



Universidad
Politécnica
de Cartagena



FACULTAD DE
CIENCIAS DE LA
E M P R E S A

LA HOJA DE CÁLCULO COMO HERRAMIENTA PARA INVESTIGACIÓN OPERATIVA

Javier Ferreira Pintado

Curso 2014/15

Director/es: *Juan Francisco Sánchez García*

Roberto Javier Cañavate Bernal

Trabajo Fin de Grado para la obtención del título de
Graduado/a en Administración y Dirección de Empresas

ÍNDICE

1. Introducción.....	2
2. Programación Lineal	4
3. Software existente	5
3.1. WinQSB.....	6
3.2. SOLVER.....	7
3.3. PHP Simplex.....	8
4. Modelo propio	9
4.1. Visión general.....	9
4.2. Uso	13
5. Ejemplos numéricos	20
5.1. Ejemplo de un problema tipo 2, problema con soluciones ilimitadas o con solución no acotada.....	20
5.2. Ejemplo de un problema tipo 3, problema con solución única	21
5.3. Ejemplo de un problema tipo 4, problema con solución óptima múltiple	24
6. Conclusiones	27
7. Bibliografía.....	29
8. Anexos.....	30
8.1. Anexo 1. (Rutina Empezar).....	30
8.2. Anexo 2. (Rutina crearTabla).....	30
8.3. Anexo 3. (Rutina Resolver).....	37
8.4. Anexo 4. (Rutina crearTabla2).....	39
8.5. Anexo 5. (Rutina tablaFinal).....	46

1. Introducción

En el presente Trabajo Fin de Grado se ha programado un sistema para resolver problemas de programación lineal con Excel. La idea surgió durante el desarrollo de la asignatura Investigación Operativa (tercer curso del Grado en ADE), ya que los estudiantes teníamos algunos problemas para instalar el software WinQSB en nuestros ordenadores personales debido a que este programa solo funciona bajo sistemas operativos Windows de 32 bits, sistema operativo que ya no suele utilizarse debido a la constante mejora de las computadoras y la aparición de sistemas operativos Windows de 64 bits. Es por ello que para ser utilizado en la mayoría de los equipos informáticos actuales se precisa de la utilización de máquinas virtuales con las que “virtualizar” un sistema operativo Windows de 32 bits, algo que el usuario medio o bien no sabe que se puede hacer o bien no es capaz de llevarlo a cabo en la mayoría de los casos.

Se trata de un problema presente en la docencia de numerosas titulaciones superiores, entre las que se incluyen diferentes grados de la Universidad Politécnica de Cartagena, ya que WinQSB es un programa utilizado por diferentes ingenierías como herramienta para resolver y modelar problemas complejos de optimización de recursos, y, por tanto, no es exclusivo de la asignatura de Investigación Operativa del Grado en Administración y Dirección de Empresas. Con la finalidad de solventar dicho problema nos planteamos la posibilidad de buscar una alternativa de software que permitiera resolver problemas de programación lineal del modo que lo hacía WinQSB, pero sin tener que precisar del uso de terceras herramientas que pudieran suponer una barrera tecnológica insalvable para muchos usuarios. Y dado que en las asignaturas de Informática de Gestión del Grado en ADE hemos aprendido que el software Microsoft Excel no solo tiene una gran potencia sino que se encuentra al alcance de la mayoría de usuarios informáticos, ¿por qué no intentar utilizarlo para solucionar problemas de programación lineal de una forma didáctica, como lo hace WinQSB?

Partiendo de esta idea, el presente Trabajo Fin de Grado tiene como objetivo principal dar solución a través del programa Microsoft Excel a problemas de programación lineal, pero de un modo didáctico similar al que emplea WinQSB, ya que mostrando todos los resultados intermedios, las iteraciones necesarias y la información que pueda ser útil para entender el método del *símplex*, se conseguiría una gran utilidad para su utilización en

docencia puesto que ayudaría a los estudiantes a entender mejor el funcionamiento del algoritmo.

Para ello hemos integrado en una hoja de cálculo de Excel el algoritmo Símplex, creando un programa automatizado en VBA (Visual Basic para Aplicaciones) para Excel, con el que hemos implementado el algoritmo, de tal forma que el usuario solo deberá conocer la información matemática que desea introducir y el programa realizará todo el proceso adaptándose perfectamente a la misma. Como se ha mencionado, se pretende que el software tenga un notable enfoque pedagógico que ayude al estudiante a familiarizarse con las diferentes etapas del método del símplex y no únicamente con su solución final.

Otro de los elementos que nos parecieron interesantes es la posibilidad de puesta en común de diversas disciplinas estudiadas durante el Grado en Administración y Dirección de Empresas, como Informática de Gestión y Matemáticas, integrándolas dentro de un mismo proyecto.

Finalmente, se tuvo en consideración que gracias a la versatilidad de Microsoft Excel podrá ser utilizado perfectamente en cualquier sistema operativo e incluso en dispositivos móviles, dando solución a uno de los problemas iniciales.

2. Programación Lineal

La Investigación Operativa es una disciplina científica moderna que utiliza métodos matemáticos, estadísticos y algoritmos para modelar y resolver problemas complejos de optimización de recursos, determinando la solución óptima y ayudando en la toma de decisiones. Su principal objetivo es obtener soluciones significativamente más eficientes (en tiempo, recursos, beneficios, costes, etc.) en comparación con aquellas decisiones tomadas de forma intuitiva o sin el apoyo de herramientas destinadas a ello.

La Programación Lineal es una rama de la Investigación Operativa que se caracteriza porque todas las expresiones matemáticas tienen carácter lineal, lo que permite la existencia de modos exactos y generales de resolución, siendo el más conocido el método del *símplex*.

El método *Símplex* es un procedimiento que fue desarrollado por George Dantzing en 1947, es el más utilizado para resolver problemas de programación lineal y es el que mayoritariamente se estudia en la enseñanza superior, de ahí que sea el que hemos escogido para su programación en Microsoft Excel. La primera implementación en una computadora de un problema de programación lineal fue en el año 1952 para resolver un problema de 71 variables y 48 ecuaciones para la que se necesitaron aproximadamente 18 horas.

El algoritmo *Símplex* es un procedimiento iterativo que se inicia con una solución válida, pero no necesariamente óptima y genera otras soluciones factibles cada vez mejores hasta encontrar la solución óptima (siempre que exista). De esta forma el algoritmo *Símplex* asegura que la solución no “empeora”, y por tanto, la siguiente solución básica factible (así se denominan las soluciones factibles que se van alcanzando con el algoritmo *Símplex*) es igual o mejor que la anterior, esto es, que el valor objetivo aumenta si el problema es de maximización, o disminuye si el problema es de minimización. El proceso concluye cuando no es posible continuar mejorando dicho valor, es decir, el valor objetivo es óptimo y, la solución correspondiente, también es óptima.

El método *Símplex* se puede utilizar para resolver cualquier problema de programación lineal de forma exacta, algunos de los más conocidos son:

- Problema de la dieta
- Problemas de transporte de tropas
- Problema del transporte de mercancías
- Problema de los árboles frutales
- Problema de la asignación del personal
- Problemas del camino mínimo
- Problemas de localización
- Problema de inversión en bolsa

3. Software existente

Actualmente existen diferentes softwares para la resolución de problemas de programación lineal, entre los que podemos destacar los siguientes:

3.1. WinQSB

WinQSB o QSB (Quantitative System Business), es un paquete de herramientas desarrolladas por el Dr. Yih-Long Chang, y es de gran ayuda para la toma de decisiones ya que contiene herramientas muy útiles para resolver distintos tipos de problemas en el campo de los Métodos Cuantitativos aplicados a la Economía y la Empresa: problemas administrativos, de producción, de recursos humanos, dirección de proyectos, etc.

Se encuentra muy extendido en la enseñanza superior debido a su facilidad de uso y sus características didácticas, como resoluciones iteración a iteración, completa información sobre las variables y restricciones en la función objetivo, etc.

Una limitación que presenta WinQSB es su funcionamiento solo bajo sistemas operativos Windows de 32 bits, sistema operativo que ya no suele utilizarse debido a la constante mejora de las computadoras y la aparición de sistemas operativos Windows de 64 bits.

El programa se encuentra formado por distintos módulos, uno para cada tipo de modelo o problema. Entre ellos se pueden destacar los siguientes:

- Linear programming and integer linear programming: Resolver problemas de programación lineal a través del algoritmo Símples (o gráficamente cuando el problema tiene únicamente dos variables) y problemas de programación lineal entera mediante el algoritmo de ramificación y acotación.
- Linear goal programming and integer linear goal programming: Resolver modelos de programación multiobjetivo con restricciones lineales.
- Quadratic programming and integer quadratic programming: Resolver problemas de programación cuadrática, es decir, problemas con función objetivo cuadrática y restricciones lineales.

- Network modeling: Resolver problemas de transporte, de asignación, del camino más corto, etc.
- Nonlinear programming: Resolver problemas no lineales sin restringir utilizando métodos de búsqueda lineal.
- PERT/CPM: Módulo de gestión de proyectos en los que hay que realizar varias actividades con relaciones de precedencia.

3.2. SOLVER

Dentro del programa Microsoft Excel es posible encontrar un poderoso complemento, Solver, el cual forma parte del conjunto de comandos conocidos como “what-if analysis tools”. Es un software de optimización que permite encontrar un valor óptimo (máximo o mínimo) en una celda denominada la celda objetivo, modificando otras celdas que se encuentren relacionadas entre sí mediante fórmulas, que estarán sujetas a restricciones o limitaciones, llamadas celdas de variables de decisión.

Microsoft Excel Solver utiliza diferentes algoritmos para encontrar soluciones óptimas entre los que destacan el código de optimización no lineal GRG2 (Generalized Reduced Gradient) desarrollado por la Universidad Leon Lasdon de Austin (Texas) y la Universidad Allan Waren (Cleveland), y el método Símples para la resolución de problemas de programación lineal implementados por John Watson y Daniel Fylstra, ambos desarrollados por la empresa Frontline System, Inc.

No obstante, Solver también presenta algunas limitaciones en su ejecución:

- Limitación en el número de restricciones. A la hora de asignar las restricciones al modelo, solo permite utilizar un máximo de 200 celdas para ello.
- En el caso de encontrarse con un problema de múltiples soluciones óptimas solo mostrará una de ellas, sin proporcionar ningún aviso de que existen más soluciones.
- Carece de cualidades pedagógicas dado que solo proporciona la solución óptima pero no las tablas y procedimientos intermedios.

3.3. PHP Simplex

PHP Simplex es una herramienta online diseñada para resolver problemas de programación lineal, siendo su uso libre y gratuito. Es posible acceder a ella a través de la siguiente URL: <http://www.phpsimplex.com/>.

Al tratarse de una herramienta pensada para ayudar a los estudiantes en su aprendizaje, no solo muestra los resultados finales sino que también muestra las operaciones intermedias para hacer más cómoda su comprensión. Además también ofrece la posibilidad de obtener la solución directamente en caso de que se solicite.

Para resolver problemas con dos variables, el programa permite elegir entre dos métodos para su resolución: método Símplex o método gráfico.

PHP Simplex presenta algunas similitudes con la herramienta desarrollada en este trabajo. No obstante, hemos intentado que nuestra implementación mejorara en algunos aspectos lo que se ofrece en la herramienta online, agregando así un valor añadido a la hoja de cálculo de Excel cuando se resuelven problemas de programación lineal. Algunos de estos aspectos son:

- No presenta todas las tablas intermedias en una misma página, limitando y perdiendo el enfoque general del problema. En la nueva hoja de cálculo creada en Excel se visualizan todas las tablas o soluciones factibles posibles dentro de la misma hoja de cálculo.
- PHP Simplex finaliza mostrando la solución óptima sin dar ninguna información adicional sobre las variables o restricciones que no se encuentren en la base. En cambio, con la nueva herramienta desarrollada en este trabajo se realiza un análisis de sensibilidad, que proporciona información sobre todas las variables y restricciones, siendo esta información relevante para la toma de futuras decisiones.

4. Modelo propio

4.1. Visión general

Como se indicaba en la introducción, en el presente Trabajo Fin de Grado se ha implementado el algoritmo Símpex en la hoja de cálculo de Excel utilizando para ello la programación VBA (Visual Basic for Applications) de forma que se consiga tener una herramienta que resuelva este tipo de problemas como en WinQSB, pero desde un punto de vista más pedagógico, con el que dar a conocer el funcionamiento de dicho algoritmo.

La programación que se ha hecho en VBA no resuelve el problema directamente, sino que crea las fórmulas necesarias con las funciones propias de Excel para que se pueda ver claramente de dónde sale cada cálculo, con lo que es a la vez una forma de resolver problemas de programación lineal y una forma de estudiar la versatilidad de las funciones que incorpora la propia hoja de cálculo.

