

Gestión de Proyectos con Excel 2010

TRUCOS ESENCIALES

RESPONSABLE EDITORIAL:

Víctor Manuel Ruiz Calderón

Susana Krahe Pérez-Rubín

DISEÑO DE CUBIERTA:

????????????????????

Gestión de Proyectos con Excel 2010

José Luis Ponz Tienda



Todos los nombres propios de programas, sistemas operativos, equipos hardware, etc. que aparecen en este libro son marcas registradas de sus respectivas compañías u organizaciones.

Reservados todos los derechos. El contenido de esta obra está protegido por la Ley, que establece penas de prisión y/o multas, además de las correspondientes indemnizaciones por daños y perjuicios, para quienes reprodujeren, plagiaran, distribuyeren o comunicaren públicamente, en todo o en parte, una obra literaria, artística o científica, o su transformación, interpretación o ejecución artística fijada en cualquier tipo de soporte o comunicada a través de cualquier medio, sin la preceptiva autorización.

© EDICIONES ANAYA MULTIMEDIA (GRUPO ANAYA, S.A.), 2010
Juan Ignacio Luca de Tena, 15. 28027 Madrid
Depósito legal:
ISBN: ??????????????
Printed in Spain
Impreso en:

Agradecimientos

Es muy difícil reseñar en unas pocas líneas a las personas que de forma directa o indirectamente han contribuido a que este nuevo libro fuera hoy día una realidad.

Las primeras palabras de agradecimiento, han de ser para mis alumnos de la escuela de Ingeniería de Edificación y del programa de postgrado en Arquitectura, Edificación, Urbanística y Paisaje de la Universidad Politécnica de Valencia, puesto que la idea de aplicar Excel a la gestión de proyectos surgió a partir de la necesidad de plasmar de forma sencilla, práctica y visual las herramientas de gestión de proyectos que no están disponibles en el software comercial y que en un contexto tan difícil y competitivo como el actual consideraba que debían de ser popularizadas y abiertas a un público no científico.

No sería honesto, si además no reconociera que este libro tiene su semilla en el profesorado del departamento de Investigación Operativa de la facultad de matemáticas de Valencia, donde descubrí las incontables aplicaciones de esta poderosa herramienta en múltiples disciplinas, no pudiendo ocultar además la gran admiración que siento por alguno de ellos.

No puedo dejar de agradecer el apoyo que he tenido en mis amigos y mentores Dr. D. Javier Benlloch Marco y Dr. D. Carlos Andrés Romano, pertenecientes a dos disciplinas muy diferentes, Arquitecto e Ingeniero Industrial respectivamente, excelentes profesionales y mejores personas, sin cuyas palabras de ánimo en los momentos difíciles del desarrollo de mi tesis doctoral y su experiencia docente hubiera sido posible este trabajo.

Para terminar, toda mi admiración y agradecimiento a las tres grandes mujeres de mi vida, que representan la experiencia, la complicidad y el orgullo de ser padre, y que ellas saben quiénes son.

Sobre el autor

José Luis Ponz Tienda es Doctor con calificación de Sobresaliente Cum Laude por la Universidad Politécnica de Valencia en el programa de doctorado de Arquitectura, Edificación, Urbanística y Paisaje, Máster en Planificación y Gestión de Procesos Empresariales por el Departamento de Investigación Operativa de la Facultad de matemáticas de Valencia y Arquitecto Técnico por la Universidad Politécnica de Valencia. Además es diplomado en Investigación Operativa y Máster en economía de la construcción.



También ha trabajado como Construction scheduler para Dragados y Construcciones SA, y ocupado diversos cargos directivos en diferentes empresas constructoras e inmobiliarias.

Actualmente trabaja como consultor de programación y procesos bajo metodología Lean Construction y es profesor del departamento de Construcciones Arquitectónicas en la Escuela de Ingeniería de Edificación y del programa de doctorado de Arquitectura, Edificación, Urbanística y Paisaje de la Universidad Politécnica de Valencia.

Índice

de contenidos

Agradecimientos	5
Sobre el autor	6
Prólogo	13
1. Un breve paseo por Excel 2010	17
1.1. Introducción a Excel 2010	17
1.2. Un primer vistazo a Excel 2010	18
1.3. Trabajar con Excel 2010.....	25
1.3.1. Escribir datos en una hoja de cálculo.....	26
1.3.2. Dar formato a los valores.....	26
1.3.2.1. Formatos fijos	27
1.3.2.2. Formatos condicionales	27
1.3.3. Imprimir libros y hojas	27
1.3.3.1. Configurar página	28
1.3.3.2. Configurar márgenes.....	28
1.3.3.3. Configurar encabezado y pie de página	29
1.3.3.4. Configurar hoja	29
1.3.4. Fórmulas y funciones	31
1.3.4.1. Referencias absolutas y relativas	31
1.3.4.2. Referencias a otras hojas del libro	33
1.3.4.3. Referencias a otras hojas de otros libros	33

2. Visual Basic VBA para Excel 2010.....	35
2.1. Introducción.....	35
2.2. Los Controles Active X	37
2.3. Procedimientos y módulos	39
2.3.1. Los procedimientos	39
2.3.2. Los módulos	40
2.4. Las variables y las operaciones.....	40
2.4.1. Tipo de variables.....	41
2.4.2. Declaración y ámbito de las variables.....	41
2.4.3. Variables multidimensionales; las matrices.....	43
2.4.4. Tipos definidos por el usuario	44
2.4.5. Operadores	44
2.5. Estructuras de decisión y de control.....	48
2.5.1. If ... Then ... Else.....	48
2.5.2. Select Case ... End Select	48
2.5.3. On...GoSub, On...GoTo	48
2.5.4. GoTo.....	49
2.5.5. GoSub ... Return	49
2.6. Estructuras de bucle.....	49
2.6.1. For ... To ... Next	50
2.6.2. While ... Wend	50
2.6.3. Do ... Loop	51
2.7. Acceso y manipulación de hojas y celdas de Excel	51
2.7.1. Objetos, métodos, propiedades y eventos	51
2.7.2. La colección Worksheets	52
2.7.3. La propiedad Cells	54
2.7.4. El objeto Range.....	54
2.8. Trabajando con formularios	56
3. Introducción a los Grafos de Proyectos	67
3.1. Una brevísima teoría de grafos	67
3.2. Tipos de Grafos de Proyecto	69
3.2.1. Grafos de actividades en flecha	70
3.2.2. Grafos de actividades en nodo	73
3.2.2.1. Relación de Final-Comienzo	74
3.2.2.2. Relación de Comienzo-Comienzo.....	74
3.2.2.3. Relación de Final-Final	75
3.2.2.4. Relación de Comienzo-Final	75
3.2.2.5. Simplificación de relaciones	76
3.3. Los tiempos de las tareas.....	76
3.3.1. Tiempo más pronto de empezar (Tmpe _i).....	77
3.3.2. Tiempo más pronto de terminar (Tmpt _i).....	77
3.3.3. Tiempo más tarde de empezar (Tmte _i)	78
3.3.4. Tiempo más tarde de terminar (Tmtt _i)	78

3.4. El algoritmo de cálculo de los tiempos	79
3.4.1. Holgura Total (Ht_i)	80
3.4.2. Holgura Libre (Hl_i)	80
3.4.3. Holgura independiente	80
3.4.4. Holgura condicional	80
3.4.5. Tarea crítica	81
3.4.6. Tarea no crítica	81
3.4.7. Camino crítico o ruta crítica	81
3.5. La representación matricial de los grafos y su cálculo	81
3.5.1. Introducción a la representación matricial	82
3.5.2. La representación matricial con Excel	84
3.5.3. El cálculo matricial de los tiempos con Excel	90
3.6. La representación del diagrama temporal	99
3.6.1. Diagramas temporales con gráficos de Excel	100
3.6.2. Diagramas temporales con importes y formatos condicionales	105
3.6.3. Diagramas temporales con fechas de calendario	115
3.6.3.1. Anexo: El cálculo de la fecha de Pascua	123
4. El Control del Proyecto	125
4.1. Introducción al control del proyecto	125
4.2. El control tradicional de Proyectos	127
4.2.1. El control tradicional por avance	127
4.2.2. El control tradicional por matriz	131
4.3. El control de Proyectos por el Método del Valor Ganado	138
4.3.1. Introducción al control por Valor Ganado	138
4.3.2. Los índices de primer orden	139
4.3.2.1. El CPTP (Coste Presupuestado del Trabajo Programado)	140
4.3.2.2. El CRTR (Coste Realizado del Trabajo Realizado)	140
4.3.2.3. El CPTR (Coste Presupuestado del Trabajo Realizado)	141
4.3.3. Los índices de segundo orden	141
4.3.3.1. La desviación de coste	141
4.3.3.2. La desviación de programa	142
4.3.3.3. La programación ganada	144
4.3.3.4. La adherencia a la programación	145
4.3.4. Los índices de tercer orden	147
4.3.4.1. Los Índices de eficiencia	148
4.3.4.2. Las proyecciones	148
4.4. Implementación del control por el Valor Ganado en Excel 2010	150
4.4.1. El CPTP en Excel 2010	150
4.4.2. El coste realizado CR y el trabajo realizado TR en Excel 2010	152
4.4.2.1. El coste realizado CR	152
4.4.2.2. El trabajo realizado TR	154
4.4.3. El CPTR en Excel 2010	155

4.4.4. Los índices de segundo y tercer orden en Excel 2010	156
4.4.4.1. Las desviaciones de coste y programa en Excel 2010	157
4.4.4.2. Los índices de rendimiento de coste y de programa en Excel 2010.....	159
4.4.4.3. La programación ganada en Excel 2010	159
4.4.4.4. La adherencia a la programación en Excel 2010.....	163
4.4.5. Los gráficos de control por valor ganado en Excel 2010	165
4.4.5.1. CPTP vs CRTR	165
4.4.5.2. CPTP vs CPTR	166
4.4.5.3. Desviación de coste y programa.....	166
4.4.5.4. Adelantos y atrasos	168
4.4.5.5. Índices de rendimiento de coste y de programa	168
4.5. Anexo al control del proyecto.....	169
5. La gestión del riesgo en los proyectos	171
5.1. Breve historia de la gestión del riesgo de los proyectos	171
5.2. El PERT tradicional	172
5.2.1. Las estimaciones del experto y la distribución β ta.....	173
5.2.2. El teorema del límite central y la distribución Normal	176
5.2.3. Los buffers de proyecto.....	177
5.2.4. Las limitaciones del PERT.....	178
5.3. La simulación de Monte Carlo	179
5.3.1. Las distribuciones de probabilidad	180
5.3.1.1. La distribución determinista.....	180
5.3.1.2. La distribución uniforme.....	180
5.3.1.3. La distribución triangular.....	181
5.3.2. La generación de duraciones aleatorias	182
5.3.2.1. La simulación de la distribución uniforme.....	183
5.3.2.2. La simulación de la distribución triangular	183
5.3.2.3. La simulación de la distribución β ta	183
5.3.3. El índice de criticidad y el valor crítico	183
5.3.3.1. El índice de criticidad.....	184
5.3.3.2. El valor crítico	184
5.3.4. La simulación de Monte Carlo con Excel 2010.....	184
5.3.5. El diagrama temporal con buffer de proyecto	190
5.4. La lógica difusa, borrosa o fuzzy	191
5.4.1. El número difuso o fuzzy	193
5.4.1.1. El núcleo del número difuso o fuzzy.....	194
5.4.1.2. El soporte inferior del número difuso o fuzzy	194
5.4.1.3. El soporte superior del número difuso o fuzzy	194
5.4.2. El alfa-corte $S_\alpha()$	194
5.4.3. La aritmética difusa o fuzzy.....	195
5.4.4. La ordenación de números difusos	195
5.4.5. La criticidad difusa.....	196

5.4.6. Implementación de la lógica difusa en Excel	197
5.4.7. El diagrama temporal difuso.....	204
5.5. Unas reflexiones finales.....	204
6. El problema de la optimización de costes	207
6.1. Introducción al TCTP	207
6.2. El problema lineal del TCTP	208
6.2.1. Los costes de producción del proyecto	208
6.2.2. Los costes de estructura del proyecto.....	211
6.2.3. Los costes totales del proyecto y su convexidad.....	213
6.2.4. La formulación del modelo matemático del problema lineal.....	213
6.2.5. El Solver de Excel	214
6.2.6. Implementación del modelo lineal en Excel 2010	217
6.3. El problema no lineal del TCTP	221
6.3.1. Las penalizaciones y bonificaciones.....	222
6.3.2. El modelado del problema no lineal	222
6.3.3. La complejidad algorítmica del problema no lineal	223
6.3.4. Implementación del modelo no lineal en Excel 2010	224
6.4. Anexo: Implementación del TCTP con enumeración explícita.....	228
7. El problema de la optimización de recursos	231
7.1. Introducción al RCPSP	231
7.2. El problema de la Nivelación	232
7.2.1. El histograma de recursos.....	232
7.2.2. La optimización por mínimos cuadrados.....	233
7.2.3. La formulación matemática de la nivelación.....	234
7.2.4. La complejidad algorítmica de la nivelación.....	235
7.2.5. La implementación de la nivelación en Excel 2010	236
7.2.5.1. El modelado de nivelación por mínimos cuadrados.....	236
7.2.5.2. El modelado de nivelación por mínimos cuadrados con recuperación de holguras	243
7.2.5.3. El modelado de nivelación multi-modo por mínimos cuadrados con recuperación de holguras.....	246
7.3. El problema de la Secuenciación	251
7.3.1. La formulación matemática del RCPSP.....	253
7.3.2. La complejidad algorítmica de la secuenciación.....	253
7.3.3. La implementación de la secuenciación en Excel 2010	254
7.3.3.1. La implementación de la secuenciación con un único modo.....	255
7.3.3.2. La implementación de la secuenciación multi-modo.....	260
7.4. Anexo.....	263
8. El Proyecto como inversión	265
8.1. Introducción.....	265
8.2. El análisis de viabilidad de un proyecto.....	266

8.2.1. Medidas estáticas y dinámicas de viabilidad.....	266
8.2.2. El Valor Actual Neto.....	267
8.2.3. Las tasas de interés y de inflación.....	268
8.2.3.1. Las tasas de interés.....	268
8.2.3.2. La tasa de inflación.....	269
8.2.4. Diferentes modos de coste pero un solo modo de cobro.....	269
8.2.5. Adelantar o retrasar la ejecución de las tareas.....	270
8.3. La implementación de la optimización de la rentabilidad con Excel 2010.....	270
8.3.1. La optimización del VAN con múltiples modos de ejecución.....	270
8.3.2. La optimización del VAN con múltiples modos de ejecución y traslados.....	278
A. Glosario de símbolos y abreviaturas.....	285
B. Bibliografía.....	291
Índice alfabético.....	299

Prólogo

Existen muchos programas de gestión de proyectos en el mercado, unos muy sencillos que tan solo me construyen un diagrama temporal, otros para usuarios que hacen un uso esporádico de estas herramientas, y luego están los programas profesionales que disponen de una gran cantidad de herramientas incorporadas para gestionar adecuadamente las tareas de un proyecto.

Lo que muy poca gente conoce, es que la gestión de proyectos es una disciplina de la Investigación Operativa que estudia cómo organizar y ejecutar de "*la mejor manera posible*" las tareas de los proyectos, para que este se ejecute en un plazo estipulado, con unos recursos disponibles y con el menor coste posible.

La expresión de "*la mejor manera posible*", es lo que se conoce como optimización, y que por desgracia para aquellos que nos dedicamos a la consultoría de empresas en gestión de proyectos echamos en falta en los programas comerciales profesionales, y que por difícil que sea de creer, hacen uso de obsoletas y limitadas metodologías aproximadas, que en algunos casos incluso ofrecen soluciones no solo pobres sino incluso incompletas e incorrectas.

Los grandes avances que se han realizado en el mundo de la gestión de proyectos paralelamente al avance de las tecnologías, parecen haber pasado incomprensiblemente desapercibidos por parte de los grandes fabricantes de software, quizás porque consideran erróneamente que estas tecnologías son demasiado complejas para el público en general y los profesionales.

Vivimos en entorno muy difícil, en un mundo cada vez más competitivo, donde la eficacia ya no es el objetivo perseguido por las empresas, sino que la constante búsqueda de la eficiencia y la mejora de la productividad es el único camino para poder competir en calidad y costes con garantías de éxito.

Para alcanzar esa eficiencia y mejora en la productividad de los proyectos, no nos podemos conformar con buenas soluciones, sino que hemos de hacer uso de los mayores avances de que dispongamos en cada momento para lograr la excelencia, el óptimo.

Este libro pretende acabar con el mito de que la gestión óptima de proyectos está restringida al ámbito académico y científico, sin aplicaciones reales en la práctica, al requerir de complejas herramientas matemáticas tan solo al alcance de algunos investigadores.

Con la aplicación de Excel, he conseguido acercar de una manera práctica y sencilla, la gestión óptima de proyectos a mis alumnos de Ingeniería de Edificación de la Universidad Politécnica de Valencia, pero aún quedaban cosas por hacer.

Con la nueva versión de Excel 2010, y la aplicación de los últimos avances en optimización evolutiva que incorpora el complemento Solver, podemos afrontar y solucionar de una manera increíblemente sencilla e intuitiva, problemas reales con grandes implicaciones prácticas, que hasta ahora parecían muy lejanas.

Este libro es una introducción a la gestión óptima de proyectos, y esta destinado a un público con algunos conocimientos previos de Excel, estudiantes de Administración y Dirección de Empresas, Ciencias Económicas, Ingenierías o Arquitectura, pero principalmente se ha pensado para profesionales que deseen ofrecer el mejor plan de proyecto a sus empresas y clientes, sin necesidad de tener conocimientos en optimización.

Además su enfoque didáctico, con una estructuración por temas en función de su complejidad, lo hace perfectamente adecuado para aquellos profesores y docentes que necesiten una guía o ayuda para sus clases.

Quedan muchos temas por tratar, principalmente sobre la organización y filosofías de gestión de proyectos, el ámbito financiero, los proyectos con relaciones generalizadas y la fragmentación de las tareas, pero seguro

que con toda la información que se ha suministrado en este libro el lector quedará más que satisfecho, descubriendo caminos que no conocía o no se atrevía a explorar.

Espero que este libro sea útil.

En materia de Planificación, jamás un movimiento inútil; en materia de estrategia ningún paso en vano.

El Arte de la Guerra.

Sun Tzu.

Siglo VI a.C.

1

Un breve paseo por Excel 2010

1.1. Introducción a Excel 2010

Solemos denominar a Excel como una hoja de cálculo, pero no seríamos justos si continuamos llamándola así. Para ser más precisos, deberíamos decir que Excel es un libro de hojas de cálculo, aunque seguiríamos sin ser justos, porque es mucho más que eso.

En sus comienzos, Excel sí que era una hoja de cálculo, pero en la actualidad es una herramienta de propósito general que hace uso de hojas de cálculo para organizar y procesar datos de diferente naturaleza de una manera tremendamente eficiente.

A través de su historia, Excel ha pasado por diferentes versiones, unas más afortunadas que otras, y a decir verdad, la primera vez que ejecuté Excel 2010 pensé que era poco más que un cambio de aspecto al no aparecer el botón de Office, pero tengo que reconocer que me equivoqué.

Excel 2010 presenta muchas mejoras que no están fácilmente a la vista, algunas de ellas, como el cambio de nombre de algunas funciones, enfadará a más de uno, y posiblemente con razón, pero dispone de tantas utilidades y funciones nuevas (o mejoras en las ya existentes) que he de reconocer que me encuentro encantado con esta nueva versión. Pero lo que más me ha impresionado ha sido su nuevo complemento Solver con la aplicación de algoritmos evolutivos para la resolución de problemas de optimización no

lineal y discreta. Ha sido algo que no me esperaba, puesto que aunque en el mercado existían complementos Solver con algoritmos evolutivos, estos eran verdaderamente caros y lejos del público en general, que impedía afrontar algunos de los problemas que veremos a lo largo de este libro.

El lector ya se habrá dado cuenta de que este libro no es para aquellos que nunca han utilizado Excel (aunque nunca es tarde para empezar), requiriendo de cierta familiarización con su entorno y metodología de trabajo, pero no obstante, para aquellos que quieran dar ese maravilloso paso y adentrarse en el mundo de la gestión de proyectos con Excel 2010 daremos un breve paseo que espero sea agradable, porque aprender algo nuevo, sobre todo si es útil, siempre es una experiencia gratificante.

1.2. Un primer vistazo a Excel 2010

La primera vez que ejecutamos Excel 2010 nos aparece una pantalla con dos partes bien diferenciadas, una parte superior que se denomina cinta de opciones, y otra inferior que son las hojas del libro caracterizada por tener la apariencia de retícula o matriz (véase la figura 1.1).

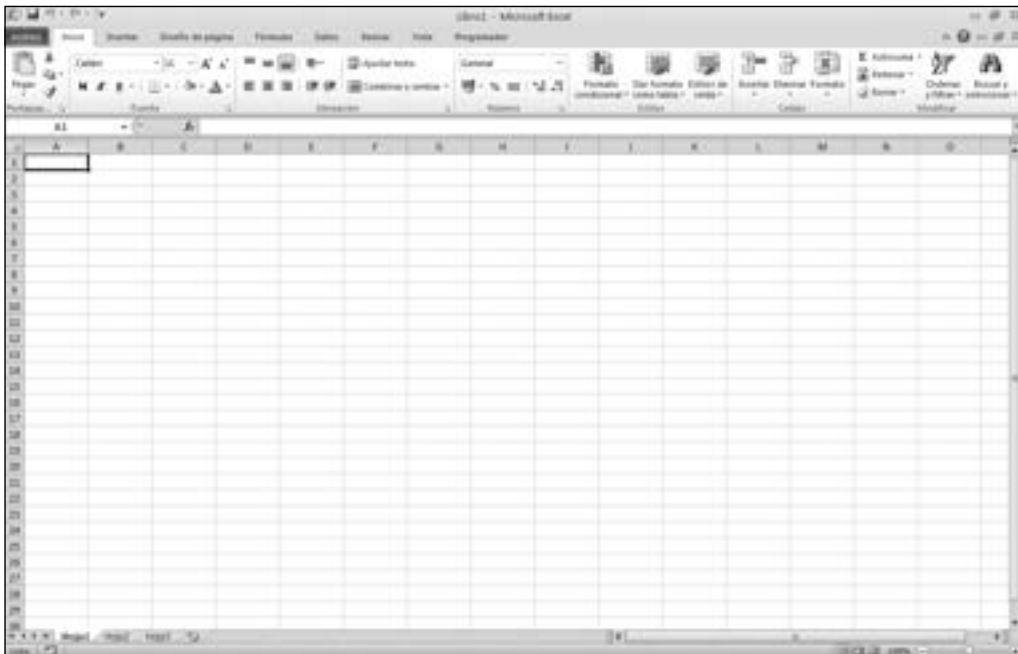


Figura 1.1. Pantalla de Excel 2010.

La zona inferior es el área de trabajo, y está formado por una retícula o matriz de celdas, organizado por filas y columnas, correspondiendo cada una de ellas a un dato (véase la figura 1.2).

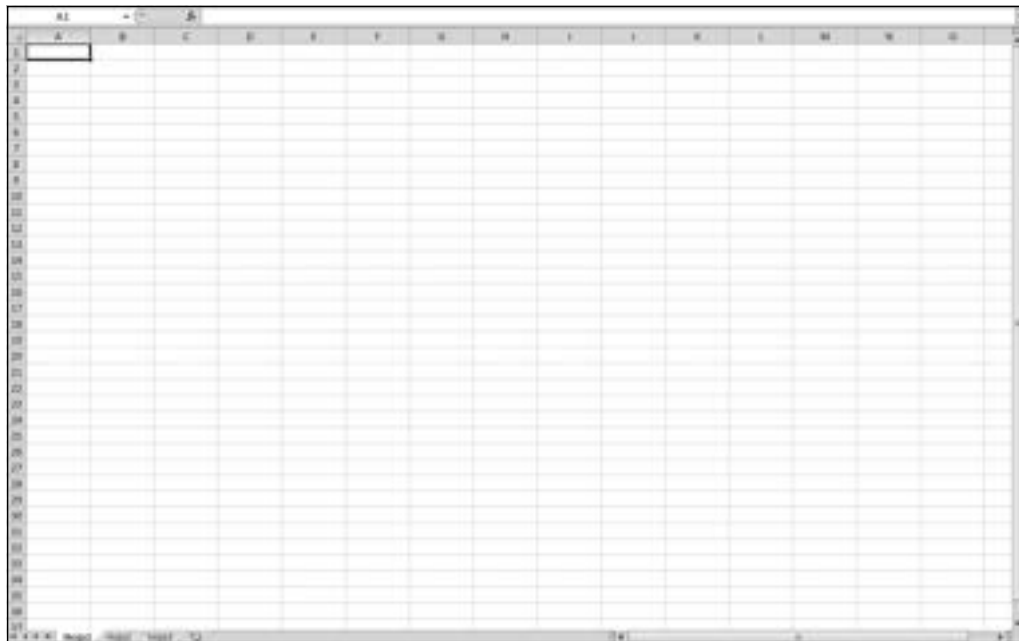


Figura 1.2. Área de trabajo de Excel 2010.

A su vez, la zona de trabajo presenta en su zona superior la barra de fórmulas, que nos indica cual es la celda activa actualmente de la hoja en la que nos encontramos, y el valor, función o fórmula que contiene (véase la figura 1.3).

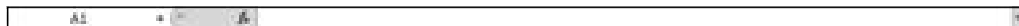


Figura 1.3. Barra de fórmulas de Excel 2010.

En la zona inferior del área de trabajo, nos encontramos con las hojas del libro que están disponibles en el libro con el que estamos trabajando, permitiendo cambiar de hojas simplemente haciendo clic sobre su pestaña correspondiente, o añadir una nueva hoja haciendo clic sobre la última pestaña, como se aprecia en la figura 1.4.

La zona superior o cinta de opciones, es donde se encuentran las utilidades para poder trabajar de forma rápida con las hojas y celdas de nuestro libro.



Figura 1.4. Hojas del libro de Excel 2010.

