***Ingeniería Investigación y Tecnología,*** volumen XVI (número 4), octubre-diciembre 2015: 551-563
ISSN 1405-7743 FI-UNAM
(artículo arbitrado)

# Algoritmo que ejecuta submodelos de un modelo matemático para mayor eficiencia

## *Algorithm that Executes Submodels of a Mathematical Model for Greater Efficiency*

Bauer-Mengelberg Juan Ricardo
*Colegio de Posgraduados, Montecillo Texcoco*
*Posgrado de Cómputo Aplicado*
*Correo: jbauer@colpos.mx*

Vega-Ruiz Rafael
*Colegio de Posgraduados, Montecillo Texcoco*
*Posgrado de Cómputo Aplicado*
*Correo: rafael.vega@colpos.mx*

Información del artículo: recibido: junio de 2014, reevaluado: julio de 2014, aceptado: junio de 2015

**Resumen**

El tema surge del servicio de información denominado FLAG (*flujo de efectivo en agronegocios*) que ofrece a sus clientes modelos matemáticos, este contiene variables del cliente y otras de su entorno de negocios. FLAG obtiene valores actualizados de estas variables con la periodicidad requerida por el cliente, quien recalcula las demás variables del modelo, lo que resulta en un impacto que el cliente obtiene de los cambios en los indicadores (otras variables) incluidos en su modelo. Se describen los métodos utilizados para incrementar la eficiencia del cálculo de un modelo matemático, que consiste en un conjunto de fórmulas con las que se obtienen valores actualizados de las variables del modelo. Cada fórmula se crea por operaciones entre otras variables del modelo. La necesidad de reducir los tiempos del proceso se debe a la ejecución de varios modelos, así como la duración total, ya que está acotada por requisitos del sistema que los invoca. Se construyen los algoritmos de modo que se minimicen las operaciones de entrada y salida. Asimismo, cuando se recalcula el modelo como consecuencia del cambio de una sola variable, solo se recalcula el submodelo correspondiente, formado por las variables afectadas directa o indirectamente por el cambio. Ejecuciones de modelos preparados para determinar la eficacia de las mejoras muestran reducciones hasta del orden de 90% en las duraciones.

**Descriptores:**

- modelo matemático
- árbol
- submodelos
- servicio de información

### *Abstract*

*The topic arises in an informing service called FLAG (cash flow in agribusinesses) that offers its clients mathematical models consisting of his own variables as well as others from his business environment. FLAG obtains updated values of the latter with the frequency determined by the client, and computes the values of the other variables, thus providing the client with the impact of the changes in the indicators he includes in his model. The methods used to increase the efficiency of the calculations of a mathematical model containing a number of formulae through which the values of its variables are computed are described. It consists of operations to be performed on its operands, variables of the model. The need to reduce processing times results from the processing of several models, where the total duration is limited by constrains of the system that invokes such executions. The algorithms are built to minimize input-output operation. Additionally, whenever the model is invoked by a change of a single variable, only the submodel, consisting of the variables that were directly or indirectly affected by the change, are calculated. Executions of models prepared to confirm the efficacy of the improvements show reductions of up to 90%.*



***Keywords:***

- *mathematical model*
- *tree*
- *submodels*
- *informing service*


## Introducción

La necesidad de minimizar la duración de la ejecución de modelos matemáticos surgió de un servicio denominado FLAG (*flujo de efectivo en agronegocios*) descrito por Bauer-Mengelberg (2010). Este servicio se ofrece a diversos clientes, quienes formulan un modelo matemático integrado por variables propias (VCL, *variables del cliente*) y variables de su entorno de negocios (VMR, *variables del mundo real*). El servicio consiste en obtener valores actualizados de las VMR y recalcular las variables que utilizan en sus modelos los clientes.

Según Gackowski (2005) los atributos que exige un servicio de este tipo en la información que proporciona a sus clientes, entre otros, se encuentra: ser interpretable, significativa, disponible a tiempo y creíble. Sin embargo, el atributo fundamental es que la información sea *timely actionable*, es decir, que el receptor pueda usar la información a tiempo. La incorporación de los datos actualizables a un modelo permite precisamente eso, comunicar al cliente el impacto de los cambios sobre sus actividades.

Generalmente, el servicio ofrece un conjunto de VMR al cliente que lo necesite, cuando alguien solicita una nueva VMR, se le agrega al conjunto. Asimismo, se definen agentes de búsqueda que serán los que, con la periodicidad requerida, obtengan valores actualizados de las variables a partir de los publicados en Internet. Cuando cambian los valores de algunas variables, se ejecutan los modelos en los que intervienen las que ya sufrieron cambios.

El modelo contiene variables calculadas, cuyos valores se obtienen mediante una fórmula que se define para ese efecto, y que tendrá como operandos otras variables, tanto del cliente como del mundo real. Los detalles de las fórmulas se proporcionan adelante.

De esta manera, cada cliente actualiza una base de datos con sus variables y fórmulas, además de actualizar los valores de las variables "de abajo", que son las que no tienen fórmula. Por ejemplo, se introducen los valores de las hectáreas en las que sembrará algún cultivo por mes, estos valores intervendrán en una fórmula de otra variable, es decir, la cosecha será el número de hectáreas multiplicado por el rendimiento ha, donde el rendimiento puede ser un dato calculado (variables de clima, tipo de semilla, fertilización, etcétera) u otro vector "de abajo".

La situación que se presenta en el FLAG muestra que el servicio puede proporcionarse a muchos clientes, en donde el cuantificador dependerá del contexto o del tipo de equipo de cómputo que utilice para ofrecerlo. Cuando los agentes detectan cambios en los valores de las VMR, en ocasiones se ejecuta un gran número de modelos, por lo tanto, la duración de estos procesos debe ser suficientemente corta para su conclusión, antes de crear la necesidad de ejecutar otros modelos como consecuencia de nuevos cambios detectados en valores de VMR, ya que se lanzan agentes sin interrupción.

El FLAG tiene una serie de mecanismos que le permiten conocer los modelos que se deben recalcular, dados los cambios en valores de VMR. Un proceso lanza los agentes (programas que consiguen datos de una página de Internet) y le comunica a otro programa periódicamente (típicamente cada minuto) dichos cambios. Este último, aplica los criterios de ejecución proporcio-

nados por cada cliente para determinar si alguno de los cambios implica el recálculo de su modelo. Para los modelos afectados se invoca el programa de ejecución de fórmulas que recalcula y regraba los valores de las variables afectadas para cada uno de ellos.

