D, uno del otro no se necesita para crecer. No obstante, D necesita de ambos para su crecimiento.

Teóricamente, una organización de este tipo trabajada a un ritmo constante, tiende a crecer exponencialmente con el paso del tiempo.

Por ejemplo, existen organizaciones que adoptan el esquema llamado *Binario*, en el cual, cada integrante se compromete a recomendar el producto e incorporar a la organización a dos personas (Ver Fig. 1.7).

Suponiendo que cada persona ingresada recomienda e incorpora a otras dos en un plazo de una semana, al término de 5 semanas tendríamos un crecimiento muy amplio:

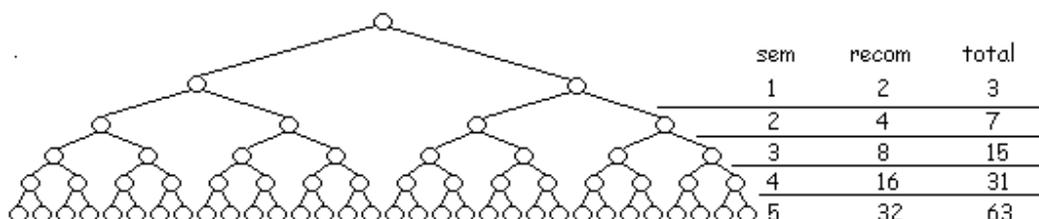


Fig. 1.7 Red binaria de comercialización

Note que en este caso las personas recién incorporadas son las próximas en incorporar más gente, de tal manera que para la semana 2, la primera persona deja de incorporar gente y el crecimiento de la organización lo realizan las 2 últimas personas incorporadas. Así, para la semana 5, las 32 personas ingresadas son las recomendadas por las que ingresaron en la semana 4; pero todas las anteriores dejan de incorporar gente.

En el verdadero esquema multinivel, como se podrá ver más adelante, no deberá existir límite de recomendados, y todos podrán seguir incorporando gente en todo momento; por lo que la organización deberá, al menos en teoría, crecer exponencialmente (Ver Fig. 1.8).

En muchas ocasiones es muy común representar las redes o sub-redes utilizando triángulos, sobre todo cuando éstas son muy grandes:

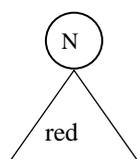


Fig. 1.8 Representación de una sub-red

En la gráfica anterior N es la persona que encabeza una organización, mientras que el triángulo red es su organización de ventas, la cual tiene la misma estructura que hemos utilizado.

En la figura 1.9 se puede observar un conjunto de sub-redes para las organizaciones de ventas:

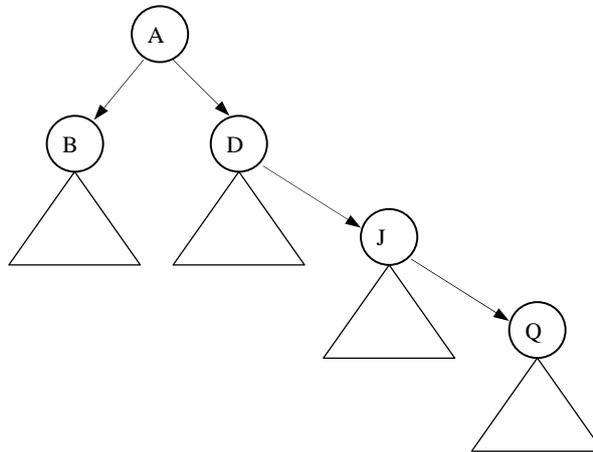


Fig. 1.9 Representación de sub-redes

Podemos notar que J y Q podrían estar incluidas dentro del triángulo conectado a D, puesto que J pertenece a D y Q pertenece a J. El hecho de separarlas es para efectos prácticos cuando se quiere mostrar que J está contenido en D y que además tiene una sub-red u organización conectada a ella; lo mismo sucede con Q.

Cada sub-red representada por un triángulo puede tener de 1 hasta m ramificaciones de personas conectadas.

#### **1.4 Elementos de una Red de Mercadeo Directo**

Toda red de mercadeo se forma por los mismos elementos y tiene la misma estructura:

- a) personas
- b) vínculos de recomendados

Los siguientes elementos son los más comúnmente utilizados por las redes de mercadeo y aunque varios autores o empresas difieren del término, en concepto significan lo mismo:

## PROMOTOR

Es la persona que promueve el consumo del producto o servicio y que además lo vende. El promotor promueve además el método del negocio con el propósito de incorporar gente a la organización.

Cada promotor ingresado es, necesariamente, un recomendado por otro miembro de la organización.

En el esquema que hemos utilizado (Fig. 1.6), A es el promotor inicial. Una vez que es incorporado B y C, estos también son promotores, porque además de comprar el producto y ser ingresados por recomendación de A, logran incorporar más gente. De esta manera, todas las personas que entran a formar parte de la organización son promotores.

TODO NEGOCIO MULTINIVEL SE BASA EN EL PRINCIPIO: “COMPRAR, CONSEGUIR QUIEN COMPRE Y HACER QUE ESTOS VENDAN”.

Bajo este principio, un promotor deberá comprar permanentemente el producto o servicio. Con esto logrará mantener activa la organización.

Cuando logra conseguir personas para que vendan el producto, contribuye al crecimiento de la empresa y de su propia organización.

Además, debe lograr que sus incorporados compren el producto para que él obtenga beneficios económicos sobre sus compras. Estos últimos a su vez, repetirán el ciclo descrito en el principio porque ellos mismos también son promotores.

## PATROCINADOR

Es la persona que, una vez recomendando el producto, servicio o negocio, logra reclutar a un promotor para su incorporación a la organización.

En el esquema utilizado (Fig. 1.10), se dice que A es patrocinador de B porque el primero recomendó al segundo. A es patrocinador de B, D y E.

Aunque H pertenece a la organización de A, no es patrocinado por A, es más bien patrocinado por D.

H es patrocinador de M únicamente y M patrocina a N y O.  
R patrocina a S.

En muchas empresas de red de mercadeo se dice también que el patrocinador es el **PATERNO** de sus patrocinados por ser éste quien logra obtener *descendencia* en la organización por medio de incorporar gente a ella. De esta manera, A es el paterno de B, D y E; mientras que estos últimos son su descendencia.

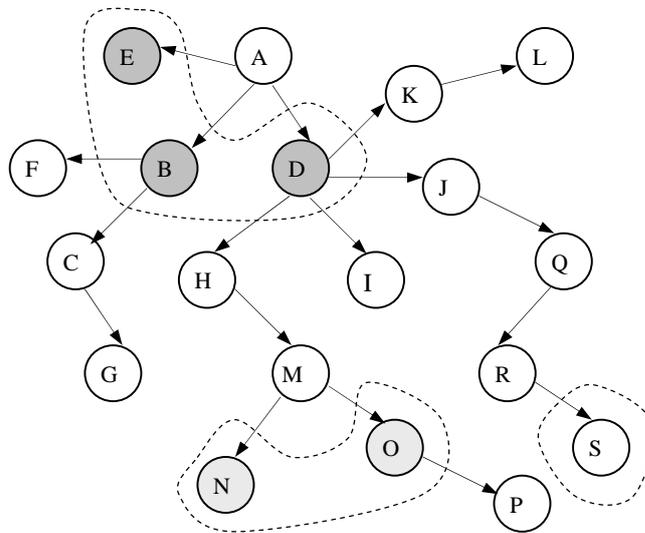


Fig. 1.10 E,B,D patrocinados por A; N,O patrocinados por M; R patrocina a S

## GRUPO

Es una organización ramificada de promotores vinculados por medio de la recomendación.

El grupo de A (Fig. 1.11) son todas las personas B...S incluyendo a A. El grupo de J es formado por J, Q, R y S. El grupo de H está formado por las personas H, M, N, O, P.

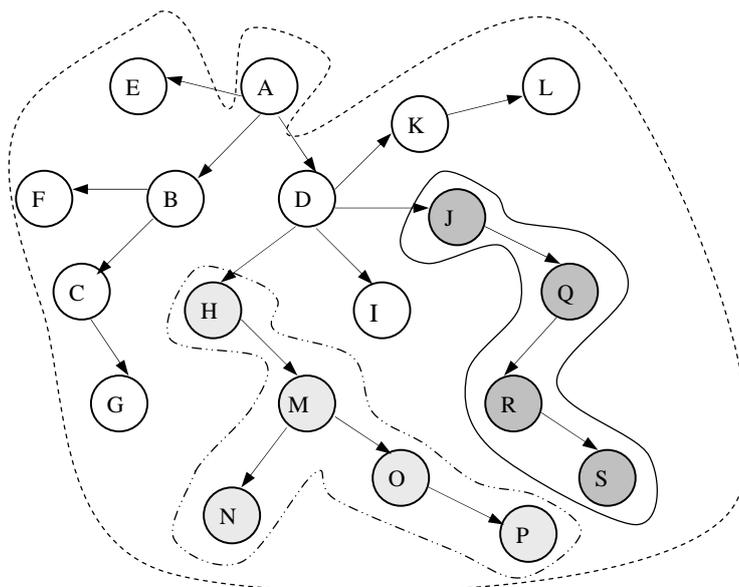


Fig. 1.11 Grupos de personas

Note que el dueño del grupo es la persona o nodo inicial de la ramificación. Algunas redes de mercadeo deciden no incluir a esta persona y forman el grupo a partir de los vínculos inmediatos a ella.

Por ejemplo, en el grupo de J consideran que está formado por Q,R y S, excluyendo a J. En realidad no existe diferencia esencial en incluir o no a la persona inicial de la ramificación. Se opta por lo general incluirla para efectos prácticos sobre todo cuando se tratan de establecer por ejemplo volúmenes de venta por grupo. En este caso casi todas incluyen las ventas realizadas por la persona inicial independientemente de si la incluyen o no en el grupo.

## LINEA FRONTAL

Es el conjunto de personas patrocinadas directamente por otra. En el siguiente ejemplo (Fig. 1.12) E, B y D forman la línea frontal de A por ser patrocinadas por A. H, I, J y K forman la línea frontal de D; mientras que F y C lo son de B.

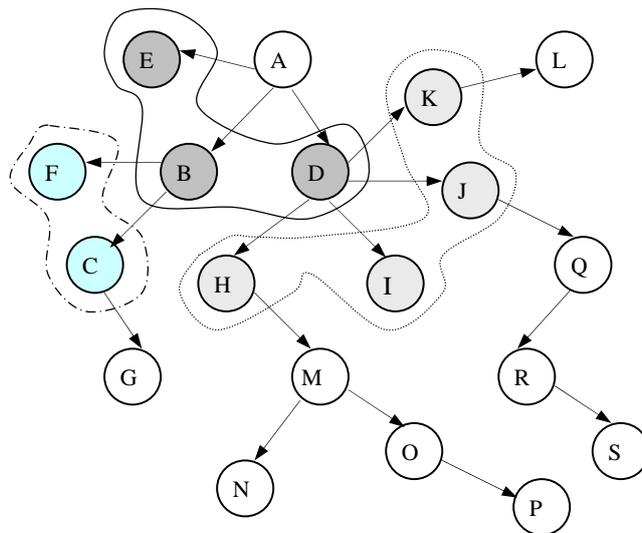


Fig. 1.12 Líneas frontales

### ASCENDENCIA (UP LINE)

Es el conjunto de personas ubicadas arriba de una que forman una trayectoria directa entre ella y otra.

La línea ascendente de H es D y A (Fig. 1.13); mientras que la línea ascendente de N está formada por M, H, D y A.

D forma parte de la línea ascendente de J pero también de Q, R, S, I, P, O, M, N, H, K y L porque se encuentra entre sus trayectorias hacia arriba.

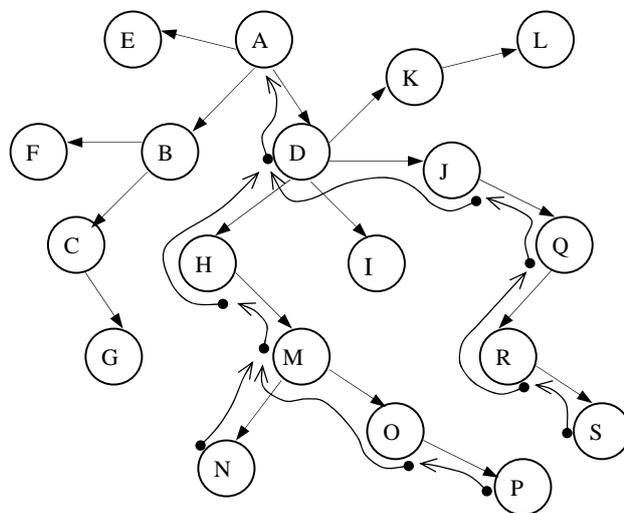


Fig. 1.13 Líneas de ascendencia

Hablando de ascendencia, es muy común formar rutas entre personas de una ramificación.

Por ejemplo:

- para llegar a la persona A desde O, hay que trazar la ruta: [M,H,D,A]
- la ascendencia de R llegando a J es: [Q,J]
- la línea ascendente de O es: [M,H,D,A]
- la ruta ascendente entre R y D es: [Q,J,D]

En muchas empresas de redes de mercadeo utilizan mucho el término **PADRE**, **ABUELO**, **HIJO**, **NIETO** para referirse al nivel cercano de ascendencia.

Por ejemplo, en la ascendencia de L, K es su padre y D es su abuelo. Al mismo tiempo, L no tiene hijos ni nietos.

Como M es padre de O, P es hijo de este último y nieto del primero. A su vez, N también es hijo de M.

Observe que cada trayectoria o ruta de ascendencia es única, es decir que partiendo de una persona  $x$  existe una sola ruta de acceso hacia la persona  $y$ .

En una estructura *Binaria*, un promotor podrá tener únicamente 2 hijos, mientras que en un esquema multinivel, podrá tener los hijos que pueda.

### DESCENDENCIA (“DOWN LINE”)

Es el conjunto de personas ubicadas debajo de una que forman una trayectoria directa entre ella y otra.

Las líneas de descendencia de H son 4: (Fig. 1.14)

1. [M]
2. [M,N]
3. [M,O]
4. [M,O,P]

J forma parte de la línea de descendencia de D, pero también de A porque se encuentra entre sus trayectorias hacia abajo.

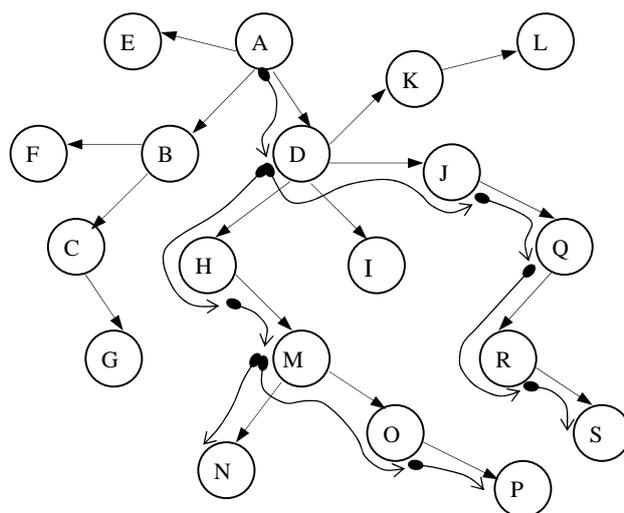


Fig. 1.14 Líneas de descendencia

Es muy común formar rutas también entre personas de una ramificación.

Por ejemplo:

- e) para llegar a la persona O desde A, hay que trazar la ruta: [D,H,M,O]
- f) la descendencia de J llegando a R es: [Q,R]
- g) la línea descendiente de M es: [N], [O] y [O,P]
- h) la ruta descendente entre D y R es: [J,Q,R]

Observe que cada persona de la organización podrá tener la descendencia que pueda (en un esquema multinivel), aunque cada una de ellas tiene una ruta única.

### HERMANOS Y PRIMOS (“CROSS LINE”)

Es el conjunto de personas que se encuentran en el mismo nivel de descendencia respecto a otra.

H, I, J y K (Fig. 1.15) son hermanos entre sí o “cross line” de D porque se encuentran en el mismo nivel de descendencia respecto a D. Son también todos ellos, primos de F y C, porque B (el padre de estos últimos) es hermano de D.

G, M, Q y L son primos por encontrarse en el mismo nivel de descendencia de D (para M, Q y L) y B (para G).

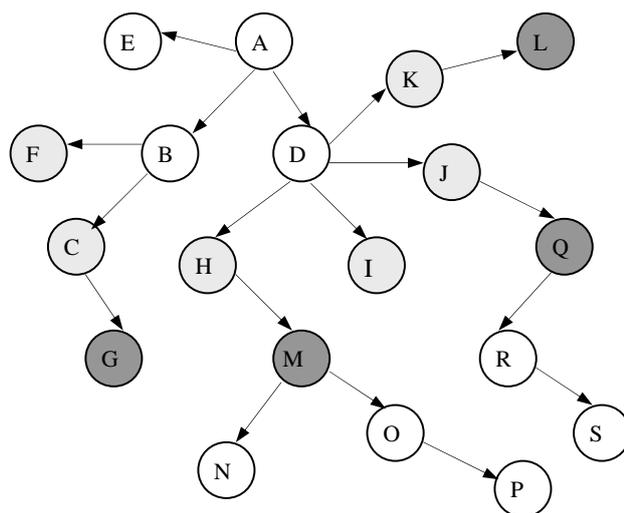


Fig. 1.15 “Cross line”

Regularmente es entre los “cross line” donde se da la única competencia en una red de mercadeo por lograr los más altos volúmenes de venta de cada una de sus organizaciones.

## GENERACIÓN

Es la posición de descendencia donde se ubica una persona que desciende de otras. Es el número de personas que forman la ruta para llegar a la persona y partiendo de la persona  $x$ .

Por ejemplo (Fig. 1.16), la línea descendente entre D y R es: [J,Q,R]. Esta ruta es formada por 3 personas incluyendo a R; por lo tanto, R forma parte de la tercera generación de D.

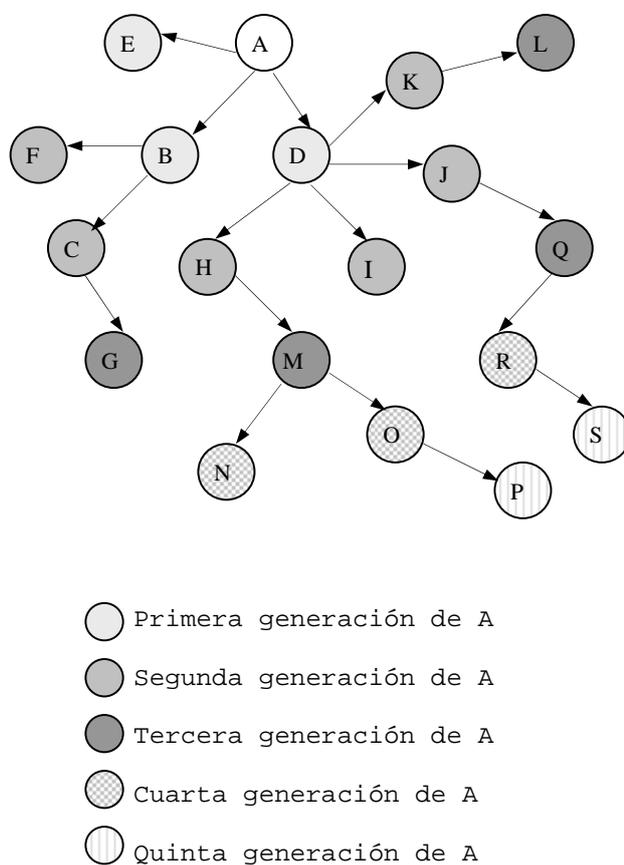


Fig. 1.16 Conteo de generaciones

Siempre que se habla de generación es necesario indicar la persona respecto a la cual se hará el conteo generacional.

Por ejemplo:

- a) O y N son primer generación de M
- b) O y N son segunda generación de H
- c) O y N son tercer generación de D
- d) O y N son cuarta generación de A
- e) Q es primer generación de J, segunda de D y tercera de A
- f) A tiene 5 generaciones

### 1.5 Tipos más comunes de redes de mercadeo

#### **MATRIX**

Es un sistema matricial con determinación previa en cuanto a la amplitud y profundidad de la red. Este tipo de empresas suelen controlar el crecimiento de la red hasta ciertos niveles buscando con ello que no se produzcan organizaciones demasiado grandes que signifiquen una migración masiva cuando una persona decide separarse de la empresa.

Estas empresas definen su crecimiento medido de  $x$  por  $y$ , donde  $x$  es el número de promotores directamente patrocinados y  $y$  es el número de niveles de profundidad que pueden alcanzar. En una empresa de  $5x10$  se tendrían 5 promotores en línea frontal por 10 generaciones hacia abajo.

#### **BINARIO**

Este sistema se define como  $2xn$ . Toda persona podrá patrocinar exclusivamente a 2 y el grado de profundidad en sus generaciones puede ser abierto. Muchas empresas utilizan este método porque suena atractivo para aquellos a quienes no les gusta reclutar gente.

Solamente tendrán que reclutar a 2 y lograr que su crecimiento sea hacia abajo. Si una persona logra reclutar a una tercera tendrá que colocarla debajo de alguna ramificación inferior.

#### **STAIRSTEP**

Es un modelo basado en el escalonamiento jerárquico. Un distribuidor escala hacia arriba los diferentes niveles en un tiempo determinado pudiendo “brincar” a su patrocinador a un nivel superior.

#### **N-ARIO**

Es un sistema definido como  $nxm$ . Toda persona es libre de patrocinar a un número  $n$  de promotores obteniendo un crecimiento generacional de  $m$  niveles indefinidos. Este es el

esquema que más reditúa a sus miembros por tener un crecimiento acorde al trabajo realizado por ellos. Muchos autores consideran este modelo como el verdadero negocio multinivel, porque sus niveles no son concebidos únicamente en el grado de profundidad generacional, sino también hacia su amplitud en la línea frontal. El modelo **BINARIO** es un subsistema del modelo **N-ARIO**. El primero es utilizado para requerir menos esfuerzo de sus miembros para hacer crecer la organización de ventas, mientras que el último es utilizado cuando se desea un crecimiento mayor de la organización.

### ***1.6 Problemática en las Redes de Mercadeo Directo***

En los esquemas tradicionales de comercialización es relativamente fácil la administración de las ventas. Sobre todo cuando se trata del pago de comisiones a los promotores y vendedores.