La cinta de opciones se organiza en fichas que se abren al hacer clic con el ratón sobre cada una de ellas, y por defecto contiene los elementos que se describen a continuación:

- **Archivo:** En la pestaña archivo de la cinta de opciones, nos encontramos con las herramientas básicas para la gestión de los archivos como **abrir**, **cerrar**, **guardar** o **nuevo**, junto a otras herramientas que nos permitirán imprimir la hoja o el libro de Excel además de cambiar ciertas opciones de configuración y complementos de Excel 2010.
- **Inicio:** Esta es la pestaña activa por defecto, y en ella se encuentran las utilidades de uso más común cuando estamos trabajando con una hoja de Excel, teniendo el aspecto de la figura 1.5.



Figura 1.5. Pestaña Inicio de la cinta de opciones de Excel 2010.

- **Portapapeles:** Gestión del portapapeles de Windows, como copiar, cortar y pegar, además tiene otras dos utilidades muy prácticas que son **Copiar formato**, y **Pegar** tan solo algunas de las características de una celda.
- **Fuente:** Para modificar el tipo de fuente, tamaño, color de fondo y color de fuente de las hojas y celdas de nuestro libro. Además podremos añadir y modificar los bordes de las celdas.
- **Alineación:** Cambio de la alineación horizontal y vertical de los valores que se presentan en las celdas. Una utilidad muy importante es **Combinar y centrar**, gracias a la cual podré unir dos o más celdas consecutivas en una sola celda.
- **Número:** Contiene los tipos formatos de los datos que se almacenan en las celdas de nuestras hojas y que podemos ajustar para que se muestren de la forma correcta o deseada. Pueden ser de diferentes tipos en función de su naturaleza, como general, número, moneda,

contabilidad, fecha, hora, porcentaje, fracción, científica, texto, especial y personalizada, aunque el tipo por defecto es el denominado Variant, que puede adoptar de cualquiera de los anteriores.

- **Estilos:** Aquí se encuentran las herramientas para dar formatos especiales, como los formatos condicionales, formatos como tabla y los estilos de visualización de las celdas.
- **Celdas:** Es utilizado para realizar las gestiones relativas a las celdas, filas y columnas de nuestra hoja, como insertar, eliminar o cambiar algunas características como el alto y el ancho de filas y columnas.
- **Modificar:** Esta pestaña es algo así como el cajón de sastre, encontrándonos con utilidades como la suma automática, la ordenación de valores por filas o columnas, o la búsqueda y/o reemplazo de valores dentro de nuestra hoja o libro.
- **Insertar:** En esta pestaña nos encontramos con las utilidades relativas a la gestión de elementos especiales de nuestro libro de Excel, como gráficos y sencillos dibujos mediante un editor propio de Excel (véase la figura 1.6).



Figura 1.6. Pestaña Insertar de la cinta de opciones de Excel 2010.

- **Tablas:** Con esta utilidad podremos agrupar las celdas de nuestra hoja en tablas y tablas dinámicas para que su gestión y análisis sea más eficiente.
- **Ilustraciones:** Aquí nos encontramos las herramientas para trabajar con imágenes, formas prediseñadas y modificables al ser parametrizadas.
- **Gráficos:** Esta es una de las herramientas más útiles de Excel, permitiendo realizar gráficos de los valores de nuestras hojas de muy distintos tipos, por ejemplo: columna, línea, circular, barra, área, dispersión, etc.
- **Minigráficos:** esta herramienta es una de las grandes novedades de Excel 2010, permitiendo incrustar pequeños gráficos en el interior de una celda.

- **Filtro:** Esta herramienta se utiliza para gestionar las conexiones de datos existentes.
- **Vínculos:** Con esta herramienta podremos añadir hipervínculos a otros archivos situados en nuestro ordenador como archivos de office, imágenes o páginas Web.
- **Texto:** Añadir cuadros de textos sencillos o artísticos con WordArt, cambiar el encabezado y pie de página, añadir firmas digitales o incrustar archivos como objetos.
- **Símbolos:** En este grupo, se encuentran dos herramientas, una para insertar símbolos y caracteres especiales, y otra un fantástico editor de ecuaciones matemáticas que también es una nueva herramienta que tan solo estaba disponible hasta ahora en Word 2007.
- **Diseño de página:** En esta pestaña se encuentran las utilidades que nos permitirán configurar algunos parámetros referentes a la visualización y formatos de impresión de las hojas de nuestro libro de Excel (véase la figura 1.7).



Figura 1.7. Pestaña Diseño de página de la cinta de opciones de Excel 2010.

- **Temas:** Los temas son conjuntos de opciones de formato que incluyen colores, fuentes y efectos, y que se aplica a toda la hoja que está activa.
- **Configurar página:** Aquí encontraremos las herramientas para personalizar los márgenes del papel, la orientación en horizontal o apaisado, el tamaño del papel, el área o rango de celdas que queremos imprimir, los saltos de página, el fondo de papel y las líneas y columnas que conforman los encabezamientos de los datos y que se repetirán en todas las hojas impresas.
- **Ajustar área de Impresión:** Con esta herramienta podemos obligar a Excel a que el área de impresión sea ajustado a un determinado número de páginas de alto o de ancho, o bien a un porcentaje sobre el tamaño original.
- **Opciones de la hoja:** Podremos determinar si aparecerán o se imprimirán las líneas de cuadrícula y los encabezados.

- **Organizar:** Con estas herramientas modificaremos el orden de visualización y otros ajustes de alineación de los objetos gráficos insertados en nuestras hojas.
- **Fórmulas:** Aquí encontraremos todas las funciones y fórmulas disponibles en Excel 2010, la posibilidad de asignar nombres especiales para rangos y celdas, o herramientas de auditoría.



Figura 1.8. Pestaña Fórmulas de la cinta de opciones de Excel 2010.

- **Biblioteca de funciones:** Las funciones son fórmulas de ecuaciones que sirven para realizar cálculos, devolver información, modificar el contenido de otras celdas, comprobar condiciones, etc. Las fórmulas se organizan según su función y pueden ser financieras, lógicas, de texto, fecha y hora, búsqueda y referencia, matemáticas, trigonométricas y otras como las estadísticas y de ingeniería.
- **Nombres definidos:** Mediante esta utilidad, podremos asignar y administrar nombres a celdas y rangos de celdas para hacer más cómodo la creación de fórmulas.
- **Auditoría de fórmulas:** Contiene un conjunto de herramientas para auditar las fórmulas escritas, como ver las celdas de las que se depende o las que dependen de la celda activa.
- **Cálculo:** Este grupo de herramientas establece las opciones de cálculo de las hojas y fuerza su cálculo.
- **Soluciones:** Éste es el conjunto de herramientas para conversiones de moneda.
- **Datos:** Ya se ha comentado anteriormente que Excel es un potente administrador y manipulador de datos, y en esta pestaña tenemos las herramientas más comunes para automatizar su gestión (véase la figura 1.9).



Figura 1.9. Pestaña Datos de la cinta de opciones de Excel 2010.

- **Obtener datos externos:** Aquí encontramos las herramientas para establecer diferentes conexiones a bases de datos desde diferentes fuentes, como Access, Web o texto.
- **Conexiones:** Aquí encontraremos las herramientas para actualizar las conexiones existentes a bases de datos.
- **Ordenar y filtrar:** Mediante este conjunto de herramientas podremos ordenar los datos que tengamos en rangos de celdas y tablas, o filtrarlos según diferentes criterios.
- **Herramientas de datos:** Este grupo de herramientas nos permite consolidar los datos de varias celdas en una, separar los datos de una celda en varias así como otras herramientas de análisis y validación.
- **Esquema:** Este conjunto de herramientas es muy útil para organizar las filas o columnas en grupos, con la posibilidad de ocultarlas o mostrarlas jerárquicamente.
- **Análisis:** Aquí encontramos la herramienta Solver, para resolver problemas de optimización, que en esta nueva versión de Excel 2010, presenta como novedad la posibilidad de utilizar algoritmos evolutivos muy avanzados.
- **Revisar:** Esta ficha reúne varios grupos de herramientas para la revisión ortográfica y traducción, incluir comentarios aclaratorios o bien proteger las hojas de modificaciones accidentales (véase la figura 1.10).



Figura 1.10. Pestaña Revisar de la cinta de opciones de Excel 2010.

- **Revisión:** En el grupo revisión encontramos herramientas de corrección ortográfica y diccionario de sinónimos.
- **Idioma:** Es una utilidad para la traducción de texto entre diferentes idiomas.
- **Comentarios:** En las hojas muy compleja, o bien que se han de compartir con otras personas, es interesante añadir comentarios a las celdas, y con este conjunto de utilidades podremos añadirlos y administrarlos.

- **Cambios:** Bien sea para evitar cambios accidentales e indeseados en las celdas, o para ocultar sus fórmulas de miradas no deseadas, podemos establecer diferentes niveles de protección protegidos con contraseña, bien a nivel de hoja o de libro.
- **Vista:** En esta pestaña, podemos encontrar diferentes utilidades para personalizar la visión de las hojas de nuestro libro de Excel (véase la figura 1.11).



Figura 1.11. Pestaña Vista de la cinta de opciones de Excel 2010.

- **Vistas de libro:** Podemos ver las hojas del libro en modo de diseño o normal, como influyen los saltos de página, o bien ver tan solo la zona de trabajo ocultando toda la cinta de opciones, incluso las pestañas.
- **Mostrar:** Conjunto de casillas de selección para mostrar una regla en modo diseño, ocultar las líneas de cuadrículas la barra de fórmulas y los títulos.
- **Zoom:** Podemos cambiar el zoom y ver los contenidos de la hoja con mayor o menor nivel de detalle.
- **Ventana:** Nos permite crear y administrar diferentes vistas de nuestra hoja, pudiendo así analizar simultáneamente zonas muy distantes entre sí.
- **Macros:** Esta utilidad también se encuentra en la pestaña Programador, y con ella podremos grabar un conjunto de acciones que realicemos y asignarle un determinado nombre y un atajo de teclado.

1.3. Trabajar con Excel 2010

Para poder empezar a trabajar con Excel 2010, tenemos que conocer como se hace referencia a las celdas de nuestras hojas.

Las celdas tienen por defecto una forma de identificarse inequívocamente, que es mediante la referencia *columnafila*, siendo la *columna* las

letras consecutivas del alfabeto (al llegar a Z continua como AA hasta la AZ, AAA hasta AZZ y así de forma consecutiva hasta XFD, que un máximo de 16.384 columnas), y la *fila* el número ordinal correspondiente hasta un máximo 1.048.576 filas (véase la figura 1.12).

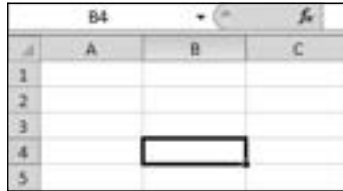


Figura 1.12. Estilo de referencia *columnafila*.

A veces se hace recomendable cambiar este sistema de referencia mediante un sistema *filacolumna*, donde *fila* es la letra F con su ordinal y la *columna* la letra C con su ordinal, para lo que ejecutaremos Archivo>Opciones>Fórmulas>Estilo de referencia F1C1, y obtendremos el resultado que se muestra en la figura 1.13.

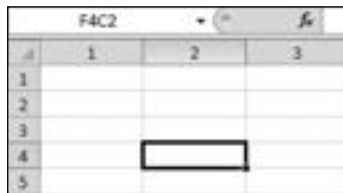


Figura 1.13. Estilo de referencia *filacolumna*.

1.3.1. Escribir datos en una hoja de cálculo

Para incluir datos en una hoja, tan solo tendremos que ubicarnos en la celda que deseemos y escribir. Excel interpretará automáticamente el tipo de dato que estamos escribiendo, diferenciando si es un número, un texto o una fecha bien sea escrita con el separador "/" o "-".

1.3.2. Dar formato a los valores

Los datos que tenemos escritos en nuestras hojas podemos hacer que se presenten de muy diversas maneras, cambiando su fuente, alineación, bordes, fondo y obligando a que adopte un tipo determinado como considerar

a un número como texto, son los denominados formatos fijos, pero también pueden adoptar un formato en función de su valor.

1.3.2.1. Formatos fijos

Los formatos fijos son los formatos que establecemos a las celdas o a las hojas de forma predeterminada, y pueden ser bien definidos por nosotros o por medio de los tipos predefinidos, muy útiles cuando trabajamos con tablas o títulos.

1.3.2.2. Formatos condicionales

Los formatos condicionales, como su propio nombre indica, establecen la forma en que se presentarán los datos en función de su valor individual, en referencia a otros valores, como resultado de una fórmula, etc.

Excel tiene unos formatos condicionales preinstalados por defecto, aunque nosotros podemos establecer diferentes reglas, que son las que se describen a continuación:

- Según sus valores.
- Según su contenido.
- Según un rango de valores.
- Según se esté por encima o por debajo de un promedio.
- Diferenciando valores únicos o duplicados.
- Según el resultado de una fórmula.

Las reglas se pueden acumular, y se ejecutan de manera consecutiva según la secuencia en que han sido establecidas, aunque esta secuencia puede ser modificada.

1.3.3. Imprimir libros y hojas

El final de todo el trabajo que hacemos con una hoja de cálculo suele consistir en su impresión, por lo tanto vamos a prestar un especial interés en este punto por la trascendencia que tiene, especialmente cuando estamos trabajando con hojas muy largas como las que son habituales en gestión de proyectos.

1.3.3.1. Configurar página

Lo primero que tendremos que hacer será configurar la página, estableciendo la orientación y el tamaño del papel de la impresora que utilizaremos, además deberemos indicar cómo queremos que Excel interprete la relación entre la hoja de cálculo y el papel impreso mediante el ajuste de escala.

Para poder establecer el ajuste de escala disponemos de dos posibilidades, una como porcentaje del tamaño normal de impresión haciendo más o menos grande el resultado que aparecerá por la impresora, y otro es obligar a Excel a que se ajuste a una determinada cantidad de hojas de papel de ancho y de alto.

Una vez ajustados estos primeros valores, es recomendable hacer clic en **Vista preliminar**, y ver el resultado final por pantalla antes de hacer la impresión definitiva (véase la figura 1.14).

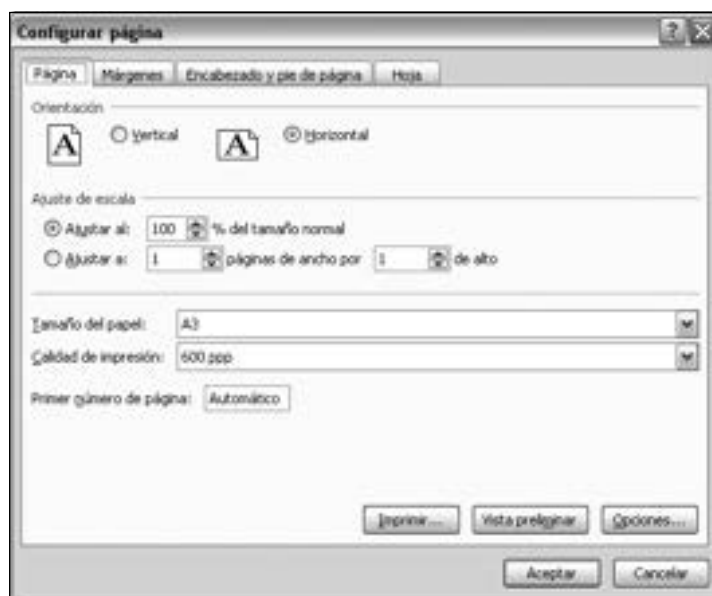


Figura 1.14. Configurar página: Página.

1.3.3.2. Configurar márgenes

Mediante el ajuste de los márgenes de papel, podremos establecer la distancia entre el borde exterior del papel en cada uno de sus lados con el área de impresión destinada a la hoja de cálculo.

También podremos obligar a Excel a que centre horizontal o verticalmente la hoja de cálculo en el papel impreso (véase la figura 1.15).

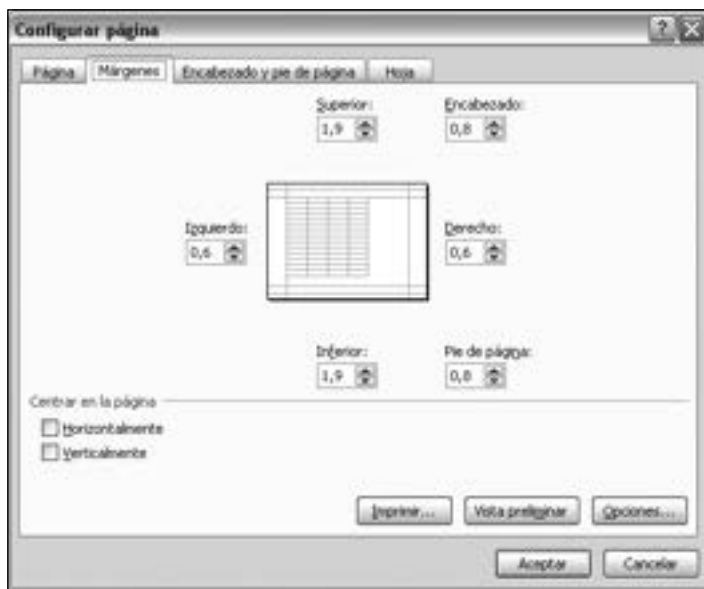


Figura 1.15. Configurar página: Márgenes.

1.3.3.3. Configurar encabezado y pie de página

Los encabezados y pies de página se ubican en la zona superior e inferior de todas las páginas impresas y no aparecen en la hoja de cálculo salvo que seleccione **vista diseño de página**. Entre los elementos que podemos incluir son imágenes, número de página, número de páginas totales, etc. (véase la figura 1.16).

1.3.3.4. Configurar hoja

Esta es la parte más importante en el momento de configurar la impresión, y muy especialmente para imprimir diagramas temporales muy largos, concretamente el establecimiento del área de impresión y la impresión de títulos. El área de impresión es el rango de celdas de toda nuestra de cálculo que deseamos imprimir.

Los títulos pueden ser en la parte superior o a la izquierda. Los títulos en la parte superior, son unas filas que se repetirán de forma relativa al área impresa, que para los diagramas temporales corresponderá a las filas del

calendario, y los títulos a la izquierda son las columnas, en nuestro caso el nombre de las tareas, duraciones y tiempos que se repetirán a la izquierda de todas las páginas (véase la figura 1.17).

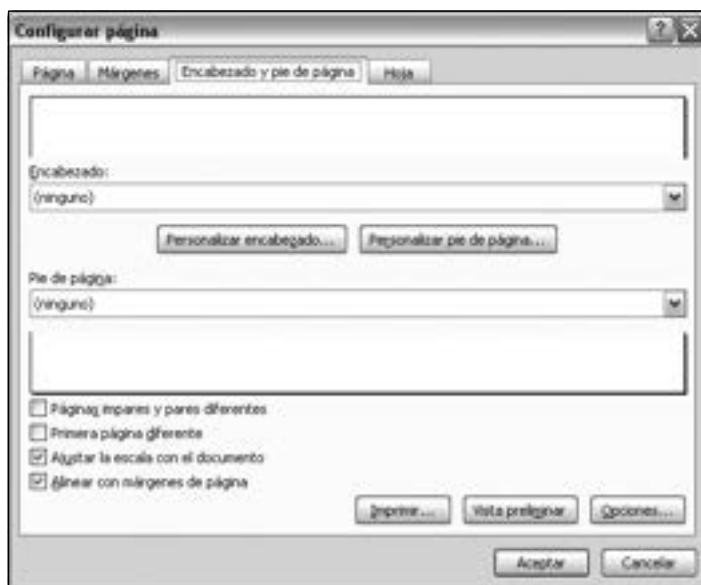


Figura 1.16. Configurar página: Encabezado y pie de página.

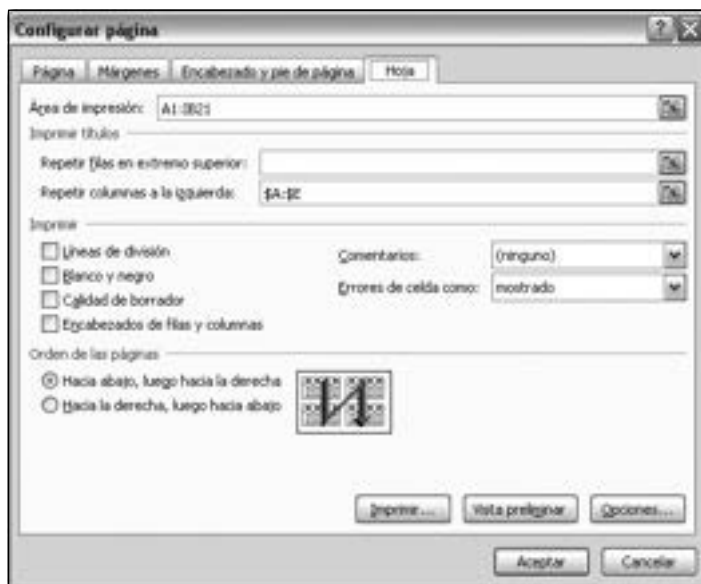


Figura 1.17. Configurar página: Hoja.

1.3.4. Fórmulas y funciones

Para terminar con este breve paseo por Excel 2010, vamos a ver cómo escribir fórmulas e insertar funciones, así como ver la diferencia entre las relaciones y absolutas, y como hacer referencia a celdas que se encuentran en otras hojas o incluso en otros libros.

La manera de escribir una fórmula es ubicarnos en la celda en la que deseamos escribirla y empezar con el signo "=" que le indica a Excel que los elementos que incluiremos a continuación son una fórmula.

La sintaxis general sería: = celda *operador* celda, pudiendo incluir operaciones anidadas por medio de paréntesis como = celda *operador* ((celda *operador* celda) *operador* celda).

Las celdas las podemos establecer bien escribiendo su referencia directamente o, mediante las teclas de cursor o más cómodamente con el uso del ratón.

Una vez escrita la fórmula, podemos revisarla y editarla de forma visual, presionando la tecla **F2** sobre la celda que contiene la fórmula, pudiendo mover las referencias coloreadas a otra posición, como se aprecia en la figura 1.18.

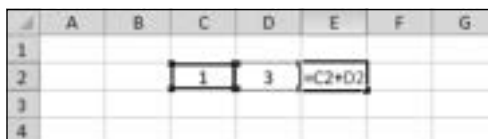


Figura 1.18. Edición de referencias de celda.

Para insertar una función, la escribiremos directamente respetando su sintaxis, o haremos uso de la utilidad de que disponemos para tal fin con **Fórmulas>Insertar función>categoría>función** (véase la figura 1.19).

1.3.4.1. Referencias absolutas y relativas

Las celdas pueden ser referenciadas con carácter relativo y absoluto. Por defecto Excel considera que las referencias a las celdas son relativas, de tal manera que si establecemos una fórmula como la suma de su celda a la izquierda y su celda superior y la copiamos a lo largo de un rango, se mantendrá siempre esta referencia. Por ejemplo, en la siguiente imagen se muestra el resultado de copiar la celda C3 con la fórmula =B3+C2, a lo largo del rango C3:G7 (véase la figura 1.20).

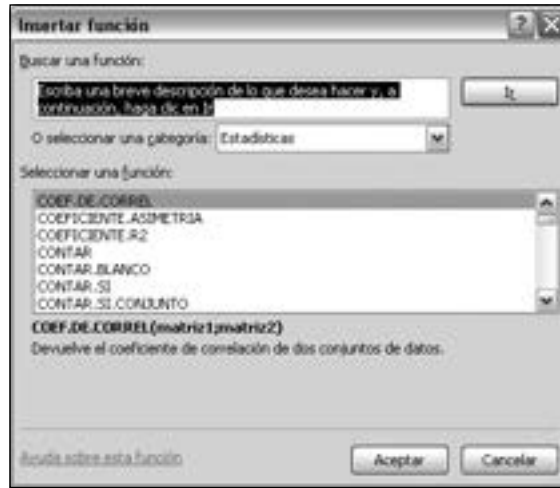


Figura 1.19. Insertar función.

	A	B	C	D	E	F	G	H
1								
2			1	2	3	4	5	
3		1	2	4	7	11	16	
4		2	4	8	15	26	42	
5		3	7	15	30	56	98	
6		4	11	26	56	112	210	
7		5	16	42	98	210	420	
8								

Figura 1.20. Referencias relativas.

El resultado no ha sido el deseado en un principio, puesto que la intención era que la referencia fuera siempre a la primera fila y columna de la tabla.

Para introducir referencia absolutas, hemos de presionar F4 después de referenciar cada celda, colocando Excel un signo "\$" antes de la fila y la columna correspondiente. De esta forma la celda C3 tendrá la fórmula $=\$B\$3+\$C\2 , que si volvemos a copiar al rango C3:G7, obtendremos el resultado mostrado en la figura 1.21.

El resultado tampoco ha sido el deseado, porque lo que queremos realmente es una referencia mixta, tomando como absoluta solamente la primera columna y la primera fila, siendo el resto relativo a su celda y columna correspondiente, debiendo rectificar la fórmula eliminando los signos "\$" correspondientes, quedando la celda C3 como $=\$B3+C\2 , y si la copiamos al rango C3:G7, tendremos por fin el resultado que muestra la figura 1.22. Que es precisamente el resultado deseado.

	A	B	C	D	E	F	G	H
1								
2			1	2	3	4	5	
3		1	2	2	2	2	2	
4		2	2	2	2	2	2	
5		3	2	2	2	2	2	
6		4	2	2	2	2	2	
7		5	2	2	2	2	2	
8								

Figura 1.21. Referencias absolutas.

	A	B	C	D	E	F	G	H
1								
2			1	2	3	4	5	
3		1	2	3	4	5	6	
4		2	3	4	5	6	7	
5		3	4	5	6	7	8	
6		4	5	6	7	8	9	
7		5	6	7	8	9	10	
8								

Figura 1.22. Referencias mixtas.

1.3.4.2. Referencias a otras hojas del libro

También podemos hacer referencia a celdas ubicadas en otras hojas respetando la siguiente sintaxis: `=nombredelahoja!celda`, o bien estableciéndola seleccionando la hoja en las pestañas inferiores de la hoja y haciendo clic en la celda correspondiente.