De haberse buscado sólo los cálculos la programación habría sido mucho más sencilla, ya que no se habría tenido que programar todas las fórmulas en las celdas de Excel, incluyendo en cada una de ellas los valores variables a tener en cuenta que se utilizan para llegar a la solución, consiguiendo así que la herramienta se adapte perfectamente a cualquier posible problema.

Al tratarse del algoritmo Símpex, los problemas se deberán introducir en forma “estándar”, esto es, de forma que su modelo matemático cumpla las siguientes condiciones:

- a) Todas las restricciones son igualdades.
- b) Todas las variables son no-negativas.
- c) Las limitaciones (lado derecho de las restricciones) son positivas.

El algoritmo Símpex necesita comenzar en una solución inicial factible, que, en este caso se conseguirá limitando el uso de restricciones al tipo “ \leq ”, añadiendo una variable de holgura para cada una de las restricciones del problema y obteniendo así una base canónica por columnas, la cual proporciona las variables asignadas a la primera solución factible del método.

Por ejemplo, dada la siguiente información:

$$\begin{aligned}
 \text{máx} \quad & z = x_1 + 2x_2 + 4x_3 \\
 \text{s. a} \quad & 3x_1 + x_2 + 5x_3 \leq 10 \\
 & x_1 + 4x_2 + x_3 \leq 8 \\
 & 2x_1 + \quad + 2x_3 \leq 7 \\
 & x_i \geq 0
 \end{aligned}$$

El problema en forma estándar se obtiene simplemente añadiendo una variable de holgura en cada una de las restricciones, con lo que se obtiene:

$$\begin{aligned}
 \text{máx} \quad & z = x_1 + 2x_2 + 4x_3 \\
 \text{s. a} \quad & 3x_1 + x_2 + 5x_3 + x_4 = 10 \\
 & x_1 + 4x_2 + x_3 + x_5 = 8 \\
 & 2x_1 + \quad + 2x_3 + x_6 = 7 \\
 & x_i \geq 0
 \end{aligned}$$

En este ejemplo la tabla inicial del símplex queda

		1	2	4	0	0	0	
	Base	x_1	x_2	x_3	x_4	x_5	x_6	b
0	x_4	3	1	5	1	0	0	10
0	x_5	1	4	1	0	1	0	8
0	x_6	2	0	2	0	0	1	7
	$c_j - z_j$	1	2	4	0	0	0	$z = 0$

cuya base inicial es x_4 , x_5 y x_6 puesto que las columnas correspondientes de la tabla forman una base canónica por columnas:

$$\begin{matrix}
 & x_4 & x_5 & x_6 \\
 \begin{pmatrix}
 1 & 0 & 0 \\
 0 & 1 & 0 \\
 0 & 0 & 1
 \end{pmatrix}
 \end{matrix}$$

Si no existiese una base canónica por columnas no se podría aplicar el algoritmo Símplex a este problema, debiendo solucionarse mediante otro método denominado “el método del símplex con variables artificiales”, “método de las dos fases” o “método de la M

grande”, que no van a ser tratados en el presente trabajo debido a las limitaciones propias de los Trabajos Fin de Grado.

Dado que cada tabla del método está asociada a una posible solución, es decir, una solución básica (vértice), es necesario comprobar mediante el test de optimalidad si se trata de la solución óptima o existe alguna mejor, en cuyo caso se procedería a cambiar de base.

El proceso para el cambio de base (y por tanto de tabla) consiste en sacar una variable de la base (variable básica) e introducir otra que no estuviese en ella (variable no básica) y para ello se tienen en cuenta los siguientes criterios:

- Si se trata de un problema de minimización se tomará la variable no básica con el menor valor de $c_j - z_j$.
- Si se trata de un problema de maximización se tomará la variable no básica con el mayor valor de $c_j - z_j$.

Si todos los valores de la columna de la tabla correspondiente a la variable escogida para entrar en la base son negativos o cero entonces el problema tiene soluciones ilimitadas (Véase en la página 12), terminando así el algoritmo. En otro caso, la variable básica que es elegida para salir de la base es aquella que posea el menor valor de la columna “ratio”.

El valor “ratio” se calcula dividiendo el valor “b” de cada fila entre el valor de la columna de la tabla escogida que corresponda a su misma fila (siempre que sea estrictamente mayor que cero, en caso contrario el valor “ratio” es ∞), determinando así la variable que sale de la base (fila) y la variable que entrará en su lugar (columna). El elemento que se encuentra en dicha intersección entre ambas variables (fila y columna) dentro de la tabla es denominado “elemento pivote”.

La siguiente tabla del símplex es obtenida realizando el cambio de base determinado y pivotando con la tabla del símplex anterior. Este proceso se realiza de forma iterativa hasta que se obtenga una tabla que contenga una solución óptima y de este modo el algoritmo Símplex se daría por finalizado.

Es posible clasificar los problemas de programación lineal en cuatro tipos diferentes si atendemos al tipo de soluciones que poseen:

- **Tipo 1: Problemas infactibles.** En este tipo de problema ocurre que el conjunto factible es vacío, es decir, no existe ninguna posibilidad válida y por tanto no puede haber ninguna que sea la mejor. Este tipo de problemas han sido descartados al pedir que las restricciones sean " \leq ", ya que la consecuencia es que al utilizar variables de holgura se obtiene una "solución básica factible" o "solución inicial posible", o lo que es lo mismo, un vértice de la región factible. De ese modo, si hay vértice significa que la región factible no es vacía porque tiene al menos una solución (ese vértice) y por tanto este caso nunca puede aparecer en nuestra programación.
- **Tipo 2: Problemas con soluciones ilimitadas o con solución no acotada.** En esta ocasión existen soluciones válidas, pero no es posible elegir la mejor de entre ellas. La razón de esto es que siempre que se elige una solución habrá otra que sea mejor que la anterior. Este tipo de solución está asociada al concepto de que el valor óptimo tiende a " $+\infty$ " o " $-\infty$ " dependiendo del objetivo de la función.
- **Tipo 3: Problemas con solución óptima única.** Existe una solución óptima y además es la única que alcanza el valor óptimo en la función objetivo, es decir, cualquier otra solución siempre es peor.
- **Tipo 4: Problemas con solución óptima múltiple.** Existe más de una solución óptima, es decir, hay al menos dos soluciones que son ambas igual de óptimas, lo cual quiere decir que tienen el mismo valor en su función objetivo.

Teniendo en cuenta la teoría de Convexidad, un problema lineal puede tener o bien cero soluciones (tipos 1 y 2), o bien exactamente una solución (tipo 3) o bien infinitas soluciones (tipo 4). En este último caso, al utilizar el algoritmo Símplex solo se obtiene una cantidad finita de opciones, debido a que dicho algoritmo solo revisa los vértices de la región factible y por tanto solo proporciona las soluciones óptimas que además son vértices, que en este tipo de problemas serán al menos dos. Es posible hallar de forma matemática todas las soluciones óptimas alternativas realizando la envolvente convexa de todas las soluciones óptimas encontradas con el algoritmo Símplex.

4.2. Uso

El proceso se encuentra implementado por completo dentro de la hoja de cálculo, de tal forma que solo se deberá indicar el número de variables y restricciones que contiene el problema e introducirlas en su lugar correspondiente. El resto es realizado por el programa de forma automática, obteniendo el resultado final y una tabla con información complementaria sobre el análisis de sensibilidad (problemas tipo 3 y 4), además se dispone en una única hoja Excel de todas las tablas e iteraciones intermedias que han sido utilizadas para llegar a la solución final.

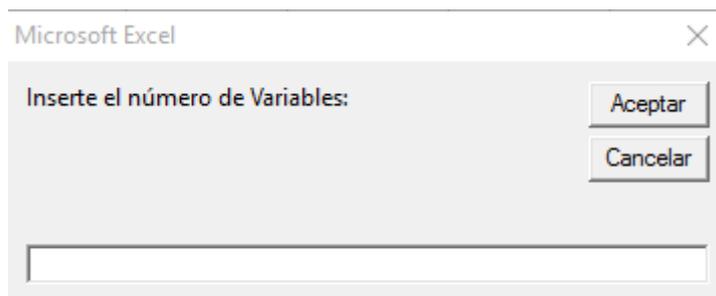
Al abrir el archivo Excel se muestra una hoja en blanco con dos botones en su parte superior izquierda: “Empezar y Resolver”. Para comenzar a utilizar el programa se deberá pulsar sobre el botón “Empezar”.



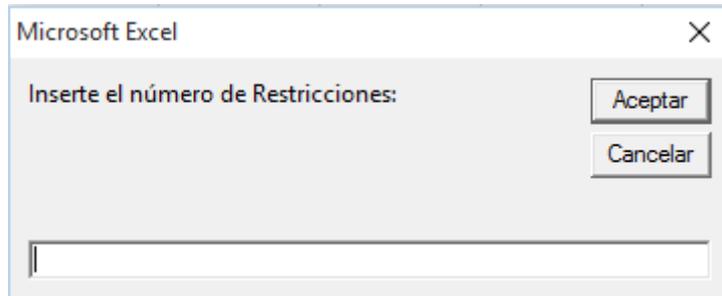
Una vez pulsado el botón, aparecerán diferentes cuadros de diálogo a través de los cuales se navegará por el menú, ejecutándose la rutina “Empezar” creada en VBA e incluida en el [anexo 1](#) de este documento.

La rutina “Empezar” dará comienzo al proceso, creando diferentes cuadros de diálogo que nos preguntarán por el número de variables y restricciones que tendrá el problema, si se trata de un problema de maximización o minimización, y para terminar nos mostrará un mensaje informativo indicándonos los datos introducidos.

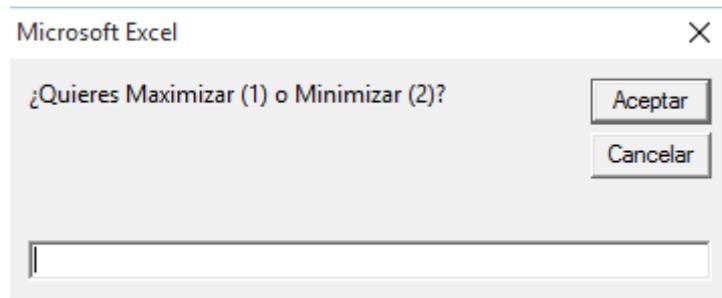
El primer cuadro que aparecerá preguntará por el número de variables que se quieren incluir en el problema.



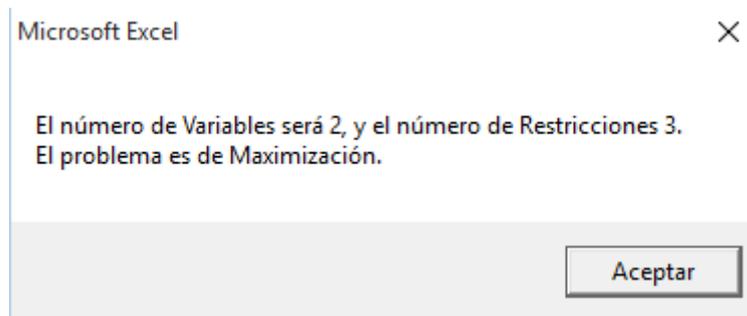
De forma similar al anterior, el programa preguntará por el número de restricciones que se quieren incluir en el problema.



Una vez determinado el número de variables y restricciones procederá a determinar si se trata de un problema de maximización o minimización, escribiendo 1 ó 2 según corresponda. En caso de omitir el valor el problema será de maximización.

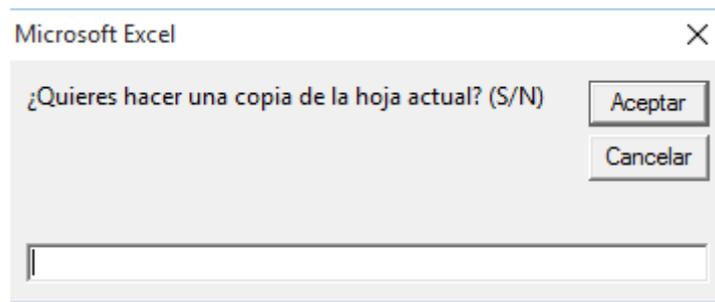


El proceso continúa mostrando otro cuadro informativo de todos los datos introducidos, por ejemplo, si se ha elige un problema donde el número de variables es 2, el número de restricciones es 3 y se trata de un problema de maximización (1), la información se muestra de la siguiente forma:



Si se trata del primer problema que se resuelve quizás no interese hacer una copia de la hoja actual, pero si se encuentra trabajando con diferentes datos el programa da la posibilidad de hacer una copia de la hoja actual y así no perder la información.

Para ello, en el último cuadro se deberá escribir “S” (Sí) para realizar una copia de la hoja o, si por lo contrario no se desea realizar copia, bastará con escribir “N” (No) o dejar el espacio en blanco. En caso de no indicar ninguna opción no se realizará copia.



Una vez terminado con el último cuadro de diálogo, Excel creará la primera tabla del *símplex* a través de la información que se le ha proporcionado. En ese mismo momento se ejecutará de forma automática la rutina “crearTabla” incluida en el [anexo 2](#) de este documento.