Sin embargo, en el esquema de las redes de mercadeo el procedimiento se complica por las siguientes razones:

- a) Además de las ventas personales, el pago de comisiones se calcula por las ventas realizadas por las descendencias de cada uno de los promotores de la organización.
- b) Existen diferentes restricciones para el conteo generacional de promotores que califican para el pago de las comisiones y bonos extra.
- c) Existen diferentes grados de calificación posibles para cada promotor, en función a su volumen de ventas por grupo; de tal manera que en una organización de ventas, un mismo promotor puede calificar para todos y cada uno de los grados (o solo para uno) dependiendo de la calificación obtenida por cada integrante de sus líneas de descendencia.
- d) En muchas redes de mercadeo se presenta el fenómeno del “roll up” o “compactación de generación”, el cual requiere de la transformación de la estructura de la red de un período a otro.

En base a lo anterior, es necesario contar con una herramienta automatizada que brinde la posibilidad de:

1. Ensamblar el árbol genealógico considerando la lógica que deberá seguir la organización (control de vínculos únicos, rutas de ascendencia y descendencia únicas, relación de recomendaciones, etc.)
2. Registrar las ventas por integrante del árbol
3. Realizar los cálculos de comisiones pertinentes en un período específico

4. Mostrar en cualquier momento las líneas de ramificación para cualquier promotor que así lo solicite.
5. Calificar automáticamente a los promotores para los siguientes grados obtenidos según la estadística de ventas.
6. Presentar los resultados de las comisiones logradas en un período específico y el detalle generacional de su integración.

Actualmente existe escasa oferta de software que considere estos requerimientos y responda a las necesidades de cada red de mercadeo en México.

En el año de 1996 la empresa Excelencia en Mercadeo, S.A. de C.V. (ECOMEX) tuvo que postergar el inicio de operaciones debido a que su organización inicial la componían más de 2000 personas y no había software disponible en México.

## Capítulo 2

### UN CASO DE UNA ORGANIZACIÓN N-ARIA

Todas las empresas que se identifican como multinivel presentan la misma estructura básica de organización como una red de mercadeo. La estructura ramificada de personas comercializadoras de sus productos es común a todas.

La diferencia esencial radica en el método de comercialización utilizado y el esquema general de participación de bonos y compensaciones a sus miembros.

Sin importar cuál sea el plan de compensaciones, todas utilizan un método habitual del cálculo de los bonos. Aunque las políticas internas las distinguen, en esencia presentan rasgos similares.

Tomando como caso práctico la empresa Excelencia en Mercadeo S.A. de C.V. (ECOMEX) expondremos su estructura de comercialización así como el plan de participación de bonos que ésta utiliza. Subrayaremos además la metodología seguida en la implementación del plan definido.

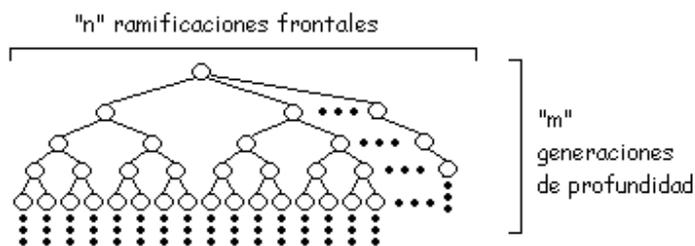
#### *2.1. ECOMEX: Su estructura comercial*

Excelencia en Mercadeo, S.A. de C.V (ECOMEX) es una red de mercadeo establecida en los años 90's. Esta empresa comercializa una gran variedad de productos naturistas. Su fundador, el Sr. Rafael García Rojas, es un empresario experto en la dirección de redes multinivel. Su crecimiento ha permitido incorporar a la organización no solamente a personas en México, sino además de los Estados Unidos, Centro y Sudamérica.

En este capítulo veremos algunas de sus características que le han permitido tener dicho crecimiento en tan poco tiempo.

Como dijimos en el **Capítulo 1**, todas las redes de mercadeo se encuentran organizadas de la misma manera. Aunque cada una de ellas utiliza sus propias políticas de participación de ganancias, su estructura es en esencia la misma.

A partir de este capítulo analizaremos la estructura general de las redes de mercadeo compuesta de  $n$  número de ramificaciones frontales por  $m$  número de generaciones de profundidad (Ver Fig. 2.1). Muchos empresarios del multinivel consideran que este tipo de estructura es el verdadero multinivel debido a que el concepto multinivel no solamente es concebido de manera vertical (grado de profundidad de la organización) sino además de manera horizontal (grado de crecimiento frontal). Se dice entonces que una red de mercadeo n-aria es de  $n \times m$  (n por m).

Fig. 2.1 Red de  $nxm$ 

ECOMEX es una red  $n$ -aria puesto que el crecimiento de todas y cada una de sus organizaciones de ventas son de un número indefinido.

## 2.2. Conceptos previos

Para un mejor entendimiento de estos conceptos diríjase al **Capítulo 1**, sección **1.4 Elementos de una Red de Mercadeo Directo**.

### Miembro

Es todo distribuidor de productos de ECOMEX que ingresa a la organización. Sin importar la posición ocupada en la empresa, todo distribuidor es un Miembro.

### Patrocinador

Es el miembro que recluta a una persona para laborar en ECOMEX.

### Posición o Nivel

Es la categoría, puesto o ubicación que obtiene un promotor en el plan de pagos.

- a) Promotor
- b) Supervisor
- c) Gerente
- d) Director
- e) Ejecutivo

El supervisor es a su vez un promotor.

El gerente es a su vez un supervisor y promotor.

El director es a su vez un gerente, supervisor y promotor.

El ejecutivo es a su vez un director, gerente, supervisor y promotor.

**Promoción**

Miembro que obtiene una posición superior en el plan de pagos al lograr los montos establecidos.

**Volumen personal de compras (VP)**

Monto total de compras realizadas en un mes a ECOMEX con su número personal individual.

**Volumen de Grupo (VG)**

Monto total de compras realizadas en un mes por un grupo de **promotores** patrocinados directamente por un Miembro.

**Volumen Total de Grupo (VTG)**

Monto total de compras realizadas por un grupo en un mes incluyendo a todos los niveles (promotores, supervisores, gerentes, directores y ejecutivos).

**Línea**

Ruta que existe entre dos miembros. Cada organización o suborganización de ventas se dice que es una línea.

**Calificar**

Un promotor es considerado calificado cuando ha cumplido con sus volúmenes de venta personales (VP) y de grupo (VG) en un mes calendario de tal manera que lo hagan acreedor a los bonos y comisiones correspondientes al nivel o posición que ocupa.

**2.3. Proceso mensual de incidencias**

El proceso mensual del cálculo de bonos y comisiones requiere básicamente de dos aspectos en la administración de la red de miembros:

Registro de miembros vinculados a la red  
 Registro de las compras realizadas por cada miembro

Por supuesto que en estos dos aspectos pueden verse envueltas varias actividades adicionales (facturación, movimientos a inventario, etc.).

Más que describir con todo detalle los métodos y procedimientos así como los formatos utilizados para el registro de operaciones en ECOMEX o en cualquier red de mercadeo, nuestro propósito es establecer los requerimientos mínimos necesarios para

solucionar el problema esencial en el cálculo de bonos y comisiones mensuales. Los métodos y procedimientos utilizados para la administración de una red de mercadeo quedan fuera del alcance de esta tesis porque son muy extensos y requieren un control muy particular y específico que varía de una empresa a otra.

En resumen, diremos que también en las redes de mercadeo se tiene la necesidad de administrar eficientemente diversas operaciones cotidianas en una comercialización de productos:

1. Control de almacenes e inventarios
2. Control de la producción (si es red de mercadeo directo)
3. Cuentas por cobrar
4. Cuentas por pagar
5. Control de bancos
6. Contabilidad

No obstante, el contar con una herramienta de software orientada a la administración de la red de distribución nos brinda muchos beneficios relacionados con los puntos anteriores sobre todo si éste se encuentra vinculado con los módulos antes referidos.

Como señalamos al principio de este tema, las preguntas planteadas en la administración de la red de miembros son:

¿Cómo establecer los vínculos entre los miembros de manera que se obtenga la red ensamblada?

¿Cómo implementar el plan de pagos en la red?

En este tema se dará respuesta a la primer pregunta, mientras que la segunda será implementada en temas posteriores en este mismo capítulo.

El proceso seguido mes con mes para obtener el resultado final de la distribución de las comisiones es muy sencillo:

1. Registrar las incidencias en la plantilla de miembros.
2. Registrar todas las órdenes de compra pagadas por cada miembro.
3. Calcular el volumen personal de compra por cada miembro.
4. Calcular el volumen de grupo (VG) y volumen total de grupo (VTG) por cada miembro.
5. Calcular bonos y comisiones del mes.
6. Emitir resultados.

### **2.3.1 Registro de la plantilla de miembros (ensamblado de la red)**

Cada vez que un miembro de la empresa recomienda el negocio a otra persona y ésta última ingresa a la empresa, deberá registrarse este movimiento. Para ello es imprescindible cierta información:

1. Número de miembro
2. Número de patrocinador

Por medio de estos dos datos podremos establecer las conexiones o vínculos que nos permitan ensamblar la red en su totalidad.

Los demás datos tales como apellido paterno, apellido materno, nombres, dirección, etc. son complementarios y aunque son muy importantes y necesarios, no los requerimos para el ensamblado de la red.

Cuando una persona ingresa a la organización se le otorga un número de identificación personal único. Pero además, es necesario vincularla con su patrocinador (Fig. 2.2), es decir, la persona quien le recomendó el negocio el cual tiene también asignado un número de identificación personal.

Para lograr esto, toda persona ingresada llena una solicitud de afiliación, la cual contiene entre otras cosas estos dos datos.

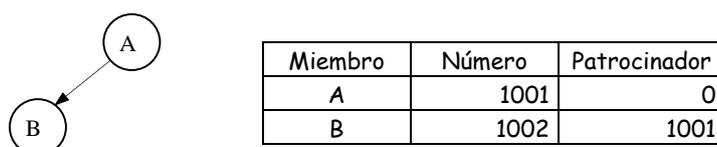


Fig. 2.2 Vínculo entre miembros

Este proceso es seguido para todo miembro que ingresa a la organización, de tal manera que al registrar a los miembros y sus recomendados se realiza el ensamblado de la red.

No hay que insistir en la consecutividad por nivel de los números de identificación personal porque, además de hacer complicada la actualización de la plantilla de miembros, es poco práctico e innecesario para el ensamblado de la red. Lo conveniente es asignar un número consecutivo cronológico conforme se ingresa a los miembros sin importar el nivel en que se sitúan en el árbol.

### 2.3.2. Registro de las compras realizadas (VP, VG y VTG)

Una vez que el miembro es ingresado a la organización podrá realizar las compras de productos a ésta indicando en toda orden su número de identificación personal asignado.

Es requisito indispensable indicar el número del miembro que realiza la compra, porque sólo de esta manera será posible obtener:

1. El volumen personal de compra (VP)
2. El volumen de grupo de los promotores que patrocina (VG)
3. El volumen total de grupo (VTG)

Las personas podrán efectuar las compras requeridas en el momento que se necesite.

Al cierre de mes (normalmente unos días antes de finalizar el mes calendario) se realiza el cálculo de las órdenes de compra para todos y cada uno de los miembros. Esto nos trae como resultado el volumen personal de compra (VP). A partir de este dato es posible calcular el volumen de grupo de sus promotores (VG) y el volumen total de grupo (VTG).

### **2.3.3. Cálculo de Bonos y Comisiones (Recorrido de la red)**

Una vez ensamblada la red y obtenido los volúmenes personales de compras (VP) se procede a realizar el cálculo del volumen de grupo (VG), volumen total de grupo (VTG), bonos y comisiones.

Recordemos que los bonos y comisiones en una Red de Mercadeo están en función a tres montos:

1. El volumen de compras personales (VP)
2. El volumen de compras realizadas por todos y cada uno de los miembros con nivel promotor que una persona patrocina directamente (VG)
3. El volumen de compras total pertenecientes a toda la organización de ventas (VTG) que cada persona patrocina (independientemente de su nivel).

Los montos comisionables de cada compra habrá que distribuirlos entre todas las ramificaciones de personas.

Para realizar esto es necesario seguir un método preciso de exploración o análisis de la red que nos permita “recorrer” los vínculos y miembros de todas las organizaciones.

Normalmente cuando se realiza esta distribución se piensa en un orden de exploración o recorrido generacional de la red para establecer los volúmenes de grupo, el conteo de generaciones, la participación de comisiones en miembros calificados, etc.

Debido a que los bonos y comisiones finales se encuentran en función a lo realizado por toda una organización de ventas es lógico realizar el cálculo de abajo hacia arriba en cada una de las ramificaciones y través de toda la red.

Siempre se deberá comenzar desde los últimos niveles hacia arriba hasta llegar al primer miembro que encabeza la organización (Fig. 2.3). Hacerlo al revés (de arriba hacia abajo) como naturalmente se ensambla la red requiere de muchos “recorridos” por cada ramificación.

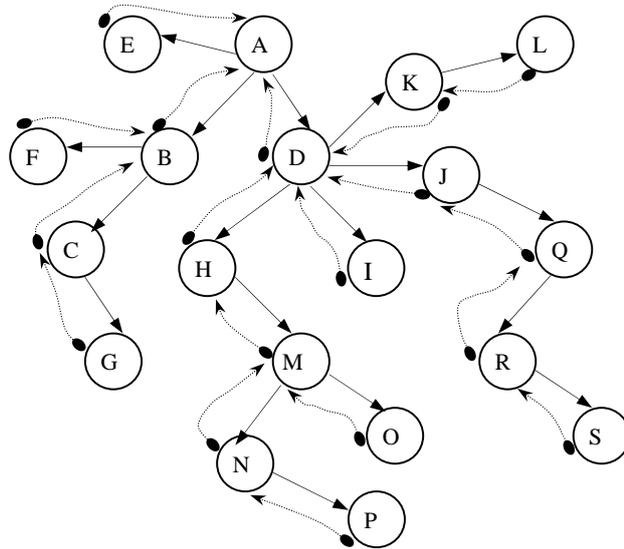


Fig. 2.3 Recorrido en la red

Este método de exploración o recorrido de la red nos brinda la manera más sencilla y rápida de realizar los cálculos pertinentes. Sin embargo, cuando la red es muy grande y los requisitos que cada empresa impone para el pago de las comisiones son varios, es muy conveniente utilizar la ayuda de un software de aplicación específico.

Al momento de realizar este cálculo es necesario comenzar dicho recorrido desde los niveles inferiores hacia arriba “aventando” la información requerida para el pago de bonos.

Comenzando por los niveles inferiores, esta información será tomada por los niveles superiores inmediatos para determinar las condiciones reunidas por estos últimos de acuerdo a las posiciones que ocupan en el plan de pagos.

Haciendo el recorrido obtendremos diversos datos, tales como:

- a) volumen personal (VP),
- b) volumen de grupo de promotores(VG),
- c) volumen total de grupo (VTG),
- d) posición o categoría,

- e) status (calificado o no calificado),
- f) acumuladores (cantidad de patrocinados frontales por posición, número de generaciones, importes pendientes por distribuir “arrastrados” desde niveles inferiores, etc.)

Toda esta información es posible transmitirla a través de la red de un miembro a otro siguiendo este método de recorrido.

#### **2.4. Niveles y Posiciones (plan de pagos)**

En ECOMEX, así como en todas las redes de mercadeo, existen diversos niveles de participación de las ganancias, los cuales son manejados como *puestos* dentro de la empresa:

- a) promotor
- b) supervisor
- c) gerente
- d) director
- e) ejecutivo

Cada puesto es obtenido en función a las ventas mensuales logradas tanto individualmente como por grupo.

Toda persona recién ingresada a la organización ocupa el puesto de promotor. A partir de ahí, puede obtener los siguientes puestos según sus estadísticas de ventas.

En otras empresas multinivel estos puestos son conocidos con diferentes nombres (perla, zafiro, rubí, etc.) aunque su propósito es el mismo.

A diferencia de las empresas con estructura tradicional, estos puestos no son jerárquicos, se utilizan más bien para establecer el logro de ventas alcanzado por cada persona. Esto implica que, por ejemplo, un gerente pueda percibir mayores ingresos que un director ejecutivo en un período determinado.

Por lo regular, en las redes de mercadeo se otorga un descuento en cada compra personal así como varios porcentajes de comisión sobre la compra realizada por su organización de ventas, los cuales varían de una empresa a otra.

Recordemos que en este tipo de empresa las comisiones son percibidas no solamente por las compras personales realizadas, sino además por las efectuadas por la organización de ventas. Si bien el crecimiento de cada organización de ventas puede ser de  $nxm$  ( $n$  ramificaciones frontales por  $m$  generaciones de profundidad), en todas las redes de mercadeo se establecen restricciones de participación sobre la descendencia de cada miembro (Véase la Tabla 2.1).

**Tabla 2.1 ECOMEX: Tabla de descuentos y comisiones**

Promotor	Supervisor	Gerente	Director	Ejecutivo
0-15%	20%	20%	20%	20%
	3%	10%	10%	10%
	3%	3%	6%	6%
	3%	3%	3%	3%
	3%	3%	3%	3%
		3%	3%	3%
			3%	3%
				3%

#### 2.4.2. Comisiones del Promotor

Además de las ganancias sobre la venta del producto (todos obtienen precio de distribuidor), el promotor también recibe los descuentos sobre compras personales (Ver Tabla 2.2). Todas y cada una de las compras hechas con su número de promotor a ECOMEX se irán acumulando mensualmente para que al cierre de mes sea otorgado un porcentaje de descuento según el monto de compras mensuales realizadas (VP):

**Tabla 2.2 ECOMEX: Tabla de descuentos a promotor**

Desde	Hasta	Porcentaje
\$0	\$5,000	0%
\$5,001	\$10,000	5%
\$10,001	\$15,000	10%
\$15,001	\$25,000	15%

Cada promotor recibe además las diferencias en porcentajes que existen entre el porcentaje que él obtuvo y el obtenido por todos y cada uno de los miembros de su descendencia.

Por ejemplo, en la siguiente organización de ventas (Ver Fig. 2.4) supongamos que todas las personas son promotores:

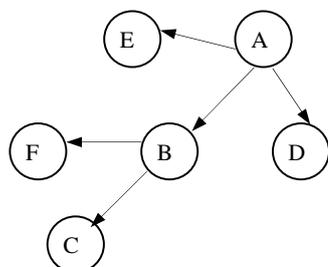


Fig. 2.4 Promotores

La tabla de descuentos y diferencias sobre descuentos quedaría establecida de la siguiente manera:

**Tabla 2.3 Descuentos y diferencias para promotores**

Promotor	Volumen de compras (VP)	Descuentos y Diferencias						Monto Total
		A	B	C	D	E	F	
A	\$14,000	10 %	10 %	5 %	10 %	0 %	10 %	$1400+200+300+50+0+150 = 2100$
B	\$2,000		0 %	0 %			0 %	$0+0+0 = 0$
C	\$6,000			5 %				300
D	\$500				0 %			0
E	\$16,000					15 %		2400
F	\$1,500						0 %	0

El promotor A obtiene un máximo de 10% (por situarse en el rango \$10,001 a \$15,000) sobre sus propias compras (\$1,400), más el 10% de las compras realizadas por B (\$200), más el 5% (la diferencia entre 10% de A y 5% logrado por C) de las compras realizadas por C (\$300), más un 0% de las compras realizadas por E (debido a que A solamente puede aspirar a un 10%, pero como E logró situarse en el rango del 15%, no existe diferencia positiva entre A y E), más el 10% de las compras realizadas por F (\$150).

Para este caso note que si B se hubiese situado en algún rango de porcentaje (digamos 10% o 15%), las diferencias de C y F no serían otorgadas a A, sino a B por ser éste quien los patrocina directamente.

Suponiendo que A no se hubiese situado en ningún rango de descuento en este período y que A fuese patrocinado por un promotor X, todo el monto obtenido por A pasaría a X si este calificase en un 10%.

El porcentaje de descuento referido, así como las diferencias son bonificados al cierre de cada mes.

### **2.4.3. Comisiones del Supervisor**

Cuando un **promotor** acumula \$25,000 con su volumen de grupo formado por todos los promotores patrocinados directamente por él (VG) en un mes calendario, es promovido a la posición de **supervisor** el día primero del siguiente mes y se le paga a partir de ese mes como Supervisor.

Para ser considerado **supervisor** calificado, el **supervisor** tiene que haber logrado un volumen personal de \$250 (VP) y tener un volumen de grupo de \$6,000 (VG) durante el mes correspondiente.

Cuando sucede esto, el **supervisor** gana:

- a) plan de compensaciones del **promotor**. El **supervisor** es a su vez un **promotor**.
- b) 5% adicional en el descuento sobre su volumen personal (VP) haciendo un total del 20%.
- c) Las diferencias entre el 20% y el porcentaje obtenido por sus **promotores** patrocinados directamente por él.
- d) 3% sobre su volumen personal de compras (VP).
- e) 3% sobre su volumen de grupo formado por todos los **promotores** patrocinados directamente por él (VG)
- f) 3% sobre el volumen de grupo (VG) de hasta 3 generaciones de **supervisores** Calificados.

De no calificar, únicamente recibirá los bonos correspondientes a **promotor** sin perder su posición de **supervisor**.

Ejemplo. En la siguiente organización de ventas (Fig. 2.5), supongamos que S es un **supervisor** Calificado:

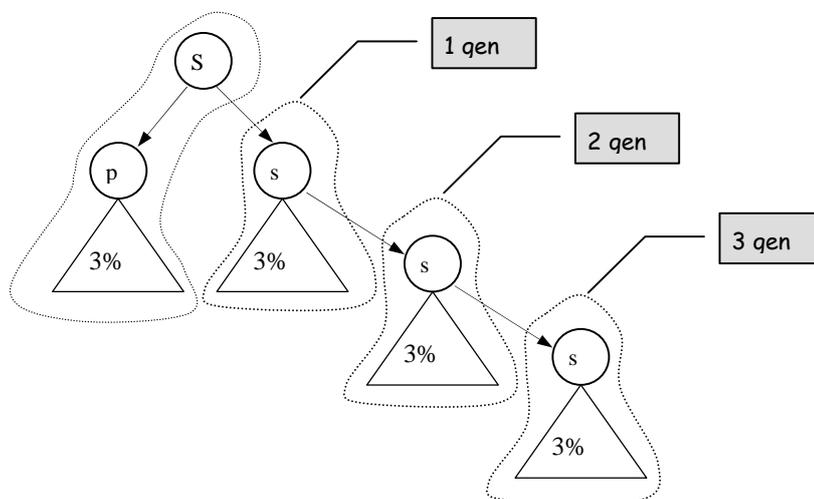


Fig. 2.5 Supervisor S calificado

S obtiene ganancias del 3% sobre su volumen personal (VP), más 3% del volumen de grupo (VG) sobre su línea de **promotores** encabezados por P, más el 3% sobre el volumen de grupo (VG) de hasta 3 generaciones de **supervisores** Calificados (s) en la línea de S.