1.3.4.3. Referencias a otras hojas de otros libros

Existe la posibilidad de hacer referencia a celdas ubicadas en otros libros con la siguiente sintaxis: `='[nombredelarchivo]nombredelahoja'!celda`, donde *nombredelarchivo* es el nombre del archivo Excel, debiendo incluir la ruta de acceso completa y la extensión del archivo donde está ubicada la celda o rango de celdas, pudiendo establecerla también de forma sencilla con el uso del ratón y la precaución de tener abiertos simultáneamente los dos libros.

2

Visual Basic VBA para Excel 2010

2.1. Introducción

VBA es el acrónimo de "*Visual Basic for Applications*", y es una integración del lenguaje de programación Visual Basic para las aplicaciones de Microsoft Office, mediante el cual se pueden crear soluciones personalizadas que complementan las funcionalidades originales.

Existen dos formas de programar en VBA, una es escribir código para un control ActiveX de tal manera que se ejecute cuando se produce una acción sobre el control, y otra es para crear nuevas funciones de Excel, que se ejecuten como las funciones preinstaladas.

Un control ActiveX es un objeto gráfico que se coloca en un formulario o directamente en una hoja del libro de Excel para mostrar o introducir datos o realizar una acción, y que será el sistema que utilizaremos.

Antes de empezar a incluir controles y trabajar con el código de VBA, deberemos de activar la ficha "Programador" en la cinta de "Opciones", para lo que:

1. Ejecutaremos Archivo>Opciones>Personalizar cinta de opciones.
2. En la lista **Fichas principales**, haremos clic en **Programador** y, a continuación, **Aceptar**.



Figura 2.1. Activar ficha Programador.

Una vez activada la ficha programador, esta tendrá el aspecto de la figura 2.2.



Figura 2.2. Aspecto de la ficha Programador.

Y haciendo clic accederemos al editor de VBA, que se encuentra dividido verticalmente en dos zonas, la de la izquierda correspondiente al explorador de proyectos, y la de la derecha del editor de código, según se aprecia en la figura 2.3.

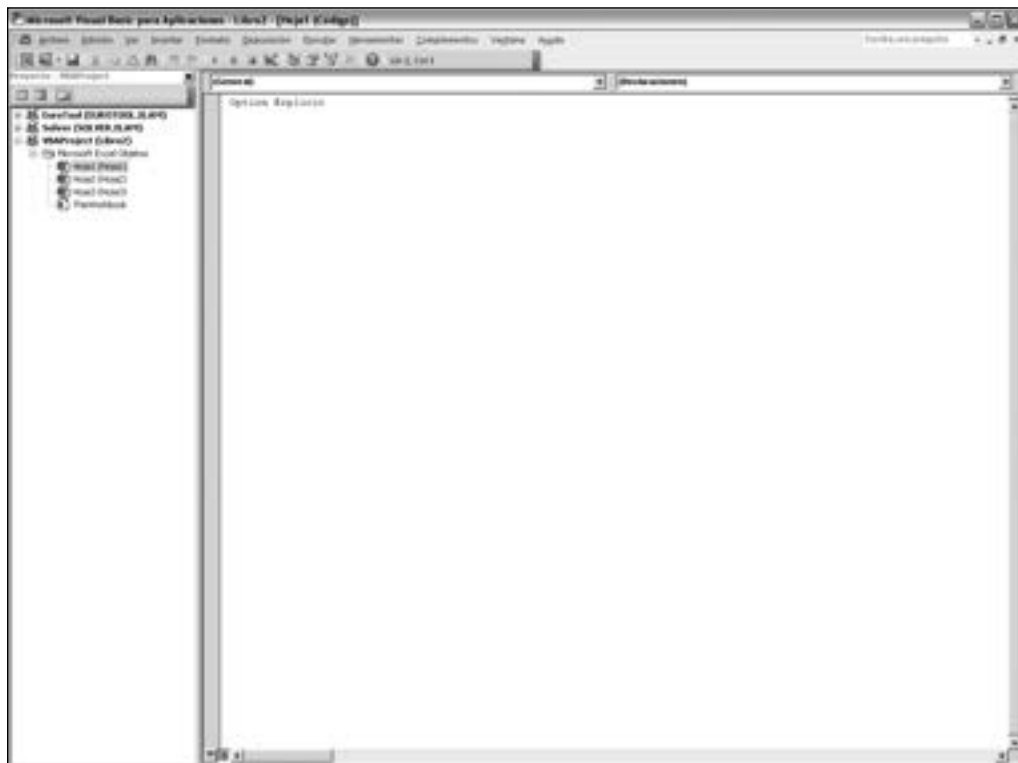


Figura 2.3. Editor de VBA para Excel.

2.2. Los Controles Active X

Un control ActiveX es un objeto gráfico que se utiliza para realizar una serie de acciones que definiremos en el código que escribiremos a tal efecto, pudiendo ser de diferente tipo en función del tipo de acción que deseamos ejecutar. Los controles más comunes son:

- **Botón de Comando.** Es el clásico botón de las aplicaciones Windows, y lo utilizaremos para ejecutar una acción asociada a un código y escrito a tal efecto, que se incluirá en su procedimiento:

```
Private Sub CommandButton1_Click()  
    'Este código se ejecuta al hacer clic sobre el CommandButton  
End Sub
```

- **Cuadro combinado.** Se utiliza para escribir nuevos valores o seleccionar uno de los existentes. Su procedimiento asociado será:

```
Private Sub ComboBox1_Change()  
    'Este código se ejecuta al cambiar el ComboBox  
End Sub
```

- **Casilla de verificación.** Se utiliza para seleccionar o no una determinada opción, y tiene dos posibles estados **Enabled** y **Locked**. Su procedimiento asociado será:

```
Private Sub CheckBox1_Click()  
    'Este código se ejecuta al hacer clic sobre el CheckBox  
End Sub
```

- **Cuadro de lista.** Se utiliza para acceder y/o cambiar los valores individuales del control. Su procedimiento asociado será:

```
Private Sub ListBox1_Click()  
    'Este código se ejecuta al hacer clic sobre el ListBox  
End Sub
```

- **Cuadro de texto.** Es utilizado para modificar un valor o mostrar información. Su procedimiento asociado será:

```
Private Sub TextBox1_Change()  
    'Este código se ejecuta al cambiar el TextBox  
End Sub
```

- **Barra de desplazamiento.** Devuelve el valor de otro control basándose en la posición del cuadro de desplazamiento. A diferencia de las barras de desplazamiento de Excel, es autónomo, y puede adoptar una posición horizontal o vertical. Su propiedad predeterminada **Value**. Su procedimiento asociado será:

```
Private Sub ScrollBar1_Change()  
    'Este código se ejecuta al cambiar el ScrollBar  
End Sub
```

- **Control de número.** Se utiliza para incrementar una variable o valor en una cantidad determinada establecida por su propiedad **Value**. Su procedimiento asociado será:

```
Private Sub SpinButton1_Change()  
    'Este código se ejecuta al cambiar el SpinButton  
End Sub
```

- **Botón de opción.** Se utiliza para seleccionar un único elemento de un grupo. Su procedimiento asociado será:

```
Private Sub OptionButton1_Click()  
    'Este código se ejecuta al hacer clic sobre el OptionButton  
End Sub
```

- **Etiqueta.** Se utiliza para presentar etiquetas de textos en las hojas y en los formularios. Su procedimiento asociado será:

```
Private Sub Label1_Click()  
    'Este código se ejecuta al hacer clic sobre el Label  
End Sub
```

- **Imagen.** Muestra una imagen en las hojas o en los formularios. Su procedimiento asociado será:

```
Private Sub Image1_Click()  
    'Este código se ejecuta al hacer clic sobre el Image  
End Sub
```

- **Botón de alternar.** Se utiliza para establecer una condición binaria del tipo si-no, verdadero-falso, activo-no activo. Su procedimiento asociado será:

```
Private Sub ToggleButton1_Click()  
    'Este código se ejecuta al hacer clic en el ToggleButton  
End Sub
```

Como se puede apreciar, al final del nombre de cada control, aparece un número, y una expresión. El número es un ordinal asignado automáticamente por VBA, y la expresión es el evento o acción ante la que es ejecutado. Cada control tiene un evento predeterminado, aunque este puede ser modificado situándonos en el módulo correspondiente y activando el menú contextual con el botón derecho del ratón seleccionar **Lista de propiedades y métodos**. Además del evento, cada control tiene unas propiedades asociadas, como el nombre, tamaño, posición, color o valor del control.

2.3. Procedimientos y módulos

2.3.1. Los procedimientos

Hasta ahora hemos visto un tipo de procedimiento especial llamado **Private Sub** asociado con los controles, pero hay otro tipo de procedimientos que son los **Sub** y **Function**.

Los procedimientos **Sub** son usados para crear parte un código que se ejecute de forma independiente y que sea llamado desde otro procedimiento. El siguiente código ejecuta el procedimiento **Saluda** llamado desde `CommandButton1_Click()`:

```
Private Sub CommandButton1_Click()  
    Saluda ("Buenos días")  
End Sub  
Sub Saluda(texto)  
    MsgBox (texto)  
End Sub
```

El procedimiento **Function** es muy similar al **Sub**, pero presenta como gran diferencia que puede ser utilizado en el lado derecho de una expresión, como la siguiente que calcula el valor de c como la suma de a y b:

```
Private Sub CommandButton1_Click()  
    a = 2  
    b = 3  
    c = Suma(a, b)  
    MsgBox ("Suma =" & c)  
End Sub  
Function Suma(a, b)  
    Suma = a + b  
End Function
```

2.3.2. Los módulos

Los diferentes procedimientos de un programa, se agrupan en módulos, y de esta forma se establecen dos niveles diferentes de código, un nivel inferior o de procedimiento y otro superior o de modulo. Cada una de las hojas de un libro de Excel es considerado como un modulo independiente, y sus procedimientos como de propios del modulo al que pertenecen salvo que se indique lo contrario. Los diferentes niveles de código son de una gran trascendencia y utilidad para controlar el ámbito de existencia de las variables.

2.4. Las variables y las operaciones

La programación con VBA es de gran ayuda para procesar de forma rápida y eficiente una gran cantidad y diversidad de datos. Estos datos son almacenados y organizados por medio de las variables, que pueden ser de diferente naturaleza en función del tipo de dato que almacenen (números, textos, fechas, etc.), y pueden ser modificados durante la ejecución del código del programa. Las variables han de tener un nombre único que la identifica dentro de su nivel de ámbito, debiendo comenzar con un carácter alfabético hasta un máximo de 255 caracteres sin puntos o caracteres de declaración de tipo.

2.4.1. Tipo de variables

Los tipos de variables más comunes en función de la naturaleza del dato que vayan a almacenar son:

- **Byte:** Números enteros positivos en el intervalo de 0 a 255 y ocupan 8 bits de memoria (1 byte).
- **Boolean:** Tipo de datos que tan sólo puede adoptar valor True (-1) o False (0). Estas variables ocupan 16 bits de memoria (2 bytes).
- **Integer:** Se usa para almacenar variables enteras y ocupa de 2 bytes de memoria en el intervalo de -32.768 a 32.767.
- **Long:** Tipo de variable para datos de números con signo y un valor comprendido entre - 2.147.483.648 y 2.147.483.647, ocupando 32 bits de memoria (4 bytes).
- **Single:** Tipo de variable para datos numéricos de precisión sencilla de punto flotante en el intervalo de valores que va desde -3,402823E38 a -1,401298E-45 para valores negativos y de 1.401298E-45 a 3.402823E38 para valores positivos. Ocupan 32 bits de memoria (4 bytes).
- **Double:** Variable para datos numéricos de doble precisión de punto flotante en el intervalo comprendido entre -1.79769313486231E308 hasta -4,94065645841247E-324 para valores negativos y que va desde 4,94065645841247E-324 a 1,79769313486232E308 para valores positivos. Ocupan 64 bits de memoria (8 bytes).
- **Variant:** Se utiliza para cualquier tipo de variable y es el tipo por defecto en caso de no ser declarada.
- **Date:** Se utiliza para almacenar fechas y horas como un número real. El valor de la izquierda de los decimales corresponde a la fecha y el valor de la derecha a la hora. Ocupan 64 bits de memoria (8 bytes).
- **String:** Éste se utiliza para almacenar cadenas de caracteres desde 1 a 65.400.

2.4.2. Declaración y ámbito de las variables

Aunque VBA no obliga a declarar explícitamente el tipo de variable con el que estamos trabajando (en cuyo caso adoptará el tipo Variant), es recomendable declararlas explícitamente y forzar a VBA a su declaración

previa mediante la inclusión de la siguiente instrucción al principio del módulo correspondiente:

```
Option Explicit
```

Puede no parecer trascendente el hecho de obligar a declarar o no las variables, pero sí que lo es.

Si cometemos un error tipográfico durante la creación del código al escribir una variable, VBA nos avisará mediante un mensaje de que esta no está declarada, pero en caso de no estar establecida la declaración obligatoria, se le asignará un valor de cero haciendo tremendamente difícil la detección del error.

El método más común de declaración es mediante la instrucción Dim, que tiene la siguiente sintaxis:

```
Dim nombre_variable([([subíndices]]) [As tipo]] ...
```

nombre_variable: Es un obligatorio, y sigue los criterios explicados para los nombres de las variables.

tipo: Es opcional, y puede ser cualquiera de los tipos explicados ya anteriormente.

El nivel del ámbito de las variables, depende de donde son declaradas. Así una variable declarada a nivel de módulo está disponible para todos los procedimientos disponibles dentro de ese modulo. Sin embargo, las variables declaradas a nivel de procedimiento solo estarán disponibles dentro de ese procedimiento.

```
Option Explicit
' Variable declarada a nivel de módulo
Dim variable_1 As Integer

Sub Procedimiento_1()
' Variable a nivel de procedimiento
Dim variable_a As Integer
End Sub

Sub Procedimiento_2()
' Variable a nivel de procedimiento
Dim variable_a As Integer
End Sub
```

En el código anterior se han declarado tres variables, una a nivel de módulo (variable_1), y otras dos a nivel de procedimiento (variable_a y variable_a), que aunque tienen el mismo nombre son diferentes, y con un ámbito exclusivo para su procedimiento correspondiente.

También podemos declarar las variables con las instrucciones **Public** y **Private**, que establecen como de amplio será su ámbito:

Public: Estarán disponibles para todos los procedimientos en todos los módulos de todas las aplicaciones, y su sintaxis es:

```
Public nombre_variable([([subíndices]))] [As tipo]] ...
```

Private: Las variables Private sólo están disponibles para el módulo en el que se declaran, pudiéndose declarar dentro de un procedimiento, y su sintaxis es:

```
Private nombre_variable([([subíndices]))] [As tipo]] ...
```

Const: Se utiliza para declarar las constantes que se pueden usar, pudiéndose además de tipo Public o Private.

```
Const Pi = 3.141616
```

2.4.3. Variables multidimensionales; las matrices

Muchas veces los datos tienen una cierta estructura dimensional como las notas de los estudiantes de un curso o las coordenadas de un plano. Este tipo de datos pueden ser almacenados y gestionados de manera muy eficiente mediante variables llamadas matrices y Arrays.

Las matrices se declaran exactamente igual que las variables, pero especificando las dimensiones de esta en los subíndices:

```
' Coordenadas de un plano de 20 por 20 elementos  
Dim Coordenadas(20, 20) As Integer
```

El primer índice de las matrices es cero, pero podemos ajustarlo a uno mediante la modificación a nivel de módulo del límite inferior predeterminado:

```
Option Base 1
```

También podemos cambiar el tamaño del último elemento de una matriz en el caso de que tengamos más datos de los inicialmente previstos, mediante el redimensionamiento a nivel de procedimiento del tamaño de la matriz. Cuando redimensionamos una matriz, los datos almacenados originalmente son borrados, por lo que a veces se hace necesario añadir la expresión **Preserve**, para que estos sean conservados, pero previamente se ha de declarar la matriz como dinámica sin poner valores en los subíndices:

```

Option Base 1
'Dimensionamos la matriz como dinámica
Dim Coordenadas() As Integer
Sub procedimiento()
    'Cambiamos el tamaño del plano a 20 por 20 elementos
    ReDim Preserve Coordenadas(20, 20)
    'Cambiamos el tamaño del plano a 20 por 30 elementos
    ReDim Coordenadas(20, 30)
End Sub

```

2.4.4. Tipos definidos por el usuario

Junto con las variables, VBA ofrece la posibilidad de organizar los datos en tipos definidos por el usuario, llamados estructura. Para hacer una declaración de tipo se utiliza la instrucción **Type ... End type**, con la siguiente sintaxis:

```

Type nombre_estructura
    nombre_elemento_1 [[[subíndices]]] As tipo
    [nombre_elemento_n [[[subíndices]]] As tipo]
...
End Type

```

La definición de una estructura de datos puede estar precedida por la instrucción **Public** o **Private**, según deseemos que sea su ámbito, pero sólo puede utilizarse en el nivel de módulo, pudiéndose declarar una variable de ese tipo en cualquier lugar dentro del alcance de la declaración.

Una estructura de datos puede usarse junto con las matrices, como por ejemplo para definir las tareas de un proyecto, conteniendo diferentes elementos según sea el tipo de dato que contenga:

```

Option Explicit
Type tarea_tipo
    codigo As String
    duración As Integer
    comienzo As Date
    final As Date
End Type
Dim tarea(100) As tarea_tipo

```

2.4.5. Operadores

Una vez que conocemos como almacenar y estructurar los datos, el siguiente paso más importante es operar con ellos, para lo cual disponemos de los operadores, símbolos que indican como manipular los y comparar los datos entre sí.

Los tipos de operadores son aritméticos, de comparación, de concatenación y lógicos.

- Los operadores aritméticos: Los operadores aritméticos son utilizados para ejecutar cálculos matemáticos, y son:

- La potencia (^): Se utiliza para elevar un número a la potencia del exponente.

```
resultado = número ^ exponente
```

- La multiplicación (*): Se utiliza para multiplicar dos números.

```
resultado = número1 * número2
```

- La división (/): Se utiliza para dividir dos números.

```
resultado = número1 / número2
```

- La suma (+): Se utiliza para sumar dos números.

```
resultado = número1 + número2
```

- La diferencia (-): Se utiliza para restar dos números.

```
resultado = número1 - número2
```

- El módulo (Mod): Divide dos números y devuelve sólo el resto.

```
resultado = número1 Mod número2
```

- La división entera (\): Divide dos números y devuelve su resultado como número entero.

```
resultado = número1 \ número2
```

- Los operadores de comparación: Los operadores de comparación son utilizados para realizar comparaciones entre los datos, y puede ofrecer como resultado **True**, **False** o **Null**:

- Igual (=).

```
expresión1 = expresión 2
```

- Menor que (<).

```
expresión 1 < expresión 2
```

- Mayor que (>).

```
expresión 1 > expresión 2
```

- Menor o igual que ($\leq$).

`expresión 1  $\leq$  expresión 2`

- Mayor o igual que ($\geq$).

`expresión 1  $\geq$  expresión 2`

- Distinto que ($\neq$).

`expresión 1  $\neq$  expresión 2`

- Los operadores de concatenación: Los operadores de concatenación se utilizan para concatenar cadenas de caracteres, y son:

- (&): Se utiliza para concatenar cadenas de caracteres alfabéticos.

`resultado = cadena1 & cadena2`

- (+): Se utiliza para concatenar cadenas de caracteres numéricos.

`resultado = cadena1 + cadena2`

- Los operadores lógicos: Los operadores lógicos se utilizan para realizar operaciones lógicas booleanas, y puede ofrecer como resultado **True**, **False** o **Null**:

- And: Se utiliza para efectuar la conjunción lógica de dos expresiones. Devuelve como resultado True solo si las dos expresiones son verdaderas, False si alguna de ellas es falsa y Null si alguna de ellas es Null.

`resultado = expresión1 And expresión2`

- Or: Se utiliza para efectuar la disyunción lógica de dos expresiones. Devuelve como resultado True si alguna de las dos expresiones es verdadero, False si las dos son falso y Null si alguna de ellas es Null.

`resultado = expresión1 Or expresión2`

- Not: Se utiliza para efectuar la negación lógica de una expresión. Devuelve como resultado True si la expresión es falsa, False si es verdadera y Null si es Null.

`resultado = Not expresión1`

- Xor: Se utiliza para efectuar la exclusión lógica entre dos expresiones. Devuelve como resultado True si una de las expresiones es

verdadera y la otra falsa, y False si las dos expresiones son verdaderas o falsas.

```
resultado = expresión1 Xor expresión2
```

- **Eqv:** Se utiliza para efectuar la equivalencia lógica entre dos expresiones.

```
resultado = expresión1 Eqv expresión2
```

- **Imp:** Se utiliza para efectuar una implicación lógica de dos expresiones.

```
resultado = expresión1 Imp expresión2
```

- **El orden de ejecución de las operaciones:** Para evitar ambigüedades cuando hay varios operadores en la misma expresión, VBA establece un orden de ejecución predeterminado según su prioridad de izquierda a derecha, resolviendo primero las expresiones con operadores aritméticos, a continuación las que tienen operadores de comparación y por último las de operadores lógicos. El orden de prioridad es:

Aritméticos	Comparación	Lógicos
Potencia (^)	Igualdad (=)	Not
Negación (-)	Desigualdad (<>)	And
Multiplicación y división (* /)	Menor que (<)	Or
División entera (\)	Mayor que (>)	Xor
Módulo (Mod)	Menor o igual que (<=)	Eqv
Suma y resta (+ -)	Mayor o igual que (>=)	Imp

Una aplicación muy útil, es que el valor de la operación aritmética, es asignado al miembro de la izquierda de la expresión después de ser realizada la expresión del miembro de la derecha, pudiendo realizar operaciones acumulativas sucesivas. En el siguiente fragmento de código, la variable contador empieza teniendo un valor de 1, pero al finalizar tiene un valor de 2, después de sumarle uno al valor inicial:

```
contador = 1
'contador empieza con valor 1
contador = contador + 1
'contador termina con valor 2
```

2.5. Estructuras de decisión y de control

Las estructuras de decisión controlan la manera en que fluye el código basándose en si se cumplen o no ciertas condiciones lógicas, es decir que controlaremos que zonas del código se ejecutan o no en función de la evaluación de las expresiones lógicas.

2.5.1. If ... Then ... Else

La estructura **If** ejecuta un bloque de código solo si la expresión lógica especificada se evalúa como verdadera. Opcionalmente podemos especificar un bloque de código en caso de que la expresión se evalúe como falsa.

```
If condición Then
    [instrucciones]
[Else
    instrucciones_else]
End If
```

2.5.2. Select Case ... End Select

La estructura **Select ... Case**, es más apropiada que **If** cuando se tienen que evaluar varias condiciones.

```
Select Case expresión
    Case condición-1
        instrucciones-1
    [Case condición -n
        instrucciones-n]
    [Case Else
        [instrucciones_else]
End Select
```

2.5.3. On...GoSub, On...GoTo

Salta a una de las líneas especificadas, dependiendo del valor de una expresión. La diferencia entre ellas es que una vez ejecutado el código del salto, mediante **GoSub** se reanuda en su punto de partida, mientras que con **GoTo** no.

```
Sub procedimiento1()
    On a GoSub linea1, linea2
    'Aquí continua después del salto
```

```
linea1:
    'Código que se ejecuta si a=1
    'vuelve atrás
linea2:
    'Código que se ejecuta si a=2
    'vuelve atrás
End Sub

Sub procedimiento2()
    On a GoTo linea1, linea2
linea1:
    'Código que se ejecuta si a=1
    'continua
linea2:
    'Código que se ejecuta si a=2
    'continua
End Sub
```

2.5.4. GoTo

Salta a la línea especificada.

```
GoTo linea1
...

linea1:
...
```

2.5.5. GoSub ... Return

Salta a una subrutina del código para volver a continuación.

```
GoSub linea1
    'Aquí continua después de ejecutar la subrutina
...

linea1:
...
Return
```

2.6. Estructuras de bucle

Las estructuras de bucle, al contrario de las estructuras de decisión, establecen la parte de código que se ejecutará varias veces, bien con una cantidad fija de repeticiones o hasta encontrar una condición de finalización determinada.

2.6.1. For ... To ... Next

Este es el caso más frecuente y sencillo de estructura de bucle, obligando a ejecutar una parte del código una determinada cantidad de veces, con la siguiente sintaxis:

```
For contador = principio To fin [Step incremento]
    [instrucciones]
[Exit For]
[instrucciones]
Next [contador]
```

El siguiente código ejecuta un bucle **For ... Next** para calcular el factorial de un número dado:

```
Sub Factorial()
    Dim i As Single
    Dim número As Single
    Dim resultado As Double
    'Calcula el factorial de un número
    número = 5: resultado = 1
    For i = 1 To número Step 1
        resultado = resultado * i
    Next
End Sub
```

2.6.2. While ... Wend

La estructura de bucle **While ... Wend**, ejecuta una porción de código mientras la condición especificada sea verdadera, con esta sintaxis:

```
While condición
    [instrucciones]
Wend
```

El siguiente código ejecuta un bucle **While ... Wend** para calcular el factorial de un número:

```
Sub Factorial()
    Dim i As Single
    Dim número As Single
    Dim resultado As Double
    'Calcula el factorial de un número
    número = 5: resultado = 1
    While i < número
        i = i + 1
        resultado = resultado * i
    Wend
End Sub
```

2.6.3. Do ... Loop

Repite un bloque de instrucciones de código cuando una condición especificada es verdadera o hasta que se convierta en verdadera, con la siguiente sintaxis:

```
Do [{While | Until} condición]
    [instrucciones]
[Exit Do]
[instrucciones]
Loop [{While | Until} condición]
```

El siguiente código ejecuta un bucle **Do ... Loop** para calcular el factorial de un número:

```
Sub Factorial()
    Dim i As Single
    Dim número As Single
    Dim resultado As Double
    'Calcula el factorial de un número
    número = 5: resultado = 1
    Do
        i = i + 1
        resultado = resultado * i
    Loop Until i = número
End Sub
```

2.7. Acceso y manipulación de hojas y celdas de Excel

Ahora que conocemos lo más fundamental del lenguaje Visual Basic, ha llegado el momento de dar un paso más y conocer la verdadera naturaleza de VBA como lenguaje orientado a objetos y aprender a programar en Excel 2010.