La eficiencia del proceso de cálculo del modelo es vital cuando se ejecutan varios modelos, por lo tanto, se buscan formas de acortar ejecuciones lo más posible, aquí se logró con tres pasos:

*Primero*: se minimizan las actividades de lectura y grabación (I-O).

*Segundo*: se implementan los submodelos, tema central de este trabajo, sustituyendo la ejecución de todas las fórmulas del modelo para recalcular únicamente los valores de las variables afectadas en forma directa o indirecta debido a cambios de valores en una sola VMR.

*Tercero:* se emplea el concepto de operando aditivo de una fórmula (solo interviene como sumando o sustraendo de la fórmula) para reemplazar el cálculo completo y no conseguir los valores de sus operandos, así se aplica la diferencia entre los valores nuevos y anteriores del operando aditivo.

Para describir cómo se implementaron estas estrategias y el impacto que tuvieron sobre los tiempos de ejecución se organizó el desarrollo, tras describir una serie de materiales que se emplearon, asimismo, se proporcionan algunos aspectos de las fórmulas del FLAG y de su ejecución, para lo cual se utilizó un ejemplo sencillo. Por otro lado, se indica el motivo de no utilizar hojas de cálculo para el servicio FLAG, que es una observación frecuente a este sistema. También se describe la ejecución de un modelo con énfasis en las operaciones de entrada y salida de datos. Posteriormente se detalla el proceso con el que se crearon los submodelos para las VMR, que son listas de las fórmulas a recalcular cuando cambia cada variable. Por último, se proporcionan algunas duraciones resultantes de simulaciones que se efectuaron para estimar el impacto sobre los tiempos de ejecución de cada uno de los tres elementos mencionados. Finalmente las conclusiones incluyen comentarios sobre el uso de los submodelos.

## Materiales y métodos

### El árbol como estructura de datos

En matemáticas y ciencias de la computación un grafo conexo es un conjunto de objetos llamados nodos, unidos por enlaces llamados aristas o arcos, que permiten representar relaciones binarias entre elementos de un conjunto. Una de las estructuras de datos más utilizadas es el árbol, que se define como un grafo conexo sin bucles (también llamados ciclos). Según Weiss (1992), un árbol consiste de un nodo distinguido *r*, llamado raíz, con cero o más subárboles T1, T2, …, Tk, cuyas raíces están conectadas por un arco dirigido a *r*.

En este trabajo se adoptó la siguiente terminología:

*subnodos:* conectados a un nodo,
*hoja:* nodo sin subnodos,
*vértice o raíz*: nodo que no es subnodo de otro y
*orden del árbol:* número máximo de subnodos que puede tener un nodo.

Entre un nodo y subnodos hay una relación llamada "padre-hijo". En FLAG se sustituyó el término *padre* por el de *dad*, ya que las primeras versiones del paquete se elaboraron en inglés.

### El FLAG usado para ilustrar los conceptos de eficiencia de procesos

Para describir el uso de submodelos de un modelo se consideró conveniente exponerlo en el contexto de un servicio informativo (*informing service*) llamado FLAG (*flujo de efectivo en agronegocios*) que proporciona a sus clientes datos del mundo real (su entorno de negocios). Como se muestra en la figura 1, este se compone por módulos, que en conjunto proporcionan el servicio.

El concepto principal que define este servicio es el modelo del cliente, donde actúan todos los procesos, entre ellos: la comunicación entre cliente y FLAG, las consultas, la actualización de variables y la ejecución de las fórmulas de un modelo.

FLAG se concibió como un servicio a varios clientes, en donde cada uno elabora un modelo del tipo descrito anteriormente. El servicio ofrece valores actualizados de diversas *variables del mundo real* (VMR) y cada cliente puede incluir algunas de ellas en sus modelos. Por ejemplo, un productor puede incluir el precio de un insumo en el modelo, así el servicio contará con un agente, que con la periodicidad necesaria, obtendrá de la fuente indicada para esa variable el valor actualizado del precio.

Como parte de las especificaciones, cada cliente proporciona los llamados criterios de ejecución, para ello, indica las variables del mundo real de su modelo, aquellas que cuando cambien los valores, harán que se ejecute el modelo. Esto permite determinar el impacto de los cambios en sus actividades o indicadores.

El criterio formulado para una VMR se especifica por un intervalo alrededor del valor anterior, es decir, si el nuevo valor cae fuera de este intervalo se ejecutará el modelo.

Los intervalos se pueden especificar por sus extremos y también como porcentajes de cambio detectados en los valores actualizados. Por ejemplo, se puede indicar que si el cambio en el valor es menor que −1 o mayor que +2 se ejecute el modelo, o hacer lo propio, pero proporcionando porcentajes del valor anterior.

Considerando que el servicio tiene *muchos* clientes, los cambios en los valores de las VMR pueden resultar en la ejecución de una gran cantidad de modelos, ya que el tiempo disponible para las ejecuciones está acotado por la periodicidad con la que se obtienen los cambios en las variables del mundo real, por lo que la eficiencia de estos procesos es crucial. Esto constituye el motivo de la investigación que se describe en donde se buscan formas de mejorar los procesos en cualquier sentido que resulten en una mayor eficiencia computacional.

## Los modelos definidos en FLAG

La figura 2 ilustra un ejemplo de un modelo del tipo FLAG. Se aprecian los nodos del árbol que están conectados entre sí. Los nodos corresponden a variables que representan algún factor de negocio o cualquier otro uso del modelo. Los subnodos (hijos) de un nodo representan los operandos de una fórmula mediante la cual se calculan los valores del nodo padre en función de los valores de sus "hijos". De ese modo, los nodos que no tienen subnodos (hojas) no tienen fórmula, y se denominan variables "de abajo".