En caso de que cualquier **supervisor** (s) en la línea de S no fuese calificado, éste no sería contado como generación para S, y el conteo de generación seguiría hacia abajo hasta encontrar otro **supervisor** Calificado. No obstante, a S se le sigue pagando el 3% sobre el VG de dicho **supervisor** no Calificado.

#### 2.4.4. Comisiones del Gerente

Cuando un **supervisor**, junto con toda su organización de ventas sin importar la posición de sus miembros acumula \$50,000 con su volumen total de grupo (VTG) en un mes calendario, es promovido a la posición de **gerente** el día primero del siguiente mes y se le paga a partir de ese mes como **gerente**.

Para ser considerado **gerente** calificado, el **gerente** tiene que haber logrado un volumen personal de \$250 (VP) y tener un volumen de grupo de \$12,000 (VG) durante el mes correspondiente.

Cuando sucede esto, el **gerente** gana:

- a) plan de compensaciones del **promotor**. El **gerente** es a su vez un **promotor**.

- b) plan de compensaciones del **supervisor**. El **gerente** es a su vez un **supervisor**.
- c) 5% adicional en el descuento sobre su volumen personal (VP) haciendo un total del 20%.
- d) Las diferencias entre el 20% y el porcentaje obtenido por sus Promotores patrocinados por él.
- e) 10% sobre su volumen personal (VP).
- f) 10% sobre su volumen de grupo formado por todos los **promotores** patrocinados directamente por él (VG).
- g) 10% sobre el volumen de grupo (VG) de hasta 3 generaciones de **supervisores** calificados.
- h) 10% sobre el volumen de grupo (VG) de sus **gerentes** no calificados en cada “down line”.

De no calificar, únicamente recibirá los bonos correspondientes a promotor y supervisor (si califica para ello) sin perder su posición de gerente.

Ejemplo. En la siguiente organización de ventas (Fig. 2.6), supongamos que G es un **gerente** calificado:

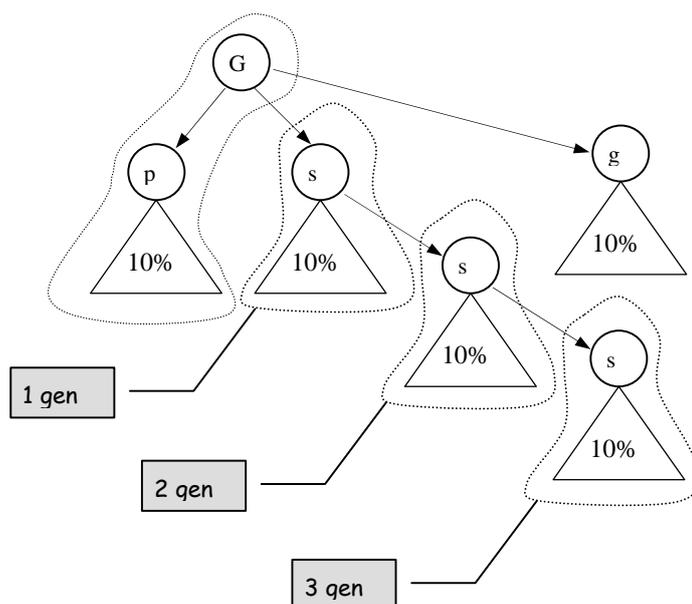


Fig. 2.6 Gerente G calificado

G obtiene ganancias del 10% sobre su volumen personal (VP), más 10% del volumen de grupo (VG) sobre su línea de **promotores** encabezados por P, más el 10% sobre el volumen de grupo (VG) de hasta 3 generaciones de **supervisores** calificados (s) en la línea de G.

En caso de que cualquier **supervisor** (s) en la línea de G no fuese calificado, éste no sería contado como generación para G, y el conteo de generación seguiría hacia abajo hasta encontrar otro **supervisor** calificado. No obstante, a G se le sigue pagando el 10% sobre el VG de dicho **supervisor** no calificado.

Adicionalmente, G recibe el 10% sobre el volumen de grupo (VG) de todos los **gerentes** (g) en sus “down-lines” que no hayan calificado como tales en ese mes calendario.

Así mismo, en caso de que G no califique en ese mes, todas sus comisiones como **gerente** que le hubiesen correspondido pasarán al primer **gerente** calificado en su up-line.

#### **2.4.5. Comisiones del Director**

Cuando un **gerente**, junto con toda su organización de ventas sin importar la posición de sus miembros acumula \$150,000 con su volumen total de grupo (VTG) en un mes calendario, es promovido a la posición de **director** el día primero del siguiente mes y se le paga a partir de ese mes como **director**.

Para ser considerado **director** calificado, el **director** tiene que haber logrado un volumen personal de \$250 (VP) y tener un volumen de grupo de \$25,000 (VG) durante el mes correspondiente.

Cuando sucede esto, el **director** gana:

- a) plan de compensaciones del **promotor**. El **director** es a su vez un **promotor**.
- b) plan de compensaciones del **supervisor**. El **director** es a su vez un **supervisor**.
- c) plan de compensaciones del **gerente**. El **director** es a su vez un **gerente**.
- d) 5% adicional en el descuento sobre su volumen personal (VP) haciendo un total del 20%.
- e) Las diferencias entre el 20% y el porcentaje obtenido por sus **promotores** patrocinados por él.
- f) 6% sobre su volumen personal (VP).
- g) 6% sobre su volumen de grupo (VG).

De no calificar, únicamente recibirá los bonos correspondientes a **promotor**, **supervisor** y **gerente** (si califica para ello) sin perder su posición de **director**.

Ejemplo. En la siguiente organización de ventas (Fig. 2.7), supongamos que D es un director calificado:

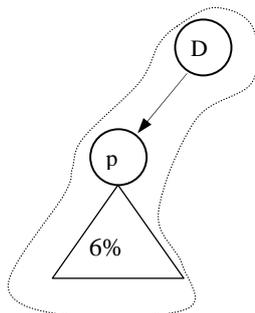


Fig. 2.7 Director D calificado

El director D recibe el 6% adicional sobre su volumen personal de compras (VP), más 6% sobre su volumen de grupo (VG).

#### 2.4.6. Comisiones del Ejecutivo

Cuando un **director**, junto con toda su organización de ventas sin importar la posición de sus miembros acumula \$300,000 con su volumen total de grupo (VTG) en un mes calendario, es promovido a la posición de **ejecutivo** el día primero del siguiente mes y se le paga a partir de ese mes como **ejecutivo**.

Para ser considerado **ejecutivo** calificado, el **ejecutivo** tiene que haber logrado un volumen personal de \$250 (VP) y tener un volumen de grupo de \$300,000 (VG) durante el mes correspondiente, además de patrocinar directamente (línea frontal) a 5 **directores** en líneas separadas.

Cuando sucede esto, el **ejecutivo** gana:

- a) plan de compensaciones del **promotor**. El **ejecutivo** es a su vez un **promotor**.
- b) plan de compensaciones del **supervisor**. El **ejecutivo** es a su vez un **supervisor**.
- c) plan de compensaciones del **gerente**. El **ejecutivo** es a su vez un **gerente**.
- d) plan de compensaciones del **director**. El **ejecutivo** es a su vez un **director**.
- e) 5% adicional en el descuento sobre su volumen personal (VP) haciendo un total del 20%.
- f) Las diferencias entre el 20% y el porcentaje obtenido por sus **promotores** patrocinados por él.
- g) 3% sobre el volumen de grupo (VG) de su 4ª generación de **supervisores** calificados
- h) Si en su volumen total de grupo (VTG) logra el monto de \$500,000 en un mes calendario recibe una cuenta de gastos de \$15,000.

De no calificar, únicamente recibirá los bonos correspondientes a **promotor**, **supervisor**, **gerente** y **director** (si califica para ello) sin perder su posición de **ejecutivo**.

En la siguiente organización de ventas (Fig. 2.8), E es un ejecutivo calificado porque patrocina directamente a 5 directores en líneas separadas:

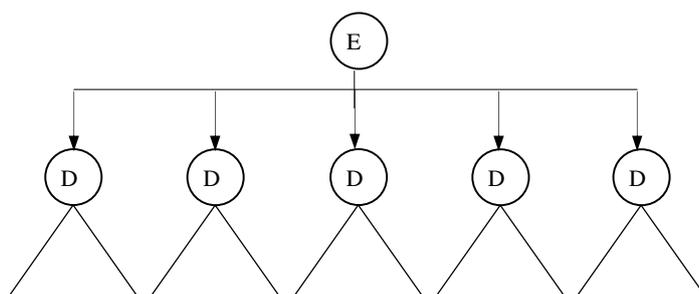


Fig. 2.8 Ejecutivo E calificado

El ejecutivo E recibe el 3% adicional sobre el volumen de grupo (VG) de la 4ª generación de supervisores calificados.

## 2.5. El conteo de generaciones

En la sección **comisiones del supervisor** de éste capítulo analizamos la manera como se realiza el pago del 3% sobre el volumen de grupo (VG) de 3 generaciones de supervisores calificados. Observamos que, en base a la definición de *generación* (**Capítulo 1**), estas son contadas a partir de la primera descendencia de Supervisores Calificados en cada línea generacional.

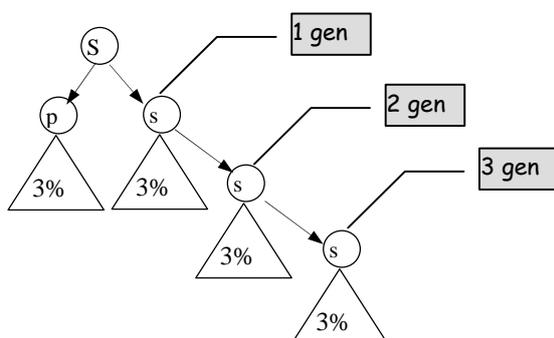


Fig. 2.9 Generaciones de S

En la gráfica anterior (Fig. 2.9) podemos observar que S cuenta con 3 generaciones de supervisores calificados (s) en una sola línea de descendencia (“down-line”). Sin embargo, en realidad no necesitan patrocinarse consecutivamente uno al otro para ser contados como *generación*.

La siguiente gráfica (Fig. 2.10) es una muestra de un ejemplo de lo que sucede en la realidad en las redes de mercadeo.

Supóngase que todos los miembros tienen el nivel de supervisores, gerentes, directores o ejecutivos (recuérdese que estos tres últimos son a su vez supervisores) y que S encabeza la organización:

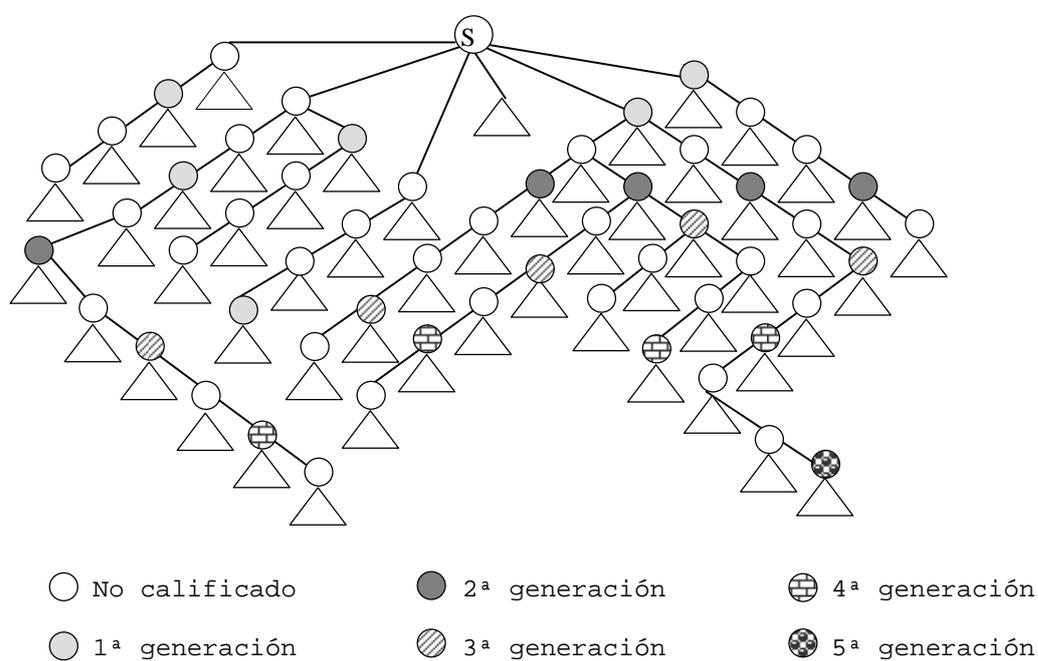


Fig. 2.10 Generaciones calificadas de S

Si todos los miembros que aparecen marcados son supervisores calificados, podríamos decir que S tiene 5 generaciones de supervisores calificados.

Note que la 1ª generación (de supervisores calificados) puede comenzar varias generaciones hacia abajo aún y cuando éstos no sean patrocinados directamente por S debido a que no se encuentran en su línea frontal.

De la misma manera, la 2ª generación (de supervisores calificados de S) comienza hasta el siguiente supervisor calificado que se encuentre en cada línea generacional. Lo mismo sucede para la 3ª, 4ª y 5ª generación.

Todo aquel supervisor  $s$  que no se encuentre calificado no será contado como generación de  $S$ ; sin embargo,  $S$  podrá obtener comisiones sobre el volumen personal de compras (VP) y volumen de grupo (VG) de  $s$ . Al no ser contado  $s$  como generación de  $S$ , le da a éste último la oportunidad de ir más abajo en cada “down-line” en la búsqueda de algún supervisor calificado.

Note también que existen miembros con varias líneas frontales. Como cada una de éstas es una organización diferente y todas forman parte a su vez de la genealogía de  $S$ , todas y cada una de las ramificaciones deberán ser tomadas en cuenta para el conteo de generaciones (Ver Fig. 2.11):

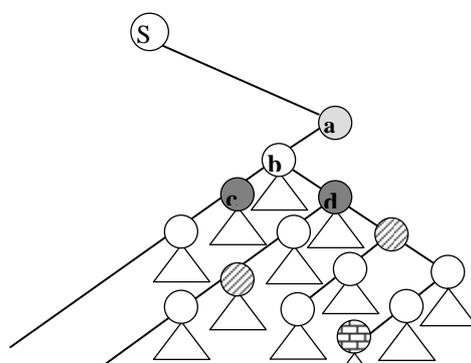


Fig. 2.11 Conteo de generaciones para  $S$

En la gráfica anterior podemos observar que la primer generación de supervisores calificados de  $S$  se encuentra en su línea frontal (miembro  $a$ ). Sin embargo, dado que el miembro  $b$  patrocina a  $c$  y  $d$  (además de la sub-red indicada), tanto  $c$  como  $d$  deberán ser contados como la 2ª generación de  $S$  debido a que cada uno de estos forman una línea de descendencia generacional de  $S$ . El miembro  $b$  no es contado como generación de  $S$  porque, aunque es un supervisor, éste no se encuentra calificado (Véase el siguiente tema: **2.6 Roll-up en miembros no calificados**).

Note también cómo surgen las dos terceras generaciones de  $S$  situadas en la línea izquierda y en la línea derecha de  $d$  respectivamente y cómo de ésta última surge la 4ª generación de supervisor calificado de  $S$ .

## 2.6. *Roll-up en miembros no calificados*



Considere por ejemplo la comisión del 10% para el gerente sobre el volumen de grupo (VG) de sus gerentes no calificados en cada “down line”.

Supongamos que A es un gerente calificado y en algunas líneas de descendencia existen gerentes calificados y no calificados.

Los bonos del 10% que A percibirá será únicamente de aquellos gerentes no calificados.

Si entre A Calificado y G no calificado existiese B calificado, entonces dichos bonos pertenecerían a éste último por encontrarse más cerca de G en su línea de ascendencia hacia A.

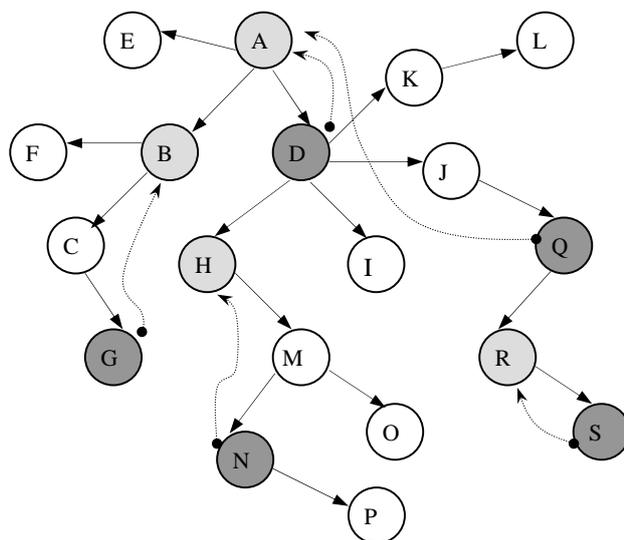


Fig. 2.13 Roll-up del Gerente

Si el miembro N estuviese calificado sería él quien se viese beneficiado del 10% de los bonos transmitidos desde P. Al no encontrarse calificado tiene que “rolarlos” hasta H, quien sí se encuentra calificado (Ver Fig. 2.13).

Esto no sucede en el caso de Q, quien por no encontrarse calificado tiene que “rolar” sus bonos hasta el miembro A porque D, aunque se encuentra en medio de la trayectoria de Q hacia A, no se hace acreedor a éstos por no estar calificado.

El miembro S no tiene que “rolar” sus bonos perdidos hasta A porque en su línea de ascendencia se encuentra R, quien por estar calificado, recibe estos bonos.

El “roll-up”, así como el esquema general del plan de pagos, también son concebidos como una política de “beneficios en cascada ascendente” por el flujo de bonos y comisiones de los niveles bajos hasta los niveles superiores de la red.

## Capítulo 3

### TEORIA DE LA ESTRUCTURA DE ÁRBOLES N-ARIOS

En capítulos anteriores hemos ilustrado gráficamente la estructura que define el modelo de organización que una red de mercadeo presenta.

Para poder realizar la implementación del modelo ramificado de una red de mercadeo es necesario comprender su estructura desde un enfoque teórico y formal.

En este capítulo partiremos de la teoría general de gráficas. Se analizará el concepto de árbol n-ario dentro de las estructuras de datos compuestas, qué elementos contiene, así como sus características formales desde el punto de vista teórico.

Se mostrará de qué manera éste modelo puede ser representado en la computadora y específicamente en una base de datos. Se verán metodologías seguidas para el recorrido de árboles y algunas de las operaciones más importantes que se practican sobre ellos; así como sus respectivos algoritmos.

#### 3.1. Teoría de Gráficas

En la sección *1.3 Estructura de una Red de Mercadeo Directo*, del **Capítulo 1** analizamos la estructura que presenta una Red de Mercadeo. Para ello utilizamos cierta simbología como representación gráfica de ésta. Establecimos los círculos para identificar a las personas que forman la organización y las flechas para denotar la recomendación y patrocinio de una persona sobre otra.

En esa sección se comienza a utilizar el concepto de “árbol” por la semejanza en estructura que una red de mercadeo presenta.

En esta sección definiremos formalmente este tipo de gráfica partiendo desde sus conceptos básicos.

##### 3.1.1. Conceptos y definiciones

Una gráfica  $G$  denotada como  $G=\{A,R\}$ , es una relación de  $R$  sobre un conjunto  $A$ . Los elementos de  $A$ , son llamados **nod**os, **puntos** o **vértices** y los elementos de  $R$  son llamados **arcos** o **líneas**.

Una relación  $R$  de un conjunto  $S$  a  $T$  es cualquier subconjunto del producto cartesiano  $S \times T$ , esto es:

$$R \subseteq \{ \langle s,t \rangle \mid s \in S \text{ y } t \in T \}$$

El producto cartesiano  $S \times T$  de dos conjuntos, es el conjunto de pares ordenados, de tal forma que la primera coordenada de cada par es un miembro de  $S$  y la segunda es un miembro de  $T$ , esto es:

$$S \times T = \{ \langle s, t \rangle \mid s \in S \text{ y } t \in T \}$$

Por ejemplo, Sea  $A = \{1, 2\}$  y  $B = \{a, b, c\}$  el producto cartesiano  $A \times B$  es:

$$A \times B = \{ \langle 1, a \rangle, \langle 1, b \rangle, \langle 1, c \rangle, \langle 2, a \rangle, \langle 2, b \rangle, \langle 2, c \rangle \}$$

Una relación  $R$  de  $A$  a  $B$ , es cualquier subconjunto  $A \times B$ , esto es:

De acuerdo a la definición de gráfica dada, consideremos que  $A$  es el conjunto de los elementos  $(n_1, n_2, n_3, \dots)$ , el producto cartesiano  $A \times A$  es  $\{ \langle n_1, n_1 \rangle, \langle n_1, n_2 \rangle, \langle n_1, n_3 \rangle, \dots, \langle n_2, n_1 \rangle, \langle n_2, n_2 \rangle, \langle n_2, n_3 \rangle, \dots, \langle n_3, n_1 \rangle, \langle n_3, n_2 \rangle, \langle n_3, n_3 \rangle, \dots \}$ . Una gráfica  $G$  es cualquier subconjunto  $A \times A$ .