2.7.1. Objetos, métodos, propiedades y eventos

VBA para Excel es un lenguaje de programación orientado a objetos, y realmente lo que hace es eso, manipular objetos. Los objetos que maneja pueden ser formularios, controles y lo más importante para nosotros, las hojas de nuestro libro de Excel y las celdas que contiene, pudiéndose agrupar en colecciones de objetos que normalmente son del mismo tipo.

Todos los objetos tienen unas propiedades, unos métodos y unos eventos, pero veamos que es cada una de las cosas:

Las propiedades de un objeto determinan su apariencia visual, su tamaño, color, posición en un formulario o su valor, y se accede a ellas de la siguiente manera.

```
objeto.propiedad = valor
valor = objeto.propiedad
```

Los métodos son programas propios del objeto que hacen algo, como mover o imprimir el objeto.

```
objeto.método
```

Los eventos una acción que se realiza sobre un objeto, como hacer clic con el ratón sobre un botón de comando, y para la que es posible escribir código como respuesta en un procedimiento.

```
objeto.evento
```

2.7.2. La colección Worksheets

Worksheets es la colección de objetos Worksheet, quizás los más importantes de VBA para Excel, y que contiene todas las hojas de cálculo de nuestro libro. La forma de trabajar es con Worksheets, es la general para cualquier objeto, con las siguientes sintaxis:

```
Worksheets[(indice)].propiedad = valor
Worksheets[(indice)].método
Worksheets[(indice)].evento
```

Pero veamos de una manera práctica como trabajar con los eventos, los métodos y las propiedades de la colección de objetos *Worksheets*.

El siguiente código se ejecuta ante un evento *Activate*, que es la activación o entrada en la "Hoja2" de nuestro libro, para posteriormente ejecutar el método *PrintPreview* (vista previa) y cambiar la propiedad *name* (nombre) de la primera hoja:

```
'El objeto actúa ante un evento
Private Sub Worksheet_Activate()
    'Ejecuta un código
    MsgBox ("has entrado en la hoja 2")
    'Ejecuta un método
    Worksheets.PrintPreview
    'Cambia una propiedad
    Worksheets(1).Name = "Esta es la hoja 1"
End Sub
```


2.7.3. La propiedad Cells

La propiedad *Cells* del objeto *Worksheets*, es la que más vamos a utilizar cuando trabajemos con VBA para Excel, porque gracias a ella podremos acceder y manipular los valores y propiedades de las celdas de las hojas de nuestro libro.

La sintaxis será:

```
Worksheets [índice].Cells(fila, columna).propiedad = valor  
valor = Worksheets [índice].Cells(fila, columna).propiedad
```

En el siguiente código cambiamos el valor de la celda de la fila 2, columna 5, y le asignamos un valor de 3, en la hoja que ocupa la posición 1, por medio de la propiedad *Value*:

```
Worksheets(1).Cells(2, 5).Value = 3
```

Es común que las hojas cambien de posición y consecuentemente de *índice*, por lo que para evitar problemas con el desplazamiento de las hojas, y que su identificación sea esté exenta de errores, el *índice* podemos sustituirlo por el nombre de la hoja entre comillas:

```
Worksheets ["nombre de la hoja"].Cells(fila, columna).propiedad = valor  
valor = Worksheets ["nombre de la hoja"].Cells(fila, columna).propiedad  
Worksheets("Hoja 1").Cells(2, 5).Value = 3
```

2.7.4. El objeto Range

El objeto *Range*, representa una celda, una fila, una columna o una selección de celdas que contienen uno o más bloques contiguos de celdas.

```
Worksheets(índice).Range(rango de celdas).propiedad = valor  
valor = Worksheets(índice).Range(rango de celdas).propiedad
```

Al rango de celdas podemos referirnos por medio de su columna y fila tal y como son expresados alfanuméricamente y por defecto, es decir "A5" o "B5:C7":

```
Worksheets(1).Range("A5").Value = 3  
Worksheets("Hoja1").Range("B5:C7").Value = 3
```

O por medio del nombre que le hemos asignado a la celda o bien rango de celdas:

```
Worksheets(1).Range("Inicio").ClearContents  
Worksheets("Hoja1").Range("Inicio").ClearContents
```

Los métodos existentes más comunes para el objeto Range son:

Nombre	Descripción
Activate	Activa una sola celda.
AddComment	Añade un comentario.
ApplyNames	Aplica nombres a las celdas del rango especificado.
ApplyOutlineStyles	Aplica estilos de esquema al rango especificado.
BorderAround	Agrega un borde y establece las propiedades.
ClearContents	Borra las fórmulas del rango.
ClearFormats	Borra el formato del objeto.
ClearOutline	Borra el esquema del rango especificado.
CreateNames	Crea nombres en el rango especificado.
Delete	Elimina el rango.
FillDown	Rellena hacia abajo.
FillLeft	Rellena hacia la izquierda.
FillRight	Rellena hacia la derecha.
FillUp	Rellena hacia arriba.
Insert	Inserta una celda o un rango de celdas.
PrintOut	Imprime el rango de celdas.
PrintPreview	Muestra una vista preliminar del objeto tal como aparecería impreso.
Run	Ejecuta la macro que se encuentra en este rango.
Select	Selecciona el rango.
Sort	Ordena los valores de un rango.
Subtotal	Crea subtotales para el rango.

Y algunas de las propiedades más comunes para el objeto Range son:

Nombre	Descripción
Column	Devuelve el número de la primera columna del rango.
Count	Devuelve el número de objetos del rango.
End	Devuelve la celda situada al final del rango.

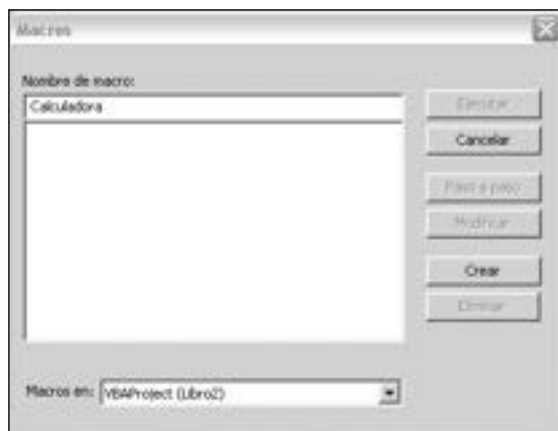


Figura 2.8. Iniciar Módulo para Macro.

Ahora volveremos a nuestro libro de Excel, y prepararemos la Macro para que sea ejecutada de forma automática, mediante **Programador>Macros>Opciones**, y escribiremos Q como Tecla de método abreviado. Ahora al pulsar simultáneamente las teclas **Control-Q**, aparecerá el formulario creado (véase la figura 2.9).

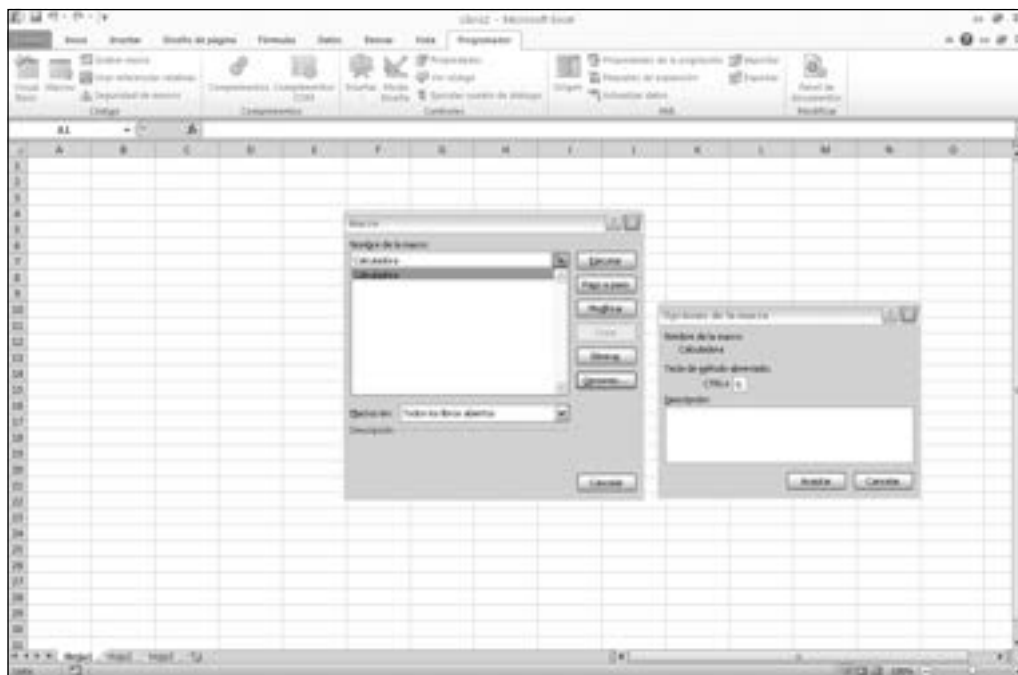


Figura 2.9. Método abreviado para Macros.

igual, cambiándoles los nombre a Tecla_Igual, Tecla_signo, Tecla_menos, Tecla_div, Tecla_mas y Tecla_por, porque así nos resultará muchísimo más fácil escribir el código correspondiente para cada una de ellas.

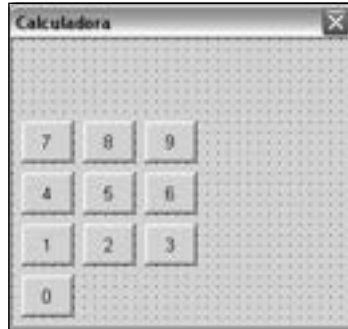


Figura 2.11. Teclas numéricas de la calculadora.

Nos quedará por añadir la pantalla en donde aparecerán los caracteres que corresponden a cada una de las teclas que vamos pulsando, y que crearemos mediante un control Cuadro de texto, al que le cambiaremos el nombre a Display, y la propiedad *Font* a un tipo de letra que nos guste, que en este caso ha sido *Small Fonts* de tamaño 24, quedando la calculadora ya terminada de diseñar con el siguiente aspecto de la figura 2.12.



Figura 2.12. Diseño final de la calculadora.

Ya estamos en condiciones de empezar a escribir el código de nuestra calculadora, y lo primero que haremos será declarar las variables a nivel de modulo:

```
Dim Entrada As String
Dim Numero As String
Dim operación As String * 1
Dim Formato_decimal As String
```

Seguro que os ha llamado la atención el hecho de solamente se han declarado variables de tipo *String*, pero no os preocupéis porque los cálculos serán correctos para lo que haremos las oportunas conversiones de cadena de texto a valor numérico.

La variable *entrada* corresponde al número que estamos escribiendo es cada momento, la *Numero* es una variable que utilizaremos para almacenar el valor escrito cuando pulsemos una tecla de operación, la variable *operación* es un *String* de tamaño fijo 1 correspondiente al tipo de operación aritmética que vamos a aplicar, y en *Formato_decimal* estableceremos el formato de separación de millares y decimales que corresponda en cada caso para el cuadro de texto.

Para establecer el formato de separación de millares, haremos que ante el evento *Show* (mostrar) del formulario, se inicialice el valor de *Formato_decimal*, a un formato de separación de miles sin decimales, escribiendo el siguiente procedimiento:

```
Private Sub UserForm_Activate()  
    Formato_decimal = "##,##"  
End Sub
```

Para que el código que más elegante, añadiremos un procedimiento que contiene un código que se repetiría a lo largo de todo el programa, y donde haremos que aparezca el valor que tiene la variable *entrada* en el cuadro de texto, pero con el formato que queremos:

```
Sub Actualiza_display()  
    Display.Text = Format(Entrada, Formato_decimal)  
End Sub
```

Antes de empezar con las teclas numéricas, es conveniente ver el procedimiento correspondiente al punto decimal. Lo primero que hace es analizar si existe ya un punto decimal, y en caso que así sea, emite un pitido de advertencia y sale del procediendo. En caso que no exista punto decimal, tendremos que cambiar el formato con que son presentados los valores en la pantalla, añadiendo a la variable *Formato_decimal* los caracteres "0." Que hace que se aparezca el punto decimal pero sin valores decimales. El código del procedimiento será:

```
Private Sub Punto_decimal_Click()  
    If InStr(Entrada, ".") <> 0 Then Beep: Exit Sub  
    If InStr(Formato_decimal, ".") = 0 Then  
        Formato_decimal = Formato_decimal & "0."  
    End If  
    Entrada = Entrada & "."  
    Actualiza_display  
End Sub
```

Para cada una de las teclas numéricas, haremos que al valor de la variable *entrada* le sea añadido el carácter correspondiente a la tecla que es pulsada.

Tendremos también en cuenta actualizar la variable *Formato_decimal*, añadiendo un carácter "0" en el caso de que exista un punto decimal en la variable *entrada*, haciendo que el código actualice la pantalla llamando al procedimiento *Actualiza_display*.

```
Private Sub Numero0_Click()  
    Entrada = Entrada & "0"  
    If InStr(Entrada, ".") <> 0 Then Formato_decimal = Formato_decimal & "0"  
    Actualiza_display  
End Sub  
Private Sub Numero1_Click()  
    Entrada = Entrada & "1"  
    If InStr(Entrada, ".") <> 0 Then Formato_decimal = Formato_decimal & "0"  
    Actualiza_display  
End Sub  
Private Sub Numero2_Click()  
    Entrada = Entrada & "2"  
    If InStr(Entrada, ".") <> 0 Then Formato_decimal = Formato_decimal & "0"  
    Actualiza_display  
End Sub  
Private Sub Numero3_Click()  
    Entrada = Entrada & "3"  
    If InStr(Entrada, ".") <> 0 Then Formato_decimal = Formato_decimal & "0"  
    Actualiza_display  
End Sub  
Private Sub Numero4_Click()  
    Entrada = Entrada & "4"  
    If InStr(Entrada, ".") <> 0 Then Formato_decimal = Formato_decimal & "0"  
    Actualiza_display  
End Sub  
Private Sub Numero5_Click()  
    Entrada = Entrada & "5"  
    If InStr(Entrada, ".") <> 0 Then Formato_decimal = Formato_decimal & "0"  
    Actualiza_display  
End Sub  
Private Sub Numero6_Click()  
    Entrada = Entrada & "6"  
    If InStr(Entrada, ".") <> 0 Then Formato_decimal = Formato_decimal & "0"  
    Actualiza_display  
End Sub  
Private Sub Numero7_Click()  
    Entrada = Entrada & "7"  
    If InStr(Entrada, ".") <> 0 Then Formato_decimal = Formato_decimal & "0"  
    Actualiza_display  
End Sub  
Private Sub Numero8_Click()  
    Entrada = Entrada & "8"  
    If InStr(Entrada, ".") <> 0 Then Formato_decimal = Formato_decimal & "0"  
    Actualiza_display  
End Sub
```

```
Private Sub Numero9_Click()
    Entrada = Entrada & "9"
    If InStr(Entrada, ".") <> 0 Then Formato_decimal = Formato_decimal & "0"
    Actualiza_display
End Sub
```

Para las teclas de los operadores aritméticos, lo primero que haremos será establecer el tipo de operador que estamos ejecutando asignado el valor correspondiente a la variable *operador*, haciendo que el valor de la variable auxiliar *Numero* sea el mismo que el del valor que tenemos actualmente en la variable *entrada*, inicializando esta como valor vacío haciéndola igual a unas comillas vacías y actualizando la pantalla.

```
Private Sub Tecla_mas_Click()
    operación = "+"
    Numero = Entrada: Entrada = ""
    Actualiza_display
End Sub
Private Sub Tecla_menos_Click()
    operación = "-"
    Numero = Entrada: Entrada = ""
    Actualiza_display
End Sub
Private Sub Tecla_por_Click()
    operación = "*"
    Numero = Entrada: Entrada = ""
    Actualiza_display
End Sub
Private Sub Tecla_div_Click()
    operación = "/"
    Numero = Entrada: Entrada = ""
    Actualiza_display
End Sub
```

Para cambiar el signo del número que estamos escribiendo utilizaremos una instrucción **Select ... Case** que analizará si el primer carácter de la cadena *entrada* es un signo menos o no.

En caso de que sea un signo menos lo que tendremos que hacer es quitarlo, actualizando por medio de una instrucción **MID** el valor de la variable *entrada*, y en caso contrario añadir este carácter al principio de la cadena.

```
Private Sub Tecla_signo_Click()
    Select Case Mid(Entrada, 1, 1)
        Case "-"
            Entrada = Mid(Entrada, 2, Len(Entrada))
        Case Else
            Entrada = "-" & Entrada
    End Select
    Actualiza_display
End Sub
```

Un elemento importante de nuestra calculadora es que tenga la posibilidad de borrar o inicializar las operaciones, para lo que tendremos que añadiré una tecla y un procedimiento que inicialice todas las variables y formatos a su estado original:

```
Private Sub Borrar_Click()
    Entrada = ""
    Numero = ""
    Formato_decimal = "##,##"
    Actualiza_display
End Sub
```

Para terminar ya solo tendremos que operar el valor que tenemos en la variable auxiliar *Numero* con el valor actual de *entrada*, en función del operador que estemos efectuando, utilizando para ello una instrucción **Select ... Case**.

Las operaciones son realizadas convirtiendo las cadenas de texto en valores numéricos mediante **VAL()**, para volverlos a transformar en cadena de texto con **Str()**.

```
Private Sub Tecla_Igual_Click()
    Select Case operación
        Case "+"
            Entrada = Str(Val(Numero) + Val(Entrada))
        Case "-"
            Entrada = Str(Val(Numero) - Val(Entrada))
        Case "*"
            Entrada = Str(Val(Numero) * Val(Entrada))
        Case "/"
            Entrada = Str(Val(Numero) / Val(Entrada))
    End Select
    Actualiza_display
    Formato_decimal = "##,##"
End Sub
```

Ya tenemos terminada nuestra calculadora, pero seguro que muchos de vosotros dudareis de la utilidad de una calculadora tan simple como macro de Excel cuando está pensada precisamente para realizar operaciones de forma muy rápida y eficiente.

Sin embargo, si añadimos una nueva tecla que haga que el valor de la pantalla sea volcado directamente en la celda activa que tenemos en nuestra hoja ya parece que sea más práctico, para lo que añadiremos un nuevo botón que llamaremos Pegar, y le escribiremos el siguiente código, que copiara el valor numérico de la variable *entrada* a la celda activa:

```
Private Sub Tecla_pegar_Click()
    ActiveCell.Value = Val(Entrada)
End Sub
```

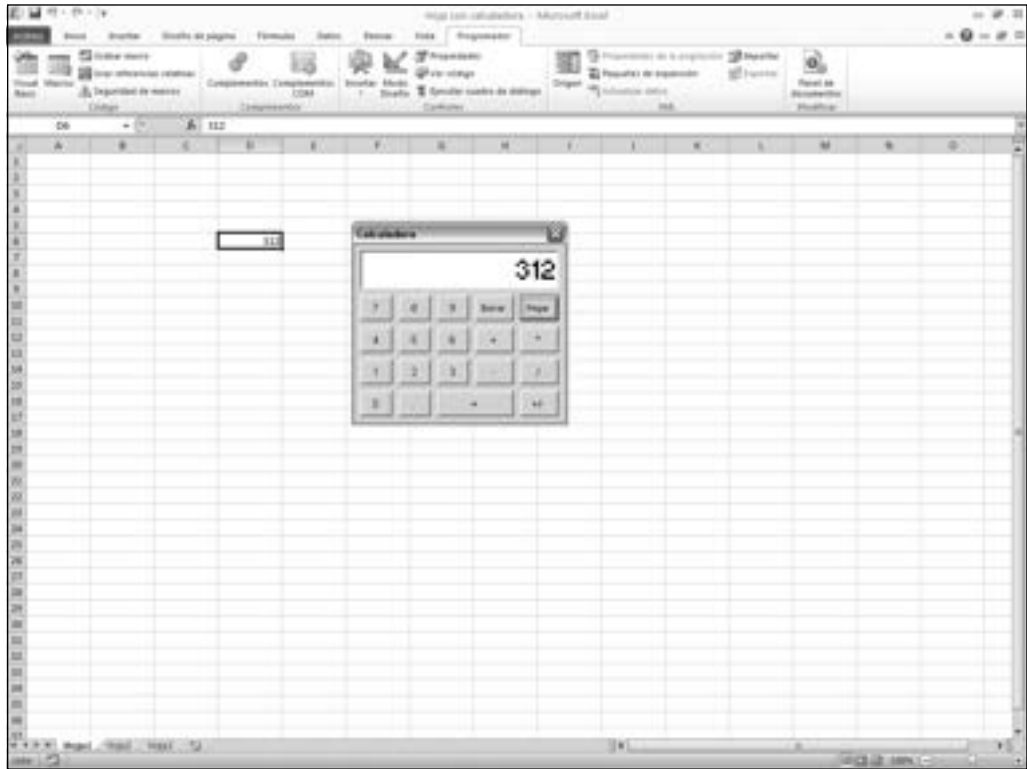


Figura 2.13. Macro calculadora ejecutándose en Excel.

NOTA

La hoja de cálculo completa se encuentra en Cap_02\Hoja con Macro de calculadora, de los ejemplos del libro.

NOTA

Si queremos que esta Macro este presente cuando iniciemos un nuevo libro en blanco, tendremos que guardar la hoja como Plantilla de Excel habilitada para macros en Archivos>Guardar como dentro de la ruta C:\Usuarios\<nombre de usuario>\AppData\Roaming\Microsoft\Plantillas, e iniciarlos como Archivo>Nuevo>Mis plantillas.

3

Introducción a los Grafos de Proyectos

3.1. Una brevísima teoría de grafos

La Gestión de Proyectos constituye un conjunto de herramientas para la solución de los diferentes problemas a los que nos encontramos cuando afrontamos la secuenciación de las tareas de un proyecto.

La más importante herramienta de trabajo utilizada es la teoría de grafos tanto para su representación grafica como para su posterior análisis y toma de decisiones. La teoría de grafos forma parte de una especialidad de las matemáticas llamada "Investigación Operativa", y concretamente estudia la representación o modelización de la realidad mediante una serie de objetos llamados vértices, nodos o sucesos, interconectados entre sí por líneas llamadas aristas, flechas o arcos y que pueden tener o no orientación o dirección asignada, así como capacidades.

Se dice que la teoría de grafos nació de la mano de Leonhard Euler en el año 1736 al plantear el "Problema de los puentes de Königsberg" (anteriormente perteneciente a la Prusia oriental y que actualmente se llama Kaliningrado) y demostrar que no era posible realizar una ruta partiendo

- **Grafos Dirigidos:** aquellos grafos en los que sus aristas tienen cierta dirección.
- **Bucle:** Arista que saliendo de un nodo o vértice regresa al mismo nodo o vértice.
- **Camino de un grafo:** sucesión finita no vacía de vértices y aristas, donde cada arista tiene por extremos los vértices adyacentes y el número de aristas será la longitud del camino.
- **Circuito Simple o Ciclo:** Es un camino del grafo donde el vértice inicial coincide con el vértice final.

A partir de las anteriores definiciones, diremos que los grafos de Proyectos son los siguientes:

1. Grafos dirigidos. Las tareas de un proyecto se organizan según una determinada secuencia de ejecución, para lo cual han de tener asignada una dirección que establezca el orden de esta secuencia.
2. Grafos sin bucles ni circuitos. Todas las tareas de un proyecto han de seguir una cierta secuencia continua con un principio y un final, no pudiéndose quedar eternamente realizando una tarea concreta (bucle) o la misma secuencia (circuito).

3.2. Tipos de Grafos de Proyecto

El primer sistema que surgió para representar tanto la secuencia de ejecución de las tareas de un proyectos como sus comienzos y finales es el denominado diagrama de Gantt. Este tipo de representación conocida a partir de 1910, adopta de nombre del ingeniero norteamericano Henry Gantt que fue quien la popularizó, aunque su autoría se la debemos a Karol Adamiecki en 1896, que la denominó Harmonigraph. El diagrama de Gantt o Harmonigraph, aunque representaba perfectamente las relaciones entre las tareas y los tiempos de comienzo y finalización de las tareas, no podemos considerarlo estrictamente como un grafo de proyecto, dejándolo en un segundo nivel pero no menos importante de diagrama.

Los grafos, tal y como se ha comentado anteriormente, son muchos más complejos y están formados por dos tipos de elementos diferentes, y que son los vértices o nodos y las aristas o flechas. Ya podrás imaginarte que en función del elemento que utilicemos para representar las tareas tendremos grafos de actividades en flecha (AoA de su abreviatura en Inglés)

si las tareas las representamos mediante las aristas del grafo y grafos de actividades en nodo (AoN) en el caso de que estas sean representadas en los nodos del grafo.

3.2.1. Grafos de actividades en flecha

Este es el primer sistema inventado para la representación de las secuencias de tareas de un proyecto mediante grafos, teniéndonos que remontar a finales de los años 50 del siglo pasado con el PERT y el desarrollo del proyecto Polaris por parte de la marina de los EEUU.

Prácticamente de manera simultánea, pero en Europa, se desarrolló el modelo CPM (*Critical Path method*) por la Du Pont Company y Remington Rand Univac.

Ambos modelos eran muy similares en su concepción pero muy diferentes en los objetivos perseguidos, mientras que el primero de ellos utilizaba discutibles herramientas estadísticas buscando minimizar el riesgo en el cumplimiento del plazo, el segundo de ellos buscaba minimizar el coste total del proyecto.

Este sistema de actividades en flecha tan solo permite relaciones del tipo de final-comienzo, así, dos tareas i y j , en la que j dependa de la finalización de i para poder comenzar, se representarían de la siguiente forma:



Figura 3.2. Grafo AoA de dos tareas.

NOTA

A partir de ahora y con carácter general cada vez que nos refiramos a una tarea i , se considerará que es o son las tareas predecesoras de j , e igualmente cada vez que nos refiramos a una tarea j se considerará que es o son las tareas sucesoras de i .