La rutina “crearTabla” creará la primera tabla del algoritmo del *símplex*, utilizando para ello la información suministrada anteriormente: el número de variables, el número de restricciones y si se trata de un problema de maximización o minimización.

Empezar		Resolver		29/09/2015 20:15				
Problema de Maximización		Base	x1	x2	x3	x4	b	Ratio
Restricc.1	0	x3			1	0		M
Restricc.2	0	x4			0	1		M
Cj-Zj			0	0	0	0	0	Z

Junto con la tabla anterior también se creará la siguiente tabla auxiliar, proporcionando toda la información referente al elemento pivote una vez se haya rellenado la primera tabla del *símplex* por completo.

Elemento pivote	
Valores	columna
	fila

Como se ha comentado anteriormente, el elemento pivote juega un papel clave en el proceso de cambio de tabla siendo, además, el elemento que señala la variable que sale (la que se encuentra en la fila del pivote) y la que entra (la que se encuentra en la columna del pivote) en la base. Por ello se ha creado una tabla informativa con la que facilitar la comprensión para cada proceso que realice el algoritmo.

Una vez que se encuentre con la primera tabla del símplex, el usuario deberá proceder a introducir la información del problema antes de continuar con el siguiente procedimiento “Resolver”.

Una vez se tenga completado el primer cuadro, y antes de pulsar el botón “Resolver”, se podrá cambiar el nombre de las variables y restricciones por otros, con el fin de que sea más cómoda su identificación y posterior trabajo con ellas.

Para asignar otro nombre a las variables tan solo se deberá de cambiar el nombre que traen por defecto “x1, x2,...” por uno nuevo. Las celdas habilitadas para realizar el cambio son las situadas en la fila horizontal superior, justo a la derecha de la celda “Base”.

Base	x1	x2	x3	x4
------	----	----	----	----

Empezar Resolver 29/09/2015 20:15

Problema de Maximización

	Base	x1	x2	x3	x4	b	Ratio
Restricc.1	0	x3		1	0		M
Restricc.2	0	x4		0	1		M
	Cj-Zj	0	0	0	0	0	Z

Para cambiar el nombre a las restricciones el procedimiento será similar al caso de las variables, la única diferencia es que en esta ocasión las celdas a modificar son las situadas a la izquierda “Restricc.1, Restricc.2,...”, de color azul.

		Base
Restricc.1	0	x3
Restricc.2	0	x4
Restricc.3	0	x5

Empezar Resolver 29/09/2015 20:15

Problema de Maximización

		Base	x1	x2	x3	x4	b	Ratio
Restricc.1	0	x3			1	0		M
Restricc.2	0	x4			0	1		M
Cj-Zj			0	0	0	0	0	Z

Es importante señalar que los cambios de nombres anteriores solo se deben realizar en la primera tabla del algoritmo, es decir, antes de pulsar el botón “Resolver”, ya que la programación repetirá automáticamente los nombres elegidos en el resto de tablas que se generen.

El valor de la función objetivo viene indicado con la letra Z, siendo la suma de cada una de las contribuciones totales que se encuentran en la base de cada tabla. Para cada una de las tablas o posibles soluciones factibles se podrá encontrar dicha información en el siguiente recuadro amarillo:

		Z	
--	--	---	--

Empezar Resolver 29/09/2015 20:15

Problema de Maximización

		Base	x1	x2	x3	x4	b	Ratio
Restricc.1	0	x3			1	0		M
Restricc.2	0	x4			0	1		M
Cj-Zj			0	0	0	0	0	Z

El siguiente paso consiste en introducir en la primera tabla del símplex toda la información referente al problema en su lugar correspondiente. Una vez completada, se procederá a pulsar el botón “Resolver” ejecutando así la siguiente rutina “Resolver” incluida en el [anexo 3](#) de este documento.

La rutina “Resolver” dará comienzo a todas las operaciones necesarias para analizar e identificar de qué tipo de problema se trata y dependiendo de ello iniciará un proceso iterativo u otro, mediante el cual se podrá completar todo el procedimiento hasta alcanzar la solución óptima de una forma automática. Para ello, se apoya en la rutina “crearTabla2”

la cual es ejecutada una vez por cada tabla nueva que necesita el algoritmo. La rutina “crearTabla2” está incluida en el anexo 4 de este documento.

Una vez que el algoritmo ha terminado, dependiendo del tipo de problema encontrado, se procederá a mostrar una información u otra:

Tipo 1: Problemas infactibles. Cómo se ha indicado anteriormente, este tipo de problemas no aparecerán con el procedimiento utilizado en este Trabajo Fin de Grado.

Tipo 2: Problemas con soluciones ilimitadas o con solución no acotada. Cuando el proceso iterativo finalice dando lugar a este tipo de problema, se avisará mediante la siguiente nota informativa.

Nota: El problema tiene solución no acotada

Tipo 3 y 4: Problemas con solución óptima única y con solución óptima múltiple. Una vez que el proceso iterativo anterior concluya dando lugar a la solución óptima del problema, se ejecutará la siguiente rutina “tablaFinal”, incluida en el anexo 5 de este documento, que creará una última tabla con toda la información relevante al problema resuelto.

Dicha tabla está compuesta por 3 partes:

1. **Solución:** Información sobre todas las variables que han sido utilizadas en el problema, incluyendo también las artificiales:
 - Solución: Nombre de las variables
 - Valor: Valor de las variables en la solución óptima
 - Cj (Coste por unidad): Coeficiente de la variable en la función objetivo
 - Contribución: Contribución total de la variable a la función objetivo
 - Bases status: Indica si la variable se encuentra o no en la base
 - Min Cj: Mínimo valor que podría tener Cj, sin que cambie la base óptima (análisis de sensibilidad)
 - Max Cj: Máximo valor que podría tener Cj, sin que cambie la base óptima (análisis de sensibilidad)

Solución	Valor	Cj	Contribución	Bases status	Min Cj	Max Cj
x1						
x2						
x3						
x4						
x5						

2. **Nota informativa:** indicará si existe una única solución óptima o pueden existir diferentes soluciones óptimas alternativas en función del tipo de problema lineal. Una solución óptima alternativa es aquella que, siendo diferente a otra solución óptima ya conocida, su valor en la función objetivo coincide con el valor óptimo de aquélla.

Nota: Existe una única solución

Nota: Pueden existir soluciones alternativas

3. **Cuadro restricciones:** Información sobre todas las restricciones que han sido utilizadas en el problema:
- Restricciones: Nombre de las restricciones
 - LHS (Left Hand Side): Valor de la restricción en la solución óptima sin tener en cuenta la variable de holgura
 - Signo: Signo de la restricción
 - RHS (Right Hand Side): Valor de la restricción en la solución óptima
 - Holgura: Valor de la variable de holgura de la restricción. Puede interpretarse como la cantidad de recurso no utilizado
 - Min RHS: Mínimo valor para b_i sin que cambie la base óptima (análisis de sensibilidad)
 - Max RHS: Máximo valor para b_i sin que cambie la base óptima (análisis de sensibilidad)

Restricciones	LHS	Signo	RHS	Holgura	Min RHS	Max RHS
Restricc.1		<=				
Restricc.2		<=				
Restricc.3		<=				

5. Ejemplos numéricos

5.1. Ejemplo de un problema tipo 2, problema con soluciones ilimitadas o con solución no acotada

Dada la siguiente información, se procederá a la creación de una tabla inicial para un problema de maximización con 2 variables y 2 restricciones:

$$\begin{aligned}
 & \text{Maximizar } 6x_1 + 4x_2 \\
 & \text{s. a. } \quad 2x_1 - 2x_2 + x_3 \leq 6 \\
 & \quad \quad 4x_1 \quad \quad + x_4 \leq 16 \\
 & \quad \quad x_1, x_2 \geq 0, \quad x_1, x_2 \in \mathbb{Z}
 \end{aligned}$$

Los datos serán añadidos en la primera tabla del símplex tal y como se muestra en la siguiente imagen, pudiendo ser modificados el nombre de las variables y restricciones.

Empezar		Resolver		29/09/2015 20:12				
Problema de Maximización		6	4	0	0			
	Base	x1	x2	x3	x4	b	Ratio	
Restricc.1	0	x3	2	-2	1	0	6	3
Restricc.2	0	x4	4	0	0	1	16	4
	Cj-Zj	6	4	0	0	0	Z	

El cuadro informativo referente al elemento pivote está indicando que la variable situada en la columna 1 (x_1) entrará en la base, mientras que la variable en la fila 1 (x_3) saldrá de ella, realizando así el pivote correspondiente.

Elemento pivote	
Valores	columna
2	1
4	fila
	1

Tras rellenar la primera tabla del símplex y pulsar sobre el botón “Resolver”, se puede observar cómo efectivamente ha salido la variable x_3 de la base, entrando en su lugar la variable x_1 , dando como resultado la siguiente tabla:

		6 4 0 0						
		Base	x1	x2	x3	x4	b	Ratio
Restricc.1	6	x1	1	-1	0,5	0	3	M
Restricc.2	0	x4	0	4	-2	1	4	1
Cj-Zj			0	10	-3	0	18	Z

Elemento pivote	
Valores	columna
-1	2
4	fila
	2

Continuando con el mismo procedimiento, en la siguiente tabla del s3mplex la variable que debe salir de la base est3 situada en la fila 2 (x4), siendo la variable de la columna 2 (x2) la que entrar3 en la base, quedando as3 la siguiente tabla:

		6 4 0 0						
		Base	x1	x2	x3	x4	b	Ratio
Restricc.1	6	x1	1	0	0	0,25	4	M
Restricc.2	4	x2	0	1	-0,5	0,25	1	M
Cj-Zj			0	0	2	-2,5	28	Z

Elemento pivote	
Valores	columna
0	3
-0,5	fila

En esta 3ltima tabla se observa que no se puede continuar con el algoritmo S3mplex ya que todas las celdas de la columna "ratio" tienen el valor "M", lo que significa que el problema tiene soluciones ilimitadas y el algoritmo de parar.

Nota: El problema tiene soluci3n no acotada

En esta situaci3n la soluci3n del problema es no acotada y, por tanto, carece de soluci3n 3ptima.

5.2. Ejemplo de un problema tipo 3, problema con soluci3n 3nica

A continuaci3n se realizar3 un ejemplo completo con el fin de explicar mejor el funcionamiento de la herramienta de forma pr3ctica.

Ejemplo: Una fábrica de muebles produce mesas y sillas a precios asequibles. El proceso de producción en ambos casos es similar y requiere un cierto número de horas de mano de obra para la carpintería y otro número de horas para la pintura y el barnizado. Cada mesa requiere 4 horas de trabajo de carpintería y 2 horas de pintura. Cada silla requiere 3 horas de carpintería y 1 hora de pintura. Durante el período de producción actual, las horas de mano de obra disponibles de carpintería y pintura son 240 y 100 horas, respectivamente. Cada mesa genera un beneficio de 7 u.m. y cada silla 5 u.m. El objetivo es determinar la mejor combinación posible de mesas y sillas con el fin de obtener el máximo beneficio posible.

Se comienza formulando matemáticamente la información del texto:

$$\begin{aligned}
 &\text{Maximizar} && 7x_1 + 5x_2 \\
 &\text{s. a.} && 4x_1 + 3x_2 + x_3 \leq 240 \\
 &&& 2x_1 + x_2 + x_4 \leq 100 \\
 &&& x_1, x_2 \geq 0, \quad x_1, x_2 \in \mathbb{Z}
 \end{aligned}$$

En primer lugar, se completará la primera tabla del s mplex con toda la informaci n. Para ello se proceder  a crear una tabla inicial para un problema de maximizaci n con 2 variables y 2 restricciones, siendo posible modificar el nombre de las variables y las restricciones por otros, como puede observarse en la imagen siguiente:

Problema de Maximizaci�n			7	5	0	0		
		Base	mesas	sillas	h1	h2	b	Ratio
h. carpint.	0	h1	4	3	1	0	240	60
h. pintura	0	h2	2	1	0	1	100	50
		Cj-Zj	7	5	0	0	0	Z

Los cambios realizados en los nombres de las variables han sido los siguientes:

X1 = "mesas" (N mero de mesas a fabricar)

X2 = "sillas" (N mero de sillas a fabricar)

X3 = "h1" (Variable de holgura 1)

X4 = "h2" (Variable de holgura 2)

Igualmente se ha modificado el nombre de las restricciones por los siguientes:

Restricc.1 = "h. carpint." (Horas de carpinter a)

Restricc.2 = "h. pintura" (Horas de pintura)

El cuadro informativo referente al elemento pivote está indicando que la variable situada en la columna 1 (x_1) entrará en la base, mientras que la variable situada en la fila 2 (x_4) saldrá de ella, realizándose así el pivote correspondiente.