Por ejemplo, sea  $A = \{a, b, c\}$  y  $R = \{ \langle a, a \rangle, \langle a, b \rangle, \langle b, c \rangle, \langle a, c \rangle, \langle c, b \rangle, \langle b, a \rangle, \langle b, b \rangle \}$ . La siguiente figura (Fig. 3.1) representa a la gráfica  $G = (A, R)$ :

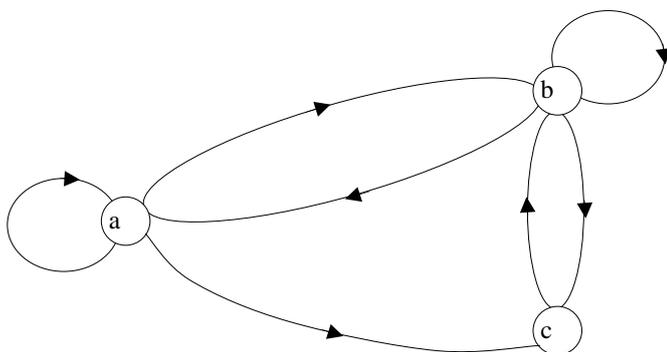


Fig. 3.1 Gráfica  $G$

### 3.1.2. Arco dirigido

Si en un arco  $e = \langle u, v \rangle$ , es importante considerar que  $u$  es el nodo inicial,  $u = \text{inic}(e)$  o sea, el nodo de donde parte el arco, y  $v$  el nodo final,  $v = \text{fin}(e)$ , el nodo a donde llega el arco, estaremos hablando de un arco dirigido (Ver Fig. 3.2):

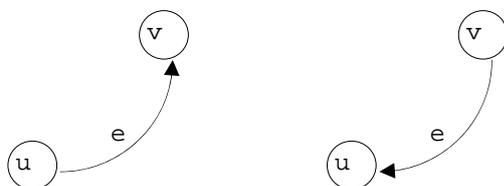


Fig. 3.2 Arco dirigido

### 3.1.3. Gráfica dirigida

Una gráfica cuyos arcos son todos dirigidos (Ver Fig. 3.3) es llamada gráfica dirigida.

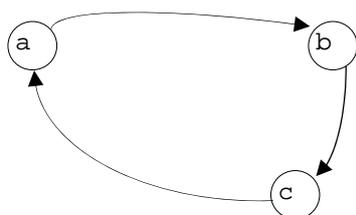


Fig. 3.3 Gráfica dirigida

### 3.1.4. Grado externo de un nodo

El grado externo de un nodo  $u$  es el número de arcos que salen de él. Esto es, el número de arcos  $e$ , tales que  $\text{inic}(e)=u$ .

### 3.1.5. Grado interno de un nodo

El grado interno de un nodo  $u$ , es el número de arcos que llegan a él. Esto es, el número de arcos  $e$ , tales que  $\text{fin}(e)=u$ .

Si en una gráfica dirigida ciertos miembros de  $R$  pueden ser colocados en una secuencia de la forma:

$$\langle a_1, a_2 \rangle \langle a_2, a_3 \rangle \langle a_3, a_4 \rangle \dots \langle a_{n-1}, a_n \rangle$$

el conjunto de arcos es llamado una trayectoria desde  $a_1$  hasta  $a_n$ . Si  $a_n=a_1$ , la trayectoria es un ciclo. Una gráfica que contiene al menos un ciclo es llamada gráfica cíclica, de otra forma es llamada gráfica acíclica (Ver Fig. 3.4).

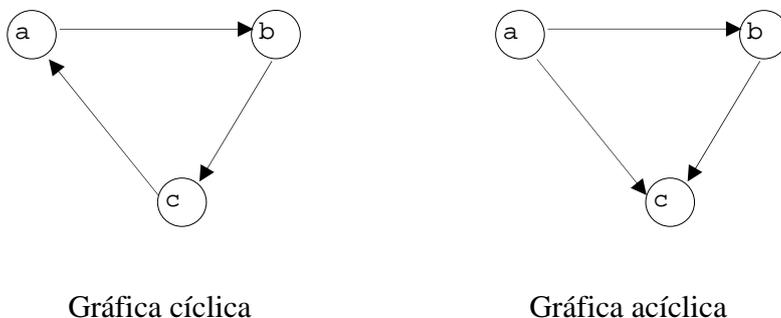


Fig. 3.4 Trayectorias en una gráfica

Cuando los arcos de la secuencia son distintos, la trayectoria es simple. Si los arcos son distintos y contienen a todos los nodos de  $A$ , la trayectoria es hamiltoniana.

La longitud de una trayectoria es el número de arcos que la componen.

### 3.1.6. Arco no dirigido

Si en un arco  $e=\langle u,v \rangle$  no es importante considerar cuál es el nodo inicial ni cuál es el nodo final, estaremos hablando de un arco no dirigido  $\langle u,v \rangle = \langle v,u \rangle$  (Ver Fig. 3.5).

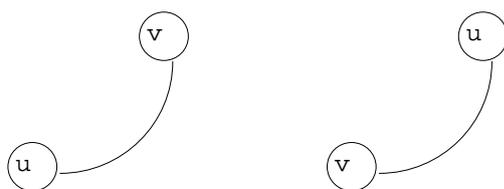


Fig. 3.5 Arco no dirigido

### 3.1.7. Lazo o loop

Un arco que une un vértice consigo mismo se llama lazo (Ver Fig. 3.6). La dirección de un lazo no tiene ningún significado y puede ser considerada como arco dirigido o no dirigido.



Fig. 3.6 Lazo o Loop

### 3.1.8. Gráfica no dirigida

Una gráfica es no dirigida (Fig. 3.7) cuando todos los arcos son no dirigidos.

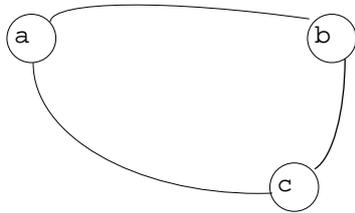


Fig. 3.7 Gráfica no dirigida

### 3.1.9. Gráfica mixta

Una gráfica es mixta cuando contiene arcos dirigidos y no dirigidos (Ver Fig. 3.8).

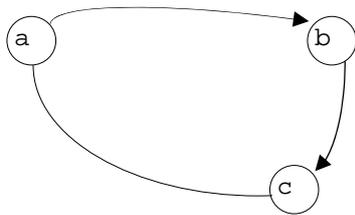


Fig. 3.8 Gráfica mixta

Con la definición de gráfica que se ha dado, no es posible conectar entre dos nodos dos arcos con el mismo sentido, ni conectar más de dos arcos, entre dos nodos.

Estas restricciones en algunos casos pueden ser eliminadas, ya que es posible encontrar que ciertos pares de nodos en algunas aplicaciones están unidos por más de dos arcos, inclusive, con el mismo sentido; tales arcos son llamados arcos paralelos (Ver Fig. 3.9).

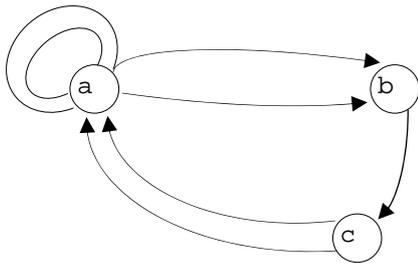


Fig. 3.9 Arcos paralelos

Una gráfica que contiene arcos paralelos se llama multigráfica. En caso de contener solamente un arco entre cualquier par de nodos se llama gráfica simple.

### 3.1.10. Arcos pesados

En muchas aplicaciones también encontramos que a los arcos de una gráfica se les asignan valores, sea ésta una gráfica dirigida o no (Ver Fig. 3.10). Estos valores, llamados peso del arco, dan origen a una gráfica pesada. El peso suele representar el costo, capacidad o propiedades del arco.

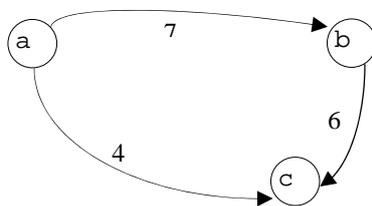


Fig. 3.10 Arcos pesados

## 3.2. Árboles

### 3.2.1. Conceptos y definiciones

Un árbol es una gráfica  $G=(A,R)$  en la que:

1. El número de nodos es igual al número de arcos más uno  $|A|=|R|+1$
2. Todos los nodos son de grado interno uno, excepto un nodo llamado la raíz, el cual es de grado interno cero.
3. No hay ciclos.
4. Cualquier trayectoria es simple.
5. Entre cualquier par de nodos sólo hay una trayectoria.
6. Cualquier arco es un arco de desconexión.

La definición proporciona las características que una gráfica debe seguir para ser un árbol. Otra forma de uso frecuente para definir un árbol es la siguiente definición recursiva:

Un árbol es un conjunto de uno o más nodos en el que hay un nodo especial, llamado la raíz del árbol, y los demás nodos son particiones en subconjuntos disjuntos  $T_1, T_2, T_3, \dots, T_n$  ( $n \geq 0$ ), cada uno de los cuales es un árbol. Cada  $T_i$  ( $1 \leq i \leq n$ ) es llamado un subárbol de la raíz.

Consideremos la siguiente gráfica  $G=(A,R)$ , donde

$A=\{a,b,c,d,e,f,\}$  y

$R=\{<a,b><a,c><a,d><c,e><c,f>\}$

La cual es representada en la siguiente figura (Fig. 3.11):

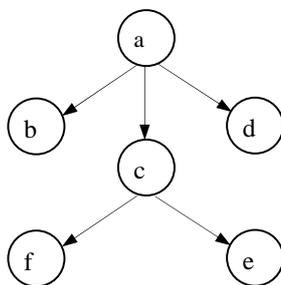


Fig. 3.11 Árbol

Para verificar si la gráfica anterior es un árbol, examinemos las particularidades de la definición.

1. El número de nodos es efectivamente el número de arcos más uno:  $6=5+1$ .
2. Cualquier nodo es de grado interno 1 excepto el nodo (a) que es la raíz.
3. No existe ningún ciclo en la gráfica.
4. Las trayectorias son todas simples.
5. Entre dos nodos cualesquiera sólo hay una trayectoria.
6. Cualquier arco que se retire de la gráfica desconecta una parte de ella.

Al ser aplicables estos conceptos a la gráfica, podemos decir que G es un árbol.

La segunda definición aplicada sobre  $A=\{a,b,c,d,e,f\}$  de forma que:

1. Si llamamos al nodo (a) la raíz del árbol y
2. los nodos restantes los particionamos en  $T_1=\{b\}$ ,  $T_2=\{c,e,f\}$  y  $T_3=\{d\}$ , obtenemos la gráfica intermedia mostrada en la Figura 3.12

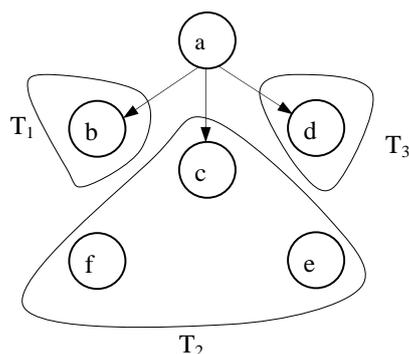


Fig. 3.12 Gráfica particionada

3. La partición  $T_2$  todavía no es un árbol. Si llamamos en esta partición al nodo (c) como la raíz y,
4. los nodos restantes los particionamos en  $T_4=\{e\}$  y  $T_5=\{f\}$ , las particiones resultantes son todas árboles (Ver Fig. 3.13)

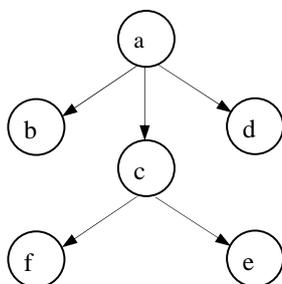


Fig. 3.13 Árbol

En la terminología que se emplea para el estudio de los árboles encontramos entre otros, los términos siguientes:

- Se define como grado o grado externo de un nodo al número de sus subárboles.
- Una hoja o nodo terminal es un nodo de grado cero.
- Un nodo ramal es un nodo de grado mayor que cero.
- El nivel de un nodo es el nivel de su antecesor directo más uno. El nivel de la raíz es 1.
- También es frecuente que los nodos de un árbol reciban nombres, tales como:
  - el nodo (a) es padre de b, c, d,
  - b, c, d, son hijos de (a)
  - b, c, d son hermanos.

### 3.2.2. Árboles ordenados, orientados y libres

Un árbol es **ordenado** cuando el orden de los subárboles es importante. Si el orden carece de importancia, se dice que el árbol es **orientado**. Cuando la dirección de los arcos se encuentra indefinida para un árbol orientado, se dice que este es un árbol **libre**.

Por ejemplo, si consideramos los siguientes tres árboles del la Fig. 3.14:

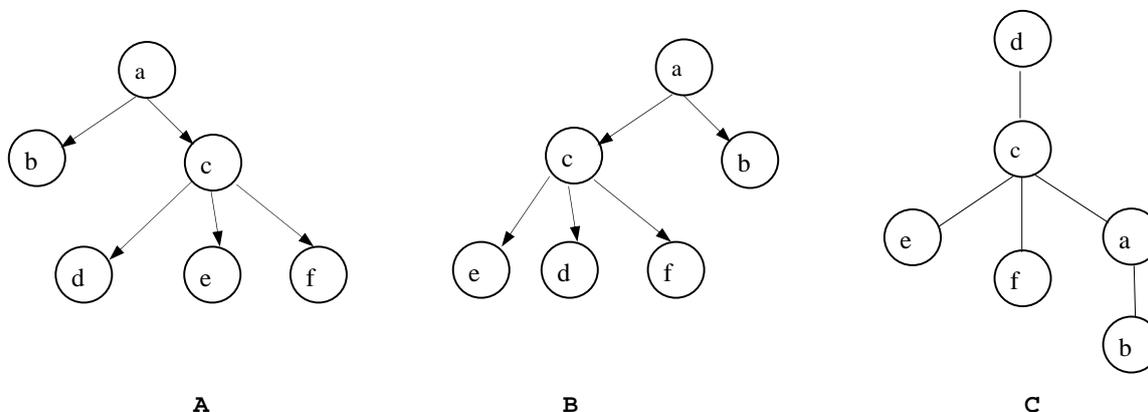


Fig. 3.14 Tipos de Árbol

notamos que, siendo:

- |                           |                            |
|---------------------------|----------------------------|
| $A \diamond B \diamond C$ | Los árboles son ordenados  |
| $A = B \diamond C$        | Los árboles son orientados |
| $A = B = C$               | Los árboles son libres     |

### 3.2.3. Representación de árboles en la computadora

Un árbol puede ser completamente especificado por una matriz de adyacencia de forma similar a una gráfica dirigida (Ver Fig. 3.15). Para especificar el nodo raíz, es posible utilizar los elementos de la diagonal principal que siempre estarán disponibles.

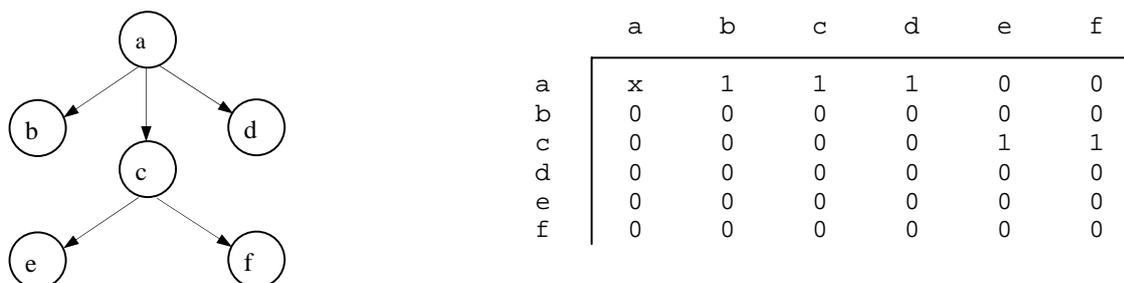


Fig. 3.15 Representación matricial

Otra alternativa es la utilización de un arreglo unidimensional (Ver Fig. 3.16) en el que cada elemento represente a cada uno de los nodos del árbol y su contenido a su antecesor directo:

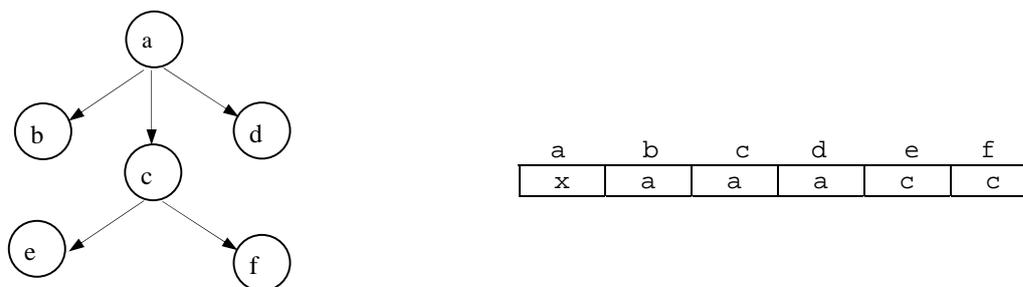


Fig. 3.16 Representación unidimensional

### 3.2.4. Recorrido en árboles binarios

Un árbol binario es un árbol en el que cualquier nodo tiene cero, uno o dos subárboles. Cuando tiene exactamente cero o dos subárboles es llamado árbol estrictamente binario.

Para este estudio no profundizaremos en este tipo de árboles, ya que estos son un caso particular de los árboles n-arios (nos interesamos más en estos últimos porque en realidad una empresa apegada más al esquema multinivel tiene más una estructura n-aria que estrictamente binaria).

Debido a que las técnicas de recorrido para un árbol se centran más en los binarios, es por esto que hacemos referencia a ellos, sin embargo, se analizará también una propuesta de recorrido para los árboles n-arios.

Recorrer un árbol es un método de visitas de los nodos con el objeto de sistematizar la recuperación de la información almacenada en los nodos de un árbol.

Una de las formas más simples de recorrer un árbol de arriba hacia abajo (top-down), es iniciar las visitas por la raíz y continuar sobre los nodos del nivel 2 de izquierda a derecha, continuar sobre los del nivel 3, hasta alcanzar el nivel n.

Cuando procedemos desde el nivel n hacia la raíz, el recorrido es de abajo hacia arriba (bottom-up) (Ver Fig. 3.17)

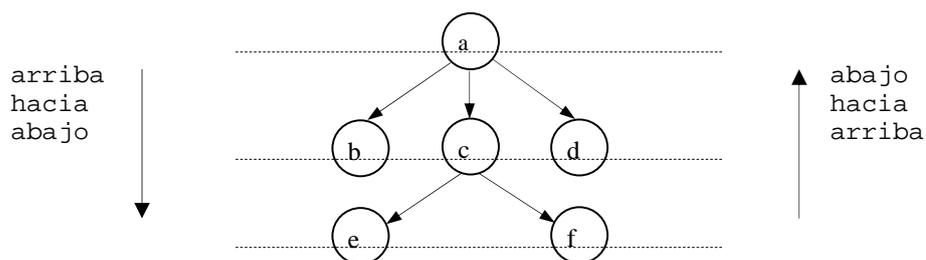


Fig. 3.17 Recorridos

Los recorridos también pueden practicarse sistematizando la visita de los subárboles:

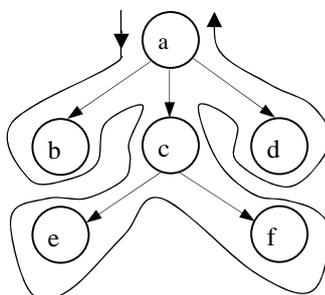


Fig. 3.18 Recorrido con visita a los nodos

Cuando los recorridos se practican de acuerdo a esta trayectoria (Fig. 3.18) podemos ver que la información contenida en un nodo puede ser recuperada antes de visitar sus subárboles, después de visitar alguno de sus subárboles o después de visitar los dos subárboles. Estas formas de recuperar la información de los nodos de un árbol caracterizan a los recorridos llamados: Preorder, Inorder, Postorder respectivamente.

Los algoritmos que realizan estos recorridos pueden ser postulados en forma recursiva o no recursiva. Por facilidad, usaremos una definición recursiva para los algoritmos.

### **RECORRIDO PREORDER**

Pasos a seguir:

1. se visita el nodo raíz
2. se recorre el subárbol izquierdo
3. se recorre el subárbol derecho

```

FUNCIÓN <PREORDER>
(
SE VISITA EL NODO RAÍZ
SI EL SUBARBOL IZQUIERDO EXISTE Y NO SE HA VISITADO
    LLAMADA A PREORDER
FIN
SI EL SUBARBOL DERECHO EXISTE Y NO SE HA VISITADO
    LLAMADA A PREORDER
FIN
RETORNAR
)

```

### **RECORRIDO INORDER**

Pasos a seguir:

1. se recorre el subárbol izquierdo
2. se visita el nodo raíz
3. se recorre el subárbol derecho

```

FUNCIÓN <INORDER>
(
SI EL SUBARBOL IZQUIERDO EXISTE Y NO SE HA VISITADO
    LLAMADA A INORDER
FIN
SE VISITA EL NODO RAÍZ
SI EL SUBARBOL DERECHO EXISTE Y NO SE HA VISITADO
    LLAMADA A INORDER
FIN
RETORNAR
)

```

## RECORRIDO POSTORDER

- Pasos a seguir:
1. se recorre el subárbol izquierdo
  2. se recorre el subárbol derecho
  3. se visita el nodo raíz

```

FUNCIÓN <POSTORDER>
(
SI EL SUBARBOL IZQUIERDO EXISTE Y NO SE HA VISITADO
  LLAMADA A POSTORDER
FIN
SI EL SUBARBOL DERECHO EXISTE Y NO SE HA VISITADO
  LLAMADA A POSTORDER
FIN
SE VISITA EL NODO RAÍZ
RETORNAR
)

```

### 3.3. Propuesta de operación para los árboles *n*-arios

#### 3.3.1. Algoritmos de recorrido para árboles *n*-arios

Los tipos de recorrido descritos anteriormente aplican normalmente para todos los árboles binarios puesto que únicamente están implicados tres nodos durante cualquiera de los recorridos en los subárboles:

1. Nodo raíz
2. Nodo izquierdo
3. Nodo derecho

Al pensar en una automatización para un árbol *n*-ario, es necesario desechar el concepto de *nodo izquierdo* y *nodo derecho*, ya que existe un número indeterminado de *m* nodos conectados a cada elemento.

Siendo esto así, el algoritmo de recorrido **INORDER** queda invalidado por no haber un “orden” izquierdo-nodo-derecho que lo justifique.