En ciertas condiciones, se hace necesario recurrir al uso de las tareas denominadas ficticias, cuyo único cometido es representar fiel y estrictamente las relaciones de dependencia de un proyecto, como en el siguiente caso formado por cuatro tareas i_1 , i_2 , j_1 y j_2 , donde j_1 depende de i_1 y j_2 depende simultáneamente de i_1 y i_2 :

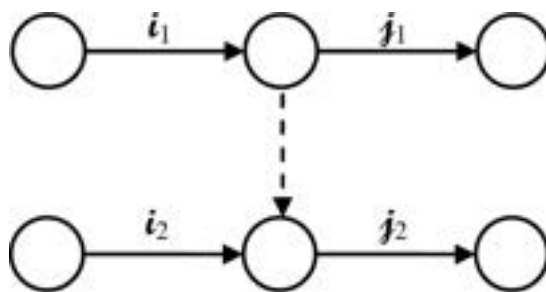


Figura 3.3. Grafo AoA con tareas ficticias.

En algunos casos, incluso se hace necesario el uso de nodos ficticios, convirtiéndose en una herramienta verdaderamente engorrosa para proyecto de una mínima complejidad. Este hecho ha provocado que este tipo de representación este actualmente totalmente en desuso.

Para ilustrar esta última afirmación, vamos a exponer el proyecto que nos acompañará a lo largo de este libro para ilustrar los diferentes casos que planteemos:

El proyecto consiste en la ejecución de las excavaciones, las cimentaciones y las estructuras de cinco viviendas unifamiliares, de tal manera que al disponer de tan solo un equipo para ejecutar las excavaciones, otro para las cimentaciones y otro para las estructuras:

Tarea	Código	depende de...
Excavación 1	Ex 1	ninguna
Excavación 2	Ex 2	Ex 1
Excavación 3	Ex 3	Ex 2
Excavación 4	Ex 4	Ex 3
Excavación 5	Ex 5	Ex 4
Cimentación 1	Ci 1	Ex 1
Cimentación 2	Ci 2	Ex 2, Ci 1
Cimentación 3	Ci 3	Ex 3, Ci 2
Cimentación 4	Ci 4	Ex 4, Ci 3
Cimentación 5	Ci 5	Ex 5, Ci 4
Estructura 1	St 1	Ci 1
Estructura 2	St 2	Ci 2, St 1

Tarea	Código	depende de...
Estructura 3	St 3	Ci 3, St 2
Estructura 4	St 4	Ci 4, St 3
Estructura 5	St 5	Ci 5, St 4

Resultando el siguiente grafo:

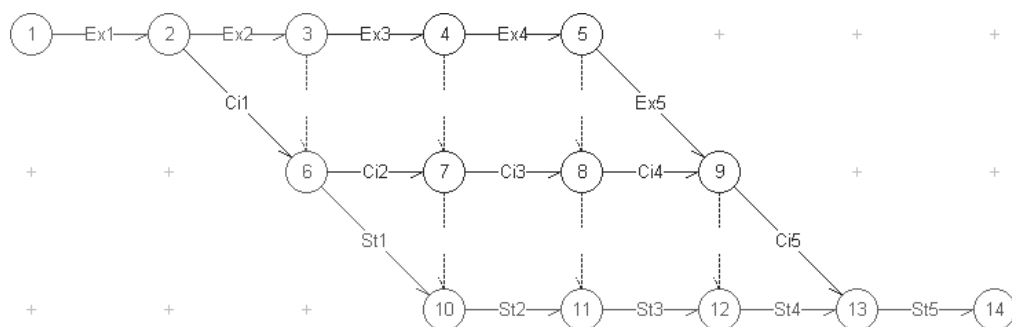


Figura 3.4. Grafo de Proyecto AoA incorrecto.

Se ha requerido de la utilización de 6 actividades ficticias para la construcción del grafo, pero si lo analizamos con detalle vemos que no se cumplen estrictamente las condiciones impuestas, sino que además estamos haciendo depender la Estructura 1 de la Excavación 2, la Estructura 2 de la Excavación 3, la Estructura 3 de la Excavación 4 y la Estructura 4 de la Excavación 5, algo que a todas luces es incorrecto y que me proporciona como en este caso resultados incorrectos. Para solucionar este problema y cumplir estrictamente las relaciones impuestas necesitaremos, hemos de añadir un nuevo nodo además de otra actividad ficticia adicional:

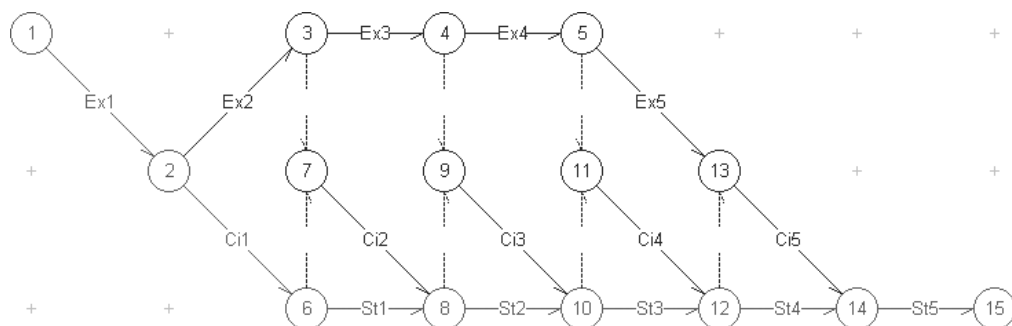


Figura 3.5. Grafo de Proyecto AoA correcto.

3.2.2. Grafos de actividades en nodo

En este sistema de representación, las actividades están representadas por los nodos del grafo, y las dependencias entre ellas por las aristas.

Este sistema nació en Europa a principios de los años sesenta de la mano del ingeniero francés B. Roy que lo llamó "Método de los Potenciales".

Paralelamente, pero en los Estados Unidos se desarrolló el "Sistema de Actividades en los Nodos" por John Fondahl, considerado como el precursor del moderno "*Precedence Diagramming Method*" (PDM).

Los primeros modelos con múltiples tipos de relaciones de precedencias aparecieron en 1964 con el manual de uso del "*Project Control System*" del IBM 1440, con J. Craig como investigador principal, aunque la autoría no está muy clara pues IBM cita ya en 1963 a HB Zahry Company de San Antonio como creador del formato de Precedencias.

Este sistema presenta muchas ventajas sobre el anterior, destacando de entre todas el hecho de que no necesitamos del uso de actividades ficticias, además de poder representar una mayor diversidad de relaciones de simultaneidad entre ellas, no exclusivamente las de final-comienzo ya comentadas anteriormente, siendo actualmente el único sistema utilizado tanto por los expertos en gestión de proyectos como por los matemáticos e ingenieros que desarrollan nuevos modelos de optimización.

Mediante el sistema de actividades en nodo, el proyecto de cinco viviendas lo representaríamos de la siguiente forma:



Figura 3.6. Grafo de Proyecto AON.

Tal y como se puede apreciar la facilidad de lectura de este tipo de representación es muchísimo mayor que con el sistema de actividades en flecha.

Los tipos de dependencias entre las tareas que pueden utilizarse con este sistema de actividades en nodo son:

3.2.2.1. Relación de Final-Comienzo

La relación de final-comienzo entre i y j ($FC(z) ij$), significa que la tarea i , ha de estar totalmente finalizada con anterioridad al comienzo de la tarea j , debiendo existir al menos un desfase de z lapsos de tiempo entre los dos sucesos, pudiendo ser este lapso cero o negativo.

Si el lapso fuera cero o positivo, la relación se leerá como "después de..." y si fuera negativo como "antes de...", siendo de aplicación al resto de relaciones que se detallan a continuación.

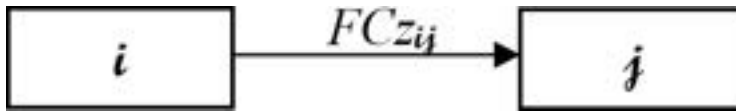


Figura 3.7. Relación final-comienzo.

Para simplificar lo representaremos de la siguiente forma:

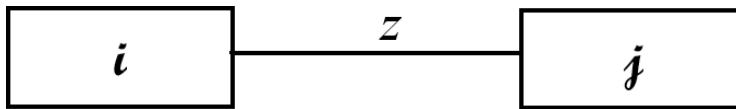


Figura 3.8. Relación final-comienzo simplificada.

Como se verá ha desaparecido la flecha de la relación al estar implícita en su localización de salida (lateral derecho) desde i y de llegada a j (lateral izquierdo).

3.2.2.2. Relación de Comienzo-Comienzo

La relación de comienzo-comienzo entre i y j ($CC(z) ij$), significa que la tarea i debe de estar iniciada, para que comience la tarea j , debiendo existir al menos un desfase de z lapsos de tiempo entre los dos sucesos, pudiendo ser este lapso cero o negativo.

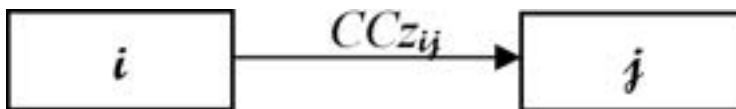


Figura 3.9. Relación comienzo-comienzo.

Para simplificar lo representaremos de la siguiente forma:

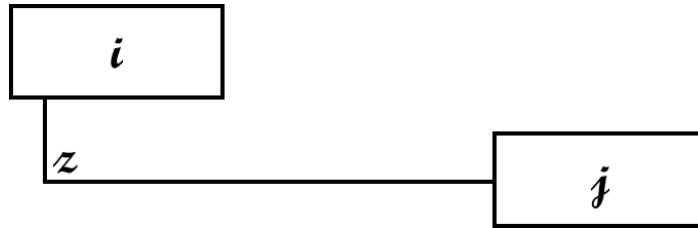


Figura 3.10. Relación comienzo-comienzo simplificada.

3.2.2.3. Relación de Final-Final

La relación de final-final entre *i* y *j* ($FF(z) ij$), significa que la tarea *i* debe de estar finalizada, para que finalice la tarea *j*, debiendo existir al menos un desfase de *z* lapsos de tiempo entre los dos sucesos, pudiendo ser cero o negativo.

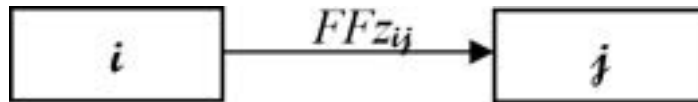


Figura 3.11. Relación final-final.

Para simplificar lo representaremos de la siguiente forma:



Figura 3.12. Relación final-final simplificada.

3.2.2.4. Relación de Comienzo-Final

La relación de comienzo-final entre *i* y *j* ($CF(z) ij$), significa que la tarea *i* debe de estar iniciada, para que finalice la tarea *j*, debiendo existir al menos un desfase de *z* lapsos de tiempo entre los dos sucesos, pudiendo ser cero o negativo. Véase la figura 3.13.

Para simplificar lo representaremos de la siguiente forma, véase la figura 3.14.

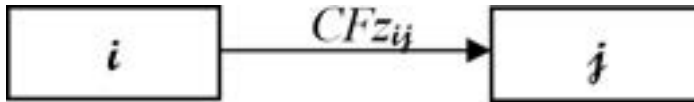


Figura 3.13. Relación comienzo-final.

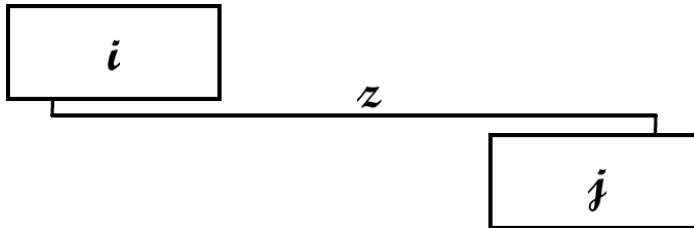


Figura 3.14. Relación comienzo-final simplificada.

3.2.2.5. Simplificación de relaciones

El tipo de relaciones más común entre las tareas de un proyecto es el de final-comienzo, pudiendo representar el resto de relaciones como relaciones de final-comienzo haciendo sencillas operaciones aritméticas.

Así una relación de comienzo-comienzo de z lapsos entre i y j , la podríamos representar como una relación final-comienzo con un lapso de tiempo igual a z menos la duración de i , y una relación de final-final de z lapsos entre i y j , la representaremos como una relación final-comienzo con un lapso de tiempo igual a z menos la duración de j .

Esta simplificación no es muy ortodoxa, además de ser tremendamente criticada por algunos autores aunque sea la utilizada por algunos de los más prestigiosos programas de gestión de proyectos disponibles en el mercado, pero con el afán de no complicar demasiado este tratado tan solo utilizaremos relaciones de final-comienzo, haciendo las transformaciones oportunas en caso de que sean necesarias.

3.3. Los tiempos de las tareas

El primer problema con el que nos encontraremos cuando afrontemos la programación de un proyecto, será el de determinar los tiempos o instantes de comienzo y finalización de las tareas que lo forman.

Este problema puede parecer difícil, pero sin embargo es tremendamente sencillo y podríamos resumirlo con la siguiente expresión:

NOTA

"El instante de finalización de una tarea será su comienzo más su duración".

El único problema que me puede surgir es determinar cuál será ese instante de comienzo si la tarea en cuestión depende de la finalización de otras tareas precedentes, pero veamos cómo se resuelve.

Si una tarea j tan solo depende de la finalización de una tarea predecesora i , el comienzo de j será el final de la tarea i , y en el caso de que tengamos más de una tarea predecesora, el comienzo de j será el final más desfavorable de todas las tareas predecesoras, y que vendrá determinado por el instante de terminación más tardío de las tareas i :

$$\text{comienzo de } j = \text{máximo de } \{\text{finales de } i\}$$

Pero para cada una de las tareas de un proyecto, no solo tendremos que determinar sus instantes de comienzo y finalización, sino que deberemos de determinar los márgenes más pronto y más tarde entre los que se encuentran los comienzos y finales.

Estos márgenes son los conocidos como los tiempos "-más-pronto" y "-más-tarde":

3.3.1. Tiempo más pronto de empezar ($Tmpe_i$)

El Tiempo más pronto de empezar una tarea, representa el instante de tiempo en el que como muy pronto puede empezar una tarea, cumpliendo las relaciones de precedencias impuestas en el proyecto.

Indistintamente lo llamaremos:

$Tmpe_i$ o bien ES_i (*Early Start*) por su universalidad en los algoritmos que expondremos.

3.3.2. Tiempo más pronto de terminar ($Tmpt_i$)

El Tiempo más pronto de terminar una tarea, representa el instante de tiempo en el que como muy pronto puede finalizar una tarea, cumpliendo las relaciones de precedencias impuestas en el proyecto.

Indistintamente lo llamaremos $Tmpt_i$ o EF_i (*Early Finish*).

3.3.3. Tiempo más tarde de empezar ($Tmte_i$)

El Tiempo más tarde de empezar una tarea, representa el instante de tiempo en el que como muy tarde puede comenzar una tarea, cumpliendo las relaciones de precedencias impuestas en el proyecto, más tarde del cual se retrasaría la finalización del proyecto.

Indistintamente lo llamaremos $Tmte_i$ o LS_i (*Latest Start*).

3.3.4. Tiempo más tarde de terminar ($Tmtt_i$)

El Tiempo más tarde de terminar una tarea, representa el instante de tiempo en el que como muy tarde puede finalizar una tarea, cumpliendo las relaciones de precedencias impuestas en el proyecto, más tarde del cual se retrasaría la finalización del proyecto.

Indistintamente lo llamaremos $Tmtt_i$ o LF_i (*Latest Finish*).

INSTANTES O TIEMPOS VS UNIDADES TEMPORALES O FECHAS

Aunque posteriormente se profundizará en la diferencia entre los dos conceptos, es importante empezar a destacar que los valores de tiempos no se refieren a unidades temporales como puedan ser los minutos, horas, días o semanas, sino que se refieren a los instantes de tiempo existentes entre esas unidades de tiempo, de tal manera que una unidad de tiempo t se encontrará entre el instante de comienzo $t-1$ y el instante de finalización t .

Los anteriores tiempos o instantes los representaremos dentro del Nodo de la tarea junto con su duración de la siguiente forma:

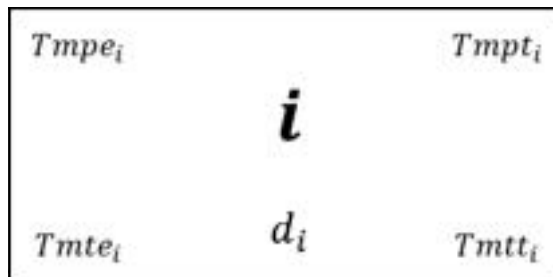


Figura 3.15. Nodo de tarea i en detalle.

3.4. El algoritmo de cálculo de los tiempos

Con todos los conocimientos anteriores, ya podemos expresar nuestro primer algoritmo, que será el del cálculo de los tiempos de los nodos o de las tareas:

El algoritmo de cálculo de los tiempos, tal y como se ha comentado anteriormente, está basado en la posibilidad de conocer el instante de finalización de una tarea una vez conocido su comienzo, y que será este más su duración.

Los algoritmos de cálculo de los tiempos tienen dos pasos, uno primero llamado cálculo adelante o *forward pass*, en la que obtenemos los tiempos más pronto de empezar y terminar y otra fase o paso denominada cálculo atrás o *backward pass* en la que calculamos los tiempos más pronto.

Así de forma general lo expresaremos:

1. Cálculo adelante.

- $Tmpe_j = \text{máximo} (\text{Inicio}; tmpt_i + z_{ij})$; Para todas las tareas j y sus predecesoras.
- $Tmpt_j = tmpe_j + d_j$; Para todas las tareas j .
- $Dur_Proyecto = \text{máximo} (tmpt_j)$; Para todas las tareas j .

2. Cálculo atrás.

- $Tmtt_i = \text{mínimo} (Dur_Proyecto; tmpe_j - z_{ij})$; Para todas las tareas i y sus sucesoras.
- $Tmte_i = tmtt_i - d_i$; Para todas las tareas i .

Donde:

- " z_{ij} " es el desfase entre i y j .
- "*Inicio*" es el tiempo previsto de inicio del proyecto.
- "*Dur_Proyecto*" es el tiempo mínimo de finalización del proyecto.

Una vez hemos obtenido los tiempos de las tareas, podemos obtener unos valores denominados holguras que serán de una importancia para la resolución de los diferentes problemas que vamos a resolver sobre los proyectos:

3.4.1. Holgura Total (Ht_i)

La holgura total de una tarea (Ht_i) es la cantidad de tiempo que podemos retrasar el comienzo de una tarea sin afectar a la duración del proyecto.

Se calcula como la diferencia entre el tiempo más tarde de terminar y el tiempo más pronto de empezar menos su duración:

$$Ht_i = Tmtt_i - Tmpe_i - d_i$$

3.4.2. Holgura Libre (HL_i)

La holgura libre de una tarea (HL_i), es la cantidad de tiempo que podemos retrasar el comienzo de una tarea sin afectar al tiempo más pronto de comienzo de las tareas sucesoras.

Se calcula de la siguiente forma:

$$HL_i = \text{mínimo} \{Tmpe_j - z_{ij} - d_i\} - Tmpe_i$$

3.4.3. Holgura independiente

La holgura independiente de una tarea, es la cantidad de tiempo que podemos retrasar el comienzo de una actividad definido por el tiempo más tarde de terminar de sus precedentes sin afectar al tiempo más pronto de comienzo de las actividades sucesoras.

3.4.4. Holgura condicional

La holgura condicional de una tarea, es la cantidad de tiempo que podemos retrasar el comienzo de una actividad definido por los tiempos más tardes de sus precedentes sin afectar el tiempo más tarde de sus sucesoras. Se calcula de la siguiente forma:

De los cuatro valores de holgura, el más importante es la holgura total, siguiéndole muy de cerca la holgura libre, siendo además especialmente trascendente cuando abordamos problemas de nivelación de recursos, relegando los valores de holgura independiente y condicional a valores meramente informativos. A partir de los valores de la holgura total, haremos una primera clasificación de las tareas como tareas críticas y no críticas:

3.4.5. Tarea crítica

Consideramos que una tarea es crítica cuando la diferencia entre el tiempo más tarde de terminar ($Tmtt_i$ o LF_i) y el tiempo más pronto de empezar ($Tmpe_i$ o ES_i) es su duración (d_i), es decir cuando su holgura total es cero.

3.4.6. Tarea no crítica

Consideramos que una tarea es no crítica cuando la diferencia entre el tiempo más tarde de terminar ($Tmtt_i$ o LF_i) y el tiempo más pronto de empezar ($Tmpe_i$ o ES_i) es mayor que su duración (d_i), es decir cuando su holgura total es distinta de cero.

Se da la circunstancia de que las tareas críticas se organizan a lo largo del grafo mediante una secuencia (o secuencias) continua que va desde el inicio del proyecto hasta el final del mismo. A esta secuencia la llamaremos Camino o ruta crítica.

3.4.7. Camino crítico o ruta crítica

El camino crítico es el camino más largo del grafo, donde todas sus relaciones y actividades son críticas y que determina la duración del proyecto. Puede haber varios caminos críticos en un grafo.

3.5. La representación matricial de los grafos y su cálculo

Hasta ahora hemos aprendido como representar los proyectos como un conjunto ordenado de pares de elementos de arcos y nodos que hemos denominado grafos mediante dos técnicas diferentes, una denominada de actividades en flecha (AoA) y otra más actual llamada de actividades en nodo (AoN), mucho más sencilla de representar y sin las engorrosas tareas ficticias. Además hemos aprendido a calcular los cuatro tiempos de las tareas, determinar sus holguras y clasificarlas como tareas críticas y tareas no críticas.

La duda que surgirá entre los lectores será:

¿Cómo "dibujar" un grafo en Excel para su tratamiento?

Pues haciendo uso de la principal característica de Excel, que es trabajar con datos almacenados en forma de celdas con filas y columnas ordenadas constituyendo matrices.

3.5.1. Introducción a la representación matricial

La característica más interesante de los grafos y que hasta ahora no se había comentado, es que los podemos representar como matrices ordenadas sin necesidad de recurrir a su representación previa mediante grafos, en donde las filas y las columnas serán las tareas de mi proyecto, y las intersecciones entre ellas serán las relaciones de dependencia existente entre ellas:

		<i>tareas precedentes</i>							
		<i>1</i>	<i>2</i>	<i>...</i>	<i>...</i>	<i>i</i>	<i>...</i>	<i>...</i>	<i>N</i>
<i>tareas</i>	<i>1</i>								
	<i>2</i>								
	<i>...</i>								
	<i>...</i>								
	<i>j</i>								
	<i>...</i>								
	<i>...</i>								
	<i>N</i>								

Figura 3.16. Representación matricial del grafo.

Si los valores de intersección de la anterior representación matricial están vacíos, significará que no existe relación entre las tareas correspondientes, y en caso contrario tendremos una relación de precedencia entre ellas, que podrá ser de cero o cualquier otro valor, de tal manera que si es cero significará que la tarea *j* comenzara una vez finalizadas las tareas precedentes, si es mayor que cero que comenzará *z* lapsos de tiempo después de finalizar la tarea correspondiente, y si es negativa antes de finalizar (véase la figura 3.17).

		<i>tareas precedentes</i>							
		1	2	...	...	i	...	...	N
<i>tareas</i>	1	Z_{11}	Z_{12}			Z_{1i}			Z_{1n}
	2	Z_{21}	Z_{22}			Z_{2i}			Z_{2n}
	...								
	...								
	j	Z_{j1}	Z_{j2}			Z_{ji}			Z_{jn}
	...								
	...								
	N	Z_{n1}	Z_{n2}			Z_{ni}			Z_{nn}

Figura 3.17. Relaciones en la matriz del grafo.

Siguiendo este criterio nuestro grafo de proyecto de cinco viviendas lo representaremos así (véase la figura 3.18).

		<i>tareas precedentes</i>														
		<i>Ex 1</i>	<i>Ex 2</i>	<i>Ex 3</i>	<i>Ex 4</i>	<i>Ex 5</i>	<i>Ci 1</i>	<i>Ci 2</i>	<i>Ci 3</i>	<i>Ci 4</i>	<i>Ci 5</i>	<i>St 1</i>	<i>St 2</i>	<i>St 3</i>	<i>St 4</i>	<i>St 5</i>
<i>tareas</i>	<i>Ex 1</i>															
	<i>Ex 2</i>															
	<i>Ex 3</i>															
	<i>Ex 4</i>															
	<i>Ex 5</i>															
	<i>Ci 1</i>															
	<i>Ci 2</i>															
	<i>Ci 3</i>															
	<i>Ci 4</i>															
	<i>Ci 5</i>															
	<i>St 1</i>															
	<i>St 2</i>															
	<i>St 3</i>															
	<i>St 4</i>															
	<i>St 5</i>															

Figura 3.18. Representación del Proyecto con la matriz.

Pero recordaremos que nuestro grafo no podía contener circuitos, y para que se de esta circunstancia, es condición suficiente que el grafo este adecuadamente ordenado, para lo cual es necesario que los valores de las relaciones tan solo afecten a la zona sombreada de la matriz.

		tarear precedentes														
		Ex 1	Ex 2	Ex 3	Ex 4	Ex 5	Gi 1	Gi 2	Gi 3	Gi 4	Gi 5	St 1	St 2	St 3	St 4	St 5
tarear	Ex 1															
	Ex 2	0														
	Ex 3		0													
	Ex 4			0												
	Ex 5				0											
	Gi 1	0														
	Gi 2		0				0									
	Gi 3			0				0								
	Gi 4				0				0							
	Gi 5					0				0						
	St 1						0									
	St 2							0					0			
	St 3								0					0		
	St 4									0					0	
	St 5										0					0

Figura 3.19. Zona de índices válidos de la matriz.

NOTA

Existe la posibilidad de aplicar un algoritmo de ordenación topológica del grafo, que aún siendo sencillo de entender y de implementar, es preferible dejarlo para algo más adelante con el fin de no abrumar con demasiados conceptos novedosos.