## FLAG y las hojas de cálculo

En las hojas de cálculo más populares se encuentra MS-EXCEL, donde la ejecución de las fórmulas es consecuencia de la modificación de una celda que las afecta. Sin embargo, hay modos de evitar que esto suceda, indicando el inicio de este cálculo mediante un comando que se incluye en la hoja para tal efecto. En los modelos implementados en FLAG solo se ejecutará el modelo

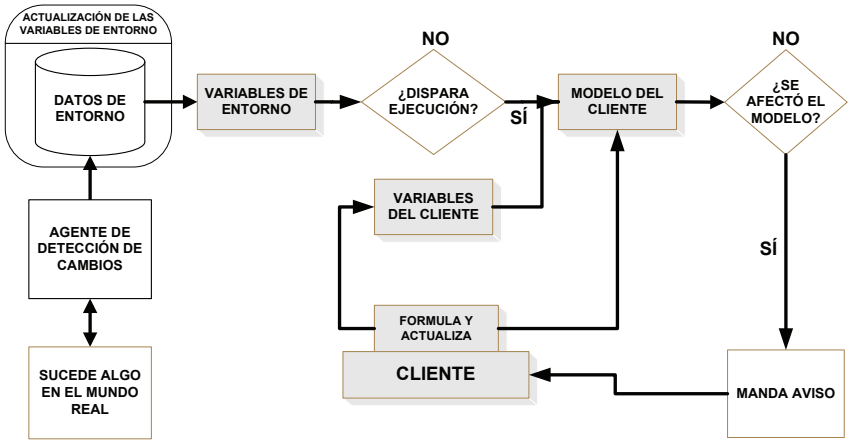



Figura 1. Servicio que proporciona el FLAG a cada uno se sus clientes

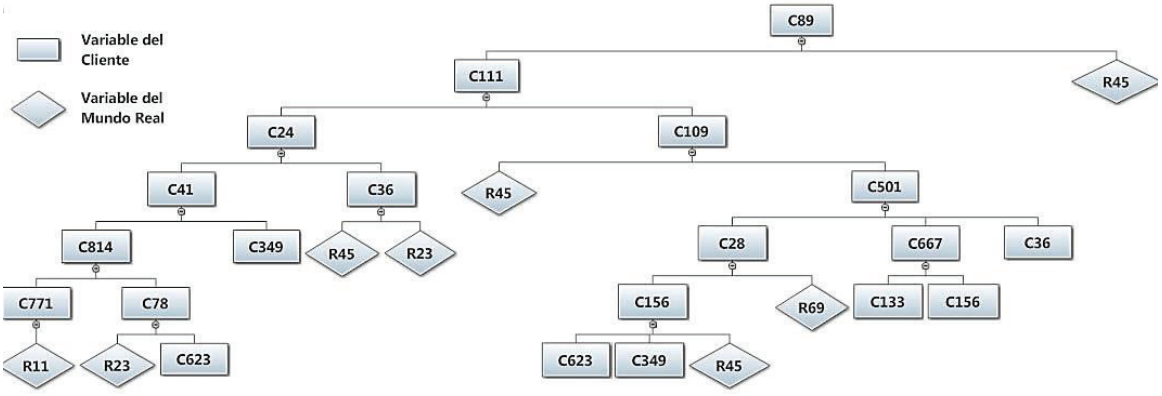



Figura 2. Ejemplo del modelo de un cliente

con el sistema cuando reciba una instrucción específica en ese sentido.

Las hojas de cálculo modernas tienen algoritmos sofisticados que les permiten decidir qué celdas deben recalcularse como consecuencia del cambio de alguna celda (Excel's Smart Recalculation, 2008). En particular, se pueden incluir instrucciones que determinen un rango de celdas con las cuales se ejecutará el modelo cuando cambien sus valores (Vbforums, 2011). En ocasiones la regla de decisión aplicada determinará que es preferible ejecutar todo el modelo para no limitar el recálculo a una parte del mismo.

En este trabajo se describe cómo se implementó este tipo de cálculos parciales a los modelos descritos; adelantamos que se utiliza un submodelo y el conjunto de fórmulas que se ejecutan para reflejar adecuadamente el cambio de una sola variable.

De la analogía con las hojas de cálculo surge una pregunta ¿Por qué no se utilizan hojas de cálculo para el servicio FLAG? Entre las causas principales se encuentran:

- Las variables en realidad son vectores de 24 valores llamados periodos. Esto complica el uso de hojas de cálculo, a pesar de que en muchas, por ejemplo en Excel, se implementan las fórmulas de arreglos (Decisión Models, 2014).
- Se admiten variables aleatorias con distribuciones discretas de hasta 4 valores, además de las variables escalares. Estas variables consisten de 24 periodos, cada uno con cuatro pares de valores (el valor y su probabilidad).
- FLAG ejecuta una serie de acciones además de las asociadas al cálculo de fórmulas. Entre otras, se encuentra la formulación de diversas consultas al modelo y la generación de alarmas, es decir, avisos urgentes al cliente. Puede resultar complicado incluir esta funcionalidad adicional en las hojas de cálculo.
- La actualización de los datos que proporciona cada cliente de sus variables "de abajo" se puede realizar directamente sobre una hoja de cálculo, pero sería más difícil para el cliente además de destruir alguna relación previamente indicada. Esto es cierto cuando modifica su modelo, ya sea agregando variables o cambiando alguna fórmula.
- Graficar los modelos implementados en hojas de cálculo sería considerablemente más complicado que hacerlo con las estructuras que usa FLAG.

Como parte del diseño del servicio FLAG se tomó la decisión de almacenar en sendos archivos las fórmulas y los valores de las variables. Para cada modelo, es decir, para cada cliente, se almacenan los valores de todas las variables en un archivo destinado a tal efecto. Hay un archivo similar para las VMR, pero este es único, ya que los valores de las variables serán los que utilicen todos los modelos que las incluyan en sus fórmulas.

La ejecución de la fórmula de una variable resulta en la actualización de sus valores, en el archivo de valores. Para cada operando de una fórmula se emplean sus valores, es decir, se cargan a memoria los del registro correspondiente.

## Las fórmulas en FLAG

La definición matemática de una fórmula que proporciona la RAE (2014) es: *ecuación o regla que relaciona objetos matemáticos o cantidades*. Un cliente define sus fórmulas especificando la variable (resultado) y los objetos (operandos), los cuales relaciona mediante operaciones. Los operandos son variables del modelo por lo que pueden, a su vez, tener una fórmula.

Los paréntesis suelen causar problemas a los que formulan modelos, especialmente cuando hay paréntesis anidados; por ello en FLAG se sustituyeron los paréntesis por "subtotales" y se construyó la fórmula agrupando operaciones en un subtotal, que de hecho tiene el mismo efecto que incluirlas en un paréntesis. Este subtotal será empleado por alguno (o más de uno) de los subtotales subsecuentes. El último subtotal proporciona el valor de la variable calculada con la fórmula.

De ese modo, una fórmula consiste de operaciones entre operandos, que pueden ser variables o subtotales calculados anteriormente. Por ejemplo, en la fórmula de la VCL156 del modelo ilustrado en la figura 1, en lugar de escribir:

VCL156 = (VMR623 + VMR349) * VMR45
se especifica: Subtotal (1) = VMR623 + VCL349
VCL156 = Subtotal (2) = subtotal (1) * VMR45

El orden en el que se ejecutan las fórmulas debe cumplir dos condiciones:

- Los operandos deben calcularse antes de que los use una formula.
- No debe haber referencias circulares (también las llamadas ciclos), término que significa que una variable interviene en una fórmula de otra variable, que directa o indirectamente interviene en la del operando.