Recorrer un árbol *n*-ario requiere que se realice una adaptación a los algoritmos PREORDER y POSTORDER:

## RECORRIDO PREORDER PARA UN ARBOL N-ARIO

Pasos a seguir:      1. se visita el nodo raíz  
                           2. se recorren todos los subárboles existentes

Sea  $n$  el número de sub-nodos conectados a un nodo raíz cualquiera:

```

FUNCIÓN <POSTORDER n-ario>
(
SE VISITA EL NODO RAÍZ
PARA  $i=1$  HASTA  $n$ , HACER:
    SI SUB-NODO( $i$ ) NO SE HA VISITADO
        LLAMADA A POSTORDER n-ario
FIN
RETORNAR
)
  
```

## RECORRIDO POSTORDER PARA UN ARBOL N-ARIO

Pasos a seguir:      1. se recorren todos los subárboles existentes  
                           2. se visita el nodo raíz

Sea  $n$  el número de sub-nodos conectados a un nodo raíz:

```

FUNCIÓN <POSTORDER n-ario>
(
PARA  $i=1$  HASTA  $n$ , HACER:
    SI SUB-NODO( $i$ ) NO SE HA VISITADO
        LLAMADA A POSTORDER n-ario
FIN
SE VISITA EL NODO RAÍZ
RETORNAR
)
  
```

Nótese que el recorrido para este tipo de árbol siempre comenzará desde un nodo cualquiera cuyo grado externo es cero, es decir, que no cuenta con arcos que salen de él hacia otros subárboles, aunque no necesariamente se encuentre ubicado en el nivel más bajo del árbol general.

Por formalidad, diremos que el recorrido en este tipo de árbol siempre será top-down (de arriba hacia abajo) y de izquierda a derecha; sin embargo, en la práctica, carece de sentido ordenar un árbol para su recorrido con este método ya que es un árbol  $n$ -ario en donde, a diferencia del árbol binario, no existe nodo izquierdo ni derecho, simplemente existen  $n$  subárboles para cada nodo.

### 3.3.2. Representación de un árbol n-ario en la computadora

En la sección 3.2.3 se plantearon los dos métodos más comunes de representar los árboles en la computadora.

Debido a que los árboles n-arios tienen un número indefinido de nodos y ramificaciones es más conveniente utilizar el arreglo unidimensional, el cual nos brinda la manera más sencilla y rápida de organizar el árbol.

El mecanismo de arreglo unidimensional (vector) en la memoria primaria de la computadora nos servirá para realizar el recorrido sobre el árbol. La dimensión de este arreglo será definida en función a la cantidad de nodos que componen el árbol.

Por ejemplo, para el siguiente árbol (Fig. 3.19) podemos establecer el mecanismo de representación por medio de un vector unidimensional:

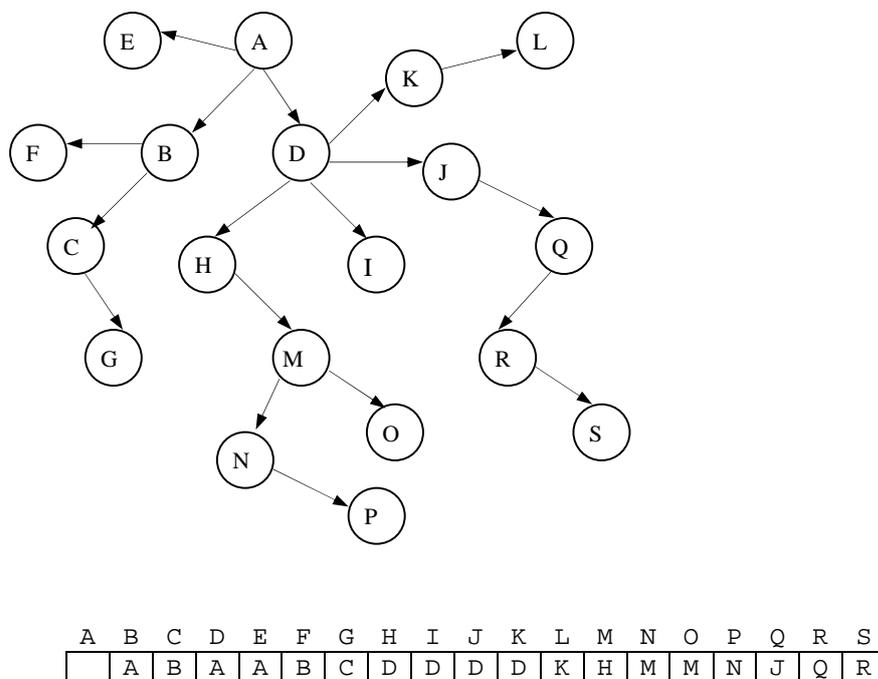


Fig. 3.19 Representación unidimensional del árbol

Este vector representa el árbol n-ario en la computadora.

Dentro del vector representamos el nodo paterno del elemento referido. Por ejemplo, el nodo paterno de B es A, el de C también es A, el de I es D, etc.

Note que el nodo paterno de A se encuentra indefinido, lo cual indica que A no tiene nodo paterno, es decir, A es el nodo raíz del árbol.

### 3.3.3. Recorrido en un árbol n-ario

Una vez establecida la representación en computadora del árbol n-ario, es necesario realizar un “ordenamiento” previo a su recorrido.

En la sección 3.2.2 se planteó el método más conveniente de representar los árboles en la computadora. Con esto obtenemos un vector (Ver Fig. 3.20) el cual hay que ordenarlo por “nodo paterno” obteniendo así de su representación original el vector resultado (Ver Fig. 3.21)

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
	A	B	A	A	B	C	D	D	D	D	K	H	M	M	N	J	Q	R

Fig. 3.20 Representación del árbol

A	B	D	E	C	F	G	H	I	J	K	M	Q	L	N	O	P	R	S
	A	A	A	B	B	C	D	D	D	D	H	J	K	M	M	N	Q	R

Fig. 3.21 Vector ordenado por nodo paterno

Nótese que el “ordenamiento” es en realidad un “agrupamiento” por nodo paterno.

En este ejemplo, donde utilizamos letras del abecedario para representar los nodos del árbol, el vector resultado coincide con un orden alfabético del nodo paterno. Sin embargo, en la práctica encontraremos que es mejor utilizar un identificador numérico para cada nodo del árbol, y como es un árbol general n-ario no necesariamente obtendremos un orden secuencial de los números identificadores del árbol, sino más bien, un “agrupamiento” de nodos.

Una vez que obtenemos el vector resultado de nodos paternos aplicamos el método deseado de recorrido.

Por ejemplo, si deseamos recorrer el árbol n-ario de la Fig. 3.19 por el método **postorder** partiendo del vector ordenado mostrado en la Fig. 3.21, obtenemos la sucesión de recorrido mostrado en la Fig. 3.22.

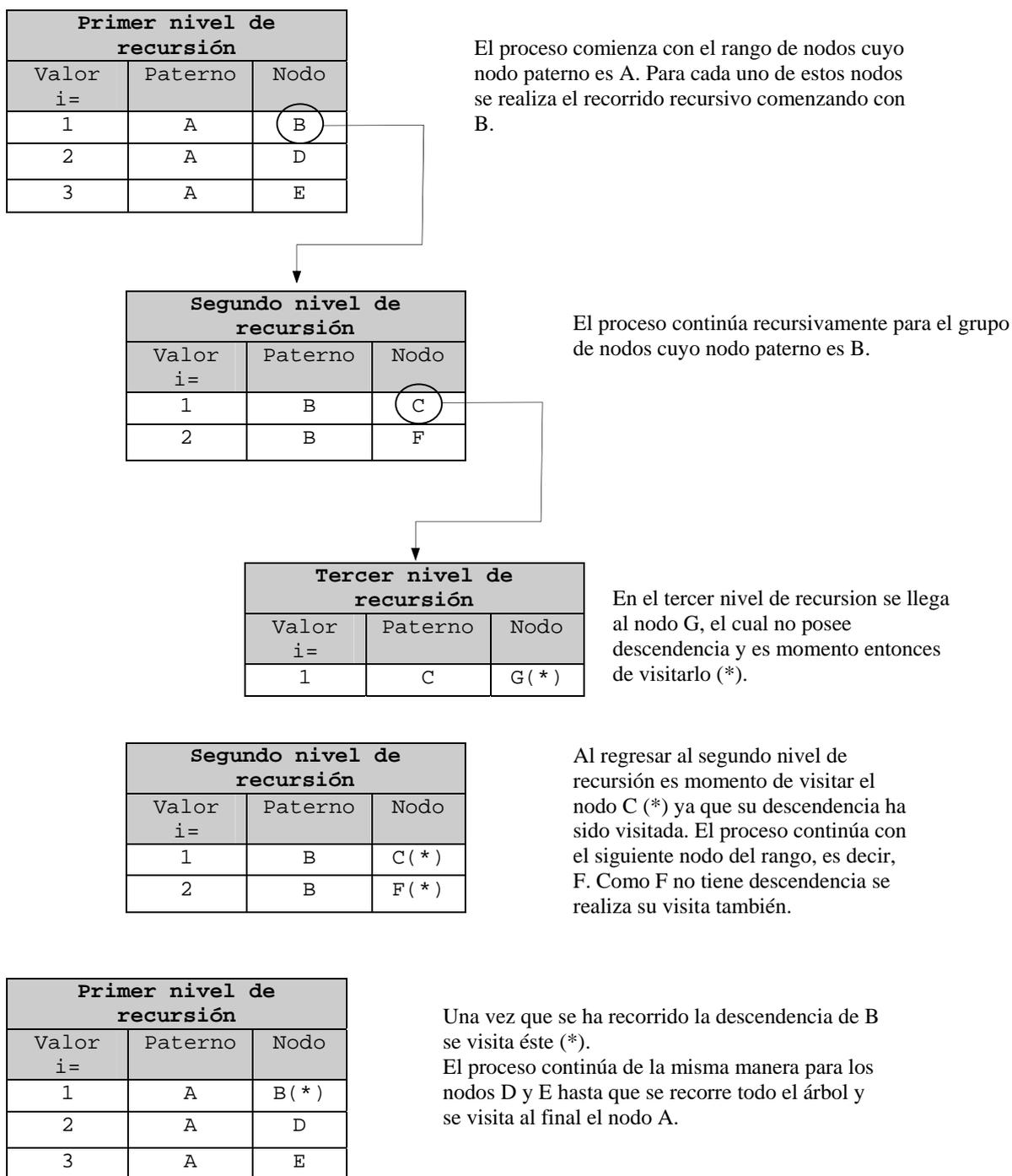


Fig. 3.22 Sucesión de recorrido del árbol

Si aplicamos el método de recorrido para el árbol de la Fig. 3.19 obtendremos un trazo de la secuencia de eventos de recorrido mostrado en la Tabla 3.1:

**Tabla 3.1 Eventos de recorrido**

Número de Evento	Evento: Pares de nodos (paterno,nodo) y visitas realizadas (*) en el recorrido
1	( ,A)
2	(A,B)
3	(B,C)
4	(C,G)
5	G*
6	C*
7	(B,F)
8	F*
9	B*
10	(A,D)
11	(D,H)
12	(H,M)
13	(M,N)
14	(N,P)
15	P*
16	N*
17	(M,O)
18	O*
19	M*
20	H*
21	(D,I)
22	I*
23	(D,J)
24	(J,Q)
25	(Q,R)
26	(R,S)
27	S*
28	R*
29	Q*
30	J*
31	(D,K)
32	(K,L)
33	L*
34	K*
35	D*
36	(A,E)
37	E*
38	A*

### 3.3.4. Sugerencias de implementación

Al trabajar con aplicaciones automatizadas de árboles n-arios se sugiere observar los siguientes aspectos:

- 1) la utilización de un **arreglo unidimensional** siempre será la mejor opción para su representación en la computadora, ya que, debido a que un árbol n-ario tiene una longitud indefinida y dinámica, el arreglo matricial requerirá mayor utilización de la memoria
- 2) el tipo de recorrido **inorder** para un árbol n-ario siempre será descartado, en su lugar pensaremos únicamente en recorridos **preorder** y **postorder**.
- 3) Al representar un árbol n-ario en la computadora es recomendable utilizar arreglos tipo registro con los siguientes campos básicos:

```
type nodo = record
```

```
    Paterno      : integer;
    Nodo         : integer;
    Visitado     : boolean;
```

```
end;
```

Esta estructura es dinámica y le permite al algoritmo trabajar con rapidez durante el recorrido. (Véase siguiente capítulo)

- 4) es conveniente identificar a cada nodo del árbol con un número consecutivo asignado automáticamente al momento de agregarlo. Esto nos ayudará a optimizar el proceso de ordenamiento para el recorrido y análisis del árbol.
- 5) diversos autores de software para administración de árboles sugieren identificar a cada nodo con un prefijo numérico (Ver Fig. 3.23), el cual indica el nivel en que el nodo se sitúa en el árbol. Por ejemplo:

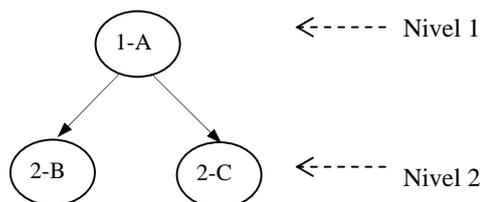


Fig. 3.23 Identificación por nivel

ésta técnica aplicada a los árboles n-arios, además de ser innecesaria, entorpece en gran medida el manejo del árbol sobre todo en los procesos de operación de nodos (altas, bajas y recorrido).

## Capítulo 4

### IMPLEMENTACION DEL MODELO DE ARBOL N-ARIO

En el **Capítulo 2** se analizó la estructura de una empresa multinivel y se relacionó con el modelo general de árboles. Se determinaron además, las operaciones que se realizan comúnmente en este tipo de empresas.

Una vez que se estudió el **Capítulo 3** podemos discernir que los modelos generales de árboles pueden ser perfectamente utilizados para la administración de empresas multiniveles.

En este capítulo se verán los mecanismos recomendados para la implementación de los modelos de árboles en las empresas con estructura multinivel. Aunque no es el propósito de esta tesis brindar un análisis desde la perspectiva de software ni mucho menos la programación y el diseño de una base de datos para su implementación, se considerarán los aspectos básicos para que el lector pueda desarrollar un sistema automatizado en el lenguaje de su preferencia. El objeto de nuestro estudio se centra principalmente en la representación del árbol de la organización en la computadora y los mecanismos de cálculo mediante el algoritmo de recorrido sugerido en el **Capítulo 3**.

#### *4.1. Utilización del programa Delphi (Borland) para ejemplificar la implementación del modelo*

En esta sección describiremos algunos de los componentes utilizados por el programa Delphi de Borland Inc. (versión 3, 5 o 7) que usaremos para ejemplificar la implementación de nuestro modelo de árboles.

Es conveniente que el lector se encuentre familiarizado al menos en un nivel básico con la programación orientada a objetos y el Lenguaje de Consulta Estructurado (SQL) para hacer más comprensible este capítulo.

En el **Capítulo 3** se analizó un ejemplo de una empresa multinivel acoplada perfectamente a la **estructura** de árboles n-arios. Allí se vio la necesidad de utilizar algunos mecanismos automatizados de registro de información para la administración de la organización (red). Si partimos del hecho de que nuestra aplicación utiliza una base de datos para registrar no sólo la plantilla de miembros de la organización, sino también los movimientos de compras de cada uno de los miembros, podremos, de acuerdo a las necesidades particulares de implementación, diseñar un modelo de base de datos y un programa de interfase con el usuario final para interactuar con esa base de datos.

Si bien depende de cada desarrollador de la aplicación el formato y lenguaje a utilizar para tal propósito, podemos decir que su información radica en algún archivo o conjunto de archivos de un disco duro.

Para hacer más sencilla nuestra implementación, utilizaremos la arquitectura por default que Borland usa para conectarse desde una aplicación hecha en Delphi a cualquier base de datos compatible (Ver Fig. 4.1).

Esta arquitectura se conoce como Borland Database Engine (BDE), y es una capa intermedia entre la base de datos y el programa que la accesa:

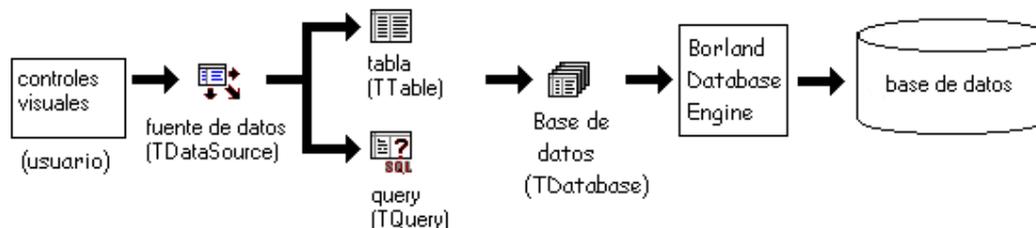


Fig. 4.1 Componentes de Borland

Los componentes que utilizaremos de Delphi para acceder a la base de datos por medio del BDE son los siguientes:

### TDataBase

Creará una conexión de acceso que vincula la base de datos registrada por el BDE y nuestra aplicación desarrollada en Delphi. La plataforma utilizada es el Borland Database Engine (BDE) y el formato original de la base de datos puede ser Paradox, dBase, Interbase, SQL Server, etc.

### TTable

Creará un acceso directo a cualquier tabla o archivo registrado en la base de datos. Cuando no existe un componente TDataBase, el TTable puede conectarse directamente con la base de datos creando un TDataBase en tiempo de ejecución.

Por ejemplo, si nuestra base de datos se encuentra en formato dBase y el archivo que contiene la plantilla de miembros se llama: 'arbol.dbf', las propiedades del componente TTable podrían ser:

```

Type Arbol:TTable;
:
Arbol.DatabaseName := 'c:\db'
Arbol.FileName      := 'arbol.dbf'
Arbol.IndexFieldName := 'Paterno'
  
```

Con estas propiedades estaremos indicando que:

- El elemento **Arbol** es del tipo **TTable**
- la base de datos se encuentra en el directorio: **c:\db**
- el archivo que accedemos será: **arbol.dbf**
- el campo de índice será: **Paterno**

## TQuery

Crea un acceso de consulta (query) SQL a uno o más archivos vinculados en la base de datos. Este componente descende de la misma clase que **TTable** (TDataSet) y es utilizado para formar consultas a la base de datos.

Por ejemplo, si deseamos obtener una lista de la plantilla de miembros podríamos definir las siguientes propiedades:

```
Type Consulta:TQuery;
:
Consulta.DatabaseName := 'c:\db'
Consulta.SQL.Add('select * from arbol.dbf order by Paterno')
```

Con estas propiedades estaremos indicando que:

- El elemento **Consulta** es de tipo **TQuery**
- La base de datos se encuentra en el directorio: **c:\db**
- La sentencia de consulta estará formada por la selección de todos los campos del archivo **arbol.dbf** ordenada por el campo **Paterno**

## TDataSource

Crea un vínculo entre algún TDataSet (TTable, TQuery) de la base de datos y los controles visuales del usuario.

### Controles visuales (TDBEdit, TDBGrid, TDBMemo, etc)

Crean mecanismos de visualización y edición de los datos contenidos en los registros y campos para alguna tabla (TTable) o consulta (TQuery).

En cualquiera de sus versiones, Delphi cuenta con estos y más componentes que nos son útiles para nuestro propósito. (Para mayor información véase el manual: **Database Application Developer's Guide** de Borland Delphi)

## 4.2. Registro de la plantilla de miembros (ensamblado de la red)

Como se vio en el **Capítulo 2**, la organización de la red de mercado se compone de todas las personas comercializadoras que la forman. Conforme se da ingreso a cada promotor se da al mismo tiempo forma al árbol de la organización.

Es precisamente en la fase de registro de los miembros en la que se ensambla la red comercial o árbol de la organización.

Debido a que la aplicación de árboles n-arios que estamos analizando requiere el manejo dinámico del árbol, es necesario que éste sea representado en la memoria permanente de la computadora, es decir, en un archivo que radique en el disco duro de ésta.

Además, existen diversas propiedades para cada nodo (miembro) requeridas en la aplicación para empresas multinivel:

- a) Clave personal
- b) Clave de patrocinador
- c) Nombre
- d) Dirección
- e) Volumen de venta personal
- f) Volumen de venta de grupo
- g) Categoría (nivel o posición en la organización)
- Etc.

Estos datos deberán ser registrados permanentemente en campos complementarios en el archivo.

Para esto es necesario que se cuente con un programa de software que registre en un archivo de base de datos estos campos. El lenguaje de programación y la arquitectura de base de datos a utilizar dependerá de las preferencias del programador y las políticas de la empresa. En nuestra ejemplificación de la implementación podremos utilizar un componente **TTable** que accese al archivo que contiene la plantilla de miembros en la base de datos. Suponiendo que tal archivo físicamente se llama “**arbol.dbf**”, el componente **TTable** contará con este nombre en su propiedad “**TableName**”.

Los campos contenidos en el archivo **Arbol** son diversos; por el momento diremos que los dos campos que nos interesan para darle ensamblado al árbol de la organización son:

- a) Clave personal (Nodo)
- b) Clave de patrocinador (Paterno)

#### 4.2.1. *Clave personal*

Es el código asignado a cada miembro (nodo) que ingresa a la organización. Es recomendable que éste tenga las siguientes propiedades:

- 1) Tipo de campo: numérico entero.
- 2) Valor asignado por el sistema automáticamente al momento de agregar el registro a la base de datos. Para esto quizás sea conveniente que se lleve el último número asignado en un archivo auxiliar.
- 3) Número consecutivo asignado cronológicamente conforme se van ingresando miembros.
- 4) Valor único para cada miembro.
- 5) Campo llave del archivo de miembros.

#### 4.2.2. *Clave de patrocinador*

Es la clave personal asignada al miembro del grupo que “recomendó” a la persona ingresada. Hablando en términos técnicos, diremos que es el nodo **paterno** del nodo que se está agregando al árbol.

Es recomendable que éste tenga las siguientes propiedades:

- 1) Tipo de campo: numérico entero.
- 2) Valor validado por el sistema automáticamente al momento de introducirlo en el campo. El programa deberá verificar que este dato se encuentre asignado a algún otro miembro de la organización.
- 3) Campo indexado junto con la clave personal.

Contando con estos dos datos bastará para que el árbol de la organización pueda ser ensamblado.

#### 4.2.3. *¿Quién es el "nodo raíz"?*

Cada red de mercadeo multinivel es encabezada por un nodo raíz, el cual en muchos casos puede ser cualquiera de estos dos:

- 1) la empresa
- 2) el socio principal

Como es de observarse, en la representación del árbol en la base de datos este es el único nodo que carece de Clave de patrocinador.

Es recomendable que, para lograr una consistencia en los datos del sistema, el nodo raíz cuente con las siguientes propiedades:

- 1) Clave personal es igual que 1
- 2) Clave de patrocinador es igual que cero o -1
- 3) A diferencia que cualquier otro miembro de la organización o nodo del árbol, el nodo raíz es el único cuya clave de patrocinador es única (solo existe un nodo raíz de grado interno igual que cero).