3.5.2. La representación matricial con Excel

Una vez que ya disponemos de todos los conceptos necesarios, vamos a proceder a representar nuestro proyecto con Excel, para lo cual lo primero que haremos será:

1. Cambiar el nombre "Hoja1" que Excel da por defecto a la primera hoja del libro por "Matriz", para lo cual situaremos el cursor del ratón sobre la pestaña "Hoja1" y pulsaremos el botón derecho aparecerá el siguiente menú contextual (véase la figura 3.20):
2. Dar el nombre "Inicio" a una celda que nos servirá para indicar a Excel donde va a empezar nuestra matriz, en nuestro caso es la celda B4, mediante Fórmulas>Asignar Nombre>Definir Nombre y teclear: **Inicio** (véase la figura 3.21).



Figura 3.20. De Hoja1 a Matriz.



Figura 3.21. Celda Inicio.

Una vez definido el nombre de la celda B4 como la celda **Inicio**, escribiremos en cada una de las celdas inmediatamente debajo todas y cada una de las tareas de mi proyecto, intentando seguir un orden secuencial.

Recomiendo utilizar para esta denominación un nombre corto de no más de 10 caracteres y que sean suficientes para identificarla inequívocamente.

Al final lo que nos tiene que quedar es algo como lo que puebe ver en la figura 3.22.

Ya estamos en condición de crear nuestro primer control ActiveX, que automatice la creación de la estructura de la matriz de forma que tan solo nos falte introducir los valores de las duraciones y de las dependencias para obtener todos los valores de los tiempos y holguras de cada una de las tareas.

Para crear el primer control, vamos a proceder de la siguiente forma Programador>Insertar>Botón de Comando (control ActiveX), tan solo hacer

clic y arrastrar con el botón izquierdo del ratón en la zona de la pantalla deseada hasta alcanzar el tamaño adecuado para el botón, aunque posteriormente pueda ser modificado (véase la figura 3.23).

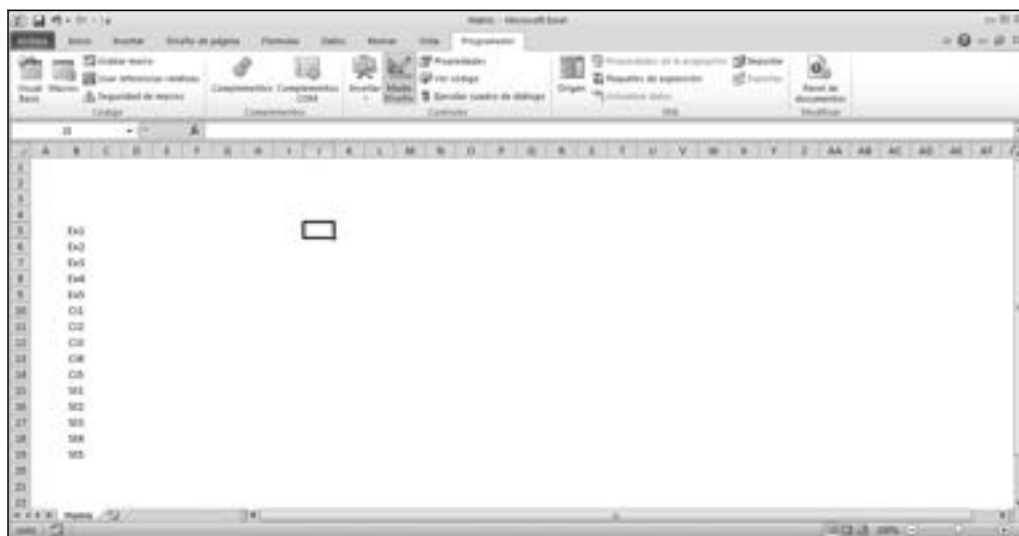


Figura 3.22. Tareas en la matriz.

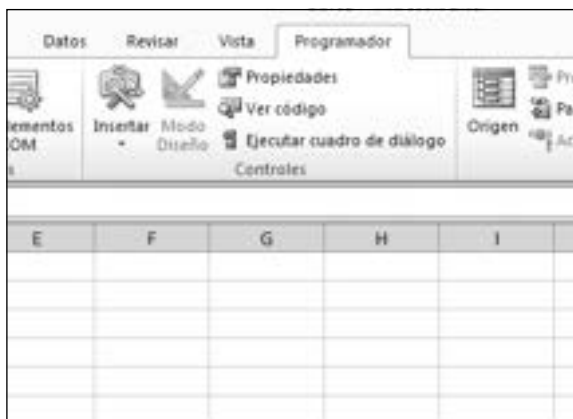


Figura 3.23. Insertar botón de comando (ActiveX).

Una vez creado el botón de comando, y mediante el botón derecho del ratón haremos aparecer el menú contextual donde seleccionaremos **Propiedades**, realizando los cambios que más se adapten a nuestros gustos. En nuestro caso, y como vamos a crear más controles le cambiaremos el texto asignado por defecto en **Caption** por **Crear**.



Figura 3.24. Propiedades de botón de comando (ActiveX).

Tan solo nos quedará indicarle al botón mediante un código en Visual Basic la secuencia de órdenes que se ejecutarán al hacer clic con el ratón sobre el botón de comando. Para introducir el código del procedimiento, activaremos la pantalla de Visual Basic haciendo doble clic con el ratón sobre el botón, apareciendo la siguiente nueva pantalla:



Figura 3.25. Ventana de Visual Basic.

Y, a continuación, en el interior del procedimiento, después de "Private Sub CommandButton1_Click ()" y antes de "End Sub" escribiremos el siguiente código:

```
'Inicializa la Matriz
Dim i As Integer
Dim j As Integer
Dim fila As Integer
Dim columna As Integer
Dim Num_tareas As Integer
Dim texto As String
With Worksheets("Matriz")

    'Establecemos la posición de la celda Inicio
    fila = Worksheets("Matriz").Range("Inicio").Row
    columna = Worksheets("Matriz").Range("Inicio").Column
    'Calcula el número de tareas
    Do
        If .Cells(fila + Num_tareas + 1, columna).Value = "" Then Exit Do
        Num_tareas = Num_tareas + 1
    Loop
    'Construye las Columnas de Duración y Tiempos
    .Cells(fila, columna + 1).Value = "Dur"
    .Cells(fila, columna + 2).Value = "Tmpe"
    .Cells(fila, columna + 3).Value = "Tmpt"
    .Cells(fila, columna + 4).Value = "Tmte"
    .Cells(fila, columna + 5).Value = "Tmtt"
    .Cells(fila, columna + 6).Value = "Htot"
    .Cells(fila, columna + 7).Value = "Hlib"

    'Construye la primera Fila de la matriz
    For i = 1 To Num_tareas
        texto = .Cells(fila + i, columna).Value
        .Cells(fila, columna + 7 + i).Value = texto
    Next

    'crea los bordes de la matriz
    For j = 0 To Num_tareas + 7
        .Cells(fila, columna + j).Borders.LineStyle = xlContinuous
    Next

    For i = 1 To Num_tareas
        For j = 0 To 7
            .Cells(fila + i, columna + j).Borders.LineStyle = xlContinuous
        Next
    Next

    For i = 0 To Num_tareas
        For j = columna + 6 To i + 6
            .Cells(fila + i, columna + j).Borders.LineStyle = xlContinuous
        Next
    Next

Next
End With
```

Una vez introducido el código, cerraremos la ventana de Visual Basic, saldremos del modo Diseño y haremos clic con el botón izquierdo del ratón,

aparecerá la estructura de la matriz de dependencias en nuestra hoja denominada Matriz:

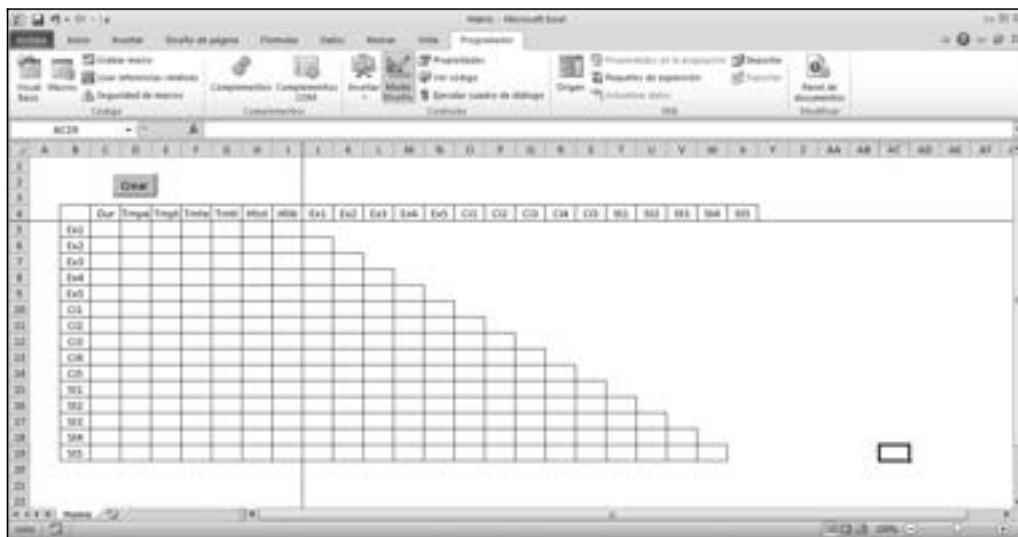


Figura 3.26. Estructura de la Matriz del proyecto.

A la matriz le hemos añadido un pequeño detalle anclando los paneles para lo cual nos hemos situado en la celda intersección de las dos primeras tareas del proyecto, y hemos seleccionado: **Vista>Inmovilizar>Movilizar Paneles**, que nos permitirá desplazarnos con total libertad por toda la hoja sin perder las referencias de filas y columnas de la celda en la que nos encontramos, algo que resulta muy útil en el caso de que el proyecto sea muy grande.

Como se puede ver en la estructura de la matriz creada, han aparecido nuevas columnas para la duración, para los tiempos y holguras, además de tantas columnas nuevas como tareas tiene nuestro proyecto. No todas las celdas de la matriz tiene los bordes dibujados, y eso es porque en el momento de crearla tan solo hemos resaltado aquellas celdas en cuya posición la dependencia introducida será correcta, algo que me facilitará mucho la depuración de errores en el proyecto.

Aunque veamos muchas celdas vacías, de la matriz, tan solo deberemos de completar la columna "Dur" correspondiente a la duración de la tarea, y las celdas correspondiente a las dependencias entre ellas, rellenado los valores de los tiempos y de las holguras mediante la creación de un nuevo botón de comando, pero antes de empezar con el código será interesante estudiar cómo se calcularán los tiempos.

3.5.3. El cálculo matricial de los tiempos con Excel

El método de cálculo de los tiempos basado en matrices que se va a exponer, es un algoritmo original que está basado en los trabajos realizados por Zaderenko, que fue el primero en establecer las reglas de cálculo de los tiempos de las tareas de un proyectos representados mediante matrices.

El trabajo de Gregori Zaderenko se podría decir que fue visionario, pues nos hemos de remontar al año 1968 (Sistemas de Programación por camino crítico: PERT-CPM-MAN SCHEDULING-RAMPS y otros métodos de elaboración y control de programas. Buenos Aires. Argentina: 001.424 Z16. Librería Mitre. 197 páginas) cuando el acceso a un ordenador estaba restringido a unos pocos científicos selectos, y que decir tiene que no existían las hojas de cálculo. No puedo ni imaginar a lo que habría hecho Zaderenko si hubiera dispuesto de las posibilidades de MS Excel 2010.

Originalmente en su algoritmo, Zaderenko añadía dos tareas ficticias, una tarea ficticia de inicio y otra de final. La tarea ficticia de inicio no era estrictamente necesaria, pero si la final, puesto que tan solo calculaba los tiempos más pronto y más tarde de comienzo, necesitando del artificio en el final para poder determinar la duración del proyecto.

El proceso es muy sencillo:

Para el cálculo de los tiempos más pronto de las tareas, tomaremos, empezando desde el principio y sucesivamente hasta el final:

1. La fila correspondiente a la tarea de la que estamos calculando los tiempos más pronto.
2. A los valores que aparecen en dicha fila (desfases de las actividades) se les suma el valor disponible en la columna adicional (Tmpt) para las filas correspondientes a esa columna.
3. El tiempo más pronto de empezar que estamos calculando será la mayor de esas sumas.
4. El Tiempo más pronto de terminar será inmediato, tan solo debaremos sumarle su duración y ya lo tenemos.

Para el cálculo de los tiempos más tarde de las tareas, procederemos exactamente igual pero al contrario, desde el final al principio, adoptando el menor de los tiempos más tarde calculados y restando finalmente la duración al valor obtenido.

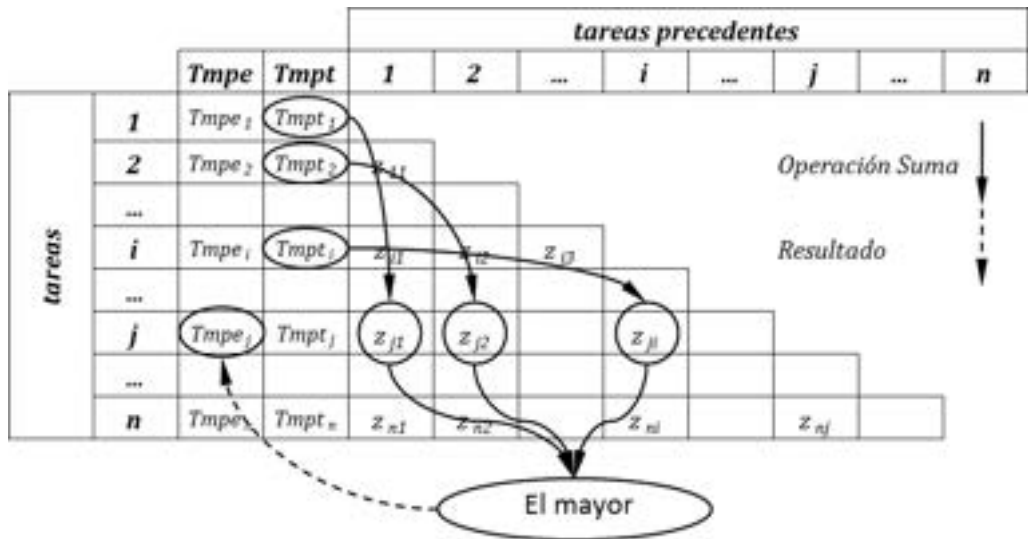


Figura 3.27. Cálculo matricial de los $Tmpe$.

Para implementar el algoritmo en Excel, deberemos crear un nuevo botón de comando (ActiveX), al que le cambiaremos el texto que aparece por defecto en **Caption** por **Calcular** y le asignaremos el siguiente código:

```
Private Sub CommandButton2_Click ()
    Call Calcular
End Sub

Sub Calcular()
    Dim i As Integer
    Dim j As Integer
    Dim fila As Integer
    Dim columna As Integer
    Dim Num_tareas As Integer
    Dim texto As String
    Dim Dur As Integer
    Dim Desfase As Integer
    Dim Tmpe_j As Integer
    Dim Tmte_j As Integer
    Dim Tmpt_j As Integer
    Dim Tmpe_i As Integer
    Dim Tmpt_i As Integer
    Dim Tmte_i As Integer
    Dim Tmtt_i As Integer
    Dim Dur_Proj As Integer

    With Worksheets("Matriz")
        'Establecemos la posición de la celda Inicio
        fila = Worksheets("Matriz").Range("Inicio").Row
        columna = Worksheets("Matriz").Range("Inicio").Column
        'Calcula el número de tareas
```

```

Do
    If .Cells(fila + Num_tareas + 1, columna).Value = "" Then Exit Do
    Num_tareas = Num_tareas + 1
Loop
'Calcula los tiempos más pronto
For j = 1 To Num_tareas
    Dur = .Cells(fila + j, columna + 1).Value
    Tmpe_j = 0
    For i = 1 To j - 1
        texto = .Cells(fila + j, columna + 7 + i).Value
        If texto <>"" Then
            Desfase = Val(texto)
            Tmpt_i = .Cells(fila + i, columna + 3).Value
            If Tmpt_i + Desfase > Tmpe_j Then Tmpe_j = Tmpt_i + Desfase
        End If
    Next
    Tmpt_j = Tmpe_j + Dur
    If Dur_Proj < Tmpt_j Then Dur_Proj = Tmpt_j
    .Cells(fila + j, columna + 2).Value = Tmpe_j
    .Cells(fila + j, columna + 3).Value = Tmpt_j
Next
'Calcula los tiempos más tarde
For i = Num_tareas To 1 Step -1
    Dur = .Cells(fila + i, columna + 1).Value
    Tmtt_i = Dur_Proj
    For j = i + 1 To Num_tareas
        texto = .Cells(fila + j, columna + 7 + i).Value
        If texto <>"" Then
            Desfase = Val(texto)
            Tmte_j = .Cells(fila + j, columna + 4).Value
            If Tmte_j - Desfase < Tmtt_i Then Tmtt_i = Tmte_j - Desfase
        End If
    Next
    Tmte_i = Tmtt_i - Dur
    .Cells(fila + i, columna + 4).Value = Tmte_i
    .Cells(fila + i, columna + 5).Value = Tmtt_i
Next
'Cálculo de Holguras
For i = 1 To Num_tareas
    'Total
    Dur = .Cells(fila + i, columna + 1).Value
    Tmpe_i = .Cells(fila + i, columna + 2).Value
    Tmtt_i = .Cells(fila + i, columna + 5).Value
    .Cells(fila + i, columna + 6).Value = Tmtt_i - Tmpe_i - Dur
    'Libre
    Tmpt_i = .Cells(fila + i, columna + 3).Value
    .Cells(fila + i, columna + 7).Value = Tmtt_i - Tmpt_i
    For j = i + 1 To Num_tareas
        texto = .Cells(fila + j, columna + 7 + i).Value
        If texto <>"" Then
            Desfase = Val(texto)
            Tmpe_j = .Cells(fila + j, columna + 2).Value
            If Tmpe_j - Desfase - Tmpt_i < .Cells(fila + i, columna +
                7).Value Then
                .Cells(fila + i, columna + 7).Value = Tmpe_j - Desfase
                    - Tmpt_i
            End If
        End If
    Next
Next

```



```
With Worksheets("Matriz")
    'Establecemos la posición de la celda Inicio
    fila = Worksheets("Matriz").Range("Inicio").Row
    columna = Worksheets("Matriz").Range("Inicio").Column
    'Calcula el número de tareas
    Do
        If .Cells(fila + Num_tareas + 1, columna).Value = "" Then Exit Do
        Num_tareas = Num_tareas + 1
    Loop
    'Borra la matriz
    For i = 0 To Num_tareas + 1
        For j = 0 To Num_tareas + 7
            .Cells(fila + i, columna + j).Value = ""
            .Cells(fila + i, columna + j).Borders.LineStyle = xlLineStyleNone
        Next
    Next
End With
```

NOTA

La hoja de cálculo completa se encuentra en Cap_03\Matriz, de los ejemplos del libro.

Llegados a este punto, ya estamos en condiciones de representar las relaciones de dependencias entre las tareas de un proyecto por muy complejo que éste sea, así como calcular los tiempos y las holguras de las tareas con la hoja de cálculo MS Excel con un sistema de representación por medio de matrices que sustituye perfectamente a la clásica representación por medio de grafos que conocíamos hasta ahora, pero puede que el sistema de cálculo que hemos utilizado no siempre cumpla con las necesidades de procesamiento posterior, cuando quiera reprogramar, optimizar y simular mi proyecto.

Los valores obtenidos para los tiempos y las holguras son valores estáticos, es decir, necesito recalcular el proyecto cada vez que modifico una desfase (valor no dependencia) o la duración de una tarea, algo que no parece ser muy práctico cuando el proyecto ha de ser procesado varios millones de veces, sería más interesante si pudiera hacer que esos valores fueran unas fórmulas que se actualizaran constantemente en vez de valores estáticos. Para Excel eso no es un problema, tan solo debo añadir un nuevo comando de botón (ActiveX) que en vez de asignar valores a las celdas en función de las dependencias, desfase y duraciones, asigne las fórmulas completas que me han llevado a esos valores.

Otra importante elemento que sería interesante añadir es el día de inicio de mi proyecto, puesto que evidentemente la programación no la hago

el día 1 del proyecto, sino que la realizo con muchos lapsos de tiempo de antelación. Y el código con las nuevas incorporaciones será:

```
Sub Formular()
    Dim i As Integer
    Dim j As Integer
    Dim fila As Integer
    Dim columna As Integer
    Dim Num_tareas As Integer
    Dim texto As String
    Dim texto_for As String
    Dim Texto_Dur_Proj As String
    Dim desfase As String
    Dim letra As String
    Dim Dur_Proj As Integer

    With Worksheets("Matriz")
        'Establecemos la posición de la celda Inicio
        fila = Worksheets("Matriz").Range("Inicio").Row
        columna = Worksheets("Matriz").Range("Inicio").Column
        'Calcula el número de tareas
        Do
            If .Cells(fila + Num_tareas + 1, columna).Value = "" Then Exit Do
            Num_tareas = Num_tareas + 1
        Loop

        'Calcula los tiempos más pronto
        For j = 1 To Num_tareas
            'De Empezar
            texto_for = "=MAX(" & "dia_Inicio"
            For i = 1 To j - 1
                If .Cells(fila + j, columna + 7 + i).Value <>"" Then
                    desfase = Letra_Col(columna + i + 6) & Trim(Str(fila + j))
                    texto_for = texto_for & ", " & Chr(columna + 67) &
                        Trim(Str(fila + i)) & "+" & desfase
                End If
            Next
            texto_for = texto_for & ")"
            .Cells(fila + j, columna + 2).formula = texto_for
            'De Terminar
            texto_for = "=" & Chr(columna + 65) + Trim(Str(fila + j))
            texto_for = texto_for & "+" & Chr(columna + 66) +
                Trim(Str(fila + j))
            .Cells(fila + j, columna + 3).formula = texto_for

            If Dur_Proj < .Cells(fila + j, columna + 3).Value Then
                Dur_Proj = .Cells(fila + j, columna + 3).Value
                Texto_Dur_Proj = Chr(columna + 67) & Trim(Str(fila + i))
            End If
        Next

        'Calcula los tiempos más tarde
        For i = Num_tareas To 1 Step -1
            'De Terminar
            texto_for = "=MIN(" & Texto_Dur_Proj
```

```

For j = i + 1 To Num_tareas
    If .Cells(fila + j, columna + 7 + i).Value <>"" Then
        desfase = Letra_Col(columna + i + 6) & Trim(Str(fila + j))
        texto_for = texto_for & ", " & Chr(columna + 68) &
            Trim(Str(fila + j)) & "-" & desfase
    End If
Next
texto_for = texto_for & ")"
.Cells(fila + i, columna + 5).formula = texto_for
'De Empezar
texto_for = "=" & Chr(columna + 69) + Trim(Str(fila + i))
texto_for = texto_for & "-" & Chr(columna + 65) +
    Trim(Str(fila + i))
.Cells(fila + i, columna + 4).formula = texto_for
Next

'Calcula las Holguras
For i = 1 To Num_tareas

    'Totales
    texto_for = "=" & Chr(columna + 69) + Trim(Str(fila + i))
    texto_for = texto_for & "-" & Chr(columna + 66) +
        Trim(Str(fila + i))
    texto_for = texto_for & "-" & Chr(columna + 65) +
        Trim(Str(fila + i))
    .Cells(fila + i, columna + 6).formula = texto_for

    'Libres
    texto_for = "=MIN ("
    texto_for = texto_for & Chr(columna + 69) + Trim(Str(fila + i))
    texto_for = texto_for & "-" & Chr(columna + 67) +
        Trim(Str(fila + i))

    For j = i + 1 To Num_tareas
        If .Cells(fila + j, columna + 7 + i).Value <>"" Then
            desfase = Letra_Col(columna + i + 6) & Trim(Str(fila + j))
            texto_for = texto_for & " , " & Chr(columna + 66) &
                Trim(Str(fila + j)) & "-" & desfase
            texto_for = texto_for & "-" & Chr(columna + 67) +
                Trim(Str(fila + i))
        End If
    Next
    texto_for = texto_for & ")"
    .Cells(fila + i, columna + 7).formula = texto_for
Next
End With
End Sub

```

También podemos personalizar la matriz, añadiendo formatos condicionales de tal forma que las tareas críticas destaquen sobre el resto de tareas, o destacando las celdas que contienen una relación de dependencia.

Para establecer un formato condicional que destaque las celdas con relaciones actuaremos de la siguiente forma:

1. Seleccionaremos el área de la matriz, que en este proyecto corresponde a las celdas de relaciones (J5:W19).
2. Inicio>Formato condicional>Nueva regla.
3. Aplicar formato únicamente a las celdas con>sin espacios en blanco.

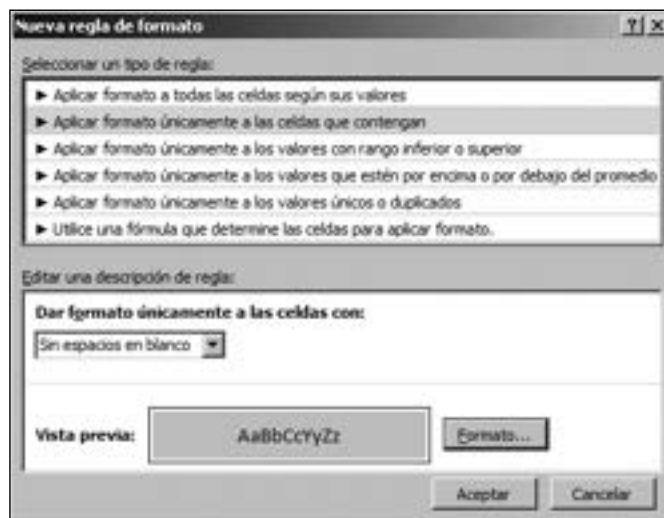


Figura 3.29. Formato condicional de las dependencias.

Y para destacar las tareas críticas, que recordaremos son las que tenían holgura total igual a cero:

1. Seleccionaremos exclusivamente la zona de datos (código, tiempos y holguras) correspondiente a las tareas (B5:I19).
2. Inicio>Formato condicional>Nueva regla.
3. Utilizaremos una fórmula que nos determine las celdas para aplicar formato.
4. Seleccionaremos la celda correspondiente a la holgura total de la primera tarea y establecer la condición igual a cero. Es importante que posteriormente modifiquemos la condición $\$H\$5=0$ por $\$H5=0$ eliminando el dólar correspondiente al cinco para hacer que la condición de la fórmula sea aplicada con carácter relativo a la holgura total de cada tarea (véase la figura 3.30).