Elemento pivote	
Valores	columna
4	1
2	fila
	2

Tras rellenar la primera tabla del símplex y pulsar sobre el botón “Resolver”, se puede observar cómo efectivamente ha salido la variable x_4 o “h2” de la base, entrando en su lugar la variable x_1 o “mesas”, dando como resultado la siguiente tabla:

			7	5	0	0		
		Base	mesas	sillas	h1	h2	b	Ratio
h. carpint.	0	h1	0	1	1	-2	40	40
h. pintura	7	mesas	1	0,5	0	0,5	50	100
		Cj-Zj	0	1,5	0	-3,5	350	Z

Elemento pivote	
Valores	columna
1	2
0,5	fila
	1

Continuando con el mismo procedimiento, en la siguiente tabla del símplex la variable que debe salir de la base está situada en la fila 1 (h1), siendo la variable de la columna 2 (sillas) la que entrará en la base, quedando así la siguiente tabla:

			7	5	0	0		
		Base	mesas	sillas	h1	h2	b	Ratio
h. carpint.	5	sillas	0	1	1	-2	40	M
h. pintura	7	mesas	1	0	-0,5	1,5	30	M
		Cj-Zj	0	0	-1,5	-0,5	410	Z

Elemento pivote	
Valores	columna
	fila

En esta última tabla, se observa que la columna “ratio” no permite mejorar la solución de la base indicando “M” en todas sus celdas, por lo que el método Símplex deberá terminar. Al tratarse de una solución del tipo 3, la herramienta mostrará el siguiente cuadro informativo sobre el análisis de sensibilidad, mostrando información sobre todas las variables:

Solución	Valor	Cj	Contribución	Bases status	Min Cj	Max Cj
mesas	30	7	210	Yes	6,666666667	10
sillas	40	5	200	Yes	3,5	5,25
h1	0	0	0	No	-M	0
h2	0	0	0	No	-M	0

La siguiente nota informativa indica el tipo de solución obtenida, siendo en este caso del tipo 3.

Nota: Existe una única solución

Para finalizar, el último cuadro mostrará toda la información relacionada con las restricciones, resultando de gran utilidad a la hora de tomar decisiones.

Restricciones	LHS	Signo	RHS	Holgura	Min RHS	Max RHS
h. carpint.	240	<=	240	0	200	300
h. pintura	100	<=	100	0	80	120

5.3. Ejemplo de un problema tipo 4, problema con solución óptima múltiple

Dada la siguiente información, se procederá a la creación de una tabla inicial para un problema de minimización con 2 variables y 2 restricciones:

$$\begin{aligned}
 & \text{Minimizar} && -x_1 - x_2 \\
 & \text{s. a.} && -x_1 + x_2 + x_3 \leq 3 \\
 & && x_1 + x_2 + x_4 \leq 5 \\
 & && x_1, x_2 \geq 0, \quad x_1, x_2 \in \mathbb{Z}
 \end{aligned}$$

Los datos serán añadidos en la primera tabla del símplex tal y como se muestra en la siguiente imagen, pudiendo ser modificados el nombre de las variables y restricciones.

Empezar Resolver 04/10/2015 13:48

Problema de Minimización

			-1	-1	0	0		
		Base	x1	x2	x3	x4	b	Ratio
Restricc.1	0	x3	-1	1	1	0	3	M
Restricc.2	0	x4	1	1	0	1	5	5
		Cj-Zj	-1	-1	0	0	0	Z

El cuadro informativo referente al elemento pivote está indicando que la variable situada en la columna 1 (x_1) entrará en la base, mientras que la variable en la fila 2 (x_4) saldrá de ella, realizando así el pivote correspondiente.

Elemento pivote	
Valores	columna
-1	1
1	fila
	2

Tras rellenar la primera tabla del símplex y pulsar sobre el botón “Resolver”, se puede observar cómo efectivamente ha salido la variable x_4 de la base, entrando en su lugar la variable x_1 , dando como resultado la siguiente tabla:

			-1	-1	0	0		
		Base	x1	x2	x3	x4	b	Ratio
Restricc.1	0	x3	0	2	1	1	8	M
Restricc.2	-1	x1	1	1	0	1	5	M
		Cj-Zj	0	0	0	1	-5	Z

Elemento pivote	
Valores	columna
	fila

En esta ocasión, tan solo han sido necesarias dos tablas para finalizar el algoritmo, como se observa en la columna “ratio”, esta no permite mejorar la solución de la base indicando “M” en todas sus celdas, por lo que el método Símplex habrá terminado.

Al tratarse de una solución del tipo 4, la herramienta mostrará el siguiente cuadro informativo sobre el análisis de sensibilidad, mostrando información sobre todas variables:

Solución	Valor	Cj	Contribución	Bases status	Min Cj	Max Cj
x1	5	-1	-5	Yes	0	M
x2	0	-1	0	No	-M	-1
x3	8	0	0	Yes	1	M
x4	0	0	0	No	-M	0

La siguiente nota informativa indica el tipo de solución obtenida, siendo en este caso del tipo 4.

Nota: Pueden existir soluciones alternativas
--

Para finalizar, el último cuadro mostrará toda la información relacionada con las restricciones, proporcionando valiosa información que podrá resultar de gran utilidad a la hora de tomar decisiones.

Restricciones	LHS	Signo	RHS	Holgura	Min RHS	Max RHS
Restricc.1	-5	<=	3	8	-5	M
Restricc.2	5	<=	5	0	0	M

6. Conclusiones

Mediante la realización de este Trabajo Fin de Grado se ha creado una herramienta informática que permite resolver problemas de programación lineal con un enfoque pedagógico, que permite entender con claridad la resolución de cualquier tipo de problema de programación lineal ya que muestra al detalle los pasos seguidos en la resolución del mismo, pudiendo ser utilizada para la docencia de algunas asignaturas universitarias.

Gracias a este desarrollo se podría permitir la migración de los docentes y estudiantes de Investigación Operativa desde el actualmente extendido WinQSB u otros programas similares, a otra herramienta de uso común y multiplataforma como es Microsoft Excel, quedando manifiesta la versatilidad que proporciona la hoja de cálculo, ya que permite que la herramienta elaborada pueda ser usada en cualquier sistema operativo que disponga de la suite informática Microsoft Office gracias a la programación en VBA.

Así mismo, al disponer del lenguaje de VBA para Excel, ha sido posible implementar dentro de la hoja de cálculo todo el algoritmo Simplex facilitando así el uso para los estudiantes, ya que la utilización de VBA permite no sólo realizar la programación de la hoja de cálculo sino también generar toda la formulación necesaria para cualquier problema que se desee hacer con ella, quedando de esta forma visible toda la formulación utilizada y no solo el resultado final.

El presente Trabajo Fin de Grado ha dado como resultado la posibilidad de utilizar la herramienta Microsoft Excel con todo su potencial para la resolución de problemas de programación lineal, siendo algo útil y necesario para el aprendizaje de asignaturas en el ámbito universitario, tanto a nivel de grado, como fuera del mismo.

Como futuras líneas de trabajo se proponen:

- La adaptación de la programación para que se puedan resolver problemas de programación lineal con cualquier tipo de restricciones, lo que implica incluir la implementación del algoritmo del simplex con variables artificiales.

- Introducir la opción de resolver aplicando la regla de Bland para evitar el ciclado¹.
- Dar la posibilidad de modificar todos los elementos de un problema ya solucionado, como cambiar coeficientes de la función objetivo o las restricciones, añadir/eliminar variables y/o restricciones, etc.
- Creación de un menú personalizado en la cinta de opciones de Excel para que el funcionamiento de la herramienta sea aún más cómodo e intuitivo para el usuario final.

¹ Ciclado es la posibilidad teórica (pero que raramente ocurre en la práctica) de que el algoritmo del simplex no alcance la solución óptima debido a que se forme un bucle infinito de tablas ninguna de las cuales contiene la solución óptima.

7. Bibliografía

BERNAL GARCÍA, J.J.; SÁNCHEZ GARCÍA, J.F., MARTÍNEZ MARÍA DOLORES, S.M. (2008). *20 herramientas para la toma de decisiones*, Madrid: Especial Directivos.

FRYE, CURTIS D. (2013). *Microsoft Excel 2013: Step by Step*, USA: O'Reilly Media.

HILLIER, F.S. AND LIEBERMAN, G.J. (2001). *Introduction to Operations Research*: McGraw-Hill.

RÍOS INSUA, S. (1990). *Investigación Operativa*: Ed. Centro de Estudios Ramón Areces.

TAHA, H.A. (2010). *Operations Research: an introduction*, University of Arkansas: Pearson Education.

TAYLOR III, B. W. (2013). *Introduction to Management Science*: Pearson Education.

WALKENBACH, J. (2013). *Excel 2013 Power Programming with VBA*, Indianapolis: Wiley Inc.

WINSTON, WAYNE L. (2011). *Microsoft Excel 2010: Data Analysis and Business Modelling*, USA: Microsoft Press.

YIH-LONG CHANG (2003). *WinQSB version 2.0: decision support software for MS-OM*. New York: Wiley Inc.

8. Anexos

8.1. Anexo 1. (Rutina Empezar)

```
Public numfilas As Integer, numColumnas As Integer
Public filaInicial As Integer, columnaInicial As Integer
Public continuar As Boolean, maximizar As Boolean
Public mensajeInicial As String

Sub Empezar()

numColumnas = InputBox("Inserte el número de Variables:")
numfilas = InputBox("Inserte el número de Restricciones:")
maxormin = InputBox("¿Quieres Maximizar (1) o Minimizar (2)?")

If maxormin = 2 Then maximizar = False Else maximizar = True

filaInicial = 4
columnaInicial = 4

mensajeInicial = "El número de Variables será " & numColumnas & ", y el
número de Restricciones " & numfilas & "." & vbCrLf & "El problema es
de "

If maximizar = True Then
    mensajeInicial = mensajeInicial & "Maximización."
Else
    mensajeInicial = mensajeInicial & "Minimización."
End If

MsgBox mensajeInicial

    'Copiar la hoja actual al final

If UCase(InputBox("¿Quieres hacer una copia de la hoja actual? (S/N)"))
= "S" Then

    Sheets("Hojal").Copy After:=Sheets(Worksheets.Count)

End If

    'Seleccionar todo y borrar

Sheets("Hojal").Select
Cells.Select
Selection.Clear
Selection.EntireRow.Hidden = False

Call crearTabla(numfilas, numColumnas, filaInicial, columnaInicial)

End Sub
```

8.2. Anexo 2. (Rutina crearTabla)

```
Sub crearTabla(numfilas As Integer, numColumnas As Integer, filaInicial
As Integer, columnaInicial As Integer)

Application.ScreenUpdating = False
```

```
Worksheets("Hoja1").Select
```

```
'Poner fecha y hora al principio
```

```
Range(Cells(filaInicial - 3, columnaInicial), Cells(filaInicial - 3,  
columnaInicial + numColumnas + numfilas - 1)).Select
```

```
ActiveCell.FormulaR1C1 = "=NOW()"
```

```
With Selection
```

```
    .HorizontalAlignment = xlCenter
```

```
    .VerticalAlignment = xlBottom
```

```
    Selection.Copy
```

```
    Selection.PasteSpecial Paste:=xlPasteValues
```

```
End With
```

```
Selection.Merge
```

```
'Crear fila superior de la tabla "Xs"
```

```
For r = 1 To numColumnas + numfilas
```

```
    Cells(filaInicial - 1, r + columnaInicial - 1).Value = "x" & r
```

```
Next r
```

```
'Poner si se trata de un problema de max or min
```

```
If maximizar = True Then
```

```
    Cells(filaInicial - 2, columnaInicial - 1).FormulaR1C1 = "Problema  
de Maximización"
```

```
Else
```

```
    Cells(filaInicial - 2, columnaInicial - 1).FormulaR1C1 = "Problema  
de Minimización"
```

```
End If
```

```
Range(Cells(filaInicial - 2, columnaInicial - 3), Cells(filaInicial -  
2, columnaInicial - 1)).Select
```

```
With Selection
```

```
    .HorizontalAlignment = xlCenter
```

```
    .Font.Underline = xlUnderlineStyleSingle
```

```
End With
```

```
Selection.Merge
```

```
'Cambiar formato para variables de holgura "hx"
```

```
Cells(filaInicial - 1, columnaInicial + numColumnas + numfilas).Value =  
"b"
```

```
Cells(filaInicial - 1, columnaInicial + numColumnas + numfilas + 1).Value  
= "Ratio"
```

```
Cells(filaInicial - 1, columnaInicial - 1).Select
```

```
With Selection
```

```
    .Value = "Base"
```

```
    Call formato_bordegrueso
```

```
End With
```

'Fila Xs "horizontal" con fondo gris y centrado

```
Range(Cells(filaInicial - 1, columnaInicial - 1), Cells(filaInicial - 1, columnaInicial + numColumnas + numfilas + 1)).Select
  With Selection
    Call fondoGris
    Call negrita
  End With
```