En FLAG ambos criterios se satisfacen mediante el uso del concepto de nivel de una fórmula. El término se explica ampliamente en el resto del trabajo, pero significa

que las fórmulas se numeran, de manera que se ejecutan en orden ascendente del nivel asignado.

Las fórmulas se almacenan en un archivo plano a medida que las define el cliente, quien generalmente no lo hará en un orden que garantice la propiedad mencionada. Por lo tanto, será necesario encontrar un orden de ejecución de las fórmulas que evite que un operando se calcule después de una fórmula que lo usa. Para ello, se ejecuta un proceso llamado *preparación de fórmulas* que logra ese objetivo, y que además se aprovecha para otros fines como se explicará a detalle en la sección de resultados.

Se decidió limitar el número de subtotales a cuatro por fórmula; del mismo modo, cada subtotal tendrá como máximo cuatro operaciones. Si una fórmula necesitara más de cuatro subtotales, se crearía una variable artificial intermedia. Los valores de estos dos máximos (cuatro) no resultan de alguna teoría o recomendación, solo fueron seleccionados como parte del diseño de FLAG, que contempla en forma prioritaria la facilidad para usar el sistema por parte de los interesados en hacerlo.

El término *sinónimo* se explicará posteriormente, pero la estructura con la que se almacenan las fórmulas es la siguiente:

- Fórmula de la variable Núm. NNN
- Nivel: se agrega durante la preparación
- Cuántos subtotales usa (puede usar menos de los 4)
- Subtotal (1 a 4)
  - Cuántos operandos: los empleados en este subtotal
  - Operación (1 a 4): solo las que se usan
  - Operandos (1 a 4): algunos operandos se reemplazarán por sinónimos.

Los operandos se indican con el tipo de operando (VCL, VMR, Sinónimo, Subtotal) y número de la variable o del subtotal.

### Operandos aditivos

El operando de una fórmula es aditivo si las operaciones que lo relacionan con los restantes operandos son la suma o la resta. Por ejemplo, en la fórmula:

V(3) = V(4) + V(5) - V(6)*V(7)

las variables V(4) y V(5) son operandos aditivos, mientras que V(6) y V(7) no lo son. Supongamos que V(4) cambió en +21 unidades, y V(3) valía (antes) 200. Entonces V(3) valdrá 221 después de calculada la fórmula, siempre que no hubieran cambiado los valores de los restantes operandos. En cambio, en la fórmula de la variable número 14:

Subtotal (1) = V(30) + V(31);  V(14) = subtotal (2) = subtotal (1) * V(32)

las variables V(30) y V(31) no son aditivas, ya que aunque lo son dentro del primer subtotal, este es un operando no aditivo de otro subtotal. De ese modo, si V(30) aumentó en 15, el subtotal (1) también aumenta en 15, pero la V(14) no. El cambio en esta será 15 * el valor de la variable (32).

Cuando esto sucede, en lugar de ejecutar toda la fórmula, se procede de la siguiente forma: supongamos que la variable C100 interviene en la fórmula de la variable C120 y se determina que lo hace de forma aditiva. Solo variables calculadas del cliente se toman en cuenta para su aditividad (si la variable es "de abajo", los cambios en los valores son resultado de tecleo, de modo que no resultan en una ejecución del modelo):

- Cuando cambia el valor de la variable C100 se almacena en memoria la diferencia entre el valor anterior y el nuevo (calculado).
- Para calcular el valor de la variable C120, se suma esa diferencia al valor que tenía la variable C120 (o se restan, si el operando fuera un sustraendo).

## Resultados

### Transformación del modelo en un árbol

Cuando el cliente define las variables y fórmulas del modelo, no se le exige determinar el orden en el que se ejecutan las fórmulas; además, en ocasiones, su modelo contendrá ciclos. Para eliminar estos últimos y preparar el modelo para su ejecución correcta, FLAG *prepara* los modelos para lograr estos objetivos. El cliente prepara sus fórmulas en un archivo y el FLAG lo transforma en otro mediante el proceso llamado: preparación del archivo de fórmulas. Este proceso se aprovecha para otras tareas.

En un árbol, un subnodo no puede ser de más de un nodo, lo que se puede interpretar como: una variable no puede estar en más de una fórmula. En general, cuando se formula un modelo, no se cumple esta condición. Por ejemplo, la superficie en has puede ser un operando de la fórmula del volumen cosechado, pero también de la cantidad de semilla que se tiene que adquirir

En FLAG, se usa el concepto de *sinónimo* de una variable para lograr la estructura deseada, es decir, cuando una variable se utiliza en más de una fórmula se

genera un sinónimo. Los sinónimos son entidades adicionales a las variables de cliente y a las VMR; toman los valores de las variables que reemplazan. En el modelo de la figura las variables VCL36, VCL156 y VCL349, VCL623, VMR023 y VMR45 requieren sinónimos, ya que son operandos de más de una fórmula.

El uso de los sinónimos permite contemplar el modelo como árbol sin imponer la restricción de que una variable no pueda estar en más de una fórmula, lo que resultaría en modelos mucho más limitados e insuficientes para diversas aplicaciones. El hecho de que sea un árbol también permite graficar el modelo y construir los submodelos, el conjunto de variables que se ven afectadas por los cambios en los valores de una sola VMR.

## Preparación del archivo de fórmulas

La preparación del modelo para su ejecución tiene los siguientes objetivos:

- Calcular los niveles de las fórmulas y con ellos una lista de fórmulas en orden ascendente de nivel.
- Para cada operando de cada fórmula se determina si se trata de un operando aditivo.
- Se reemplazan los operandos "repetidos" por sinónimos, mismos que se numeran en orden progresivo a medida que se les necesita.
- Se crea un nuevo archivo de fórmulas, donde se guardarán en orden, pero con los operandos sustituidos por sinónimos cuando sea necesario.
- Se agrega el submodelo de cada VMR que utiliza el modelo, es decir, una lista de fórmulas que se ejecutarán cuando la invocación del modelo se deba a cambios de una única VMR.

A continuación, se describe brevemente el programa que crea el archivo final a partir del archivo de fórmulas preparado por el cliente, aquí se dividió la descripción en varios pasos, mismos que corresponden al orden en el que se ejecutan las diversas funciones necesarias.