Y la nueva matriz completamente finalizada quedará como se muestra en la figura 3.31.

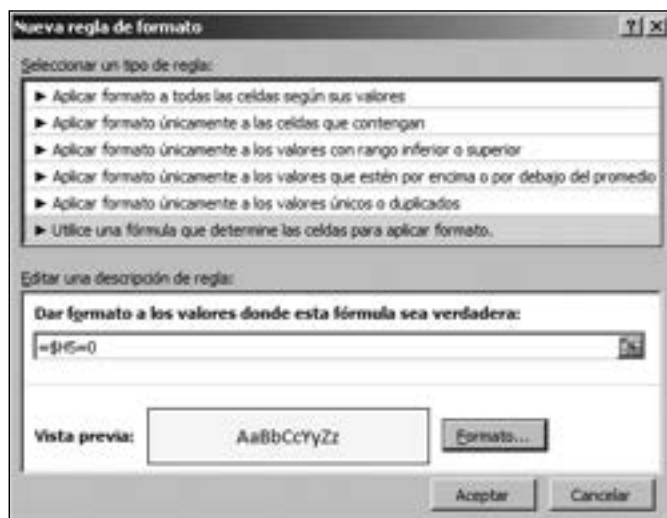


Figura 3.30. Formato condicional de las tareas críticas.

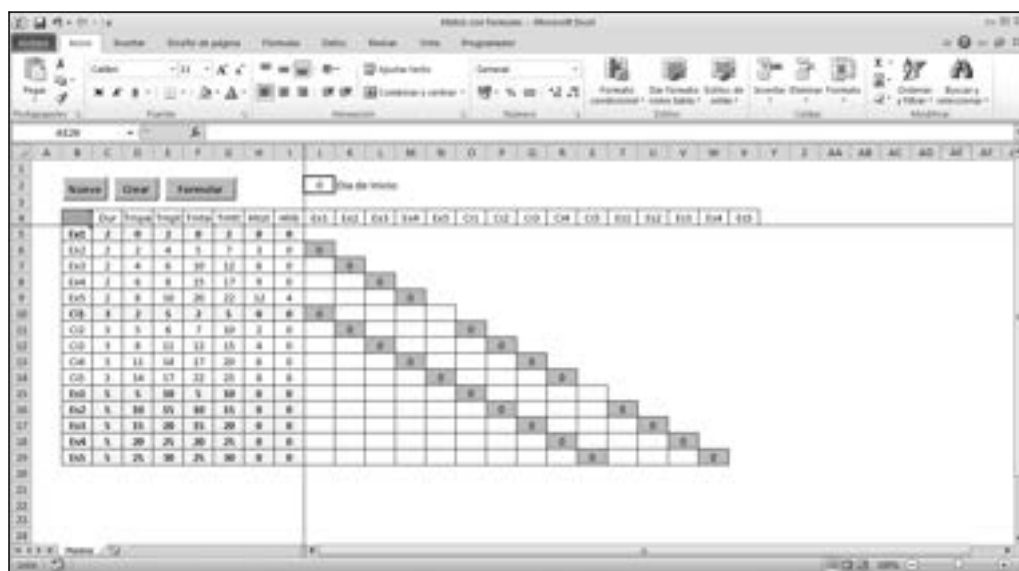


Figura 3.31. Matriz con fórmulas y formatos condicionales terminada.

NOTA

La hoja de cálculo completa se encuentra en Cap_03\Matriz con fórmulas, de los ejemplos del libro.

3.6. La representación del diagrama temporal

Una vez que hemos determinado los tiempos más pronto y más tarde de empezar y terminar, así como las holguras de cada una de las tareas de mi proyecto, hemos de representarlos ahora de una forma fácilmente entendible por terceros.

Se ha de tener muy presente, que la gestión proyectos es una herramienta muy útil y versátil, utilizada por una gran variedad de profesionales de especialidades diferentes, que no necesariamente han de ser expertos en las herramientas y en los modelos matemáticos que utilizamos para programarlos.

Es muy importante, además de ser algo frecuentemente olvidado, que gran parte del éxito de nuestro trabajo en la programación de un proyecto no solo reside en que las estimaciones que realicemos sean muy fiables y los cálculos tremendamente rigurosos y consistentes, es fundamental que la información que ofrezcamos sea visualmente atractiva e intuitiva su interpretación.

Por lo tremendamente sencilla e intuitiva que resulta su lectura por parte de cualquier profano, la herramienta más popular de la gestión de proyectos es el diagrama temporal, conocido popularmente como diagrama de Gantt, aunque recordaremos que su verdadera autoría se la debemos a Karol Adamiecki que lo llamo Harmonigraph.

Excel me proporciona una gran variedad de posibilidades para representar un diagrama temporal en función de la utilidad que le quiera dar o la complejidad que necesite.

Puedo necesitar un sencillo, discreto y atractivo diagrama temporal que utilizaré para incrustar como figura en una memoria, puedo necesitar un diagrama temporal más completo para poder calcular un flujo de caja y su rentabilidad financiera, o uno más complejo que me sirva para poder controlar el avance y la toma de medidas correctoras, y por supuesto me puede interesar que aparezcan o no fechas de calendario o días teóricos.

Vamos a exponer a continuación algunas ideas para su representación con Excel, que por supuesto no son las únicas ni quizás las mejores, que se pueden combinar entre si y que esperamos servirán de punto de partida para que el lector elabore sus propios y personalizados diagramas temporales totalmente interactivos con la información suministrada y los modelos aplicados.

3.6.1. Diagramas temporales con gráficos de Excel

Gracias a las posibilidades graficas de Excel, lo primero que se le ocurrirá al lector es hacer uso de dichas utilidades para representar el diagrama temporal, y vamos a exponer a continuación como se hace:

1. Seleccionaremos el rango de celdas correspondientes a los códigos de las tareas y las duraciones, y que en nuestro caso es B5:C19.
2. Crearemos el gráfico mediante: Insertar>Barra>Barra apilada.



Figura 3.32. Gráfico de barra apilada.

Hemos creado el germen de nuestro diagrama temporal, pero ahora le hemos de dar forma.

Lo primero que haremos será poner las tareas en orden, que como se puede ver aparecen en orden inverso al deseado, para lo cual:

1. Seleccionaremos los códigos de las tareas haciendo clic sobre ellos con el botón izquierdo del ratón.
2. Activaremos luego el menú contextual mediante el botón derecho del ratón.
3. Dar formato a eje...
4. Opciones de eje.
5. Seleccionaremos: Categorías en orden Inverso.

El diagrama temporal ya empieza a tener la forma que deseamos, pero tan solo tenemos los valores de las duraciones, y necesitamos representar cuando empiezan esas tareas y la holgura que tienen.

Para conseguirlo, tan solo necesitamos añadir esos datos a nuestro gráfico, de la siguiente forma:

1. Seleccionaremos las barras del gráfico mediante el botón izquierdo del ratón.
2. Activaremos el menú contextual con el botón derecho del ratón.
3. Seleccionar datos...
4. Editar, y escribiremos **Duraciones**.

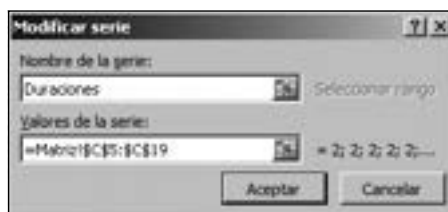


Figura 3.33. Editar duraciones.

5. Agregar, Escribiremos **Tmpe** y seleccionar el área D5:D19.

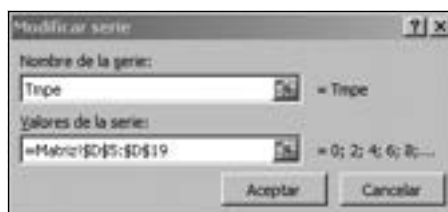


Figura 3.34. Añadir datos de los Tmpe.

6. Agregar, Escribimos **Holgura total** y entonces seleccionamos el área H5:H19.

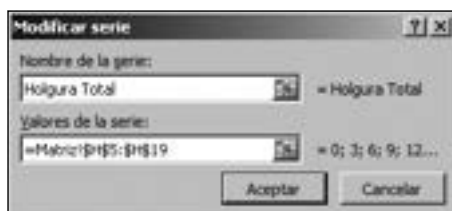


Figura 3.35. Añadir datos de la holgura total.

7. Ordenaremos los conjuntos de datos de tal manera que queden en el orden siguiente:

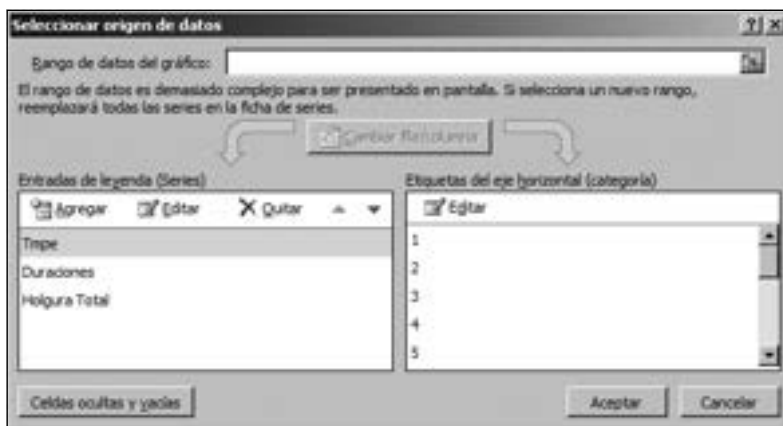


Figura 3.36. Ordenar conjunto de datos.

Y obtendremos como resultado el diagrama temporal que nos muestra la figura 3.37.

Aunque no lo parezca nuestro diagrama temporal está ya prácticamente terminado, tan solo nos quedará pendiente eliminar las leyendas y quitar las barras correspondientes al Tmpe, pero no eliminándolas, sino ocultándolas quitándoles el color tanto de relleno como de los bordes.

Para cambiar los colores a las barras, seleccionaremos la barra correspondiente y que en nuestro caso es la de los Tmpe con el botón izquierdo del ratón, abriremos entonces el menú contextual y a continuación seleccionaremos **Dar formato a serie de datos...**, para posteriormente cambiar el relleno con **Relleno>Sin relleno**, y **Color del borde>Sin línea**, como muestra la figura 3.38.

Por supuesto que es posible cambiar los colores de las barras correspondientes a las duraciones y a las holguras, pero eso lo dejaremos a la imaginación o necesidades del lector según las preferencias o imagen que deseemos transmitir.

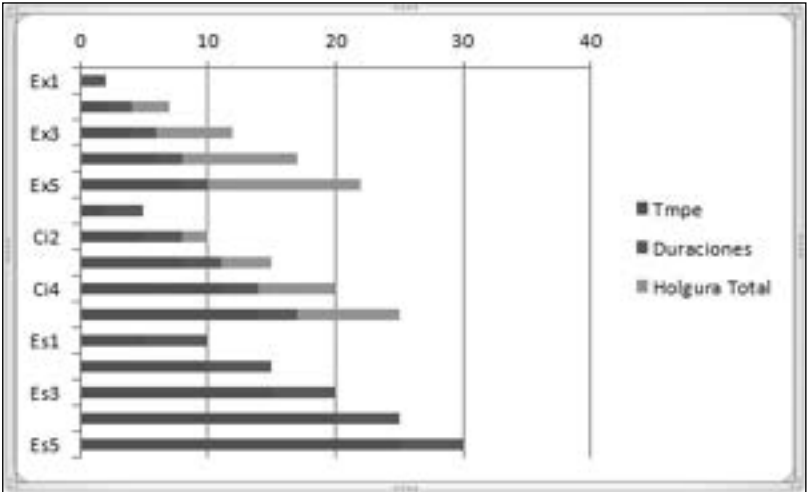


Figura 3.37. Diagrama temporal previo.



Figura 3.38. Matriz con diagrama temporal.

El Diagrama temporal no tiene porque estar como un objeto flotante y posiblemente hasta molesto en la hoja de la matriz, podemos hacer que constituya una Hoja por sí mismo, seleccionándolo y mediante el menú contextual: **Mover gráfico>Hoja nueva**, y escribir **Según los Tmpe**.

Hemos puesto en el nombre de la hoja "Según los Tmpe" porque como se habrá imaginado el avispa lector, también puedo hacer el diagrama temporal "Según los tiempos más tarde", tan solo tendré que repetir el proceso anterior pero esta vez cambiando el orden de visualización, haciendo que primero aparezcan los Tmpe, luego la Holgura total y al final la duración, obteniendo un nuevo diagrama temporal de mi proyecto según los "Tiempos más tarde" con un sencillo y útil artificio.



Figura 3.39. Hoja con diagrama temporal según los tiempos más tarde.

NOTA

La hoja de cálculo completa se encuentra en Cap_03\Diagrama temporal con gráficos, de los ejemplos del libro.

3.6.2. Diagramas temporales con importes y formatos condicionales

Otra necesidad que me puede surgir es tener que distribuir temporalmente ciertas cantidades que pueden corresponder a Importes de ventas o de costes, valores que necesitaré temporalizar y totalizar para cada de las unidades temporales de mi proyecto, y que posteriormente podré utilizar para obtener los flujos de caja estáticos o incluso dinámicos descontados al tipo de interés y de inflación previsto.

La manera de resolverlo también es muy sencilla, pero de una forma completamente diferente, y será trabajando con las celdas de Excel.

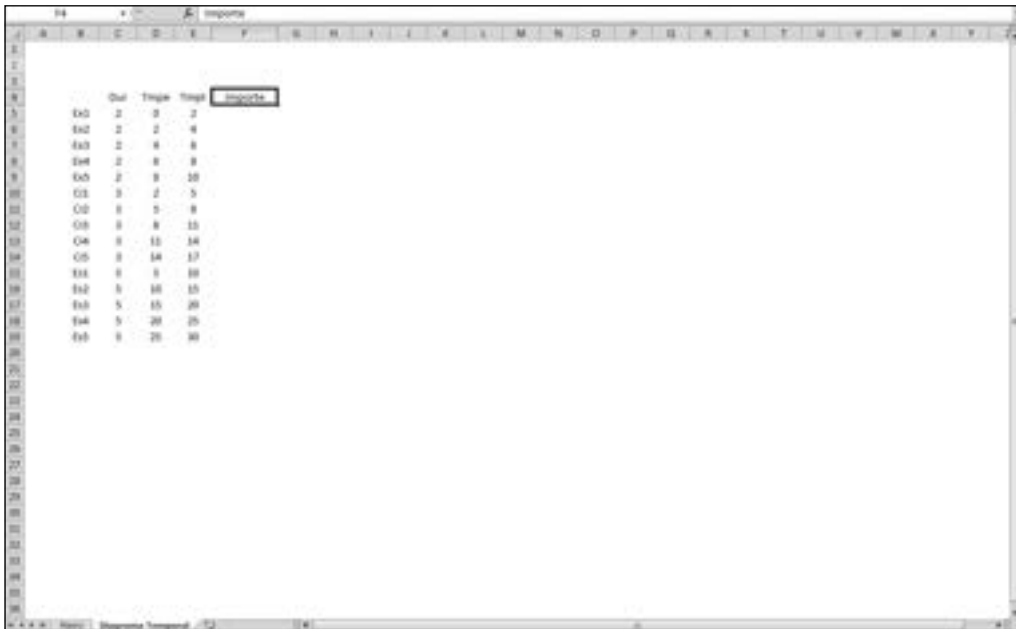
Lo primero que tendremos que hacer es insertar una nueva hoja de cálculo a la que llamaré "Diagrama Temporal", y en el caso de que dispongas de hojas sin utilizar en el libro de Excel, como por ejemplo la Hoja2, puedes sencillamente cambiarle el nombre por "Diagrama Temporal", que como recordarás se hace mediante el menú contextual, que se activa con el botón derecho del ratón una vez situado encima de la pestaña de la Hoja.

Ahora nos situaremos en la celda B5, e introduciremos una fórmula de la siguiente forma:

1. Escribiremos: **=** (el signo igual le indica a Excel que a continuación vamos a introducir una fórmula).
2. Haremos clic sobre la pestaña de la hoja llamada "Matriz" y seleccionar la celda B5 de esa hoja.
3. Pulsaremos la tecla **Enter** o **Intro**.
4. Una vez en este punto, el texto correspondiente a la fórmula de la celda B5 de la hoja "Diagrama temporal" debe ser: **"=Matriz!B5"**, y tendrá que aparecer el texto "Exc 1" como valor de esa misma celda.
5. Repetiremos la operación hasta la celda B19. En vez de repetir la fórmula cada vez, puedes hacerlo más rápido mediante **Copiar>Pegar** de la celda B5 en el rango B6:B19, o más rápido aún con la opción de **Arrastrar>Copiar celdas**.

La primera columna del encabezamiento de nuestro diagrama temporal ya está finalizado, ahora tendremos que repetir los pasos anteriores con las columnas correspondientes a las duraciones (Dur), la de los tiempos más pronto de empezar (Tmpe) así como de los tiempos más pronto de terminar (Tmpt).

Una vez finalizada la primera parte del encabezamiento, tan solo me restará por añadir la columna correspondiente a los importes que deseo temporalizar, debiendo obtener como resultado esta estructura previa:



	Qui	Tiempo	Importe
Ex0	2	8	2
Ex2	2	2	4
Ex3	2	8	8
Ex4	2	8	8
Ex5	2	8	10
Ex1	3	2	5
Ex0	8	5	8
Ex8	8	8	15
Ex4	8	11	14
Ex5	8	14	17
Ex1	8	8	10
Ex2	5	10	15
Ex3	5	15	20
Ex4	5	20	25
Ex5	8	20	30

Figura 3.40. Hoja con encabezado del diagrama temporal.

El siguiente paso consistirá el crear los días de calendario:

1. Lo primero será facilitarnos el trabajo posterior inmovilizando los paneles, para eso nos situaremos en la celda G5 y con **Vista>Inmovilizar>Inmovilizar paneles** los dejaremos fijos, pudiéndonos ya desplazar con libertad por toda la hoja sin perder las referencia principales.
2. Nos situaremos en la celda G4 y luego escribiremos **1** como valor de la celda.
3. Ubicándonos en la celda H4, escribiremos la siguiente fórmula **=G4+1**, que podremos escribir directamente, bien mediante el cursor e **Intro** o con el ratón y haciendo clic con el botón izquierdo.
4. Mediante **Copiar>Pegar** o **Arrastrar>Copiar** celdas copiaremos la fórmula en tantas columnas como deseemos, y que en nuestro caso será en un mínimo de 30 columnas (que es la duración del proyecto), es decir hasta la columna AJ, aunque podemos elegir más y hacerlo más grande si lo deseamos, por ejemplo hasta la columna AO.

A continuación vamos a escribir los siguientes valores como importes de las tareas del proyecto:

Tarea	Celda	Importe a escribir
Ex 1	F5	15
Ex 2	F6	15
Ex 3	F7	15
Ex 4	F8	15
Ex 5	F9	15
Ci 1	F10	25
Ci 2	F11	25
Ci 3	F12	25
Ci 4	F13	25
Ci 5	F14	25
St 1	F15	35
St 2	F16	35
St 3	F17	35
St 4	F18	35
St 5	F19	35

Una vez escritos los anteriores importes, tendremos que ubicarlos no solo en el instante temporal que le corresponde, sino que además ha de ser la parte proporcional a la duración de la tarea.

Como parece un poco complejo de entender, lo mejor será verlo en dos etapas, una primera que consistirá en establecer la parte proporcional, y otra que será la de la ubicación.

La primera es la más sencilla, la parte proporcional del importe correspondiente a una unidad temporal no será más que el importe dividido por la duración de la tarea, así, 300 euros de importe entre 10 días resultará a 30 euros diarios.

La segunda es la más compleja porque hemos de hacer uso de la diferencia de concepto entre instante temporal y unidad temporal, debiendo establecer si una determinada unidad temporal se encuentra entre los tiempos de una tarea expresados en instantes temporales.

Para entenderlo mejor veamos la siguiente figura correspondiente a una unidad temporal t . Esta unidad temporal t se encontrará entre los instantes $t-1$ y t :

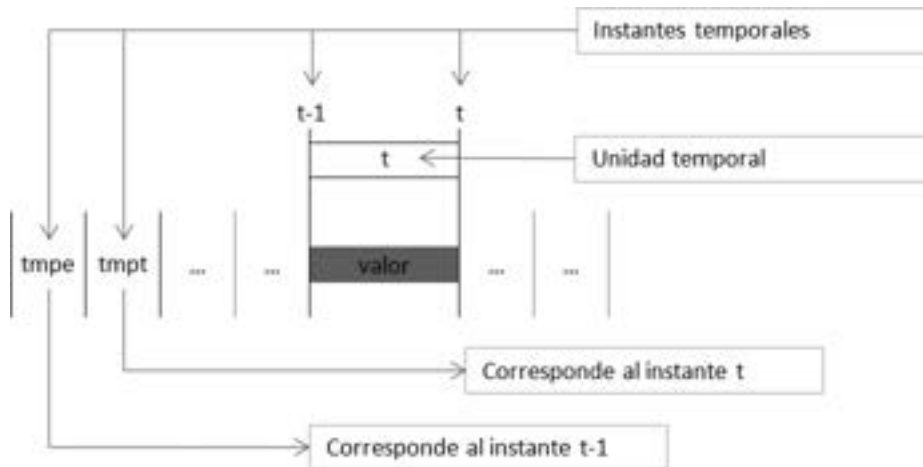


Figura 3.41. Instantes vs unidades temporales.

Teniendo claros los anteriores conceptos, establecer si en una determinada celda hemos de ubicar su importe proporcional correspondiente es muy sencillo, tan solo deberá cumplir dos condiciones simultáneamente:

1. El instante del tiempo más pronto de terminar de la tarea tendrá que ser menor o igual que el instante $t-1$, que corresponde a la unidad temporal t ($Tmpe \leq t-1$).
2. El instante del tiempo más pronto de terminar tendrá que ser mayor o igual que el instante t , el cual corresponde a la unidad temporal t ($Tmpt \geq t$).

Darle esa orden a Excel es muy sencillo, porque existe una fórmula diseñada para tal fin:

SI (prueba lógica; [valor si verdadero]; [valor si falso])

El problema nos surge porque en nuestro caso se han de cumplir dos condiciones de forma simultánea, y que vamos a llamar condición 1 y condición 2, pero tan solo tendremos que anidar la fórmula anterior de la siguiente forma:

SI (condición 1; SI (condición 2; [valor si verdadero]; [valor si falso]); [valor si falso])

La condición 1 será $T_{mpe} \leq t-1$, la condición 1 será $T_{mpt} \geq t$, [valor si verdadero] será Importe dividido por la duración y [valor si falso] un valor nulo que representaremos mediante unas dobles comillas vacías (sin nada en su interior, ni espacios en blanco). La fórmula completa para la celda G5 será:

=SI (\$D5<=G\$4-1; SI (\$E5>=G\$4; \$F5/\$C5;"");")

Es muy importante que se respeten los símbolos \$ que me indican que la referencia a la fila o columna se ha de realizar de manera absoluta y no relativa, porque de esta forma tan solo necesitaremos formular una celda, ya que posteriormente la copiaremos a lo largo de todas las celdas del diagrama, quedándonos la estructura siguiente:



Figura 3.42. Estructura básica del diagrama temporal.

Ahora tan solo me quedará para terminar la primera parte del diagrama temporal resaltar aquellas celdas que contienen algún valor, para lo que, seleccionando las celdas correspondientes al diagrama temporal G5:AO19, y tal como hemos aprendido anteriormente, aplicaremos el siguiente formato condicional:

1. Formato condicional>Nueva regla.
2. Dar formato únicamente a las celdas con>Sin espacios en blanco.
3. Aplicamos el formato que deseamos.

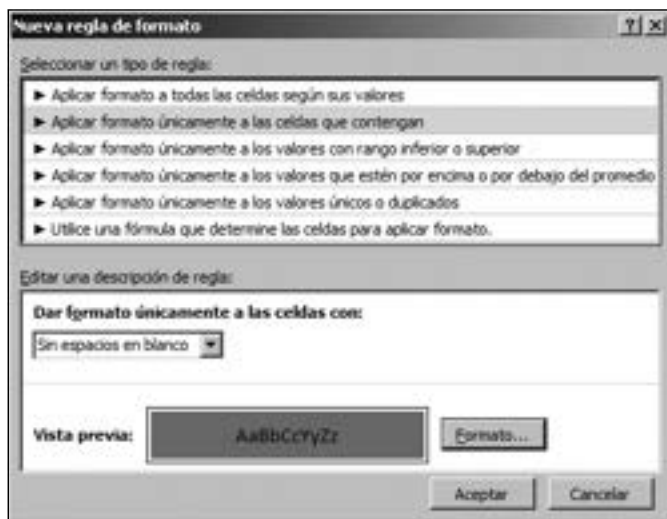


Figura 3.43. Formato condicional del diagrama temporal.

En este momento podemos considerar que el diagrama temporal está finalizado, restando tan solo totalizar para cada periodo los importes proporcionales de las tareas que se ejecutan en él, para lo que añadiremos una nueva fila, en este caso la 20, en la escribiremos la siguiente fórmula (celda G20):

=SUMA (G5:G19), fórmula que posteriormente copiaremos a lo largo de todas las columnas del calendario.

Terminaremos añadiendo una nueva fila (la 21) donde reflejaremos los valores acumulados hasta un determinado periodo, de tal manera que el valor de sea celda será la suma del importe del periodo correspondiente y el acumulado anterior.

Tan solo he de tener la precaución de que la primera columna, como no tiene acumulado anterior, su valor acumulado será el propio valor del periodo, y que en nuestro caso para el periodo 1 correspondiente a la celda G21, la fórmula será: =G20, y para el periodo 2 o celda H21 la fórmula será =H20+G21, y que copiare a lo largo de todo el calendario.

La hoja con el diagrama temporal ya completamente finalizada tendrá el aspecto que muestra la figura 3.44.