#### 4.2.4. *Cómo evitar los nodos "huérfanos"*

En el ensamblado de un árbol de organización es común enfrentarse a la problemática de los “nodos huérfanos”. Estos son aquellos nodos cuya clave de patrocinador es indefinida. Esto sucede principalmente cuando un nodo o miembro de la red es removido de ésta por diversas razones (separación de la empresa, incapacidad permanente, muerte, etc.) causando que toda su descendencia, es decir, los miembros que patrocina, quede “huérfana”.

Para evitar esto, es recomendable contar con una función de software que “reasigne” o “conecte” cada nodo de descendencia con su “abuelo”.

Para ilustrarlo, considérese el siguiente árbol en su estado original:

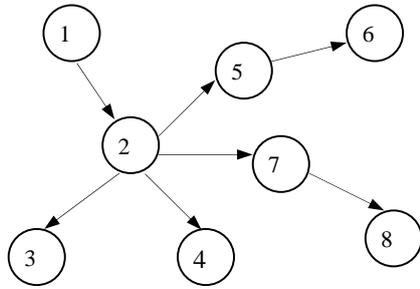


Fig. 4.2 Árbol ensamblado

Si removemos el nodo 2 al árbol de la Fig. 4.2, los nodos 3, 4, 7 y 5 quedarían “huérfanos”:

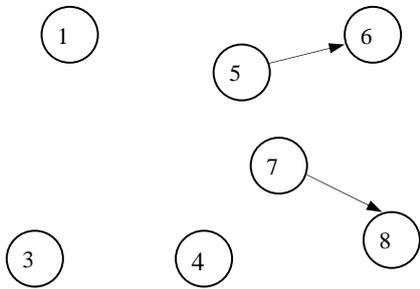


Fig. 4.3 Árbol desensamblado

Para evitar esta inconsistencia en el árbol podemos “reasignar” o “conectar” los nodos 3, 4, 7 y 5 con su “abuelo”, el nodo 1:

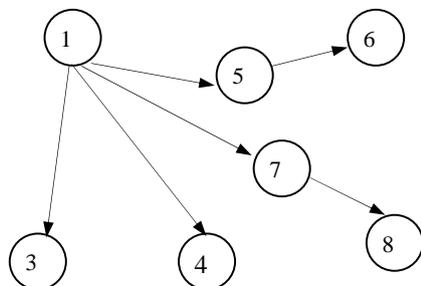


Fig. 4.4 Árbol reconectado

Esta técnica es la más recomendable a seguir por las empresas multinivel cuando existe una separación de un miembro de su organización debido a que en términos comerciales es conveniente que si una persona patrocina a otra que es dada de baja, la primera (quien dio origen al grupo) cuenta con más derechos de “adoptar” a su descendencia.

Otra política que se sigue es “ascender” a cualquiera de los nodos “hijos” al lugar que ocupaba el “padre” de estos, aunque esto tenga la desventaja de que se creen conflictos y rivalidades entre los nodos “hijos” poniendo en riesgo la estabilidad comercial del grupo.

#### 4.2.4.1. Ejemplo de procedimiento para adoptar descendencias de nodos eliminados

Los siguientes procedimientos son un ejemplo de cómo reasignar la descendencia de un nodo que es eliminado del árbol.

Estos se presentan realizados en programación **Delphi-pascal** de **Borland** utilizando dos componentes de base de datos:

**Arbol: (TTable).** Es el archivo que contiene la plantilla de miembros. Se asume el hecho de que el usuario del sistema ha posicionado el puntero de registro de este archivo en el registro a eliminar y ha confirmado la operación.

**Query\_Descendencia: (TQuery).** Es el conjunto de nodos que son descendencia del registro a eliminar en el archivo **Arbol**.

La propiedad SQL de este componente es la sentencia de selección que obtiene el rango de registros (descendencia) que se reasignarán hacia su “abuelo”:

```
select Paterno from arbol.dbf where Paterno=:xClavePaterno
```

El parámetro se identifica por **:ClavePaterno** y es asignado en el procedimiento **Reasigna\_Descendencia** del Procedimiento 1 por la función interna **ParamByName** previo a su apertura. Este procedimiento es ejecutado toda vez que un nodo va a ser eliminado del árbol (Ver Procedimiento 2).

### Procedimiento 1: Adopción de descendencia para nodos huérfanos

```
procedure Reasigna_Descendencia(ClaveNodo,ClaveAbuelo:integer)
begin
    Query_Descendencia.ParamByName('ClavePaterno').Value:=ClaveNodo;
    Query_Descendencia.Open;
    while not Query_Descendencia.Eof do
        begin
            Query_Descendencia.Paterno:=ClaveAbuelo;
            Next;
        end;
    end;
end;
```

### Procedimiento 2: Eliminación de un nodo del árbol

```
procedure EliminaMiembro;
begin
    Reasigna_Descendencia(Arbol.Nodo, Arbol.Paterno);
    Arbol.Delete;
end;
```

### 4.3. Registro de las compras realizadas (obtención del VP)

En el **Capítulo 2** se describió el proceso que normalmente sigue una empresa multinivel para evaluar el plan de compensaciones a los miembros de la red a partir de los montos logrados en las compras personales.

El dato primario para hacer esto es el volumen personal (**VP**) de compra, que como ya se vio, es el monto total de compras que un miembro realiza a su nombre, sean estas para consumo personal o para venta.

Por lo general, en una empresa multinivel este volumen es actualizado mes con mes calendario.

La implementación del registro de compras realizadas consiste en elaborar un mecanismo automatizado que permita registrar este dato y asociarlo a cada miembro de la organización.

Para lo anterior, es necesario contar con un módulo del sistema de administración que registre la siguiente información en un archivo de base de datos:

- 1) Clave personal
  - 2) Fecha de la compra
  - 3) Modo de pago
  - 4) Importe de la compra
- y en general, cualquier información requerida.

Para el propósito de nuestro estudio únicamente subrayaremos la importancia del dato: **Importe de la compra**

El sistema utilizado deberá ser capaz de registrar cualquier número de órdenes de compra realizadas por cada miembro mes con mes.

Algunas empresas multinivel registran una sola orden mensual de compra para cada miembro, mientras que otras requieren el registro de diversas órdenes de compra por miembro durante el mes de operaciones.

Sea cual fuere la necesidad, lo importante es contar con un mecanismo automatizado que contabilice el Importe de la compra y lo acumule en algún campo de la plantilla de miembros (véase la sección: **4.2. Registro de la plantilla de miembros (ensamblado de la red)** en este capítulo).

Esta operación puede ser realizada de diversas maneras; las más comunes son:

1. Acumulación del Importe de compra al momento de registrar cada orden.
2. Sumatoria del Importe de compra por cada orden registrada en el archivo de órdenes mensuales. Esta sumatoria es realizada al final del período previo al cálculo de comisiones.

En cualquiera de estos dos casos el origen del dato es el mismo: el Importe de compra registrado en un archivo de base de datos y cualquiera que sea el proceso seleccionado su implementación es sencilla.

El mecanismo seleccionado dependerá de las preferencias del programador. Mientras que la opción 1 nos brinda la obtención del Importe de compra al día, tiene como desventaja la desaplicación de aquellas órdenes que sean canceladas o corregidas por cualquier motivo. La opción 2 nos brinda la ventaja de realizar una sola suma en el archivo de órdenes mensuales, mientras que una desventaja puede ser el consumo de tiempo de sumatoria previo al proceso de cálculo mensual.

Con las ventajas que actualmente brindan los sistemas basados en SQL es recomendable utilizar la opción 1, apoyada con una serie de procedimientos y funciones que permitan la actualización al día de la información.

#### **4.4. Cálculo de Bonos y Comisiones (Recorrido de la red)**

En esta sección veremos la forma de implementar el algoritmo de recorrido del árbol de la organización analizado en el **Capítulo 3**.

Para la implementación del algoritmo será necesario considerar los siguientes aspectos:

1. Se cuenta con un archivo de base de datos que contiene la plantilla de miembros (nodos) de la organización (árbol).
2. El árbol de la organización se encuentra totalmente ensamblado, es decir, cumple con los requisitos de árbol general considerados en el **Capítulo 3** y carece de inconsistencias tales como la presencia de nodos “huérfanos”.
3. La información necesaria para el cálculo se encuentra completa (Importe de compra mensual por miembro, status, etc.) y se dispone de campos auxiliares para registrar los valores calculados.

Según lo analizado en el **Capítulo 2** mediante este proceso se pueden obtener los importes auxiliares necesarios para el cálculo final de bonos y compensaciones. Estos importes pueden ser:

1. Volumen de Grupo formado por los **promotores** patrocinados directamente por un miembro (VG)
2. Volumen Total de Grupo (VTG)
3. Número de miembros patrocinados directamente
4. Cantidad de generaciones

Toda esta información y la que se requiera, puede ser proporcionada con la aplicación del algoritmo de recorrido. Es por esta razón que se recomienda un recorrido **postorder**, el cual realiza la visita de la raíz una vez que se recorren sus subárboles.

##### **4.4.1. Implementación del algoritmo de recorrido del árbol**

El proceso de recorrido sobre el árbol n-ario consiste en realizar una consulta al archivo ordenado por el campo **Paterno** desde el primer registro (raíz) hasta el último. Cada registro contenido en el archivo representa a cada nodo del árbol.

Durante este proceso serán visitados todos y cada uno de los nodos que componen el árbol.

La visita a un nodo consiste en recuperar su información registrada en campos complementarios con el propósito de establecer nueva información que será utilizada en la toma de decisiones durante el proceso.

El algoritmo de recorrido sugerido en el **Capítulo 3** implica un proceso recursivo; sin embargo, en la implementación real tendremos que utilizar un mecanismo no recursivo

de recorrido debido a que un árbol n-ario es dinámico e indefinido en su longitud, aunque finito. Si deseáramos utilizar la función de recursión tendríamos que “subir” el árbol del archivo en un arreglo tipo record en memoria, el cual tendría básicamente la siguiente estructura:

```
Type
  Arbol = record
    Nodo                : integer;
    Paterno             : integer;
    Status              : integer;
    VP                  : float;
    VG                  : float;
    CantidadPromotores  : integer;
    CantidadSupervisores : integer;
    :
    :
    etcétera
  end;
```

Esto podría ser realizado siguiendo estos pasos:

1. Obtención del árbol agrupado por campo Paterno (quizás en un query de resultado).
2. Crear un registro de memoria para cada registro de archivo
3. Realizar el recorrido recursivo en el arreglo de registro de memoria
4. Depositar en el archivo los datos obtenidos en el proceso de recorrido para cada registro de memoria actualizando su correspondiente registro en archivo.

La única ventaja que esta técnica nos brinda es la rapidez del recorrido en el punto 3 ya que se aplica un algoritmo recursivo a un arreglo de memoria.

Sin embargo, como dijimos, realmente desconocemos por anticipado la magnitud que pueda tener el árbol de la organización; si bien puede pensarse en una empresa de 3,000 miembros, cabe la posibilidad de que en un momento dado sea de 3,000,000 o muchos más.

Debemos ser capaces de garantizarle al usuario que no tendrá problemas de estabilidad de su software por falta de recursos de memoria cuando el número de miembros sea demasiado grande.

Debido a ello, es recomendable realizar el recorrido directamente en el archivo de base de datos que contiene la plantilla de miembros o apoyarnos en mecanismos rápidos que el SQL nos puede proveer.

Ciertamente, esta técnica puede resultar un poco más tardada porque el recorrido no se realiza directamente en la memoria, sino en un archivo externo. No obstante, aún sigue siendo recomendable que el usuario dedique unos minutos más una sola vez al mes para obtener sus resultados a que nuestro software sufra una caída por falta de recursos, lo cual

dejaría totalmente paralizado al usuario y haría inservible nuestro programa. Como ya dijimos, actualmente existen herramientas de SQL con procesos en servidor y caché local de la computadora de proceso que pueden brindarnos más rapidez y “compensar” el uso de esta última técnica.

Con esto en mente, el proceso no recursivo de recorrido podría ser realizado en dos pasos:

1. Obtención del árbol agrupado por campo Paterno (quizás en un query de resultado).
2. Recorrido no recursivo sobre el árbol obtenido

Para la implementación del algoritmo de recorrido se ha elegido el lenguaje **Delphi-Pascal** de **Borland** en cualquiera de sus versiones.

El paso 1 puede ser realizado con un componente TQuery cuya sentencia SQL podría ser:

```
select Nodo, Paterno, Status, VP,... from arbol.dbf order by Paterno
```

La ejecución de esta sentencia nos brinda como resultado un árbol ordenado o agrupado por nodo paterno listo para su recorrido ( véase el **Capítulo 3** en la sección **3.3.3. Recorrido en un árbol n-ario a partir de su representación en la computadora**).

Durante el proceso de recorrido podríamos auxiliarnos de otro componente TTable por ejemplo para realizar la actualización (visita) de los nodos del árbol.

#### **4.4.1.1. Elementos básicos utilizados durante el proceso de recorrido**

Durante el recorrido del árbol será necesario contar con los elementos mostrados en la Tabla 4.1:

**Tabla 4.1 Elementos utilizados en la implementación**

Nombre de Elemento	Tipo	Descripción
Arbol	TTable	Componente que utiliza el archivo Arbol.dbf indexado por el campo Paterno sobre el cual se realizará el recorrido
ArbolNodo	integer	Campo del elemento Arbol que identifica la clave del nodo
ArbolPaterno	Integer	Campo del elemento Arbol que identifica la clave de patrocinador del elemento Arbol.Nodo
FindPaterno(nodo)	función	Función que realiza la búsqueda del nodo en el campo Paterno de Arbol. Si la clave de nodo se encuentra, el apuntador de registro es posicionado en el, de lo contrario el apuntador no se mueve.
FindNodo(nodo)	función	Función que realiza la búsqueda del nodo en el campo Nodo de Arbol. Si la clave de nodo se

		encuentra, el apuntador de registro es posicionado en el, de lo contrario el apuntador no se mueve.
Nodo_Raiz	integer	Clave de identificación del nodo raíz del Arbol. Se sugiere que esta sea una constante igual que 1 y que se encuentre identificada quizás en un archivo de configuración externo además del archivo de la plantilla de miembros "Arbol"
Ramas_Raiz	integer	Número de ramas del nodo raíz. Este dato puede ser obtenido con la ejecución de la función <code>ObtieneNumeroDeRamasRaiz</code> la cual utiliza un componente <code>TQuery</code> con la sentencia de consulta SQL: <b>select count(Paterno) as Ramas from Arbol.dbf where Paterno=:Nodo</b>
Paterno_Raiz	integer	Clave de identificación del paterno del nodo raíz del Arbol. Se sugiere que esta sea una constante igual que cero o -1 y que se encuentre identificada quizás en un archivo de configuración externo
VisitaNodo(nodo)	función	Procedimiento que realiza la visita del nodo pasado como parámetro.
XArbolN	TTable	Componente que utiliza el archivo Arbol indexado por el campo <code>Nodo</code> para realizar la visita del nodo deseado
XArbolP	TTable	Componente que utiliza el archivo Arbol indexado por el campo <code>Paterno</code> para realizar la acumulación de valores provenientes del nodo visitado en el componente <code>XArbolN</code>

#### 4.4.1.2. Programación no recursiva para el recorrido del árbol

El siguiente código (Programa 1) es un ejemplo de una implementación en Delphi pascal del algoritmo no recursivo para el recorrido del árbol:

#### Programa 1: Recorrido no recursivo del árbol

```
unit recorrido; {nombre de unidad: recorrido.pas}

interface

uses
  {sección de programas incluidos por default}
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, DBTables, StdCtrls;

type
  TRecorridoForm = class(TForm) {forma que contiene los elementos y rutinas}
    XArbolN: TTable; {archivo arbol.dbf indexado por Nodo para visita}
    XArbolP: TTable; {archivo arbol.dbf indexado por Paterno para visita}
    Iniciar: TButton; {botón que clickea el usuario para realizar cálculo}
    Arbol: TTable; {archivo arbol.dbf sobre el cual se hará el recorrido}
    ArbolNodo: TIntegerField; {campo Nodo de Arbol}
  end;
end;
```

```

ArbolPaterno: TIntegerField; {campo Paterno de Arbol}
RamasRaiz: TQuery;         {consulta para obtener cantidad de ramas de nodo raíz}
procedure Establece_las_bases; {procedimiento que establece el ambiente operativo}
function ObtieneNumeroDeRamasRaiz(const Nodo:integer):integer; {regresa número de ramas
del nodo raíz}
function FindPaterno(const Nodo:integer):boolean; {realiza la búsqueda de Nodo en el
campo Paterno}
function FindNodo(const Nodo:integer):boolean; {realiza la búsqueda de Nodo en el
campo Nodo}
procedure IniciarClick(Sender: TObject); {evento que inicia el proceso de
cálculo}
procedure VisitaNodo(const Nodo:integer); {realiza la visita al Nodo}
private
{ Private declarations }
public
{ Public declarations }
end;

var
  RecorridoForm: TRecorridoForm;

implementation

{$R *.dfm}

function TRecorridoForm.ObtieneNumeroDeRamasRaiz(const Nodo:integer):integer;
begin
  { Función que regresa como resultado el número de ramas pertenecientes al
  nodo raíz del árbol (grado externo).
  El proceso se realiza mediante una sentencia SQL que simplemente cuenta
  el número de ocurrencias del nodo Paterno para los nodos cuyo nodo Paterno
  es el nodo raíz del árbol. }
  RamasRaiz.DatabaseName := 'c:\db';
  RamasRaiz.SQL.Add('select count(Paterno) as Ramas from Arbol.db where Paterno=:Nodo');
  RamasRaiz.ParamByName('Nodo').Value:=Nodo; {valoriza el parámetro llamado 'Nodo'}
  RamasRaiz.Open;
  Result:=RamasRaiz.FieldByName('Ramas').AsInteger;
  RamasRaiz.Close;
end;

function TRecorridoForm.FindPaterno(const Nodo:integer):boolean;
var
  Registro_Actual: TBookmark; {marcador de registro del archivo Arbol}
begin
  { Función que regresa VERDADERO en caso de que el parámetro Nodo
  se encuentre en el campo índice Paterno. Recuérdese que el archivo Arbol
  se encuentra indexado de origen por el campo Paterno.
  Si Nodo no existe como llave del archivo el puntero de éste es regresado
  a su posición guardada en la marca Registro_Actual }
  try
    Registro_Actual:=Arbol.GetBookmark; {marca el actual registro de Arbol}
    Result:=Arbol.FindKey([Nodo]); {encontrar la llave Nodo en campo Paterno}

    {si no fué encontrado el campo, el apuntador de Arbol se regresa a su posición original}
    if not Result then Arbol.GotoBookmark(Registro_Actual);

  finally
    Arbol.FreeBookmark(Registro_Actual); {libera marca de archivo}
  end;
end;

function TRecorridoForm.FindNodo(const Nodo:integer):boolean;
var
  Registro_Actual: TBookmark; {marcador de registro del archivo Arbol}
begin
  { Función que regresa VERDADERO en caso de que el parámetro Nodo
  se encuentre en el campo índice Nodo. Recuérdese que el archivo Arbol
  se encuentra indexado de origen por el campo Paterno, así que el campo
  de indexación es cambiado al campo Nodo.
  Si Nodo no existe como llave del archivo el puntero de éste es regresado

```

```

a su posición guardada en la marca Registro_Actual}
try
  Registro_Actual:=Arbol.GetBookmark; {marca el actual registro de Arbol}
  Arbol.IndexFieldNames := 'Nodo';
  Result:=Arbol.FindKey([Nodo]); {encontrar la llave Nodo en campo Paterno}

  {si no fué encontrado el campo, el apuntador de Arbol se regresa a su posición original}
  if not Result then Arbol.GotoBookmark(Registro_Actual);

finally
  Arbol.IndexFieldNames := 'Paterno'; {regresa indice a su estado original}
  Arbol.FreeBookmark(Registro_Actual); {libera marca de archivo}
end;
end;

procedure TRecorridoForm.VisitaNodo(const Nodo:integer);
begin
  { Función que realiza la visita al nodo pasado como parámetro
  Para realizar la visita habrá que posicionarse en el registro del Nodo
  y realizar la actualización de datos correspondientes }
  XArbolN.FindKey([Nodo]);
  { sección que contiene los criterios de la visita al nodo }
  { :
    }
  { :
    }
end;

procedure TRecorridoForm.Establece_las_bases;
begin
  {Elemento Arbol para realizar el recorrido}
  Arbol.DatabaseName := 'c:\db';
  Arbol.TableName := 'Arbol.dbf';
  Arbol.IndexFieldNames := 'Paterno';
  Arbol.Open;

  {Elemento XArbolN para realizar la visita al nodo}
  XArbolN.DatabaseName := 'c:\db';
  XArbolN.TableName := 'Arbol.dbf';
  XArbolN.IndexFieldNames := 'Nodo';
  XArbolN.Open;

  {Elemento XArbolP para "mandar" datos al nodo paterno del nodo visitado}
  XArbolP.DatabaseName := 'c:\db';
  XArbolP.TableName := 'Arbol.dbf';
  XArbolP.IndexFieldNames := 'Paterno';
  XArbolP.Open;
end;

{ Realiza el proceso de recorrido sobre el árbol así como también el cálculo
de los bonos y comisiones (visita del nodo).
Este proceso es ejecutado cuando el usuario "clickea" el botón Iniciar contenido
en la forma RecorridoForm
}

procedure TRecorridoForm.IniciarClick(Sender: TObject);
var
  Nodo_Raiz : integer; {clave del nodo raíz del árbol}
  Paterno_Raiz : integer; {clave del nodo paterno de la raíz del árbol}
  Ramas_Raiz : integer; {cantidad de ramas (grado externo) del nodo raíz del árbol}
  wRamasRecorridas : integer; {contador de ramificaciones recorridas para el nodo raíz}
  wPaterno : integer; {valor temporal del nodo Paterno en proceso de recorrido}
  wNodo : integer; {valor temporal del nodo en proceso de recorrido}
begin
  Establece_las_bases; {procedimiento que establece el ambiente operativo}
  Nodo_Raiz := 0; {opcionalmente puede ser tomado de un archivo de configuración}
  Paterno_Raiz := -1; {la raíz no tiene nodo paterno}
  wRamasRecorridas := 1;
  wPaterno := ArbolPaterno.Value;
  FindPaterno(Nodo_Raiz);

```

```

wPaterno:=ArbolPaterno.Value;
Ramas_Raiz:=ObtieneNumeroDeRamasRaiz(Nodo_Raiz);
while wRamasRecorridas<=Ramas_Raiz
do begin
  while (wPaterno=ArbolPaterno.Value)
  do begin
    wNodo:=ArbolNodo.Value;
    if FindPaterno(wNodo)
    then wPaterno:=ArbolPaterno.Value
    else
    begin
      VisitaNodo(ArbolNodo.Value);
      if wPaterno=Nodo_Raiz
      then wRamasRecorridas:=wRamasRecorridas+1;
      Arbol.Next;
    end;
  end;
  FindNodo(wPaterno);
  VisitaNodo(ArbolNodo.Value);
  wPaterno:=ArbolPaterno.Value;
  if wPaterno=Nodo_Raiz then wRamasRecorridas:=wRamasRecorridas+1;

  {asegura la detención del proceso si se ha visitado el nodo raíz del árbol}
  if wPaterno=Paterno_Raiz then wRamasRecorridas:=Ramas_Raiz+1;

  Arbol.Next;
end;
end;

end.

