Algoritmo de promociones

Fue implementado en SQL Server, y tiene el nombre de “CreatePromos”, este algoritmo se implementó como probabilístico numérico. Y únicamente posee un parámetro “CantidadPromos”, el cual es un entero que se usa para indicar la cantidad de promociones que se desea que el algoritmo genere.  
Su funcionamiento es el siguiente:

De forma aleatoria se toma un descuento a un artículo (que oscila del 1% al 100%) y una condición.   
En el sistema existen 3 tipos de condiciones:   
a) Condiciones por artículos: Solicita que el usuario haya comprado una “n” cantidad de un artículo.  
b) Condiciones por monto gastado: Solicita que el usuario haya gastado, en una sola compra una cantidad superior o equivalente al que solicita la condición.  
c) Condiciones por puntos: Solicita que el cliente tenga la cantidad de puntos necesarios para “cajear” la promoción.

Cabe destacar que el tipo de condición es elegida totalmente al azar.

Dentro de un ciclo, el cual se mantendrá mientras el tamaño de la tabla sea inferior al del parámetro ingresado (“CantidadPromos”), se va a generar un número flotante random, el cual indicará si el porcentaje de “perdida” para el vendedor será:   
a) Totalmente al azar: 1% - 100%, del cual hay un 5% de probabilidades de que se dé.  
b) Descuento mayor: 1% - 50%, del cual hay un 30% de probabilidades de que se dé.  
c) Regular: 1% - 25%, del cual hay un 70% de probabilidades de que se dé.  
  
Por ejemplo en el “Regular”, la promoción tendrá como requisito que de la compra total del cliente este haya gastado lo suficiente, tal que el descuento que se le va a dar al cliente oscile entre un 1% a un 25% del total de la compra.  
Por ello, si un cliente desea un descuento de 2500 colones, las condiciones han de garantizar que el cliente haya comprado al menos 7500 colones en algún otro artículo, el total de la compra o en puntos.

Luego el algoritmo, calculará la cifra exacta de ganancia que podría, potencialmente, generar la promoción; así como la perdida (descuento).   
Estos valores se van a almacenar en las variables “MinimoCondicion” y “MontoDescuento”.

Y finalmente, el algoritmo almacenará la promoción en la base de datos. (En la tabla “promociones”). Aunque puesto que una solo promoción puede tener uno o más condiciones y/o descuentos, se usarán 4 tablas compuestas para generar las relaciones N-N.

Casos de prueba

* Creación de clientes.
* Ingreso de categorías para los artículos.
* Creación de artículos.
* Creación de descuentos.
* Creación de condiciones.
* Compra de artículos.
* Recomendación de promociones por parte del sistema.
* Uso de promociones en una compra.
* Búsqueda de artículos por nombres semejantes (usando la función LIKE de SQL).

La prueba se realizó con:

* 25 categorías.
* 25 artículos.
* 25 condiciones por artículos.
* 25 condiciones por monto.
* 25 condiciones por puntos.
* 50 descuentos.
* 100 promociones.

A nivel de base de datos, y mediante el uso de Scripts de llenado automáticos.

Específicamente usando las stored procedures:   
CreateCategories CreateConditionsByAmount  
CreateArticles CreateConditionsByPoints  
CreateClients CreateDiscounts  
CreateConditionsByArticles CreatePromos

Los cuales, cada uno posee un único parámetro, correspondiente a la cantidad de estos; que el usuario desee que el sistema generare automáticamente.

Por otra parte para el ingreso manual de información a la base de datos a las tablas, se tiene los procedimientos:  
NewCategory NewArticle NewFactura AddDetailsToFacturaPromocion  
NewClient NewDiscount AddDetailsToFactura

De estos procedimientos es importante mencionar que las compras son realizadas con NewFactura, acción a la cual se le agregan artículos o promociones cajeables mediante los procedimientos AddDetailsToFactura y AddDetailsToFacturaPromocion.

Finalmente para consultas se tiene los siguientes stored procedures:  
ArticlesByName, el cual muestra cualquier artículo cuyo nombre sea similar al valor ingresado.

PossiblePromos, el cual muestra todas las promociones, a las cuales el cliente satisface todas las condiciones, en una compra determinada.