'Crear cuadro "Z"

```
Cells(filaInicial + numfilas, columnaInicial + numColumnas + numfilas + 1).Value = "Z"
```

'Formato Z

```
Range(Cells(filaInicial + numfilas, columnaInicial + numfilas + numColumnas), Cells(filaInicial + numfilas, columnaInicial + numfilas + numColumnas + 1)).Select

  With Selection
    Call formato_bordefino
    Call formato_bordegrueso
    Call fondoAmarillo
    Call negrita
  End With
```

'Crear columna izquierda de la tabla "Xs"

```
Range(Cells(filaInicial, columnaInicial - 1), Cells(filaInicial + numfilas - 1, columnaInicial - 1)).FormulaArray = "=TRANSPOSE(R[-1]C[" & numColumnas + 1 & "]:R[-1]C[" & numColumnas + numfilas + 1 & "])"
```

'Crear columna izquierda de la tabla "Restricciones"

```
For r = 1 To numfilas
  Cells(r - 1 + filaInicial, columnaInicial - 3).Value = "Restricc." & r
Next r
```

'Formato columna "Restricciones"

```
Range(Cells(filaInicial, columnaInicial - 3), Cells(filaInicial + numfilas - 1, columnaInicial - 3)).Select

  With Selection
    Call fondoAzul
    Call formato_bordefino
    Call formato_bordegrueso
  End With
```

'Fila Xs "vertical" con fondo gris y centrado

```
Range(Cells(filaInicial, columnaInicial - 1), Cells(filaInicial + numfilas - 1, columnaInicial - 1)).Select

  With Selection
    Call fondoGris
    Call negrita
    Call formato_bordefino
    Call formato_bordegrueso
  End With
```

```
End With

'Formato celda "Cj-Zj"

Cells(filaInicial + numfilas, columnaInicial - 1).Select

With Selection
    .Value = "Cj-Zj"
    Call negrita
End With

'Crear cuadro "Elemento Pivote"

Cells(filaInicial - 2, columnaInicial + numColumnas + numfilas + 3).Value
= "Elemento pivote"

Range(Cells(filaInicial - 2, columnaInicial + numColumnas + numfilas +
3), Cells(filaInicial - 2, columnaInicial + numColumnas + numfilas +
4)).Select

With Selection
    Call fondoNaranja
    Call negrita
    Call formato_bordegrueso
End With
Selection.Merge

Cells(filaInicial - 1, columnaInicial + numColumnas + numfilas +
3).Select

With Selection
    .Value = "Valores"
    Call fondoGris
    Call negrita
End With

Cells(filaInicial - 1, columnaInicial + numColumnas + numfilas +
4).Select
With Selection
    .Value = "columna"
    Call fondoGris
    Call negrita
End With

Cells(filaInicial + 1, columnaInicial + numColumnas + numfilas +
4).Select
With Selection
    .Value = "fila"
    Call fondoGris
Call negrita
End With

'Formato "valores"

Range(Cells(filaInicial - 1, columnaInicial + numColumnas + numfilas +
3), Cells(filaInicial + numfilas - 1, columnaInicial + numColumnas +
numfilas + 3)).Select

With Selection
    Call formato_bordefino
    Call formato_bordegrueso
```

End With

'Formato "columna"

Range(Cells(filaInicial - 1, columnaInicial + numColumnas + numfilas + 4), Cells(filaInicial, columnaInicial + numColumnas + numfilas + 4)).Select

With Selection

Call formato_bordefino

Call formato_bordegrueso

End With

'Formato "fila"

Range(Cells(filaInicial + 1, columnaInicial + numColumnas + numfilas + 4), Cells(filaInicial + 2, columnaInicial + numColumnas + numfilas + 4)).Select

With Selection

Call formato_bordefino

Call formato_bordegrueso

End With

'Centrar Rango tabla principal completa

Range(Cells(filaInicial - 2, columnaInicial - 3), Cells(filaInicial + numfilas, columnaInicial + numfilas + numColumnas + 4)).Select

With Selection

.HorizontalAlignment = xlCenter

End With

'Primera fila "Función Objetivo". Borde grueso, cuadrados interiores y color naranja

Range(Cells(filaInicial - 2, columnaInicial), Cells(filaInicial - 2, columnaInicial + numfilas + numColumnas - 1)).Select

With Selection

Call formato_bordefino

Call formato_bordegrueso

Call fondoNaranja

End With

'Columna izquierda "Valores que entran a la tabla". Borde grueso, cuadrados interiores y color naranja

Range(Cells(filaInicial, columnaInicial - 2), Cells(filaInicial + numfilas - 1, columnaInicial - 2)).Select

With Selection

Call formato_bordefino

Call formato_bordegrueso

Call fondoNaranja

End With

'Formato cuadro de las variables

Range(Cells(filaInicial - 1, columnaInicial), Cells(filaInicial + numfilas - 1, columnaInicial + numfilas + numColumnas - 1)).Select

```
With Selection
    Call formato_bordefino
    Call formato_bordegrueso
End With
```

'Columna "b"

```
Range(Cells(filaInicial - 1, columnaInicial + numfilas + numColumnas),
Cells(filaInicial + numfilas - 1, columnaInicial + numfilas +
numColumnas)).Select
```

```
With Selection
    Call formato_bordefino
    Call formato_bordegrueso
End With
```

'Columna "Ratio"

```
Range(Cells(filaInicial - 1, columnaInicial + numfilas + numColumnas +
1), Cells(filaInicial + numfilas - 1, columnaInicial + numfilas +
numColumnas + 1)).Select
```

```
With Selection
    Call formato_bordefino
    Call formato_bordegrueso
End With
```

'Fila inferior "Entre Cj-Zj y Z"

```
Range(Cells(filaInicial + numfilas, columnaInicial), Cells(filaInicial
+ numfilas, columnaInicial + numfilas + numColumnas - 1)).Select
```

```
With Selection
    Call formato_bordefino
    Call formato_bordegrueso
End With
```

'Crear tabla Base Canónica inicial

```
For r = 1 To numfilas
    For s = numColumnas + 1 To numColumnas + numfilas
        If r = s - numColumnas Then
            Cells(r + filaInicial - 1, s + columnaInicial -
1).Value = 1
        Else
            Cells(r + filaInicial - 1, s + columnaInicial -
1).Value = 0
        End If
    Next s
Next r
```

'Formula Entre "Cj-Zj y Z"

```
For r = 1 To numfilas + numColumnas
```

```
Cells(filaInicial + numfilas, r + columnaInicial - 1).FormulaR1C1 = "R[" & -numfilas - 2 & "]C-SUMPRODUCT(R[" & -numfilas & "]C[" & -(r + 1) & "]:R[-1]C[" & -(r + 1) & "], R[" & -numfilas & "]C:R[-1]C)"
```

Next r

'Formula "Z"

```
Cells(filaInicial + numfilas, columnaInicial + numColumnas + numfilas).FormulaR1C1 = "=SUMPRODUCT(R[" & -numfilas & "]C[" & (-numfilas - numColumnas - 2) & "]:R[-1]C[" & (-numfilas - numColumnas - 2) & "], R[" & -numfilas & "]C:R[-1]C)"
```

'Poner 0 en la función objetivo (base canónica, siempre es 0)

```
Range(Cells(filaInicial - 2, columnaInicial + numColumnas), Cells(filaInicial - 2, columnaInicial + numfilas + numColumnas - 1)).Value = 0
```

'Poner 0 en columna izquierda (siempre es 0 en la primera tabla)

```
Range(Cells(filaInicial, columnaInicial - 2), Cells(filaInicial + numfilas - 1, columnaInicial - 2)).FormulaArray = "=TRANSPOSE(R[-2]C[" & numColumnas + 2 & "]:R[-2]C[" & numfilas + numColumnas + 1 & "])"
```

'Formulas cuadro "ELEMENTO PIVOTE"

'Formula columna "Valores"

If maximizar Then

```
Range(Cells(filaInicial, columnaInicial + numfilas + numColumnas + 3), Cells(filaInicial + numfilas - 1, columnaInicial + numfilas + numColumnas + 3)).FormulaArray = "=IF(MAX(R[" & numfilas & "]C[" & -numfilas - numColumnas - 3 & "]:R[" & numfilas & "]C[-4])>0,INDEX(RC[" & -numfilas - numColumnas - 3 & "]:R[" & numfilas & "]C[-4]),,MATCH(MAX(R[" & numfilas & "]C[" & -numfilas - numColumnas - 3 & "]:R[" & numfilas & "]C[-4]),R[" & numfilas & "]C[" & -numfilas - numColumnas - 3 & "]:R[" & numfilas & "]C[-4]),0)),""")"
```

Else

```
Range(Cells(filaInicial, columnaInicial + numfilas + numColumnas + 3), Cells(filaInicial + numfilas - 1, columnaInicial + numfilas + numColumnas + 3)).FormulaArray = "=IF(MIN(R[" & numfilas & "]C[" & -numfilas - numColumnas - 3 & "]:R[" & numfilas & "]C[-4])<0,INDEX(RC[" & -numfilas - numColumnas - 3 & "]:R[" & numfilas & "]C[-4]),,MATCH(MIN(R[" & numfilas & "]C[" & -numfilas - numColumnas - 3 & "]:R[" & numfilas & "]C[-4]),R[" & numfilas & "]C[" & -numfilas - numColumnas - 3 & "]:R[" & numfilas & "]C[-4]),0)),""")"
```

End If

'Formula "columna"

If maximizar Then

```
Cells(filaInicial, columnaInicial + numColumnas + numfilas + 4).FormulaR1C1 = "=IF(MAX(R[" & numfilas & "]C[" & -numfilas -
```

```

numColumnas - 4 & "]:R[" & numfilas & "]C[-5])>0,MATCH(MAX(R[" &
numfilas & "]C[" & -numfilas - numColumnas - 4 & "]:R[" & numfilas
& "]C[-5]),R[" & numfilas & "]C[" & -numfilas - numColumnas - 4 &
"]:R[" & numfilas & "]C[-5],0),"")"

Else

Cells(filaInicial, columnaInicial + numColumnas + numfilas +
4).FormulaR1C1 = "=IF(MIN(R[" & numfilas & "]C[" & -numfilas -
numColumnas - 4 & "]:R[" & numfilas & "]C[-5])<0,MATCH(MIN(R[" &
numfilas & "]C[" & -numfilas - numColumnas - 4 & "]:R[" & numfilas
& "]C[-5]),R[" & numfilas & "]C[" & -numfilas - numColumnas - 4 &
"]:R[" & numfilas & "]C[-5],0),"")"

End If

'Formula "fila"

Cells(filaInicial + 2, columnaInicial + numColumnas + numfilas +
4).FormulaR1C1 = "=IF(MAX(R[-2]C[-3]:R[" & numfilas - 3 & "]C[-
3])>0,MATCH(MIN(R[-2]C[-3]:R[" & numfilas - 3 & "]C[-3]),R[-2]C[-3]:R["
& numfilas - 3 & "]C[-3],0),"")"

'Formulas columna "Ratio"

Range(Cells(filaInicial, columnaInicial + numfilas + numColumnas + 1),
Cells(filaInicial + numfilas - 1, columnaInicial + numfilas + numColumnas
+ 1)).FormulaR1C1 = "=IF(OR(LEN(RC[2])=0,RC[2]<=0),"M",RC[-1]/RC[2])"

End Sub

```

8.3. Anexo 3. (Rutina Resolver)

```

Sub Resolver()

'Realizar el test de no acotación y no dejar que empiece

If Application.CountIf(Range(Cells(filaInicial, columnaInicial +
numfilas + numColumnas + 1), Cells(filaInicial + numfilas - 1,
columnaInicial + numfilas + numColumnas + 1)), ">=0") = 0 Then

Cells(filaInicial + numfilas + 3, columnaInicial).Value = "Nota:
El problema tiene soluciones ilimitadas"

Range(Cells(filaInicial + numfilas + 3, columnaInicial),
Cells(filaInicial + numfilas + 3, columnaInicial + numfilas +
numColumnas + 1)).Select

    With Selection
        .HorizontalAlignment = xlCenter
        Call formato_bordegrueso
    End With
    Selection.Merge

Else

continuar = True
r = 1

Do While continuar = True