### Paso 1. Sinónimos y aditividad (primera pasada)

El término "pasada" se usa en programación para indicar que se leen todos los datos en forma secuencial. En este caso, se lee cada fórmula en el orden en el que aparece dentro del archivo de fórmulas. Para abreviar procesos posteriores se construye un arreglo de todas las fórmulas en memoria.

El objetivo de este primer paso es comenzar el cálculo del nivel de las fórmulas, reemplazar operandos repetidos por sus sinónimos y determinar la aditividad de los operandos de cada fórmula. Para ello, se construyen tres arreglos con las estructuras que se muestran en las tablas 1, 2 y 3, respectivamente.

De esta manera, se procesa cada fórmula del arreglo creado anteriormente a partir de las fórmulas del archivo. Conviene recordar que solo las variables del cliente pueden tener una fórmula y para cada una:

- Si la variable con la que se calcula todavía no se encuentra en el arreglo de variables del cliente, se agrega.
- Se actualiza el campo "tiene fórmula" a verdadero.

Tabla 1. Los campos de cada variable del cliente que se agrega al arreglo

| Variables del cliente | |
|---|---|
| | Posición en archivo final (de la fórmula de esta variable) |
| | Tiene fórmula: si no tiene, es de "arriba", raíz del árbol |
| | Dad: la variable de la cual es operando de su fórmula |
| | Nivel: nivel de ejecución calculado para la fórmula de esta variable |
| | Es-aditiva: -1, 0,1 según el operando es restando, no es aditivo, sumando |
| | En cuántas fórmulas aparece |

Tabla 2. Los campos de cada variable del mundo real que se agrega al arreglo de estas

| Variables del mundo real | |
|---|---|
| | Núm. variable: el número en el modelo del cliente |
| | Original VMR: el número en la base del FLAG |
| | Dad: la variable de la cual es operando de su fórmula |
| | En cuántas fórmulas aparece |

Tabla 3. Los campos de cada sinónimo que se crea y agrega al arreglo de estos

| Sinónimo |
|---|
| Variable que reemplaza: el número de esa variable |
| Tipo de variable: indica si la variable que reemplaza es del cliente o del mundo real |
| Dad: la variable de la cual es operando de su fórmula |
| Dad: la variable de la cual es operando de su fórmula |

- Para cada variable que aparece como operando de cada subtotal de la fórmula, se actualizan los datos en el arreglo correspondiente a su tipo (del cliente o del mundo real), es decir, se suma 1 al campo "En cuántas fórmulas aparece" y si el resultado es mayor que 1 se crea un sinónimo de esa variable y se agrega al arreglo de sinónimos (tabla 3) con los datos que le corresponden, se actualiza el campo "variable que reemplaza" con la variable del operando actual y "*dad*" con el número de la variable de la fórmula en proceso; finalmente se reemplaza en la fórmula el operando por su sinónimo.

Para cada operando de tipo VCL, se determina si interviene en la fórmula como operando aditivo, examinando primero el subtotal en el que aparece, después se hace lo propio con los subtotales en los que se encuentra ese subtotal como operando. De este modo se asigna un valor al campo es-aditiva: 0 si no es aditivo, 1 si es sumando y −1 si es sustraendo. Si un operando está en más de un subtotal no es aditivo. Se hace lo mismo con los sinónimos una vez que agregaron al arreglo de estos.

Para las VMR no se determina si intervienen como operandos aditivos, ya que al invocar una ejecución solo se conocen los valores "nuevos" de estas variables, de modo que no se cuenta con una diferencia aplicable a la variable calculada con la fórmula.

## Paso 2. Cálculo de niveles

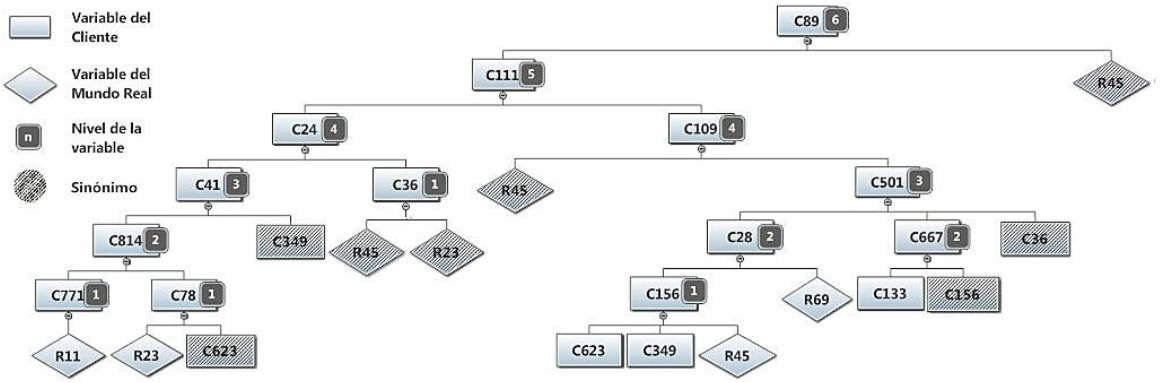
Se procesan todas las fórmulas con el número de pasadas que sean necesarias. Se calcula:

Nivel de la fórmula = Max (nivel de sus operandos) + 1

Cada vez que cambia se actualiza el nivel de la variable en el arreglo de variables del cliente. Este proceso se repite hasta que ya no haya cambios de nivel. Si el número de lecturas excede el número de fórmulas del modelo, hay por lo menos una referencia circular (bucle) en el modelo.

Para determinar las variables involucradas en el ciclo, se ejecuta el mismo proceso varias veces, pero ahora almacenando la información de los operandos que originan el cambio de nivel. Se interrumpe el proceso tras desplegar en el monitor del usuario las fórmulas y sus operandos, que intervienen en el ciclo para que el encargado de formular el modelo pueda eliminarlo. Estos procesos también detectan la presencia de más de un ciclo.

Al término del cálculo de niveles, se actualiza el nivel de cada fórmula en el arreglo de fórmulas en memoria, a partir del arreglo de variables del cliente. En la figura 3 se muestran los sinónimos y los niveles de cada




| La lista de fórmulas en orden ascendente de nivel | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C36 | C78 | C156 | C771 | C28 | C667 | C814 | C41 | C501 | C24 | C109 | C111 | C89 |

Figura 3. El modelo tras calcular los niveles y reemplazar operandos repetidos por sus sinónimos y la lista de fórmulas en orden construida

fórmula del modelo, así como la lista de fórmulas en el orden que se producen en el siguiente paso.