```

El Programa 1 descrito anteriormente presenta el recorrido del árbol de la organización de miembros en una forma no recursiva.

Durante el proceso de recorrido cada nodo es visitado por medio de la función **Visita\_Nodo()**. Esta función hasta el momento únicamente posiciona el puntero de otro componente hacia el mismo archivo **Arbol.dbf** en el nodo a visitar:

```
XArbolN.FindKey([Nodo]);
```

A partir de ahí, la operación de visita dependerá en gran medida del esquema de cálculo implementado.

Por ejemplo:

En este procedimiento la visita al nodo deseado puede consistir en:

1. Obtener el volumen personal de ventas (VP) para el nodo visitado.
2. Determinar si en base a su VP el miembro califica para la percepción de bonos.
3. Determinar si el miembro visitado es sujeto a nivelación (ascenso promocional) de acuerdo a las políticas de calificación.
4. Contabilizar para su nodo paterno el volumen de grupo (VG) de éste último.
5. Contar para su nodo paterno el número de generaciones por nivel que ocupa el miembro visitado.

6. Determinar las cantidades no asignadas al nodo paterno por no cumplir este último con las políticas de calificación (Roll-up).

Todas estas operaciones de visita al nodo pueden ser realizadas y muchas más.

#### 4.4.1.3. Cálculo del volumen de grupo (VG) y volumen total de grupo (VTG)

En la sección 2.3.2. *Registro de las compras realizadas (VP, VG y VTG)* se determinaron los importes a considerar previos a la aplicación de la política de pagos y compensaciones a los miembros de la organización. Se ha mencionado también que el volumen personal de compras de cada miembro se puede acumular al momento de registrar una orden a nombre de dicho miembro (véase la sección 4.3. *Registro de las compras realizadas (obtención del VP)*). Entonces, durante el proceso de visita a un nodo podemos asumir que el volumen personal (VP) de dicho miembro ya se encuentra registrado en algún campo del archivo Arbol.dbf; de ahí que únicamente habrá que “mandar” el VP hacia el patrocinador del nodo visitado.

Con esto en mente, podemos escribir el segmento del código de acumulación (Ver Programa 2) para el VG y VTG en el procedimiento **VisitaNodo()**:

#### Programa 2: Visitando el nodo

```

procedure TRecorridoForm.VisitaNodo(const Nodo:integer);
begin
  {----- OPERACIONES PARA EL NODO VISITADO -----}

  xArbolN.FindKey([Nodo]); {Posiciona el apuntador en el nodo a visitar}
  xArbolN.Edit;           {Pone archivo en modo de edición}

  {el volumen del grupo de promotores incluye su compra personal}
  xArbolN.FieldByName('VG').Value := xArbolN.FieldByName('VG').Value +
    xArbolN.FieldByName('VP').Value;

  {el volumen total del grupo incluye su compra personal}
  xArbolN.FieldByName('VTG').Value := xArbolN.FieldByName('VTG').Value +
    xArbolN.FieldByName('VP').Value;

  xArbolN.Post; {Guarda los cambios realizados}

  {----- OPERACIONES PARA EL NODO PATERNO DEL VISITADO -----}

  if xArbolP.FindKey([xArbolN.FieldByName('Paterno').Value]) then
    begin
      xArbolP.Edit;

      {el volumen de grupo de su patrocinador se forma unicamente por el volumen de
      los promotores (nivel 1) que patrocina directamente}
      if xArbolN.FieldByName('Nivel').Value = 1
        then xArbolP.FieldByName('VG').Value := xArbolP.FieldByName('VG').Value +
          xArbolN.FieldByName('VP').Value;

      {calcula volumen total de grupo de su patrocinador}
      xArbolP.FieldByName('VTG').Value := xArbolP.FieldByName('VTG').Value +
        xArbolN.FieldByName('VP').Value;

      if xArbolP.State = dsEdit then xArbolP.Post; {guarda cambios}
    end;
end;

```

Al momento de visitar el nodo (Ver Programa 2) podemos aprovechar el proceso para realizar una visita a su nodo paterno (instrucción: `xArbolP.FindKey([xArbolN.FieldByName('Paterno').Value])`), puesto que las políticas de pago para éste último están en función del primero.

#### 4.4.1.4. *Cálculo de comisiones del Promotor (descuentos y diferencias por descuento)*

En esta sección se verá la manera de implementar los mecanismos del plan de pagos y compensaciones referentes a los descuentos realizados sobre el volumen personal de compra (VP) por cada miembro, los cuales fueron descritos en el **Capítulo 2**, sección: **2.4 Niveles y Posiciones (plan de pagos)**.

Para tal propósito se utilizan las tablas de descuentos y bonos programados en el procedimiento correspondiente (recuérdese que todo miembro de la red es a su vez un promotor independientemente del nivel que tenga; de esta manera, un supervisor es a su vez un promotor, un gerente es a su vez un promotor y un supervisor, un director es a su vez un promotor, supervisor y gerente, etc.).

La rutina de programación se ha desglosado en diferentes procedimientos de acuerdo al nivel que ocupa el nodo del miembro que es visitado.

De esta manera, el procedimiento `VisitaNodo()` (Ver Programa 2) que realiza el cálculo de descuentos y diferencias de descuentos queda programado de la manera como lo muestra el Programa 3:

#### **Programa 3: Visitando el nodo (incorporando los descuentos)**

```

procedure TRecorridoForm.VisitaNodo(const Nodo:integer);
var PjeDescuento:double; {porcentaje de descuento sobre VP}
    Disponible:double; {porcentaje a repartir en ascendencia}
begin
  {----- OPERACIONES PARA EL NODO VISITADO -----}

  xArbolN.FindKey([Nodo]); {Posiciona el apuntador en el nodo a visitar}
  xArbolN.Edit;           {Pone archivo en modo de edición}

  {el volumen del grupo de promotores incluye su compra personal}
  xArbolN.FieldByName('VG').Value := xArbolN.FieldByName('VG').Value +
    xArbolN.FieldByName('VP').Value;

  {el volumen total del grupo incluye su compra personal}
  xArbolN.FieldByName('VTG').Value := xArbolN.FieldByName('VTG').Value +
    xArbolN.FieldByName('VP').Value;

  {determina el porcentaje de descuento sobre VP del nodo visitado}
  PjeDescuento:= Bono_Descuento(xArbolN.FieldByName('Nivel').Value,
    xArbolN.FieldByName('VP').Value);
  xArbolN.FieldByName('BonoDescuento').Value :=
    xArbolN.FieldByName('VP').Value * PjeDescuento;

```

```

xArbolN.Post; {Guarda los cambios realizados}

{----- OPERACIONES PARA EL NODO PATERNO DEL VISITADO -----}

if xArbolP.FindKey([xArbolN.FieldByName('Paterno').Value]) then
begin
  xArbolP.Edit;

  {el volumen de grupo de su patrocinador se forma unicamente por el volumen de
  los promotores (nivel 1) que patrocina directamente}
  if xArbolN.FieldByName('Nivel').Value = 1
    then xArbolP.FieldByName('VG').Value := xArbolP.FieldByName('VG').Value +
      xArbolN.FieldByName('VP').Value;

  {calcula volumen total de grupo de su patrocinador}
  xArbolP.FieldByName('VTG').Value := xArbolP.FieldByName('VTG').Value +
    xArbolN.FieldByName('VP').Value;

  {determina porcentaje disponible a distribuir en la ascendencia}
  Disponible := 0.20-PjeDescuento;
  if (Disponible>0)
    then
      Bono_DiferenciaDscto(PjeDescuento,Disponible); {calcula diferencias dsctos}
  if xArbolP.State = dsEdit then xArbolP.Post; {guarda cambios}
end;
end;

```

La función `Bono_Descuento()` del Programa 4 simplemente determina el porcentaje de descuento correspondiente según la ubicación del volumen personal de compra (VP) en el rango de importes:

#### Programa 4: Bonos y descuentos

```

function TRecorridoForm.Bono_Descuento( const Nivel:integer; VP: double):double;
begin
  { Dependiendo el nivel ocupado en el plan de pagos, calcula el porcentaje
  de descuento correspondiente para el miembro requerido.
  Los rangos de VP podrían ser tomados de un archivo externo para que estos
  puedan ser editables por el usuario. }
  Result:=0;
  if (Nivel=1) {según las políticas de pago, solo los nivel 1 (promotores) pueden}
  then {tener descuentos menores al 20%}
  begin
    if (VP>=5001) and (VP<=10000)
    then Result:=0.5 else
    if (VP>=10001) and (VP<=15000)
    then Result:=0.10 else
    if (VP>=15001)
    then Result:=0.15;
  end
  else Result:=0.20; {cualquier nivel <>1 tiene descuento del 20%}
end;

```

Como puede verse, además del volumen de grupo (VG) y volumen total de grupo (VTG) ahora el programa calcula también el porcentaje de descuento sobre su volumen personal (VP) para el nodo visitado. En caso de que el nivel del miembro nodo visitado sea *promotor*, existen “diferencias” de descuentos que habrá que distribuir a través de su línea de ascendencia. Esto es realizado por la función `Bono_DiferenciaDscto()` del Programa 5:

## Programa 5: Diferencias de descuentos

```

procedure TRecorridoForm.Bono_DiferenciaDscto(var PjeDescuento, Disponible:double);
var PjeAsignado:double; {Porcentaje que corresponde al descuento otorgado}
    PjeDiferencia:double; {Diferencia entre los descuentos otorgados a nodo y paterno}
begin
while (Disponible>0) do {mientras exista porcentaje disponible a distribuir...}
begin
    {determina porcentaje correspondiente al nodo}
    PjeAsignado:=Bono_Descuento( xArbolP.FieldByName('Nivel').Value,
        xArbolP.FieldByName('VP').Value);
    {determina si existe diferencias entre el nodo inferior y actual}
    PjeDiferencia:=PjeAsignado-PjeDescuento;
    if PjeDiferencia>0 then
    begin
        xArbolP.Edit;
        {otorga diferencia a nodo actual}
        xArbolP.FieldByName('BonoDiferenciaDscto').Value :=
            xArbolP.FieldByName('BonoDiferenciaDscto').Value +
            (xArbolN.FieldByName('VP').Value * PjeDiferencia);
        xArbolP.Post;
        Disponible := Disponible - PjeDiferencia;
    end;
    PjeDescuento:=PjeAsignado;
    {si aún existe porcentaje disponible para distribuir,
    "sube" un nivel en ascendencia}
    if Disponible>0 then xArbolP.FindKey([xArbolP.FieldByName('Paterno').Value]);
end;
end;

```

#### 4.4.1.5. Cálculo de comisiones del Supervisor

Según lo analizado en el **Capítulo 2**, sección: **2.4.2. Comisiones del Supervisor**, el cálculo de comisiones para el nivel supervisor está sujeto a dos condiciones. Para ser considerado supervisor calificado, el supervisor tiene que haber logrado en un mes calendario:

1. un volumen de compra personal por \$250 (VP)
2. un volumen de grupo de \$6,000 (VG)

Ambas condiciones son sencillas de implementar con tan solo agregar un bloque condicionante al código de cálculo al momento de determinar si el miembro es sujeto al bono de supervisor:

```
If (xArbolN.FieldByName('VP').Value >= 250) and
(xArbolN.FieldByName('VG').Value >= 6000) then
  Begin
  :
  :
  End;
```

De acuerdo al plan de pagos descrito en la sección referida, los bonos correspondientes al supervisor se integran por los siguientes montos:

- a) plan de compensaciones del **promotor**. El **supervisor** es a su vez un **promotor**.
- b) 5% adicional en el descuento sobre su volumen personal (VP) haciendo un total del 20%.
- c) Las diferencias entre el 20% y el porcentaje obtenido por sus **promotores** patrocinados directamente por él.
- d) 3% sobre su volumen personal de compras (VP).
- e) 3% sobre su volumen de grupo formado por todos los **promotores** patrocinados directamente por él (VG)
- f) 3% sobre el volumen de grupo (VG) de hasta 3 generaciones de **supervisores** calificados.

La implementación del cálculo para los conceptos a, b y c ya se encuentra incorporada en la sección **4.4.1.4. Cálculo de comisiones del promotor (descuentos y diferencias por descuento)** de este capítulo.

Los conceptos d y e se pueden implementar de la manera como lo muestra el Programa 6:

#### Programa 6: Comisiones del supervisor

```
procedure TRecorridoForm.VisitaNodo(const Nodo:integer);
var :
  :
  BonoSupervisorDisponible : double; {bono para supervisor}
begin
  {----- OPERACIONES PARA EL NODO VISITADO -----}
```

```

xArbolN.FindKey([Nodo]); {Posiciona el apuntador en el nodo a visitar}
xArbolN.Edit;      {Pone archivo en modo de edición}
:
:
{bonos como SUPERVISOR}
BonoSupervisorDisponible := 0;
if xArbolN.FieldByName('Nivel').AsInteger in [2,3,4,5] then
begin
  BonoSupervisorDisponible := xArbolN.FieldByName('VG').Value * 0.03;
  if (xArbolN.FieldByName('VP').Value >= 250) and
    (xArbolN.FieldByName('VG').Value >= 6000) then
  begin
    xArbolN.FieldByName('BonoSupervisor').Value :=
      xArbolN.FieldByName('BonoSupervisor').Value + BonoSupervisorDisponible;
  end;
end;

xArbolN.Post; {Guarda los cambios realizados}

{----- OPERACIONES PARA EL NODO PATERNO DEL VISITADO -----}

if xArbolP.FindKey([xArbolN.FieldByName('Paterno').Value]) then
begin
  xArbolP.Edit;
  :
  {determina porcentaje disponible a distribuir en la ascendencia}
  :
  {Bonos de SUPERVISOR. Manda en línea de ascendencia}
  if BonoSupervisorDisponible > 0
  then SubeBonoSupervisor(BonoSupervisorDisponible);

  if xArbolP.State = dsEdit then xArbolP.Post; {guarda cambios}
end;
end;

```

Para realizar la implementación del concepto **f** se utiliza el procedimiento `SubeBonoSupervisor()` del Programa 7, el cual “sube” en ascendencia el bono disponible para todos los supervisores calificados que encuentre solo con la cuenta de 3 generaciones de supervisores calificados:

### Programa 7: Cálculo de la 3ª generación

```

procedure TRecorridoForm.SubeBonoSupervisor(const BonoSupervisorDisponible:double);
var GeneracionSupervisor: integer; {contador de generaciones para supervisor}
begin
  GeneracionSupervisor:=1;
  {asegura posicionamiento en nodo paterno del visitado}
  xArbolP.FindKey([xArbolN.FieldByName('Paterno').Value]);
  {hacer mientras el número de generaciones sea <= 3 y no se
  sobrepase el inicio del árbol}
  while (GeneracionSupervisor <= 3) and
    not xArbolP.BOF do
  begin
    {El miembro es supervisor si su nivel se encuentra
    en el rango 2..5 (supervisor, gerente, director, ejecutivo),
    y es calificado si su VP>=250 y su VG>=6000}
    if (xArbolP.FieldByName('Nivel').AsInteger in [2,3,4,5]) and
      (xArbolP.FieldByName('VP').Value >= 250) and
      (xArbolP.FieldByName('VG').Value >= 6000) then
    begin
      xArbolP.FieldByName('BonoSupervisor').Value :=
        xArbolP.FieldByName('BonoSupervisor').Value + BonoSupervisorDisponible;
      GeneracionSupervisor := GeneracionSupervisor+1;
    end;
    {"sube" en línea de ascendencia buscando supervisores calificados}
  end;
end;

```

```

    xArbolP.FindKey([xArbolP.FieldByName('Paterno').Value]);
end;
end;

```

Nótese que dicho procedimiento evalúa la factibilidad de asignar el bono en función de los criterios de calificación. Uno de estos criterios está condicionado por el número de generaciones de supervisores calificados. Para ello utilizamos el contador generacional `GeneracionSupervisor`. La cuenta de generaciones se realiza, no de arriba hacia abajo como lógicamente pudiera pensarse, sino de abajo hacia arriba “subiendo” el bono en línea de ascendencia.

Durante el recorrido en la línea de ascendencia para “subir” el bono a todos los supervisores que se encuentren calificados nos enfrentamos a un problema. Nótese que una condición para la asignación del bono implica el cálculo previo del volumen de grupo (VG) de cada miembro visitado por el procedimiento `SubeBonoSupervisor()` del Programa 7. Sin embargo, como estamos realizando el recorrido ascendente en una sola línea no podemos saber en ese momento el valor final del VG para cada nodo visitado, ya que existen otras líneas de descendencia paralelas a la que estamos recorriendo compuestas por nodos que aún no han sido visitados.

La solución consiste en recorrer dos veces el árbol:

1. Primer recorrido: para obtener el volumen de grupo (VG)
2. Segundo recorrido: para realizar el cálculo de bonos

De esta manera obtenemos los datos primarios que incluyen el VG de cada nodo y posteriormente los utilizamos para el cálculo de los bonos.

El primer recorrido lo podemos aprovechar para realizar el cálculo de aquellos bonos que no se requieran “subir” en línea de ascendencia y que en ésta línea se utilice el VG de cada nodo que la compone. Por ejemplo, en este recorrido bien podríamos calcular los bonos correspondientes al nivel **promotor**, que consisten en los descuentos y diferencias de descuentos, ya que si bien “suben” las diferencias en los porcentajes de descuento, en ningún nodo de la línea de ascendencia es requerido el VG para condicionar el pago del bono.

#### **4.4.1.6. Cálculo de comisiones del Gerente**

Según lo analizado en el **Capítulo 2**, sección: **2.4.3. Comisiones del Gerente**, el cálculo de comisiones para el nivel **gerente** está sujeto a dos condiciones. Para ser considerado gerente calificado, el **gerente** tiene que haber logrado en un mes calendario:

1. un volumen de compra personal por \$250 (VP)
2. un volumen de grupo de \$12,000 (VG)

Ambas condiciones son sencillas de implementar con tan solo agregar un bloque condicionante al código de cálculo al momento de determinar si el miembro es sujeto al bono de **gerente**:

```
If (xArbolN.FieldByName('VP').Value >= 250) and
(xArbolN.FieldByName('VG').Value >= 12000) then
  Begin
  :
  :
  End;
```

De acuerdo al plan de pagos descrito en la sección referida, los bonos correspondientes al gerente se integran por los siguientes montos:

- a) plan de compensaciones del **promotor**. El **gerente** es a su vez un **promotor**.
- b) plan de compensaciones del **supervisor**. El **gerente** es a su vez un **supervisor**.
- c) 5% adicional en el descuento sobre su volumen personal (VP) haciendo un total del 20%.
- d) Las diferencias entre el 20% y el porcentaje obtenido por sus promotores patrocinados por él.
- e) 10% sobre su volumen personal (VP).
- f) 10% sobre su volumen de grupo formado por todos los **promotores** patrocinados directamente por él (VG).
- g) 10% sobre el volumen de grupo (VG) de hasta 3 generaciones de **supervisores** calificados.
- h) 10% sobre el volumen de grupo (VG) de sus **gerentes** no calificados en cada “down line”.

La implementación del cálculo para los conceptos a, b, c y d ya se encuentra referida en la sección **4.4.1.4. Cálculo de comisiones del promotor (descuentos y diferencias por descuento)** y **4.4.1.5. Cálculo de comisiones del supervisor** de este capítulo, ya que el **gerente** es a su vez un **promotor** y también un **supervisor**.