```

```
Call crearTabla2(numfilas, n
umColumnas, filaInicial + (numfilas + 5) * r, columnaInicial)
r = r + 1
Loop

If Application.CountIf(Range(Cells(filaInicial + (numfilas + 5) *
(r - 1), columnaInicial + numfilas + numColumnas + 1),
Cells(filaInicial + (numfilas + 5) * (r - 1) + numfilas - 1,
columnaInicial + numfilas + numColumnas + 1)), ">0") = 0 Then

    If maximizar = True Then

        If Application.CountIf(Range(Cells(filaInicial +
(numfilas + 5) * (r - 1) + numfilas, columnaInicial),
Cells(filaInicial + (numfilas + 5) * (r - 1) + numfilas,
columnaInicial + numfilas + numColumnas - 1)), ">0") > 0
Then

            Cells(filaInicial + (numfilas + 5) * r - 2,
columnaInicial).Value = "Nota: El problema tiene
solución no acotada"

            Range(Cells(filaInicial + (numfilas + 5) * r - 2,
columnaInicial), Cells(filaInicial + (numfilas + 5) *
r - 2, columnaInicial + numfilas + numColumnas +
1)).Select

                With Selection
                    .HorizontalAlignment = xlCenter
                    Call formato_bordegrueso
                End With
                Selection.Merge

            Else

                Call tablaFinal(filaInicial + (numfilas + 5) *
r - 2, columnaInicial)

            End If

        Else

            If Application.CountIf(Range(Cells(filaInicial +
(numfilas + 5) * (r - 1) + numfilas, columnaInicial),
Cells(filaInicial + (numfilas + 5) * (r - 1) +
numfilas, columnaInicial + numfilas + numColumnas -
1)), "<0") > 0 Then

                Cells(filaInicial + (numfilas + 5) * r - 2,
columnaInicial).Value = "Nota: El problema tiene
solución no acotada"

                Range(Cells(filaInicial + (numfilas + 5) * r -
2, columnaInicial), Cells(filaInicial +
(numfilas + 5) * r - 2, columnaInicial + numfilas
+ numColumnas + 1)).Select

                    With Selection
                        .HorizontalAlignment = xlCenter
                        Call formato_bordegrueso
                    End With
                    Selection.Merge

                Else


```

```
        Call tablaFinal(filaInicial + (numfilas + 5) *
            r - 2, columnaInicial)
    End If
End If
End If
End Sub
```

8.4. Anexo 4. (Rutina crearTabla2)

```
Sub crearTabla2(numfilas As Integer, numColumnas As Integer,
filaInicial As Integer, columnaInicial As Integer)

Application.ScreenUpdating = False

Worksheets("Hojal").Select

For r = 1 To numColumnas + numfilas

    Cells(filaInicial - 1, r + columnaInicial - 1).FormulaR1C1 =
        "=R[" & -numfilas - 5 & "]C"

Next r

    'Copiar columna "Restricciones"

For r = 1 To numfilas

    Cells(filaInicial + r - 1, columnaInicial - 3).FormulaR1C1 =
        "=R[" & -numfilas - 5 & "]C"

Next r

    'Formato columna "Restricciones"

Range(Cells(filaInicial, columnaInicial - 3), Cells(filaInicial +
numfilas - 1, columnaInicial - 3)).Select

    With Selection
        Call fondoAzul
        Call formato_bordefino
        Call formato_bordegrueso
    End With

Cells(filaInicial - 1, columnaInicial + numColumnas + numfilas).Value
= "b"
Cells(filaInicial - 1, columnaInicial + numColumnas + numfilas +
1).Value = "Ratio"
Cells(filaInicial - 1, columnaInicial - 1).Select

    With Selection
        .Value = "Base"
        Call formato_bordegrueso
    End With

'Fila Xs "horizontal" con fondo gris y centrado
```

```
Range(Cells(filaInicial - 1, columnaInicial - 1), Cells(filaInicial - 1, columnaInicial + numColumnas + numfilas + 1)).Select
    With Selection
        Call fondoGris
        Call negrita
    End With
```

'Crear cuadro "Z"

```
Cells(filaInicial + numfilas, columnaInicial + numColumnas + numfilas + 1).Value = "Z"
```

'Formato Z

```
Range(Cells(filaInicial + numfilas, columnaInicial + numfilas + numColumnas), Cells(filaInicial + numfilas, columnaInicial + numfilas + numColumnas + 1)).Select
```

```
    With Selection
        Call formato_bordefino
        Call formato_bordegrueso
        Call fondoAmarillo
        Call negrita
    End With
```

'Fila Xs "vertical" con fondo gris y centrado

```
Range(Cells(filaInicial, columnaInicial - 1), Cells(filaInicial + numfilas - 1, columnaInicial - 1)).Select
```

```
    With Selection
        Call fondoGris
        Call negrita
        Call formato_bordefino
        Call formato_bordegrueso
    End With
```

'Formato celda "Cj-Zj"

```
Cells(filaInicial + numfilas, columnaInicial - 1).Select
```

```
    With Selection
        .Value = "Cj-Zj"
        Call negrita
    End With
```

'Crear cuadro "Elemento Pivote"

```
Cells(filaInicial - 2, columnaInicial + numColumnas + numfilas + 3).Value = "Elemento pivote"
```

```
Range(Cells(filaInicial - 2, columnaInicial + numColumnas + numfilas + 3), Cells(filaInicial - 2, columnaInicial + numColumnas + numfilas + 4)).Select
```

```
    With Selection
        Call fondoNaranja
        Call negrita
        Call formato_bordegrueso
    End With
    Selection.Merge
```

```
Cells(filaInicial - 1, columnaInicial + numColumnas + numfilas +
3).Select
  With Selection
    .Value = "Valores"
    Call fondoGris
    Call negrita
  End With

Cells(filaInicial - 1, columnaInicial + numColumnas + numfilas +
4).Select

With Selection
  .Value = "columna"
  Call fondoGris
  Call negrita
End With

Cells(filaInicial + 1, columnaInicial + numColumnas + numfilas +
4).Select

  With Selection
    .Value = "fila"
    Call fondoGris
    Call negrita
  End With

  'Formato "valores"

Range(Cells(filaInicial - 1, columnaInicial + numColumnas + numfilas +
3), Cells(filaInicial + numfilas - 1, columnaInicial + numColumnas +
numfilas + 3)).Select

  With Selection
    Call formato_bordefino
    Call formato_bordegrueso
  End With

  'Formato "columna"

Range(Cells(filaInicial - 1, columnaInicial + numColumnas + numfilas +
4), Cells(filaInicial, columnaInicial + numColumnas + numfilas +
4)).Select

  With Selection
    Call formato_bordefino
    Call formato_bordegrueso
  End With

  'Formato "fila"

Range(Cells(filaInicial + 1, columnaInicial + numColumnas + numfilas +
4), Cells(filaInicial + 2, columnaInicial + numColumnas + numfilas +
4)).Select

  With Selection
    Call formato_bordefino
    Call formato_bordegrueso
  End With

  'Centrar Rango tabla principal completa
```

```
Range(Cells(filaInicial - 2, columnaInicial - 3), Cells(filaInicial + numfilas, columnaInicial + numfilas + numColumnas + 4)).Select
```

```
With Selection
    .HorizontalAlignment = xlCenter
End With
```

'Primera fila "Función Objetivo". Borde grueso, cuadrados interiores y color naranja

```
Range(Cells(filaInicial - 2, columnaInicial), Cells(filaInicial - 2, columnaInicial + numfilas + numColumnas - 1)).Select
```

```
With Selection
    Call formato_bordefino
    Call formato_bordegrueso
    Call fondoNaranja
End With
```

'Columna izquierda "Valores que entran a la tabla". Borde grueso, cuadrados interiores y color naranja

```
Range(Cells(filaInicial, columnaInicial - 2), Cells(filaInicial + numfilas - 1, columnaInicial - 2)).Select
```

```
With Selection
    Call formato_bordefino
    Call formato_bordegrueso
    Call fondoNaranja
End With
```

'Formato cuadro de las variables

```
Range(Cells(filaInicial - 1, columnaInicial), Cells(filaInicial + numfilas - 1, columnaInicial + numfilas + numColumnas - 1)).Select
```

```
With Selection
    Call formato_bordefino
    Call formato_bordegrueso
End With
```

'Columna "b"

```
Range(Cells(filaInicial - 1, columnaInicial + numfilas + numColumnas), Cells(filaInicial + numfilas - 1, columnaInicial + numfilas + numColumnas)).Select
```

```
With Selection
    Call formato_bordefino
    Call formato_bordegrueso
End With
```

'Columna "Ratio"

```
Range(Cells(filaInicial - 1, columnaInicial + numfilas + numColumnas + 1), Cells(filaInicial + numfilas - 1, columnaInicial + numfilas + numColumnas + 1)).Select
```

```
With Selection
    Call formato_bordefino
    Call formato_bordegrueso
End With
```

End With

'Fila inferior "Entre Cj-Zj y Z"

```
Range(Cells(filaInicial + numfilas, columnaInicial), Cells(filaInicial + numfilas, columnaInicial + numfilas + numColumnas - 1)).Select
```

With Selection

 Call formato_bordefino

 Call formato_bordegrueso

End With

'Formula Entre "Cj-Zj y Z"

```
For r = 1 To numfilas + numColumnas
```

```
    Cells(filaInicial + numfilas, r + columnaInicial - 1).FormulaR1C1 = "=R[" & -numfilas - 2 & "]C-SUMPRODUCT(R[" & -numfilas & "]C[" & -(r + 1) & "]:R[-1]C[" & -(r + 1) & "], R[" & -numfilas & "]C:R[-1]C)"
```

```
Next r
```

'Formula "Z"

```
Cells(filaInicial + numfilas, columnaInicial + numColumnas + numfilas).FormulaR1C1 = "=SUMPRODUCT(R[" & -numfilas & "]C[" & (-numfilas - numColumnas - 2) & "]:R[-1]C[" & (-numfilas - numColumnas - 2) & "], R[" & -numfilas & "]C:R[-1]C)"
```

'Poner 0 en la función objetivo (parte base canónica)

```
Range(Cells(filaInicial - 2, columnaInicial + numColumnas), Cells(filaInicial - 2, columnaInicial + numfilas + numColumnas - 1)).Value = 0
```

'Copiar datos de la función objetivo principal

```
For r = 1 To numColumnas + numfilas
```

```
    Cells(filaInicial - 2, r + columnaInicial - 1).FormulaR1C1 = "=R[" & -numfilas - 5 & "]C"
```

```
Next r
```

'Columna izquierda (AQUI CAMBIA, YA NO ES SIEMPRE 0)

```
For r = 1 To numfilas
```

```
    Range(Cells(filaInicial, columnaInicial - 2), Cells(filaInicial + numfilas - 1, columnaInicial - 2)).FormulaArray = "=IF((ROW()-" & filaInicial - 1 & ")=R[" & -numfilas - 5 + 2 & "]C[" & numfilas + numColumnas + 6 & "],INDEX(R[" & -numfilas - 7 & "]C[2]:R[" & -numfilas - 7 & "]C[" & numfilas + numColumnas + 1 & "],1,R[" & -numfilas - 5 & "]C[" & numfilas + numColumnas + 6 & "]),R[" & -numfilas - 5 & "]C:R[" & -6 & "]C)"
```

```
    Range(Cells(filaInicial, columnaInicial - 1), Cells(filaInicial + numfilas - 1, columnaInicial - 1)).FormulaArray = "=IF((ROW()-" & filaInicial - 1 & ")=R[" & -numfilas - 5 + 2 & "]C[" & numfilas + numColumnas + 5 & "],INDEX(R[" & -numfilas - 6 & "]C[1]:R[" & -numfilas - 6 & "]C[" & numfilas + numColumnas &
```

```
"),1,R[" & -numfilas - 5 & "]C[" & numfilas + numColumnas + 5 &
")],R[" & -numfilas - 5 & "]C:R[" & -6 & "]C)"
```

Next r

```
'Formulas cuadro "ELEMENTO PIVOTE"
'Formula columna "Valores"
```

If maximizar Then

```
Range(Cells(filaInicial, columnaInicial + numfilas + numColumnas
+ 3), Cells(filaInicial + numfilas - 1, columnaInicial +
numfilas + numColumnas + 3)).FormulaArray = "=IF(MAX(R[" &
numfilas & "]C[" & -numfilas - numColumnas - 3 & "]:R[" &
numfilas & "]C[-4])>0,INDEX(RC[" & -numfilas - numColumnas - 3 &
"]:R[" & numfilas & "]C[-4],,MATCH(MAX(R[" & numfilas & "]C[" &
-numfilas - numColumnas - 3 & "]:R[" & numfilas & "]C[-4]),R[" &
numfilas & "]C[" & -numfilas - numColumnas - 3 & "]:R[" &
numfilas & "]C[-4],0)),""")"
```

Else

```
Range(Cells(filaInicial, columnaInicial + numfilas + numColumnas
+ 3), Cells(filaInicial + numfilas - 1, columnaInicial +
numfilas + numColumnas + 3)).FormulaArray = "=IF(MIN(R[" &
numfilas & "]C[" & -numfilas - numColumnas - 3 & "]:R[" &
numfilas & "]C[-4])<0,INDEX(RC[" & -numfilas - numColumnas - 3 &
"]:R[" & numfilas & "]C[-4],,MATCH(MIN(R[" & numfilas & "]C[" &
-numfilas - numColumnas - 3 & "]:R[" & numfilas & "]C[-4]),R[" &
numfilas & "]C[" & -numfilas - numColumnas - 3 & "]:R[" &
numfilas & "]C[-4],0)),""")"
```