### Paso 3. Creación de un arreglo de variables con fórmulas en orden ascendente de nivel

Se recorre el arreglo de fórmulas, de las que ya se ha calculado su nivel. Se agregan a un nuevo arreglo (el de fórmulas en orden ascendente), insertando cada una en la posición que le corresponde con el ordenamiento por nivel y número de variable.

Con el fin de proporcionar información sobre el archivo de fórmulas preparado para la ejecución del modelo, se describen algunos aspectos de este archivo. Su diseño resultó de la búsqueda de eficiencia en la ejecución del modelo, a pesar de que el impacto del diseño adoptado es mínimo (se usa una vez por ejecución) pero el hecho de que sustituye el uso de tablas de la base de datos es significativo.

El archivo, es un archivo plano que se usa como "binario", y se divide en secciones:

- Una parte fija, que contiene información del resto del archivo (de hecho, apuntadores a las demás secciones).
- Un arreglo que consta de los números de las fórmulas en orden creciente de nivel: es de longitud variable (depende del número de fórmulas del modelo). Se agrega la posición de cada fórmula en el archivo: esto permite leer las fórmulas (almacenadas en la siguiente del archivo) sin efectuar operaciones para determinar el segmento del archivo que la contiene.
- Las fórmulas mismas una tras otra con todas sus especificaciones, se graban a partir del arreglo de fórmulas en memoria, en donde se incluye el nivel de la fórmula y se reemplazan los operandos "repetidos" por sus sinónimos.

### Paso 4. Se graba el arreglo de fórmulas ordenadas en el archivo de fórmulas

Para cada una se indica la posición de esta fórmula en el archivo final de fórmulas (el byte en cual comienza).

### Paso 5. Se agregan las fórmulas (que están en memoria) al archivo

Se recorre el "arreglo de fórmulas en orden" y se graba cada una en el archivo, usando el campo "Posición en archivo final" actualizado en el paso anterior.

### Paso 6. El armado del submodelo correspondiente a cada VMR

El submodelo correspondiente a una VMR consiste en el conjunto de variables cuyos valores pueden cambiar como resultado de cambios en los valores de esa VMR. Se recorre el arreglo de VMR armado en el paso 1 (tabla 2) y se prepara un arreglo que denominamos arreglo índice de submodelos que contiene la información de cada VMR del arreglo. Posteriormente se completan los datos con la cantidad de fórmulas del submodelo y la posición de la lista de sus fórmulas en el archivo.

Para cada VMR, se actualiza un "arreglo de arreglos", donde el arreglo secundario corresponde a la lista de fórmulas del submodelo. En una sección posterior dedicada precisamente a este tema de proporción se muestra la descripción detallada de la rutina con la que se arman los submodelos.

### Paso 7. Se graba el arreglo índice de submodelos creado en el paso 6 en el archivo

En el registro base del archivo también se actualiza la posición inicial a partir de la cual se grabó este índice de submodelos. Al terminar el proceso para cada VMR, se agrega el "arreglo de arreglos" al archivo.

Con esto concluye el proceso de preparación del archivo de fórmulas para la ejecución del modelo. En realidad, el programa tiene una función adicional relacionada con las alarmas (avisos urgentes) solicitadas por los clientes, tema que no se incluyó, ya que es independiente del asunto de artículo. Los interesados en este tema pueden consultar detalles en Bauer y Velázquez (2010). Se agregan al archivo los criterios de alarma del cliente, mismos que están en su base de datos. El propósito de incluirlos aquí es el de no tener que usar esta base de datos.

## Armado del submodelo de una VMR

El modelo que ilustra la figura 4 se incluyó con el único fin de ilustrar el proceso de creación de un submodelo, por lo que no es necesaria la interpretación de las diversas variables como parte de una situación real. Un signo "+" o "−" en los conectores de las variables con sus "*dads*" indica que esa variable interviene de forma aditiva en la fórmula de dicho padre.

En la figura 5, donde se describe el proceso de armado del submodelo, se utilizó la siguiente notación:

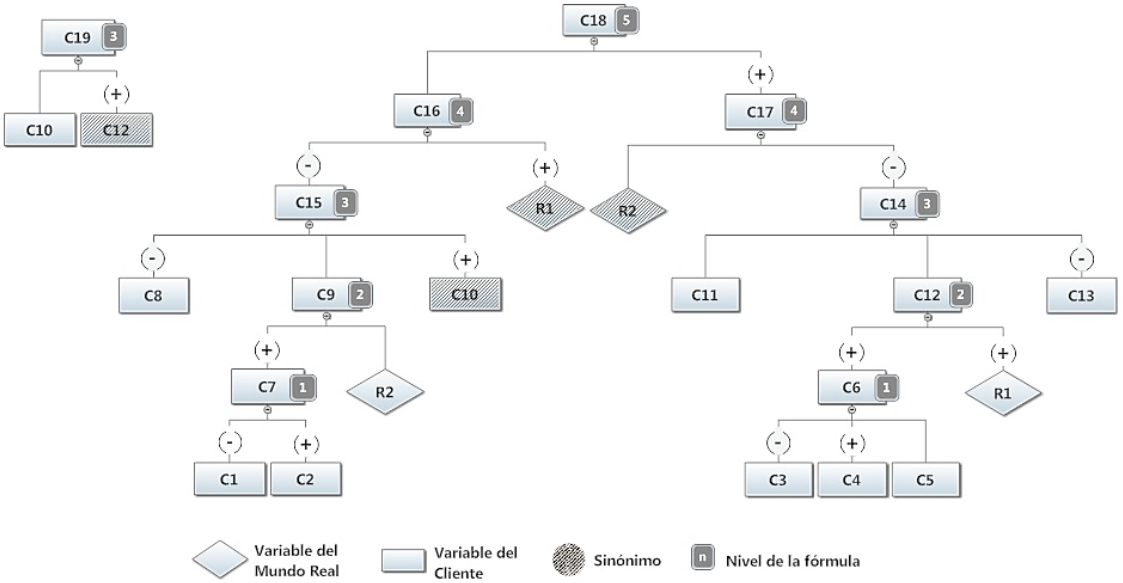



Figura 4. Modelo utilizado para ilustrar el algoritmo que prepara un submodelo

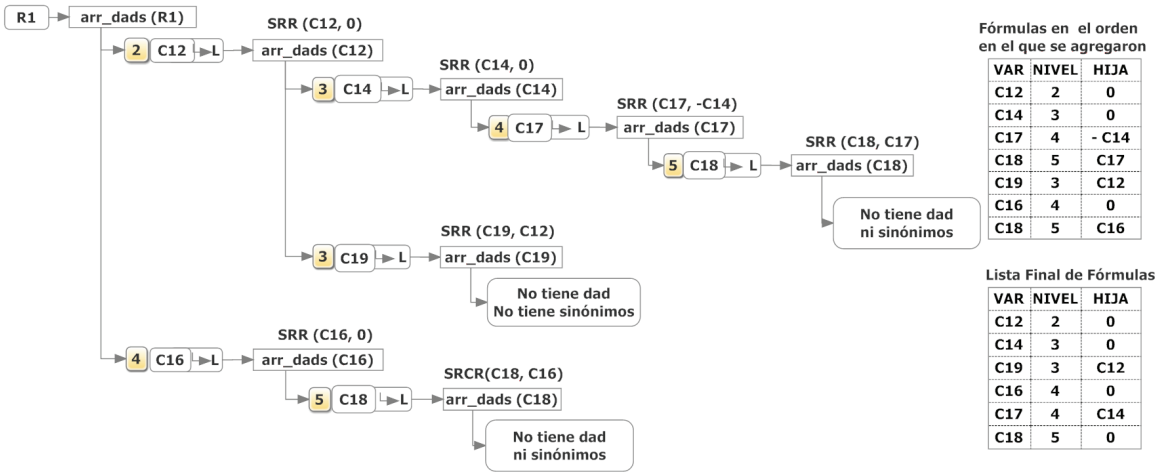



Figura 5. El algoritmo que construye la lista de fórmulas del submodelo de la VMR R1 y la lista de fórmulas que produce, además de la que resulta de ordenarla por nivel y eliminar fórmulas repetidas

- Arr_dads: los "*dads*" de los sinónimos de la variable. Para cada fórmula de la lista se incluye el número de variable, el nivel de la fórmula y la hija que invoca dicha fórmula.
- → L: agregar esta fórmula a la lista de fórmulas del submodelo.
- SRR (Ci, H) invoca la subrutina recursiva con los argumentos:
  Ci= variable
  H = número de la variable que invocó la ejecución de Ci.

Los valores del parámetro H son:

- Cero si H no es operando aditivo.
- Si es sumando, se usa el número de la variable.
- Si es restando (sustraendo) se cambia el signo del número de la variable.

En el diagrama de la figura 5 se muestra la lista de fórmulas conforme se agregaron al submodelo, esto es, en el orden en el que se generaban. También se agregó una

lista final de fórmulas, la cual debe tener las siguientes características:

- Se especifica "la hija" si esta interviene en modo aditivo en la fórmula que la invoca (de lo contrario, se indica hija = 0). También se mencionó que las VMR no se consideran aditivas.
- Las fórmulas deben estar en orden ascendente de nivel. Observe que los niveles de las fórmulas se calcularon previamente y se registraron en el arreglo de fórmulas.
- Si una fórmula se repite en este arreglo, se elimina la repetida y se indica "hija = 0". Se podría invocar dos veces el cálculo si ambas tuvieran hijas aditivas, pero se decidió no contemplar esta situación (ver C18 en el modelo de la figura 3).

## Ejecución de modelos

La ejecución de las fórmulas consiste en:

- Leer los valores de la variable a calcular.
- Efectuar las operaciones de cada subtotal, para cada subtotal se cargan los valores de los operandos.
- Cuando se haya calculado el último subtotal, se regraban los valores de la variable calculada con los del último subtotal (figura 6).

Si la fórmula tiene la indicación de que la invocó un operando aditivo, no se efectúan las operaciones: solamente se aplica la diferencia de valores de la variable hija a los de la variable calculada (esa diferencia se ha calculado previamente en el mismo proceso).

## Simulación para cuantificar la reducción de las duraciones de los procesos

Para determinar el impacto de las mejoras sobre la duración de la ejecución de los modelos se formularon dos: Mod1 con 44 variables y 33 fórmulas y Mod2 con 332 variables y 264 fórmulas. En ambos se incluyeron VMRs con sinónimos.

Se mencionó que uno de los aspectos que se contemplaron para reducir el tiempo de proceso era minimizar las operaciones de entrada y salida. Cuando se usan los valores de una variable (como variable a calcular o como operando) se dejan en memoria para evitar obtenerlas nuevamente de disco, cuando aparezcan como operandos en otra fórmula como sinónimos (un operando es un sinónimo de la variable en cuestión). Esto es lo que se indica como "guardar valores ya leídos".

Las ejecuciones son demasiado breves para comparar los algoritmos (duran pocos milisegundos) por lo que se invocaron ambos modelos 1000 veces. Además se determinó la duración de ejecutar el Mod1 500 veces y el Mod2 500 veces (para reflejar una situación en la que hay que recalcular modelos de distintas dimensiones). Para comprobar que la duración de N ejecuciones del mismo modelo es lineal (es N veces mayor a la de una sola ejecución del modelo), se ejecutaron los modelos 500 veces y las duraciones eran prácticamente la mitad de las correspondientes a 1000 repeticiones. Las duraciones (en segundos con 3 decimales) que se muestran en la figura 6 se midieron desde el momento en que inicia el cálculo del primer modelo hasta el fin del proceso de todos los modelos. Para determinar el impacto de cada una de las mejoras incorporadas a la nueva versión, se repitieron los procesos, incluyendo y excluyendo cada una de ellas.

En la Figura 7 se muestran los resultados de todas las combinaciones de opciones. En el uso de este programa, se ejecuta la simulación para cada combinación de opciones y se agregan las duraciones a un arreglo como el que se muestra en la imagen.