Los conceptos **e** y **f** se pueden implementar como lo muestra el Programa 8:

### Programa 8: Bonos del gerente

```
procedure TRecorridoForm.VisitaNodo(const Nodo:integer);
var :
  :
  BonoGerenteDisponible : double; {bono para gerente}
begin
  {----- OPERACIONES PARA EL NODO VISITADO -----}

  xArbolN.FindKey([Nodo]); {Posiciona el apuntador en el nodo a visitar}
  xArbolN.Edit;           {Pone archivo en modo de edición}
  :
  {bonos como SUPERVISOR}
  :

  {bonos como GERENTE}
  if xArbolN.FieldByName('Nivel').AsInteger in [3,4,5] then
  begin
    BonoGerenteDisponible := xArbolN.FieldByName('VG').Value * 0.10;
    if (xArbolN.FieldByName('VP').Value >= 250) and
```

```

(xArbolN.FieldByName('VG').Value >= 12000) then
begin
  xArbolN.FieldByName('BonoGerente').Value :=
    xArbolN.FieldByName('BonoGerente').Value + BonoGerenteDisponible;
  BonoGerenteDisponible := 0;
end;
end;

xArbolN.Post; {Guarda los cambios realizados}

{----- OPERACIONES PARA EL NODO PATERNO DEL VISITADO -----}

if xArbolP.FindKey([xArbolN.FieldByName('Paterno').Value]) then
begin
  xArbolP.Edit;
  :
  {Bonos de SUPERVISOR. Manda en línea de ascendencia}
  :
  {Bonos de GERENTE. Manda en línea de ascendencia}
  if BonoGerenteDisponible > 0
  then SubeBonoGerente(BonoGerenteDisponible);

  if xArbolP.State = dsEdit then xArbolP.Post; {guarda cambios}
end;
end;

```

Para realizar la implementación del concepto **h** se utiliza el procedimiento `SubeBonoGerente()`, el cual “sube” en ascendencia el bono disponible de aquel **gerente** no calificado hasta encontrar un **gerente** calificado que perciba dicho bono (Ver Programa 9):

### Programa 9: Bonos de gerentes no calificados

```

procedure TRecorridoForm.SubeBonoGerente(const BonoGerenteDisponible:double);
begin
  {asegura posicionamiento en nodo paterno del visitado}
  xArbolP.FindKey([xArbolN.FieldByName('Paterno').Value]);
  {busca gerente calificado mientras no se sobrepase el inicio del árbol}
  while not xArbolP.BOF do
  begin
    {El miembro es gerente si su nivel se encuentra
    en el rango 3..5 (gerente, director, ejecutivo),
    y es calificado si su VP>=250 y su VG>=12000}
    if (xArbolP.FieldByName('Nivel').AsInteger in [3,4,5]) and
      (xArbolP.FieldByName('VP').Value >= 250) and
      (xArbolP.FieldByName('VG').Value >= 12000) then
    begin
      xArbolP.FieldByName('BonoGerente').Value :=
        xArbolP.FieldByName('BonoGerente').Value + BonoGerenteDisponible;
      Exit; {una vez asignado el bono interrumpe ciclo}
    end;
    {"sube" en línea de ascendencia buscando un gerente calificado}
    xArbolP.FindKey([xArbolP.FieldByName('Paterno').Value]);
  end;
end;

```

Para implementar el concepto **g** debemos identificar el nivel que ocupa el nodo visitado; si éste es **supervisor**, **gerente**, **director** o **ejecutivo**, habrá que captar el monto del 10% sobre su **VG** (sin importar si éste se encuentra calificado o no) y “subirlo” en línea de ascendencia hasta 3 generaciones de **supervisores** calificados (Ver Programa 10):

## Programa 10: Bono sobre la 3ª generación de supervisores

```

procedure TRecorridoForm.VisitaNodo(const Nodo:integer);
var :
:
:   BonoSupervisorParaGerente : double; {bono de supervisor para gerente}
begin
{----- OPERACIONES PARA EL NODO VISITADO -----}
:
:   {bonos como SUPERVISOR}
:
:   {bonos como GERENTE}
:
:   {bono de SUPERVISOR para GERENTE}
:   BonoSupervisorParaGerente := 0;
:   if xArbolN.FieldByName('Nivel').AsInteger in [2,3,4,5]
:   then
:       BonoSupervisorParaGerente := xArbolN.FieldByName('VG').Value * 0.10;
:
:   {----- OPERACIONES PARA EL NODO PATERNO DEL VISITADO -----}

if xArbolP.FindKey([xArbolN.FieldByName('Paterno').Value]) then
begin
:
:   {Bonos de GERENTE. Manda en línea de ascendencia}
:
:   {Bono de SUPERVISOR para GERENTE. Manda en línea de ascendencia}
:   if BonoSupervisorParaGerente > 0
:       then SubeBonoSupervisorParaGerente(BonoSupervisorParaGerente);
:
:   end;
end;
end;

```

El procedimiento `SubeBonoSupervisorParaGerente()` toma el bono disponible `BonoSupervisorParaGerente` y lo “sube” hasta 3 generaciones de **supervisores** calificados asignándolo únicamente a los niveles **gerente**, **director** y **ejecutivo**, siempre que estos se encuentren calificados como **gerente** (Ver Programa 11):

## Programa 11: “Sube bono del gerente”

```

procedure TRecorridoForm.SubeBonoSupervisorParaGerente(const BonoSupervisorParaGerente:double);
var GeneracionSupervisor: integer; {contador de generaciones para supervisor}
begin
GeneracionSupervisor:=1;
{asegura posicionamiento en nodo paterno del visitado}
xArbolP.FindKey([xArbolN.FieldByName('Paterno').Value]);
{hacer mientras el número de generaciones sea <= 3 y no se
sobrepase el inicio del árbol}
while (GeneracionSupervisor <= 3) and
not xArbolP.BOF do
begin
{El miembro es supervisor si su nivel se encuentra
en el rango 2..5 (supervisor, gerente, director, ejecutivo),
y es calificado si su VP>=250 y su VG>=6000}
if (xArbolP.FieldByName('Nivel').AsInteger in [2,3,4,5]) and
(xArbolP.FieldByName('VP').Value >= 250) and
(xArbolP.FieldByName('VG').Value >= 6000) then
begin
{siendo supervisor calificado cuenta como generación}
GeneracionSupervisor := GeneracionSupervisor+1;

{El miembro es gerente si su nivel se encuentra
en el rango 3..5 (gerente, director, ejecutivo),
y es calificado si su VP>=250 y su VG>=12000}

```

```

if (xArbolP.FieldByName('Nivel').AsInteger in [3,4,5]) and
(xArbolP.FieldByName('VP').Value >= 250) and
(xArbolP.FieldByName('VG').Value >= 12000) then
begin
xArbolP.FieldByName('BonoGerente').Value :=
xArbolP.FieldByName('BonoGerente').Value + BonoSupervisorParaGerente;
end;

end;
{"sube" en línea de ascendencia buscando supervisores calificados}
xArbolP.FindKey([xArbolP.FieldByName('Paterno').Value]);
end;
end;

```

Nótese que cuando no existe algún **gerente** calificado en la línea de ascendencia, el procedimiento simplemente “cuenta” el número de generaciones de los **supervisores** calificados para asegurarse que la repartición del bono de **gerente** no rebase las 3 generaciones. Cuando un **supervisor** no se encuentra calificado éste no es contado como generación y el contador de generaciones `GeneracionSupervisor` no es incrementado, logrando así que el recorrido sobre la línea de ascendencia “suba” más niveles en el árbol en la búsqueda de **gerentes** calificados.

#### 4.4.1.7. Cálculo de comisiones del Director

Según lo analizado en el **Capítulo 2**, sección: **2.4.4. Comisiones del Director**, el cálculo de comisiones para el nivel **director** está sujeto a dos condiciones. Para ser considerado **director** calificado, el **director** tiene que haber logrado en un mes calendario:

1. un volumen de compra personal por \$250 (VP)
2. un volumen de grupo de \$25,000 (VG)

De acuerdo al plan de pagos descrito en la sección referida, los bonos correspondientes al **director** se integran por los siguientes montos:

- a) plan de compensaciones del **promotor**. El **director** es a su vez un **promotor**.
- b) plan de compensaciones del **supervisor**. El **director** es a su vez un **supervisor**.
- c) plan de compensaciones del **gerente**. El **director** es a su vez un **gerente**.
- d) 5% adicional en el descuento sobre su volumen personal (VP) haciendo un total del 20%.
- e) Las diferencias entre el 20% y el porcentaje obtenido por sus **promotores** patrocinados por él.
- f) 6% sobre su volumen personal (VP).
- g) 6% sobre su volumen de grupo (VG).

La implementación del cálculo para los conceptos a, b, c, d y e ya se encuentra referida en la sección **4.4.1.4. Cálculo de comisiones del promotor (descuentos y diferencias por descuento)**, **4.4.1.5. Cálculo de comisiones del supervisor** y **4.4.1.6. Cálculo de**

**comisiones del gerente** de este capítulo, ya que el **director** es a su vez un **promotor**, **supervisor** y también un **gerente**.

Los conceptos **f** y **g** se pueden implementar como lo muestra el Programa 12:

### Programa 12: Comisiones del director

```

procedure TRecorridoForm.VisitaNodo(const Nodo:integer);
var :
:
begin
{----- OPERACIONES PARA EL NODO VISITADO -----}
:
{bono de SUPERVISOR para GERENTE}
:
{bono como DIRECTOR}
if (xArbolN.FieldByName('Nivel').AsInteger in [4,5]) and
(xArbolN.FieldByName('VP').Value >= 250) and
(xArbolN.FieldByName('VG').Value >= 25000)
then
xArbolN.FieldByName('BonoDirector').Value :=
xArbolN.FieldByName('VG').Value * 0.06;
:
:
end;

```

#### 4.4.1.8. Cálculo de comisiones del Ejecutivo

Según lo analizado en el **Capítulo 2**, sección: **2.4.5. Comisiones del Ejecutivo**, el cálculo de comisiones para el nivel **ejecutivo** está sujeto a tres condiciones. Para ser considerado **ejecutivo** calificado, el **ejecutivo** tiene que haber logrado en un mes calendario:

1. un volumen de compra personal por \$250 (VP)
2. un volumen de grupo de \$300,000 (VG)
3. patrocinar directamente (línea frontal) a 5 **directores** en líneas separadas.

De acuerdo al plan de pagos descrito en la sección referida, los bonos correspondientes al ejecutivo se integran por los siguientes montos:

- a) plan de compensaciones del **promotor**. El **ejecutivo** es a su vez un **promotor**.
- b) plan de compensaciones del **supervisor**. El **ejecutivo** es a su vez un **supervisor**.
- c) plan de compensaciones del **gerente**. El **ejecutivo** es a su vez un **gerente**.
- d) plan de compensaciones del **director**. El **ejecutivo** es a su vez un **director**.
- e) 5% adicional en el descuento sobre su volumen personal (VP) haciendo un total del 20%.
- f) Las diferencias entre el 20% y el porcentaje obtenido por sus **Promotores** patrocinados por él.
- g) 3% sobre el volumen de grupo (VG) de su 4<sup>a</sup> generación de **supervisores** calificados

- h) Si en su volumen total de grupo (VTG) logra el monto de \$500,000 en un mes calendario recibe una cuenta de gastos de \$15,000.

La implementación del cálculo para los conceptos a, b, c, d, e y f ya se encuentra referida en la sección **4.4.1.4. Cálculo de comisiones del promotor (descuentos y diferencias por descuento)**, **4.4.1.5. Cálculo de comisiones del supervisor**, **4.4.1.6. Cálculo de comisiones del gerente** y **4.4.1.7. Cálculo de comisiones del Director** de este capítulo, ya que el **ejecutivo** es a su vez un **promotor, supervisor, gerente** y también un **director**.

Para el cálculo de los conceptos **g** y **h** es necesario contar con la cantidad de **directores** que un **ejecutivo** patrocina directamente en su línea frontal. Este dato puede ser obtenido en el primer recorrido del árbol al momento de realizar la visita del nodo paterno (Ver Programa 13):

### Programa 13: Comisiones del ejecutivo

```

procedure TRecorridoForm.VisitaNodo(const Nodo:integer);
var :
:
begin
:
{----- OPERACIONES PARA EL NODO PATERNO DEL VISITADO -----}

if xArbolP.FindKey([xArbolN.FieldByName('Paterno').Value]) then
begin
:
{realiza conteo de directores en nodo paterno cuando el nivel
del nodo visitado es Director}
if xArbolN.FieldByName('Nivel').Value = 4
then xArbolP.FieldByName('CantDirectores').Value :=
xArbolP.FieldByName('CantDirectores').Value + 1;
:
end;
end;

```

Para implementar el concepto **g** que calcula un bono adicional como **ejecutivo** sobre la 4ª generación de **supervisores** calificados es necesario realizar un ajuste en el procedimiento ya descrito con anterioridad (Ver Programa 7: Cálculo de la 3ª generación) para los **supervisores**. Este ajuste se muestra en el Programa 14:

### Programa 14: Adaptación del Programa 7 para el Cálculo de la 4ª generación para el ejecutivo

```

procedure TRecorridoForm.SubBonoSupervisor(const BonoSupervisorDisponible:double);
var GeneracionSupervisor: integer; {contador de generaciones para supervisor}
begin
GeneracionSupervisor:=1;
{asegura posicionamiento en nodo paterno del visitado}
xArbolP.FindKey([xArbolN.FieldByName('Paterno').Value]);
{hacer mientras el número de generaciones sea <= 4 y no se
sobrepase el inicio del árbol}
while (GeneracionSupervisor <= 4) and
not xArbolP.BOF do

```

```

begin
  if { El miembro es supervisor si su nivel se encuentra
      en el rango 2..5 (supervisor, gerente, director, ejecutivo),
      y es calificado si su VP>=250 y su VG>=6000
      El bono se paga solo hasta la 3er generación }
    (
      (xArbolP.FieldByName('Nivel').AsInteger in [2,3,4,5]) and
      (xArbolP.FieldByName('VP').Value >= 250) and
      (xArbolP.FieldByName('VG').Value >= 6000) and
      (GeneracionSupervisor <= 3)
    )
    OR
    { El miembro es ejecutivo si su nivel = 5
      y es calificado si su VP>=250 y su VG>=300000 y
      la cantidad de directores patrocinados directamente
      por él es >= 5
      El bono se paga solo en la 4a generación }
    (
      (xArbolP.FieldByName('Nivel').Value = 5) and
      (xArbolP.FieldByName('VP').Value >= 250) and
      (xArbolP.FieldByName('VG').Value >= 300000) and
      (xArbolP.FieldByName('CantDirectores').Value >= 5) and
      (GeneracionSupervisor = 4)
    )
  then
    begin
      xArbolP.FieldByName('BonoSupervisor').Value :=
        xArbolP.FieldByName('BonoSupervisor').Value + BonoSupervisorDisponible;
      GeneracionSupervisor := GeneracionSupervisor+1;
    end;
    {"sube" en línea de ascendencia buscando supervisores calificados}
    xArbolP.FindKey([xArbolP.FieldByName('Paterno').Value]);
  end;
end;

```

Nótese que el procedimiento ahora explora hasta la 4ª generación de **supervisores** y en caso de encontrar un **ejecutivo** que califique, calcula el bono correspondiente.

Por último, implementaremos el concepto **h** que realiza el cálculo de la cuenta de gastos para el ejecutivo calificado. Este es muy sencillo de programar adaptando nuevamente el procedimiento `VisitaNodo()` mostrado en el Programa 15:

### Programa 15: Bono adicional para el ejecutivo

```

procedure TRecorridoForm.VisitaNodo(const Nodo:integer);
var :
  :
begin
  {----- OPERACIONES PARA EL NODO VISITADO -----}
  :
  {bono como DIRECTOR}
  :
  {bono como EJECUTIVO}
  if (xArbolN.FieldByName('Nivel').AsInteger = 5) and
    (xArbolN.FieldByName('VP').Value >= 250) and
    (xArbolN.FieldByName('VG').Value >= 500000) and
    (xArbolN.FieldByName('CantDirectores').Value >= 5)
  then
    xArbolN.FieldByName('BonoEjecutivo').Value := 15000;
  :
  :
end;

```

#### 4.4.1.9. El conteo de generaciones y Roll-up en miembros no calificados

En el **Capítulo 2**, secciones **2.5 El conteo de generaciones** y **2.6 Roll-up en miembros no calificados** se describió el criterio a seguir para realizar el conteo generacional de miembros calificados durante el proceso de recorrido en el árbol.

En ésta sección solo haremos referencia a la manera implícita en que estos conceptos han sido implementados en los procedimientos correspondientes.

En el procedimiento `VisitaNodo()` (Ver los Programas: 3,6,8,10,12 y 15) podemos observar no solamente la visita al nodo correspondiente durante el recorrido, sino además la exploración en línea ascendente sobre el nodo paterno, abuelo, bisabuelo, etcétera; siguiendo según el número de generaciones correspondientes al contexto de cálculo.

Por ejemplo, una vez que se visita el nodo  $x$ , la sección:

```
{Bonos de SUPERVISOR. Manda en línea de ascendencia}
if BonoSupervisorDisponible > 0
then SubeBonoSupervisor(BonoSupervisorDisponible);
```

realiza la ubicación en el nodo paterno con el propósito de comenzar una nueva línea de recorrido a fin de “buscar” a los miembros de la generación ascendente partícipes de los porcentajes del monto (VG) logrado por  $x$ . Esta exploración es realizada por el procedimiento `SubeBonoSupervisor()` de los Programas:7 y 14 utilizando para ello el contador de generación `GeneracionSupervisor`.

El conteo generacional se realiza “de abajo hacia arriba” y el contador `GeneracionSupervisor` únicamente es incrementado si el nodo de la línea de ascendencia visitado cumple con las condiciones de calificación. En caso contrario, dicho nodo no es contado como generación de sus ascendentes, dando oportunidad de que el monto de participación `BonoSupervisorDisponible` del nodo  $x$  llegue a más profundidad en línea de ascendencia.

Es precisamente en éste momento cuando se presenta el llamado “*roll-up*” o “*compactación de niveles*”. Al encontrarse durante la exploración de la línea de ascendencia algún miembro no calificado, el proceso “rola hacia arriba” el monto que hubiera correspondido a él sobre  $x$  hacia otro miembro que sí califique. En el momento en que se encuentra ese miembro calificado merecedor del bono, el contador generacional es incrementado y el proceso se detiene cuando se ha llegado al límite de generaciones participantes de dicho bono o bien, hasta el nodo raíz del árbol.

Lo mismo sucede en el procedimiento `SubeBonoSupervisorParaGerente()` del Programa 11, el cual realiza el cálculo del 10% sobre el volumen de grupo (VG) de hasta 3 generaciones de **supervisores** calificados.

## CONCLUSIONES Y RECOMENDACIONES

Existe un mayor número de empresas que intentan orientarse hacia el esquema multinivel dado el éxito que este modelo de comercialización ha demostrado.

La empresa multinivel ofrece el mejor sistema de organización de un grupo de ventas porque no solamente se forma por vendedores de un producto, sino además éstos son consumidores de tal producto, reclutadores de personal de ventas y formadores de organizaciones de ventas. Esta organización de ventas solo es posible cuando es definida por una estructura ramificada de miembros, es decir, un modelo de árbol general, n-ario o lo que los administradores de ventas llaman: multinivel.

Para los administradores de redes de mercadeo multinivel resulta muy difícil implementar el modelo en la organización si no se manejan al menos en el aspecto teórico, los conceptos fundamentales de las estructuras de árboles generales o n-arios. Y más difícil aún es el diseñar un software que logre la implementación del modelo sin entender estos conceptos.

Se ha observado que para los administradores de ventas resulta muy difícil hacerse entender cuando de presentación del esquema de pagos a su personal de ventas se trata. En la medida como manejen la parte teórica de la estructura de árboles les resultará más sencillo presentar dicho esquema.

Para que una red de mercadeo directo sea verdaderamente multinivel debe permitir su crecimiento no solo en profundidad, sino también en su amplitud, es decir, que una persona pueda patrocinar a cuantos miembros desee.

El modelo matemático de árboles generales o n-arios puede ser completamente implementado en cualquier red de mercadeo directo. No es necesario restringir a la empresa al modelo binario únicamente por que exista mayor información técnica para este tipo de árboles.

Para realizar la implementación del modelo en la computadora será necesario, en la mayoría de los casos, realizar una serie de adaptaciones al modelo utilizado, sobre todo en el proceso de recorrido del árbol.

Regularmente, en la implementación habrá que considerarse los siguientes aspectos:

1. La relación miembro-patrocinador será definida por un arreglo unidimensional representado en la computadora por los campos *nodo-paterno*.
2. Se debe registrar al menos el acumulado personal de ventas por miembro (VP) cada vez que una remisión o factura sea registrada por el sistema; esto contribuirá al mejor desempeño del algoritmo de recorrido en el módulo de cálculo.
3. Recorrer al menos dos veces el árbol. La primera para obtener los montos de ventas globales (VG, VTG) por organización (nodo) y la segunda para que en base a esos montos se realice el cálculo de bonos y comisiones correspondientes.

4. En cada visita a los nodos deberá realizarse una exploración en la línea de ascendencia de dicho nodo para distribuir los montos correspondientes en las generaciones respectivas.
5. Se pueden aprovechar algunas características del SQL para evitar el doble recorrido del árbol implementando un mecanismo de acumulación generacional de forma ascendente. Por ejemplo, se puede utilizar el siguiente script de definición para el archivo **ARBOL** como una tabla creada en Interbase:

```

CREATE TABLE ARBOL
(
  NODO      INTEGER NOT NULL,
  PATERNO  INTEGER NOT NULL,
  VP       NUMERIC(15, 2),
  VTG      NUMERIC(15, 2),
  PRIMARY KEY (NODO, PATERNO)
);

SET TERM ^;

CREATE TRIGGER ARBOL_AU1 FOR ARBOL
ACTIVE BEFORE UPDATE POSITION 0
as
begin
  if (OLD.VP <> NEW.VP) then
    begin
      NEW.VTG = OLD.VTG - OLD.VP + NEW.VP;
    end
  end
^

CREATE TRIGGER ARBOL_AU2 FOR ARBOL
ACTIVE AFTER UPDATE POSITION 1
as
begin
  if (OLD.VTG <> NEW.VTG) then
    begin
      update arbol set VTG = VTG - OLD.VTG + NEW.VTG
      where nodo = OLD.paterno;
    end
  end
^

COMMIT WORK ^
SET TERM ;^

```

El trigger ARBOL\_AU1 realiza la acumulación del campo VP en el campo VTG en cada momento que se actualiza el VP por una operación de registro de venta personal para cualquier miembro.

El trigger ARBOL\_AU2 detecta una actualización al campo VTG y causa que el campo VTG del patrocinador sea a su vez actualizado. Esto crea un llamado recursivo del trigger hasta llegar a la raíz del árbol.

Estos mecanismos bien podrían aprovecharse para realizar un pre-cálculo desde el momento en que se registran las ventas por miembro.

6. Para realizar la promoción de miembros a los siguientes niveles jerárquicos en la empresa se puede utilizar un proceso de cierre mensual que simplemente utilice los volúmenes de ventas logrados y cualquier información estadística y actualice el campo NIVEL del archivo arbol según la calificación del miembro.

7. Es necesario inicializar los campos VP, VG y VTG en cero durante el proceso de cierre mensual, de manera que estos queden preparados para la corrida de bonos del mes siguiente. Es recomendable llevar un registro histórico de movimientos quizás por mes en un archivo extra.

Un siguiente paso en la implementación de nuestro modelo sería el manejo recursivo del algoritmo de recorrido del árbol. Esto podrá ser posible si aprovechamos las ventajas que nos brindan las herramientas modernas de proceso de información aunado con equipos de mayores capacidades tanto en memoria como en procesador.

**BIBLIOGRAFÍA:**

LEONARD, Ken. "What is Network Marketing? The online resource for network marketers", Netscape, USA, February 1994

CISNEROS, Guzmán, Juan Salvador. "Antecedentes, fundamentos y alcances del MLM", Tesis de Maestría en Administración, Universidad Autónoma de Aguascalientes, México, Julio 1996

EUÁN, Jorge I., CORDERO B., Luis G., "Estructura de datos", Dirección General de Publicaciones de la U.N.A.M., México, 1984.

GARCIA ROJAS, Rafael. "Manual de Métodos y Procedimientos", Excelencia en Mercadeo, S.A. de C.V. México, Diciembre 1995.

JORDÁN, Cristina, TORREGROSA, Juan R. "Introducción a la teoría de grafos y sus algoritmos", Reverté SM., Universidad Politécnica de Valencia, España, 1996

COLMAN, BUSBY, ROSS, "Estructuras de Matemáticas Discretas Para La Computación", Prentice Hall, 1997