End If

```
'Formula "columna"
```

If maximizar Then

```
Cells(filaInicial, columnaInicial + numColumnas + numfilas +
4).FormulaR1C1 = "=IF(MAX(R[" & numfilas & "]C[" & -numfilas -
numColumnas - 4 & "]:R[" & numfilas & "]C[-5])>0,MATCH(MAX(R[" &
numfilas & "]C[" & -numfilas - numColumnas - 4 & "]:R[" &
numfilas & "]C[-5]),R[" & numfilas & "]C[" & -numfilas -
numColumnas - 4 & "]:R[" & numfilas & "]C[-5],0),""")"
```

Else

```
Cells(filaInicial, columnaInicial + numColumnas + numfilas +
4).FormulaR1C1 = "=IF(MIN(R[" & numfilas & "]C[" & -numfilas -
numColumnas - 4 & "]:R[" & numfilas & "]C[-5])<0,MATCH(MIN(R[" &
numfilas & "]C[" & -numfilas - numColumnas - 4 & "]:R[" &
numfilas & "]C[-5]),R[" & numfilas & "]C[" & -numfilas -
numColumnas - 4 & "]:R[" & numfilas & "]C[-5],0),""")"
```

End If

```
'Formula "fila"
```

```
Cells(filaInicial + 2, columnaInicial + numColumnas + numfilas +
4).FormulaR1C1 = "=IF(MAX(R[-2]C[-3]:R[" & numfilas - 3 & "]C[-
```

```
3])>0,MATCH(MIN(R[-2]C[-3]:R[" & numfilas - 3 & "]C[-3]),R[-2]C[-3]:R[" & numfilas - 3 & "]C[-3],0),"")"
```

'Formulas columna "Ratio"

```
Range(Cells(filaInicial, columnaInicial + numfilas + numColumnas + 1),  
Cells(filaInicial + numfilas - 1, columnaInicial + numfilas +  
numColumnas + 1)).FormulaR1C1 =  
"=IF(OR(LEN(RC[2])=0,RC[2]<=0),"M",RC[-1]/RC[2])"
```

'Formula Filas Pivote y Filas no Pivote

```
For fila = 1 To numfilas
```

```
Dim prueba As Integer
```

```
filaPivote = Cells(filaInicial - numfilas - 3, columnaInicial +  
numfilas + numColumnas + 4).Value
```

```
If fila <> filaPivote Then
```

```
For columna = 1 To numfilas + numColumnas + 1
```

```
Cells(filaInicial + fila - 1, columnaInicial + columna -  
1).FormulaR1C1 = "=R[" & -numfilas - 5 & "]C-(INDEX(R" &  
filaInicial - numfilas - 5 & "C" & columnaInicial & ":R" &  
filaInicial - 6 & "C" & columnaInicial + numfilas +  
numColumnas & "," & fila & ", R" & filaInicial - numfilas -  
5 & "C" & columnaInicial + numfilas + numColumnas + 4 &  
")*INDEX(R" & filaInicial & "C" & columnaInicial & ":R" &  
filaInicial + numfilas - 1 & "C" & columnaInicial +  
numfilas + numColumnas & ",R" & filaInicial - numfilas - 3  
& "C" & columnaInicial + numfilas + numColumnas + 4 & ",))"
```

```
Next columna
```

```
Else
```

```
For columna = 1 To numfilas + numColumnas + 1
```

```
Cells(filaInicial + fila - 1, columnaInicial + columna -  
1).FormulaR1C1 = "=R[" & -numfilas - 5 & "]C/INDEX(R" &  
filaInicial - numfilas - 5 & "C" & columnaInicial & ":R" &  
filaInicial - 6 & "C" & columnaInicial + numfilas +  
numColumnas & ", R" & filaInicial - numfilas - 3 & "C" &  
columnaInicial + numfilas + numColumnas + 4 & ", R" &  
filaInicial - numfilas - 5 & "C" & columnaInicial +  
numfilas + numColumnas + 4 & ")"
```

```
Next columna
```

```
End If
```

```
Next fila
```

'Controlar si hemos acabado las iteraciones

```
If maximizar = True Then
```

```
If Application.CountIf(Range(Cells(filaInicial + numfilas,  
columnaInicial), Cells(filaInicial + numfilas, columnaInicial +  
numfilas + numColumnas - 1)), ">0") > 0 Then  
continuar = True
```

```
Else
```

```
continuar = False
```

```
End If
```

```
Else
```

```
If Application.CountIf(Range(Cells(filaInicial + numfilas,  
columnaInicial), Cells(filaInicial + numfilas, columnaInicial +  
numfilas + numColumnas - 1)), "<0") > 0 Then
```

```
        continuar = True
    Else
        continuar = False
    End If
End If

End Sub
```

8.5. Anexo 5. (Rutina tablaFinal)

```
Sub tablaFinal(filaInicial As Integer, columnaInicial As Integer)

    'Columna "Solución"

    Cells(filaInicial, columnaInicial).FormulaR1C1 = "Solución"
    Range(Cells(filaInicial + 1, columnaInicial), Cells(filaInicial +
    numfilas + numColumnas, columnaInicial)).FormulaArray = "=TRANSPOSE(R["
    & -numfilas - 5 & "]C:R[" & -numfilas - 5 & "]C[" & numfilas + numColumnas
    - 1 & "])"

    'Columna "Valor"

    Cells(filaInicial, columnaInicial + 1).FormulaR1C1 = "Valor"
    Range(Cells(filaInicial + 1, columnaInicial + 1), Cells(filaInicial +
    numfilas + numColumnas, columnaInicial + 1)).FormulaR1C1 =
    "=IFERROR((INDEX(R" & filaInicial - numfilas - 3 & "C" & columnaInicial
    + numfilas + numColumnas & ":R" & filaInicial - 4 & "C" & columnaInicial
    + numfilas + numColumnas & ",MATCH(RC[-1],R" & filaInicial - numfilas -
    3 & "C" & columnaInicial - 1 & ":R" & filaInicial - 4 & "C" &
    columnaInicial - 1 & ",0))),0)"

    'Columna "Cj"

    Cells(filaInicial, columnaInicial + 2).FormulaR1C1 = "Cj"
    Range(Cells(filaInicial + 1, columnaInicial + 2), Cells(filaInicial +
    numfilas + numColumnas, columnaInicial + 2)).FormulaArray =
    "=TRANSPOSE(R[" & -numfilas - 6 & "]C[-2]:R[" & -numfilas - 6 & "]C[" &
    numfilas + numColumnas - 3 & "])"

    'Columna "Contribución"

    Cells(filaInicial, columnaInicial + 3).FormulaR1C1 = "Contribución"
    Range(Cells(filaInicial + 1, columnaInicial + 3), Cells(filaInicial +
    numfilas + numColumnas, columnaInicial + 3)).FormulaR1C1 = "=RC[-2]*RC[-
    1]"

    'Columna "Bases Status"

    Cells(filaInicial, columnaInicial + 4).FormulaR1C1 = "Bases status"
    Range(Cells(filaInicial + 1, columnaInicial + 4), Cells(filaInicial +
    numfilas + numColumnas, columnaInicial + 4)).FormulaR1C1 =
    "=IF(ISERROR(MATCH(RC[-4],R" & filaInicial - numfilas - 3 & "C" &
    columnaInicial - 1 & ":R" & filaInicial - 4 & "C" & columnaInicial - 1
    & ",0))),""No"",""Yes"")"

    'Columna "Min Cj"

    Cells(filaInicial, columnaInicial + 5).FormulaR1C1 = "Min Cj"
```

```

Range(Cells(filaInicial + (numfilas * 2) + numColumnas + 7,
columnaInicial), Cells(filaInicial + (numfilas * 3) + (numColumnas * 2)
+ 6, columnaInicial + numfilas + numColumnas - 1)).FormulaR1C1 =
"=IF((IF(COUNTIF(R" & filaInicial - numfilas - 3 & "C" & columnaInicial
- 1 & ":R" & filaInicial - 4 & "C" & columnaInicial - 1 & ",R" &
filaInicial - numfilas - 4 & "C")>0,-9.99E+307,IF(INDEX(R" & filaInicial
- numfilas - 3 & "C:R" & filaInicial - 4 & "C,MATCH(R[" & -(numfilas *
2) - numColumnas - 6 & "]C" & columnaInicial & ",R" & filaInicial -
numfilas - 3 & "C" & columnaInicial - 1 & ":R" & filaInicial - 4 & "C"
& columnaInicial - 1 & ",0))>0,INDEX(R" & filaInicial - numfilas - 3 &
"C:R" & filaInicial - 4 & "C,MATCH(R[" & -(numfilas * 2) - numColumnas
- 6 & "]C" & columnaInicial & ",R" & filaInicial - numfilas - 3 & "C" &
columnaInicial - 1 & ":R" & filaInicial - 4 & "C" & columnaInicial - 1
& ",0))),-9.99E+307)))<0,IF(COUNTIF(R" & filaInicial - numfilas - 3 & "C"
& columnaInicial - 1 & ":R" & filaInicial - 4 & "C" & columnaInicial -
1 & ",R" & filaInicial - numfilas - 4 & "C")>0,-9.99E+307,IF(INDEX(R" &
filaInicial - numfilas - 3 & "C:R" & filaInicial - 4 & "C,MATCH(R[" & -
(numfilas * 2) - numColumnas - 6 & "]C" & columnaInicial & ",R" &
filaInicial - numfilas - 3 & "C" & columnaInicial - 1 & ":R" & filaInicial
- 4 & "C" & columnaInicial - 1 & ",0))>0,INDEX(R" & filaInicial - numfilas
- 3 & "C:R" & filaInicial - 4 & "C,MATCH(R[" & -(numfilas * 2) -
numColumnas - 6 & "]C" & columnaInicial & ",R" & filaInicial - numfilas
- 3 & "C" & columnaInicial - 1 & ":R" & filaInicial - 4 & "C" &
columnaInicial - 1 & ",0))),-9.99E+307)),R[" & -(numfilas * 2) -
numColumnas - 6 & "]C" & columnaInicial + 2 & "+R" & filaInicial - 3 &
"C/(IF(COUNTIF(R" & filaInicial - numfilas - 3 & "C" & columnaInicial -
1 & ":R" & filaInicial - 4 & "C" & columnaInicial - 1 & ",R" & filaInicial
- numfilas - 4 & "C")>0,-9.99E+307,IF(INDEX(R" & filaInicial - numfilas
- 3 & "C:R" & filaInicial - 4 & "C,MATCH(R[" & -(numfilas * 2) -
numColumnas - 6 & "]C" & columnaInicial & ",R" & filaInicial - numfilas
- 3 & "C" & columnaInicial - 1 & ":R" & filaInicial - 4 & "C" &
columnaInicial - 1 & ",0))>0,INDEX(R" & filaInicial - numfilas - 3 &
"C:R" & filaInicial - 4 & "C,MATCH(R[" & -(numfilas * 2) - numColumnas
- 6 & "]C" & columnaInicial & ",R" & filaInicial - numfilas - 3 & "C" &
columnaInicial - 1 & ":R" & filaInicial - 4 & "C" & columnaInicial - 1
& ",0))),-9.99E+307))))"

```

```

Range(Cells(filaInicial + 1, columnaInicial + 5), Cells(filaInicial +
numfilas + numColumnas, columnaInicial + 5)).FormulaR1C1 =
"=IF(IFERROR(MAX(R[" & (numfilas * 2) + numColumnas + 6 & "]C[-5]:R[" &
(numfilas * 2) + numColumnas + 6 & "]C" & columnaInicial + numfilas +
numColumnas - 1 & "),""-M""))=-9.99E+307,""-M""",IFERROR(MAX(R[" &
(numfilas * 2) + numColumnas + 6 & "]C[-5]:R[" & (numfilas * 2) +
numColumnas + 6 & "]C" & columnaInicial + numfilas + numColumnas - 1 &
"),""-M""))"

```

'Columna "Max Cj"

```
Cells(filaInicial, columnaInicial + 6).FormulaR1C1 = "Max Cj"
```

```

Range(Cells(filaInicial + (numfilas * 3) + (numColumnas * 2) + 8,
columnaInicial), Cells(filaInicial + (numfilas * 4) + (numColumnas * 3)
+ 7, columnaInicial + numfilas + numColumnas - 1)).FormulaR1C1 =
"=IF((IF(COUNTIF(R" & filaInicial - numfilas - 3 & "C" & columnaInicial
- 1 & ":R" & filaInicial - 4 & "C" & columnaInicial - 1 & ",R" &
filaInicial - numfilas - 4 & "C")<0,9.99E+307,IF(INDEX(R" & filaInicial
- numfilas - 3 & "C:R" & filaInicial - 4 & "C,MATCH(R[" & -(numfilas *
3) - (numColumnas * 2) - 7 & "]C" & columnaInicial & ",R" & filaInicial
- numfilas - 3 & "C" & columnaInicial - 1 & ":R" & filaInicial - 4 & "C"
& columnaInicial - 1 & ",0))<0,INDEX(R" & filaInicial - numfilas - 3 &
"C:R" & filaInicial - 4 & "C,MATCH(R[" & -(numfilas * 3) - (numColumnas
* 2) - 7 & "]C" & columnaInicial & ",R" & filaInicial - numfilas - 3 &