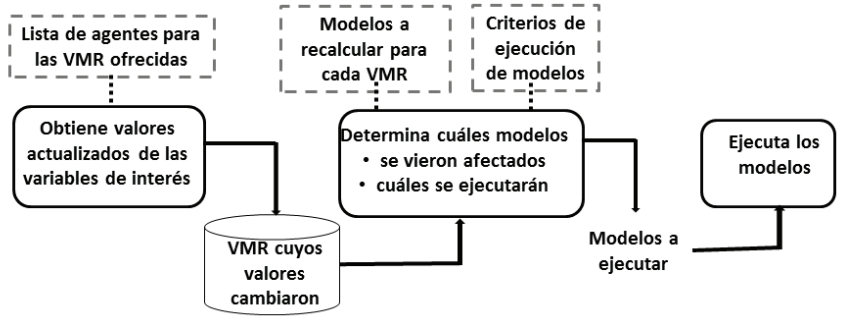
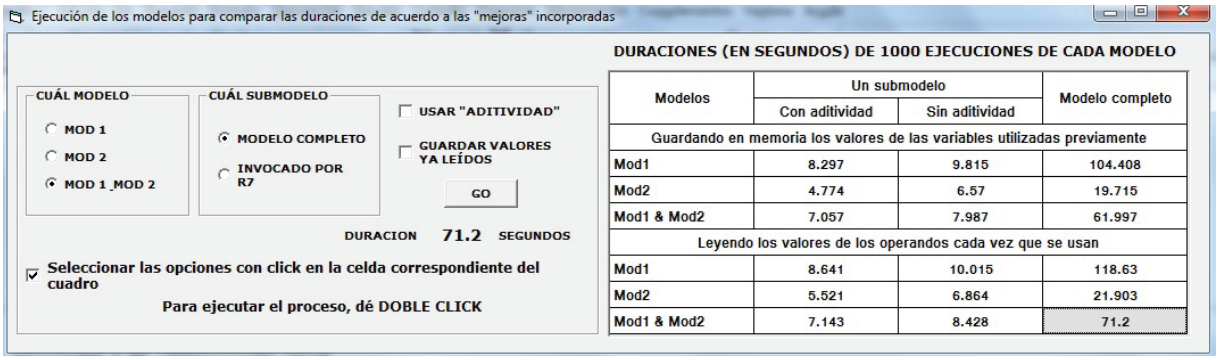



Figura 6. El motor del FLAG formado por tres procesos




DURACIONES (EN SEGUNDOS) DE 1000 EJECUCIONES DE CADA MODELO

| Modelos | Un submodelo | | Modelo completo |
|---|---|---|---|
| | Con aditividad | Sin aditividad | |
| Guardando en memoria los valores de las variables utilizadas previamente | | | |
| Mod1 | 8.297 | 9.815 | 104.408 |
| Mod2 | 4.774 | 6.57 | 19.715 |
| Mod1 & Mod2 | 7.057 | 7.987 | 61.997 |
| Leyendo los valores de los operandos cada vez que se usan | | | |
| Mod1 | 8.641 | 10.015 | 118.63 |
| Mod2 | 5.521 | 6.864 | 21.903 |
| Mod1 & Mod2 | 7.143 | 8.428 | 71.2 |

Figura 7. Forma utilizada para invocar las ejecuciones de las fórmulas de los modelos con las mejoras incorporadas para reducir las duraciones de los procesos

## Conclusiones

Las simulaciones indican que se logró el objetivo principal de reducir significativamente la duración en las ejecuciones. Por un lado, los cambios hechos para evitar la lectura repetida de valores de operandos por sí solos redujeron las duraciones en aproximadamente 10%. El uso de los submodelos naturalmente produce una reducción mucho mayor, en especial cuando el modelo tiene operandos aditivos o cuando tiene muchas fórmulas. De ese modo, esta versión del paquete FLAG es mucho más eficiente que su predecesora, lo que a su vez, significa que se podría ofrecer el servicio a un número mayor de clientes.

Los algoritmos utilizados para crear y ejecutar los submodelos son aplicables a otro tipo de modelos, lo que le da generalidad al trabajo realizado. En particular, el nombre del FLAG surgió de la conveniencia de poder ofrecer ese servicio a agronegocios, pero ningún componente del servicio (ni de los aspectos técnicos) limita su aplicación a cualquier otra situación: la naturaleza de los modelos que permiten utilizar submodelos reside en su estructura, y no depende del tipo de fórmulas o de las operaciones que se implementan.

## Referencias


Bauer-Mengelberg J.R. An informing service based on models defined by its clients. *Informing Science: the International Journal of an Emerging Transdiscipline,* volumen 13, 2010: 87-119.

Bauer-Mengelberg J.R. y Velázquez G.D. Timely informing clients of the impact of changes in their business environment. *Issues in Informing Science and Information Technology,* volumen 7, 2010: 377-392.

Decision Models, *Array formulae* [en línea] [fecha de consulta: 16 de febrero de 2014]. Disponible en: http://www. decisionmodels.com/optspeedj.htm

Decision Models, *Excel's smart recalculation*, 2008 [en línea] [fecha de consulta: 4 de junio de 2014]. Disponible en: http://www.decisionmodels.com/calcsecrets.htm

Gackowski Z. Informing systems in business environments: a purpose-focused view. *Informing Science Journal,* volumen 8, 2005: 101-122.

RAE. Diccionario de la Real Academia Española, *Definición: "Fórmula"*, 2014 [en línea] [fecha de consulta: 19 de febrero de 2014]. Disponible en: http://lema.rae.es/drae/?val= fórmula.

Vbforums. Specify a range for worksheet_calculate method, 2011, [en línea] [fecha de consulta: 2 de junio de 2014]. Disponible en: http://www.vbforums.com/showthread.php? 659881-RESOLVED-Specify-a-range-for-worksheet_calculate-method

Weiss M.A. *Data structures and algorithm analysis,* 2a ed., Reedwood City, The Benjamin/Cummings Publishing Company, 1996, pp. 87-93, 133-138.



**Este artículo se cita:**

**Citación estilo Chicago**

Bauer-Mengelberg Juan Ricardo, Rafael Vega-Ruiz. Algoritmo que ejecuta submodelos de un modelo matemático para mayor eficiencia. *Ingeniería Investigación y Tecnología*, XVI, 04 (2015): 551-563.

**Citación estilo ISO 690**

Bauer-Mengelberg J.R., Vega-Ruiz R. Algoritmo que ejecuta submodelos de un modelo matemático para mayor eficiencia. *Ingeniería Investigación y Tecnología*, volumen XVI (número 4), octubre-diciembre 2015: 551-563.

## Semblanzas de los autores

*Juan Ricardo Bauer-Mengelberg*. Licenciado en matemáticas por la Universidad de Buenos Aires, Argentina. PhD en estadística e investigación de operaciones por la University of Wisconsin, Madison donde impartió cursos de programación estocástica en la misma universidad. En México, además de ser profesor del Colegio de Posgraduados, una institución dedicada a la enseñanza e investigación en agronomía, pero que cuenta con departamentos de estadística y computación aplicada, ha ocupado diversos puestos siempre en el campo de sistemas de información, tema del cual ha sido consultor toda su vida profesional.

*Rafael Vega-Ruiz*. Ingeniero egresado de la carrera de irrigación por la Universidad Autónoma Chapingo. Obtuvo el grado de maestría en ciencias en el Colegio de Postgraduados con la tesis titulada: "Desarrollo completo del sistema FLAG". Actualmente es consultor en sistemas de información.