```

```
"C" & columnaInicial - 1 & ":R" & filaInicial - 4 & "C" & columnaInicial - 1 & ",0)),9.99E+307)))>0,IF(COUNTIF(R" & filaInicial - numfilas - 3 & "C" & columnaInicial - 1 & ":R" & filaInicial - 4 & "C" & columnaInicial - 1 & ",R" & filaInicial - numfilas - 4 & "C"<0,9.99E+307,IF(INDEX(R" & filaInicial - numfilas - 3 & "C:R" & filaInicial - 4 & "C,MATCH(R[" & -(numfilas * 3) - (numColumnas * 2) - 7 & "]C" & columnaInicial & ",R" & filaInicial - numfilas - 3 & "C" & columnaInicial - 1 & ":R" & filaInicial - 4 & "C" & columnaInicial - 1 & ",0))<0,INDEX(R" & filaInicial - numfilas - 3 & "C:R" & filaInicial - 4 & "C,MATCH(R[" & -(numfilas * 3) - (numColumnas * 2) - 7 & "]C" & columnaInicial & ",R" & filaInicial - numfilas - 3 & "C" & columnaInicial - 1 & ":R" & filaInicial - 4 & "C" & columnaInicial - 1 & ",0)),9.99E+307)),R[" & -(numfilas * 3) - (numColumnas * 2) - 7 & "]C" & columnaInicial + 2 & "+R" & filaInicial - 3 & "C/(IF(COUNTIF(R" & filaInicial - numfilas - 3 & "C" & columnaInicial - 1 & ":R" & filaInicial - 4 & "C" & columnaInicial - 1 & ",R" & filaInicial - numfilas - 4 & "C"<0,9.99E+307,IF(INDEX(R" & filaInicial - numfilas - 3 & "C:R" & filaInicial - 4 & "C,MATCH(R[" & -(numfilas * 3) - (numColumnas * 2) - 7 & "]C" & columnaInicial & ",R" & filaInicial - numfilas - 3 & "C" & columnaInicial - 1 & ":R" & filaInicial - 4 & "C" & columnaInicial - 1 & ",0))<0,INDEX(R" & filaInicial - numfilas - 3 & "C:R" & filaInicial - 4 & "C,MATCH(R[" & -(numfilas * 3) - (numColumnas * 2) - 7 & "]C" & columnaInicial & ",R" & filaInicial - numfilas - 3 & "C" & columnaInicial - 1 & ":R" & filaInicial - 4 & "C" & columnaInicial - 1 & ",0)),9.99E+307))))"
```

```
Range(Cells(filaInicial + 1, columnaInicial + 6), Cells(filaInicial + numfilas + numColumnas, columnaInicial + 6)).FormulaR1C1 = "=IF(IFERROR(MIN(R[" & (numfilas * 3) + (numColumnas * 2) + 7 & "]C[-6]:R[" & (numfilas * 3) + (numColumnas * 2) + 7 & "]C" & columnaInicial + numfilas + numColumnas - 1 & "],RC[-4]) = 9.99E+307,""M"",IFERROR(MIN(R[" & (numfilas * 3) + (numColumnas * 2) + 7 & "]C[-6]:R[" & (numfilas * 3) + (numColumnas * 2) + 7 & "]C" & columnaInicial + numfilas + numColumnas - 1 & "],RC[-4]))"
```

'Ocultar Tabla operaciones Min/Max Cj

```
Range(Cells(filaInicial + (numfilas * 2) + numColumnas + 7, columnaInicial), Cells(filaInicial + (numfilas * 4) + (numColumnas * 3) + 7, columnaInicial + numfilas + numColumnas - 1)).Select
```

```
With Selection.Font
    .ThemeColor = xlThemeColorDark1
    .TintAndShade = 0
End With
```

'Ocultar las filas

```
Rows("" & filaInicial + (numfilas * 2) + numColumnas + 7 & ":" & filaInicial + (numfilas * 4) + (numColumnas * 3) + 7 & "").Select
```

```
Selection.EntireRow.Hidden = True
'Cuadro "Nota:"
```

```
numCeros = Application.WorksheetFunction.CountIf(Range(Cells(filaInicial - 3, columnaInicial), Cells(filaInicial - 3, columnaInicial + numfilas + numColumnas - 1)), 0)
```

```
numRestr = Application.WorksheetFunction.Count(Range(Cells(filaInicial - numfilas - 3, columnaInicial - 2), Cells(filaInicial - 4, columnaInicial - 2)))
```

```
resultCeros = numCeros - numRestr
```

```
Cells(filaInicial + numfilas + numColumnas + 2, columnaInicial + 1).FormulaR1C1 = "=IF(" & resultCeros & " = 0, ""Nota: Existe una única solución"", ""Nota: Pueden existir soluciones alternativas"")"
```

```
Range(Cells(filaInicial + numfilas + numColumnas + 2, columnaInicial), Cells(filaInicial + numfilas + numColumnas + 2, columnaInicial + 6)).Select
```

```
With Selection  
    .HorizontalAlignment = xlCenter  
    Call formato_bordegrueso  
End With  
Selection.Merge
```

'Columna "Restricciones"

```
Cells(filaInicial + numfilas + numColumnas + 4, columnaInicial).FormulaR1C1 = "Restricciones"  
Range(Cells(filaInicial + numfilas + numColumnas + 5, columnaInicial), Cells(filaInicial + (numfilas * 2) + numColumnas + 4, columnaInicial)).FormulaArray = "=R" & filaInicial - numfilas - 3 & "C" & columnaInicial - 3 & ":R" & filaInicial - 4 & "C" & columnaInicial - 3 & ""
```

'Columna "LHS"

```
Cells(filaInicial + numfilas + numColumnas + 4, columnaInicial + 1).FormulaR1C1 = "LHS"  
Range(Cells(filaInicial + numfilas + numColumnas + 5, columnaInicial + 1), Cells(filaInicial + (numfilas * 2) + numColumnas + 4, columnaInicial + 1)).FormulaR1C1 = "=RC[2]-RC[3]"
```

'Columna "Signo"

```
Cells(filaInicial + numfilas + numColumnas + 4, columnaInicial + 2).FormulaR1C1 = "Signo"  
Range(Cells(filaInicial + numfilas + numColumnas + 5, columnaInicial + 2), Cells(filaInicial + (numfilas * 2) + numColumnas + 4, columnaInicial + 2)).FormulaR1C1 = "<="
```

'Columna "RHS"

```
Cells(filaInicial + numfilas + numColumnas + 4, columnaInicial + 3).FormulaR1C1 = "RHS"  
Range(Cells(filaInicial + numfilas + numColumnas + 5, columnaInicial + 3), Cells(filaInicial + (numfilas * 2) + numColumnas + 4, columnaInicial + 3)).FormulaArray = "=R4C" & 4 + numfilas + numColumnas & ":R" & 4 + numfilas - 1 & "C" & 4 + numfilas + numColumnas & ""
```

'Columna "Holgura"

```
Cells(filaInicial + numfilas + numColumnas + 4, columnaInicial + 4).FormulaR1C1 = "Holgura"
```

```
Range(Cells(filaInicial + numfilas + numColumnas + 5, columnaInicial + 4), Cells(filaInicial + (numfilas * 2) + numColumnas + 4, columnaInicial + 4)).FormulaR1C1 = "=IFERROR((INDEX(R" & filaInicial - numfilas - 3 & "C" & columnaInicial + numfilas + numColumnas & ":R" & filaInicial - 4 & "C" & columnaInicial + numfilas + numColumnas & ",MATCH(R[" & -4 - numfilas & "]C[-4],R" & filaInicial - numfilas - 3 & "C" & columnaInicial - 1 & ":R" & filaInicial - 4 & "C" & columnaInicial - 1 & ",0))),0)"
```

'Columna "Min RHS"

```
Cells(filaInicial + numfilas + numColumnas + 4, columnaInicial + 5).FormulaR1C1 = "Min RHS"
```

```
For fila = 1 To numfilas
```

```
Cells(filaInicial + numfilas + numColumnas + fila + 4, columnaInicial + 5).FormulaArray = "=IF(MAX(IF(OFFSET(R" & filaInicial - numfilas - 4 & "C" & columnaInicial + numColumnas & ",1," & fila - 1 & "," & numfilas & ",1)<=0,-9.99E+307,RC[-2]-R" & filaInicial - numfilas - 3 & "C" & columnaInicial + numfilas + numColumnas & ":R" & filaInicial - 4 & "C" & columnaInicial + numfilas + numColumnas & "/OFFSET(R" & filaInicial - numfilas - 4 & "C" & columnaInicial + numColumnas & ",1," & fila - 1 & "," & numfilas & ",1)))=-9.99E+307,""-M"",MAX(IF(OFFSET(R" & filaInicial - numfilas - 4 & "C" & columnaInicial + numColumnas & ",1," & fila - 1 & "," & numfilas & ",1)<=0,-9.99E+307,RC[-2]-R" & filaInicial - numfilas - 3 & "C" & columnaInicial + numfilas + numColumnas & ":R" & filaInicial - 4 & "C" & columnaInicial + numfilas + numColumnas & "/OFFSET(R" & filaInicial - numfilas - 4 & "C" & columnaInicial + numColumnas & ",1," & fila - 1 & "," & numfilas & ",1))))"
```

```
Next fila
```

'Columna "Max RHS"

```
Cells(filaInicial + numfilas + numColumnas + 4, columnaInicial + 6).FormulaR1C1 = "Max RHS"
```

```
For fila = 1 To numfilas
```

```
Cells(filaInicial + numfilas + numColumnas + fila + 4, columnaInicial + 6).FormulaArray = "=IF(MIN(IF(OFFSET(R" & filaInicial - numfilas - 4 & "C" & columnaInicial + numColumnas & ",1," & fila - 1 & "," & numfilas & ",1)>=0,9.99E+307,RC[-3]-R" & filaInicial - numfilas - 3 & "C" & columnaInicial + numfilas + numColumnas & ":R" & filaInicial - 4 & "C" & columnaInicial + numfilas + numColumnas & "/OFFSET(R" & filaInicial - numfilas - 4 & "C" & columnaInicial + numColumnas & ",1," & fila - 1 & "," & numfilas & ",1)))=9.99E+307,""M"",MIN(IF(OFFSET(R" & filaInicial - numfilas - 4 & "C" & columnaInicial + numColumnas & ",1," & fila - 1 & "," & numfilas & ",1)>=0,9.99E+307,RC[-3]-R" & filaInicial - numfilas - 3 & "C" & columnaInicial + numfilas + numColumnas & ":R" & filaInicial - 4 & "C" & columnaInicial + numfilas + numColumnas & "/OFFSET(R" & filaInicial - numfilas - 4 & "C" & columnaInicial + numColumnas & ",1," & fila - 1 & "," & numfilas & ",1))))"
```

```
Next fila
```

'Formato Centrar + Borde grueso + borde fino cuadro 1

```
Range(Cells(filaInicial, columnaInicial), Cells(filaInicial + numfilas + numColumnas, columnaInicial + 6)).Select
```

```
With Selection
    .HorizontalAlignment = xlCenter
    Call formato_bordefino
    Call formato_bordegrueso
End With
```

'Formato Centrar + Borde grueso + borde fino cuadro 2

```
Range(Cells(filaInicial + numfilas + numColumnas + 4, columnaInicial), Cells(filaInicial + (numfilas * 2) + numColumnas + 4, columnaInicial + 6)).Select
```

```
With Selection
    .HorizontalAlignment = xlCenter
    Call formato_bordefino
    Call formato_bordegrueso
End With
```

'Formato fila superior primer cuadro color naranja y negrita

```
Range(Cells(filaInicial, columnaInicial), Cells(filaInicial, columnaInicial + 6)).Select
```

```
With Selection
    Call fondoNaranja
    Call negrita
End With
```

'Formato columna izquierda color gris y negrita

```
Range(Cells(filaInicial + 1, columnaInicial), Cells(filaInicial + numfilas + numColumnas, columnaInicial)).Select
```

```
With Selection
    Call fondoGris
    Call negrita
End With
```

'Formato fila superior segundo cuadro color naranja y negrita

```
Range(Cells(filaInicial + numfilas + numColumnas + 4, columnaInicial), Cells(filaInicial + numfilas + numColumnas + 4, columnaInicial + 6)).Select
```

```
With Selection
    Call fondoNaranja
    Call negrita
End With
```

'Formato columna izquierda segundo cuadro color azul

```
Range(Cells(filaInicial + numfilas + numColumnas + 5, columnaInicial), Cells(filaInicial + (numfilas * 2) + numColumnas + 4, columnaInicial)).Select
```

```
With Selection
    Call fondoAzul
End With
```

End Sub