El diagrama temporal que vemos en la figura 3.44, lo podemos mejorar incluyendo características que lo hagan parecer más profesional y sofisticado, y la primera que vamos a incorporar será la representación de las holguras.



Tal y como hicimos anteriormente, formularé las celdas para que contengan los valores de los tiempos más tarde de terminar, valores que ya fueron calculados previamente y que se encuentran en la hoja Matriz.

No hace falta reformular nuevamente las celdas de la hoja al insertar filas o columnas, porque Excel ya se encarga de reajustarlas automáticamente.

Seleccionando ahora las celdas correspondientes al ámbito del diagrama temporal, añadiremos un nuevo formato condicional, pero ahora lo vamos

a hacer de una forma diferente para que el lector pueda ver las diferentes posibilidades que le ofrece Excel para sus proyectos:

1. Seleccionaremos las celdas H5:AP19.
2. Formato condicional>Nueva regla.
3. Dar formato a los valores donde esta fórmula sea verdadera:
4. Y escribiremos: **=Y (\$D5<=H\$4-1; \$F5>=H\$4)**; mediante la aplicación de la función Y, evaluaremos las condiciones del interior del paréntesis, obteniendo VERDADERO si ambas expresiones son verdaderas y FALSO en caso contrario.

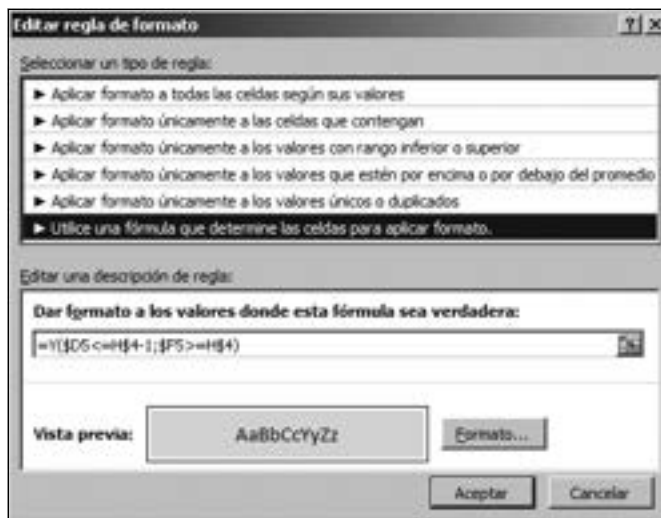


Figura 3.45. Formato condicional para las holguras.

Y nos quedará este nuevo diagrama temporal en donde se ven perfectamente reflejadas las holguras totales de cada una de las tareas (véase la figura 3.46).

Otro elemento que a priori parece imprescindible para mi proyecto es un gráfico en el que se visualicen los flujos de caja por periodos y acumulados. La manera de hacerlo es tremendamente sencilla:

1. Seleccionaremos los valores del flujo de caja por periodo y acumulados H20:AP21.
2. Insertar>Columna>Columna agrupada.

Aparecerá el siguiente gráfico en nuestra hoja (véase la figura 3.47).

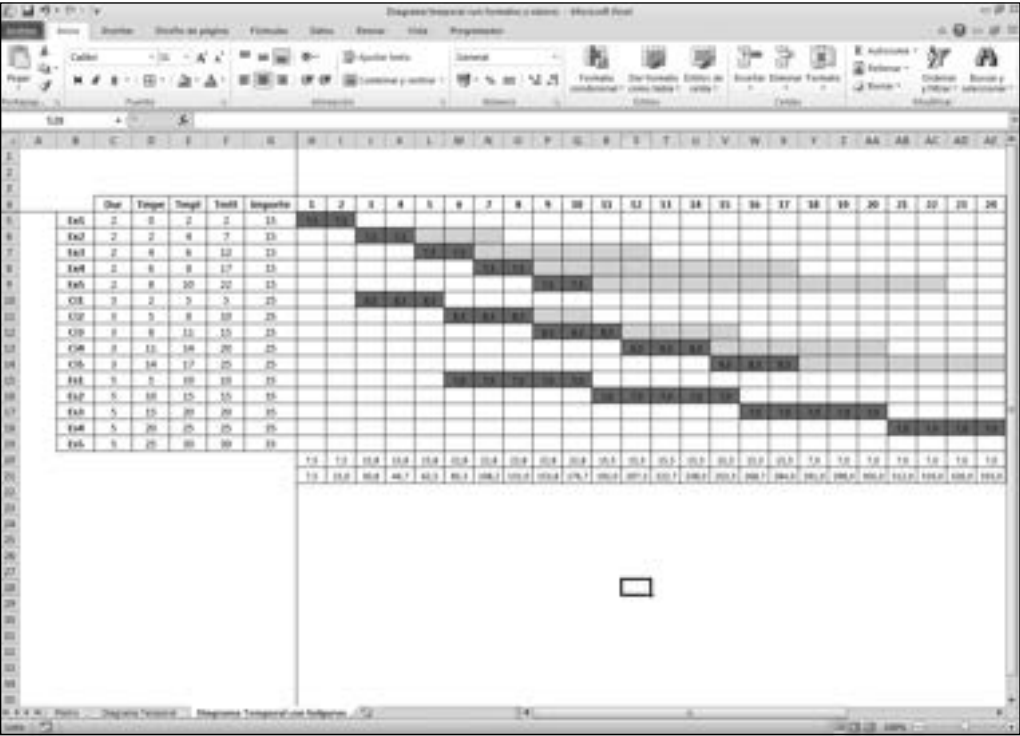


Figura 3.46. Diagrama temporal con holguras.

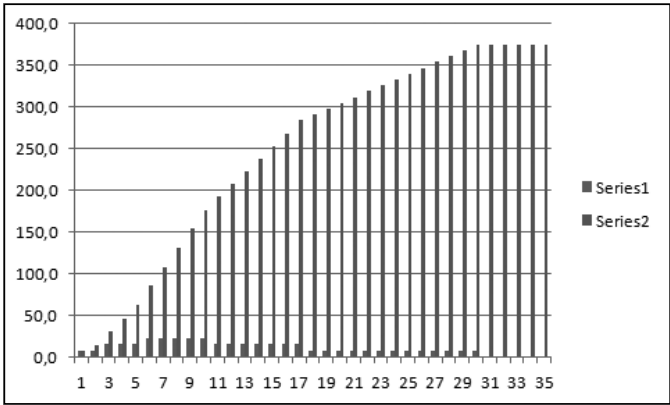


Figura 3.47. Gráfico previo de Flujos de caja.

Aunque los valores que me ofrece el gráfico son totalmente correctos, la lectura es bastante deficiente, debiendo de hacer unos cuantos cambios. Lo primero que haré será cambiar el nombre de las series por lo que realmente representan: **Por periodos** y **A origen**, y que recordaremos

hacíamos seleccionando las leyendas y con el menú contextual **Seleccionar datos>Editar**. Pero lo que realmente va a hacer cambiar el gráfico va a ser cambiar el tipo de gráfico de los valores a origen por un gráfico de líneas, seleccionando la serie a origen y con el menú contextual **Cambiar tipo de gráfico de series>Línea**. La escala no parece que cumpla con nuestras exigencias, unos valores, los "A origen" se ven perfectamente, pero los valores "Por periodos", parecen más bien manchitas, algo que solucionaremos seleccionando la serie "A origen" y con el menú contextual **Dar formato a serie de datos>Opciones de serie>Eje secundario**.

Si muevo el gráfico a una hoja y le añado un título obtendré:

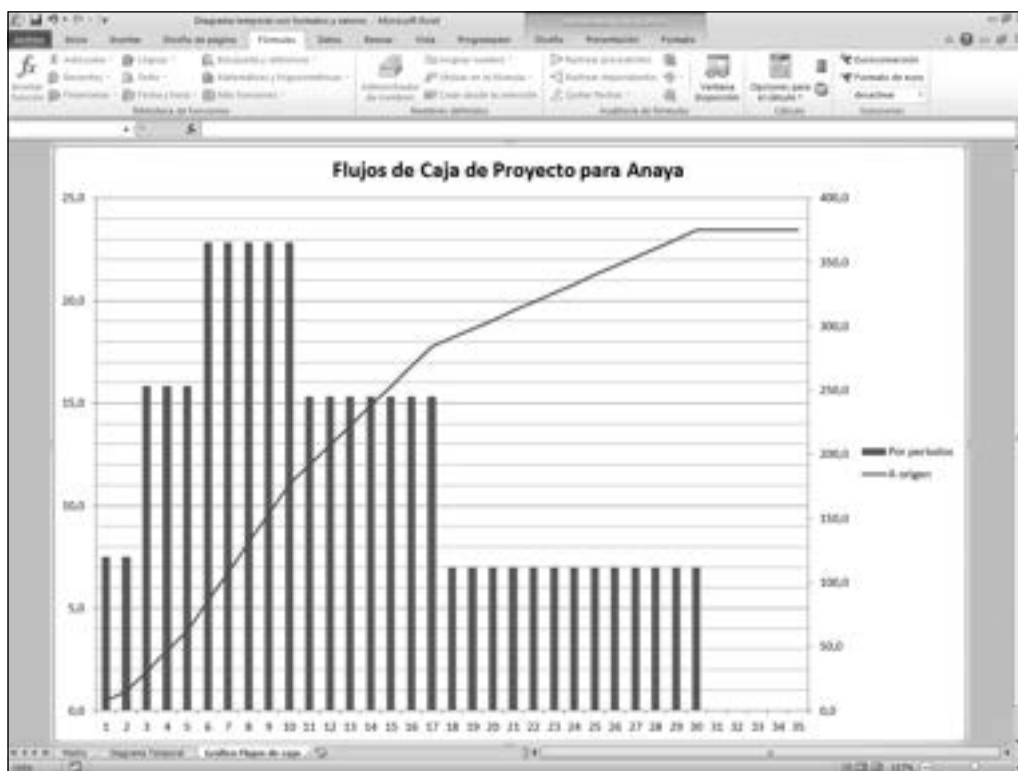


Figura 3.48. Gráfico de Flujos de caja.

NOTA

La hoja de cálculo completa se encuentra en Cap_03\Diagrama temporal con formatos y valores, de los ejemplos del libro.

3.6.3. Diagramas temporales con fechas de calendario

Para terminar con este capítulo, vamos a ver cómo afrontar el problema de cómo incorporar fechas reales de calendario a nuestro diagrama temporal. Hasta ahora hemos visto cómo calcular las fechas de comienzo de terminación de cada una de las tareas de nuestro proyecto, y cómo representarlas en un diagrama temporal, pero los proyectos reales están basados en calendarios, con fechas reales de comienzo, días festivos e incluso esos entrañables días de puente.

Lo primero que tendré que hacer para trabajar con calendarios es establecer la fecha de inicio del proyecto así como la fecha en la que nos encontramos actualmente, pero también deberé de establecer las fechas no laborables y festivas del proyecto, es decir, determinar los festivos nacionales, provinciales y locales, los días de puente que aún no siendo festivos no se trabajará y que afectan al proyecto, así como la configuración de un fin de semana tipo del proyecto.

Este nuevo diagrama temporal, utiliza los formatos condicionales que hemos visto hasta ahora y una fórmula de Excel que me va a permitir determinar las fechas que tienen lugar después de una determinada cantidad de días laborables. Esta fórmula es:

`DIA.LAB.INTL (fecha inicial; días_lab; [fin_de_semana]; [festivos])`

Y tiene los siguientes argumentos (valor que proporciona información a una acción, un evento, un método, una propiedad, una función o bien un procedimiento.):

- `fecha_inicial`: Obligatorio. Es La fecha inicial, y que en nuestro caso será el tiempo más pronto de empezar de la tarea.
- `días_lab`: Obligatorio. Es el número de días laborables antes o después de la `fecha_inicial`. Un valor positivo nos dará como resultado una fecha futura; un valor negativo nos proporciona una fecha pasada. En nuestro caso será la duración de la tarea.
- `fin_de_semana`: Opcional. Indica los días de la semana que corresponden a días de la semana y no se consideran días laborables. El `fin_de_semana` puede expresarse como número, pero nosotros vamos a utilizar la representación de cadena. Los valores de las cadenas de `fin_de_semana` tendrán siete caracteres de longitud y cada carácter de la cadena representará un día de la semana comenzando por el

lunes. Los días laborables serán representados con 1 y los no laborables con 0.

- festivos: Opcional. Un conjunto de fechas que se deben excluir del calendario de días laborables.

Para poder trabajar adecuadamente, necesitaré crear una hoja a la que le pondré el nombre de "Datos Calendario", donde estableceré la configuración del fin de semana tipo, un rango de celdas donde almacenaré los días festivos y la fecha de inicio del proyecto. En esa hoja además incluiré la fecha actual y haré que Excel me presente automáticamente la fecha de finalización del proyecto. Esa hoja tendrá la siguiente forma:

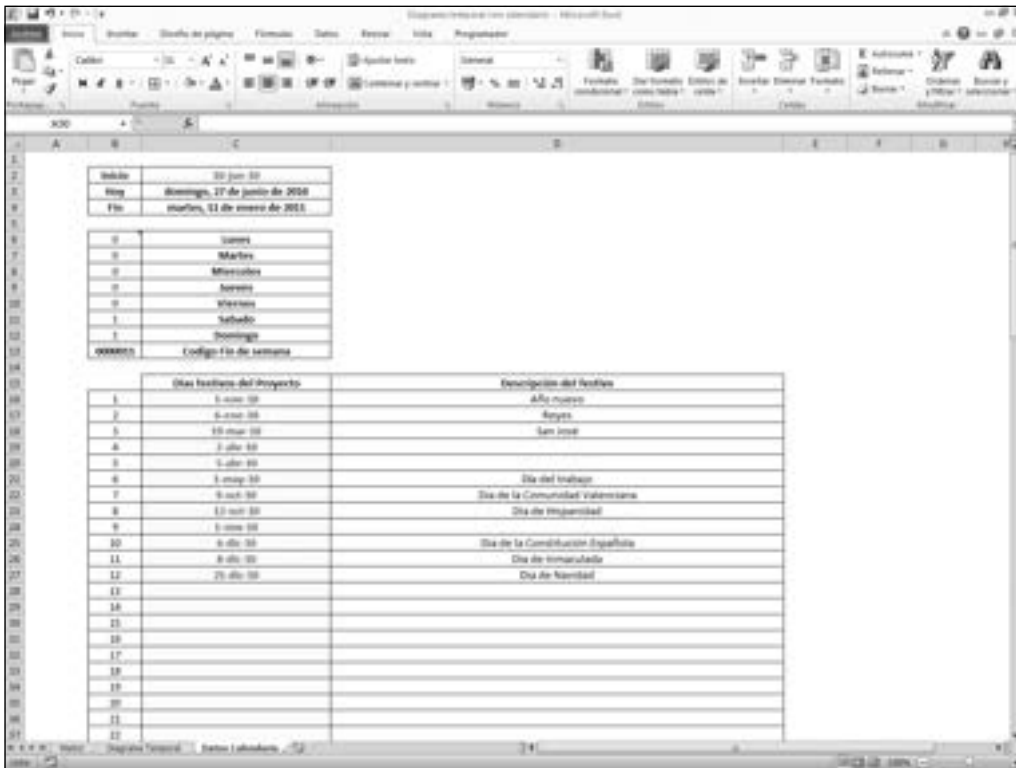


Figura 3.49. Hoja "Datos de calendario".

Donde cada uno de sus elementos importante son los que se describen a continuación:

- Inicio o Fecha de Inicio del Proyecto: Corresponde a la celda C2, y en ella escribiré la fecha prevista para el comienzo de los trabajos.

Deberá estar en formato de fecha corto con la estructura DD/MM/AAAA, donde cada una de las letras corresponde al dígito de día, mes o año. El 10 de junio de 2010 lo escribiré: **10/06/2010**. La celda deberá tener formato fecha y le asignaré el nombre "Fecha_Inicio".

- Hoy o Fecha Actual: Corresponde a la celda C3, y en ella escribiré: =HOY (), para que Excel me presente automáticamente el número de serie de la fecha actual. La celda deberá tener formato fecha larga y le asignaré el nombre "Fecha_Hoy".
- Fin o Fecha de finalización del Proyecto: Corresponde a la celda C4, y su valor vendrá determinado por una fórmula que establezca el valor máximo de los tiempos más tarde de terminar de todas las tareas, y concretamente será: =MAX ('Diagrama Temporal'! D5:D19), pero todavía no escribas nada, espera a que tengamos comenzada la hoja "Diagrama temporal".
- Código fin de semana: Corresponde a la celda B13, que contiene la fórmula =CONCATENAR (B6; B7; B8; B9; B10; B11; B12), que lo que hace es convertir los textos que introduciré en las celdas B6:B12 en un solo (recordar que solo puede adoptar valores de 1 y 0, correspondiendo el 1 a un día no laborable y el 0 a un día laborable. La celda deberá tener formato texto y le asignaré el nombre "Cod_Fin_D".
- Festivos: Es el rango de celdas C16:C115, a que le habré asignado el nombre "festivos". En este rango escribiré los festivos que afecten al ámbito del proyecto. No es necesario que los festivos sigan ningún orden, pudiendo este ser totalmente arbitrario.

Para crear nuestro diagrama temporal con fechas de calendario, empezaremos insertando la hoja "Diagrama temporal" que nos permita incorporar las funciones y conocimientos que hemos visto previamente. La hoja deberá tener el aspecto que muestra la figura 3.50.

Para darle forma a la hoja comenzaré formulando las fechas de inicio y final de las tareas, así, empezando por la celda C5, escribiré: =DIA.LAB.INTL (Fecha_Inicio; Matriz! D5; Cod_Fin_D; Festivos), y en la celda D5: =DIA.LAB.INTL (Fecha_Inicio; Matriz! E5-1; Cod_Fin_D; Festivos).

El resultado que obtendré será el que muestra la figura 3.51.

No, no están mal las formulas, son correctas aunque no en la forma que deseo, porque me están mostrando el número de serie que representa la fecha. Para que tome el formato adecuado seleccionaré el rango y con el menú contextual: **Formato de celdas>Fecha**, y seleccionaré aquel formato de entre los disponibles que más se ajuste a mis gustos o necesidades.

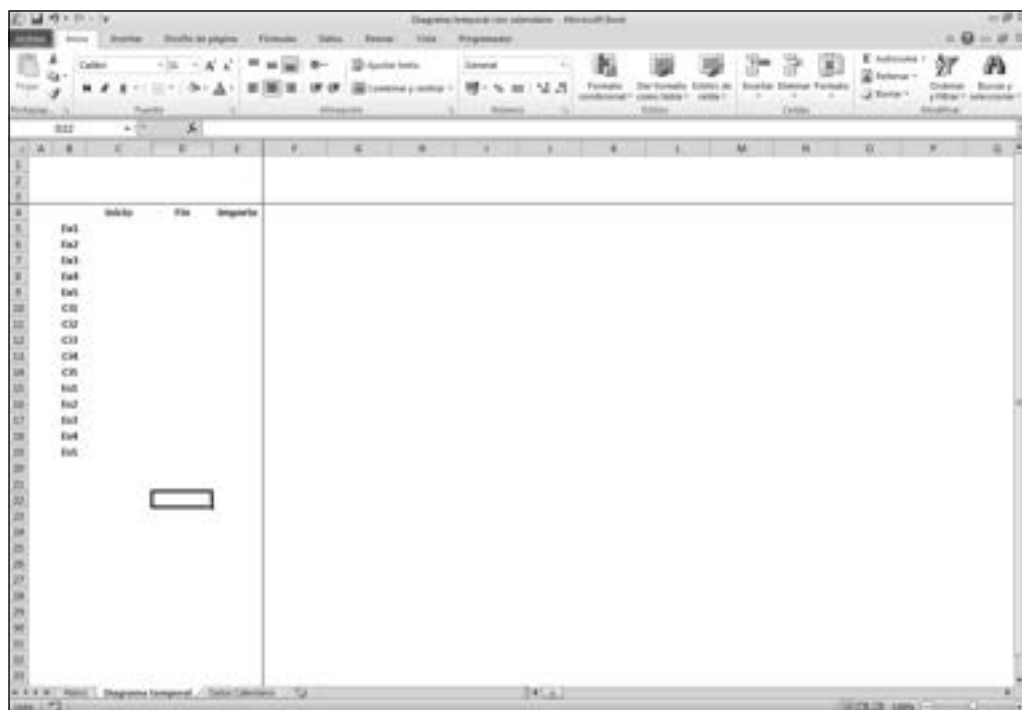


Figura 3.50. Hoja "Diagrama temporal" con fechas en blanco.

	A	B	C	D	E
1					
2					
3					
4			Inicio	Fin	Importe
5		Ex1	40339	40352	15,00
6		Ex2	40353	40366	15,00
7		Ex3	40367	40380	15,00
8		Ex4	40381	40394	15,00
9		Ex5	40395	40408	15,00
10		C11	40353	40373	25,00
11		C12	40374	40394	25,00
12		C13	40395	40415	25,00
13		C14	40416	40436	25,00
14		C15	40437	40457	25,00
15		Es1	40374	40408	35,00
16		Es2	40409	40443	35,00
17		Es3	40444	40479	35,00
18		Es4	40480	40515	35,00
19		Es5	40519	40554	35,00
20					

Figura 3.51. Celdas con números de serie de fechas.

Ahora ya puedes regresar a la hoja "Datos calendario" y entonces escribir la fórmula que nos quedó pendiente en la celda C4 correspondiente al Fin o Fecha de finalización del Proyecto.

Ya tengo las fechas de comienzo y finalización de las tareas en la forma correcta, ahora vamos a continuar con el calendario que aparecerá en mi calendario, para lo que haré uso de las filas 3 y 4. Se podrá apreciar que la fila 3 es un poco más alta, pero no es la única diferencia. A la fila 3 le he de dar un alto adecuado para poder alojar en su interior las fechas, y que cambiaré seleccionando la fila, y con el menú contextual Alto de fila... y escribir 45.

Posteriormente volveré a activar el menú contextual y con Formato de Celdas>Alineación y escribir 90 grados (también lo puedo cambiar con la rueda de orientación), y además le cambiaré el formato de general a fecha corta como se ha visto antes.

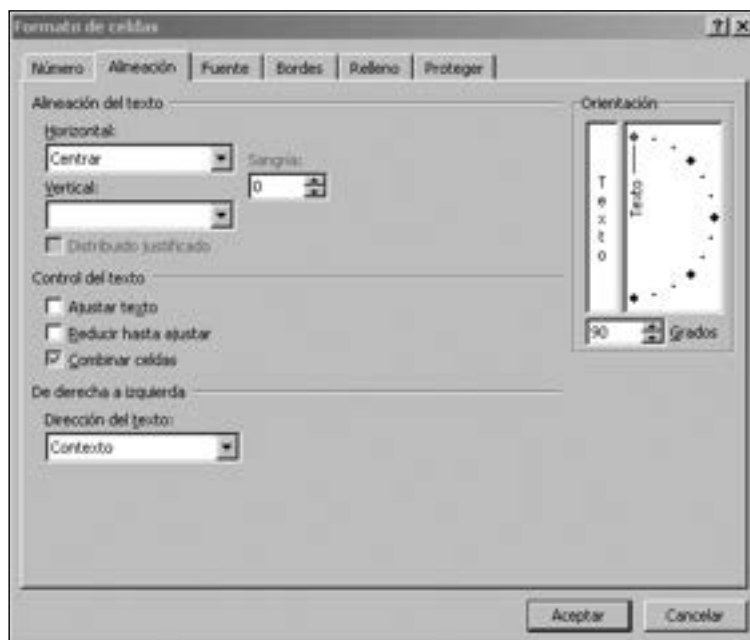


Figura 3.52. Cambio de Orientación de celdas calendario.

Ahora procederé a formular las filas tres y cuatro de las columnas F y G, escribiendo **=Fecha_Inicio** en la celda F3 y **1** en la celda F4; en la celda G3 escribiré **=F3+1**, y en la celda G4 **=F4+1**.

Finalizado esto, copiaré y pegaré las celdas G3:G4 a lo largo de todo el ámbito del calendario y obtendré:

Date		Time		Day		Month		Year		Hour		Minute		Second		Millisecond		Microsecond		Nanosecond		Picosecond		Femtosecond		Attosecond		Zeptosecond		Yoctosecond		Rontosecond		Quectosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextosecond		Sextose	
------	--	------	--	-----	--	-------	--	------	--	------	--	--------	--	--------	--	-------------	--	-------------	--	------------	--	------------	--	-------------	--	------------	--	-------------	--	-------------	--	-------------	--	--------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	-------------	--	---------	--

Figura 3.53. Celdas calendario con formato.

NOTA

Como se puede apreciar, debajo de cada día del calendario aparece el número de orden de la fecha en el calendario natural, en el caso de que se desee el número de orden de la fecha en el calendario laboral, cambiaremos la fórmula de la celda G4 por =DIAS.LAB (Fecha_Inicio; G3; Festivos).

Ya solo me quedará formular y dar formato a las celdas del diagrama. La fórmula la escribiremos en la celda F5 y será:

=SI(DIAS.LAB.INTL(F\$3;F\$3;Cod_Fin_D;Festivos)=1;SI(\$C5<=F\$3;SI(\$D5>=F\$3;\$E5/Matriz!\$C5;"");"."))

Como se ve, la fórmula es muy parecida a las vistas anteriormente, con la diferencia de que se ha añadido una función de evaluación condicional al principio, con la que comprobamos que la fecha es laborable, para que en caso contrario Excel escriba un punto en la celda en lugar del valor del importe proporcional, indicando que es un día festivo o de fin de semana. Como los signos de \$ lo he colocado correctamente, puedo copiar y pegar esta fórmula por todo el rango de celdas, y tendré:

- Inicio>Formato condicional>Nueva regla.
- Dar formato a los valores donde esta fórmula sea verdadero.
- ='Datos Calendario'!\$C\$3=F\$3

Con todos los formatos ya finalizados y las filas de importes "Por periodo" y "A origen" incluidas, me aparecerá un diagrama temporal totalmente profesional, donde no solo se reflejarán correctamente los fines de semana según el formato que haya establecido, sino que además tendrá en cuenta los días no laborables del proyecto.

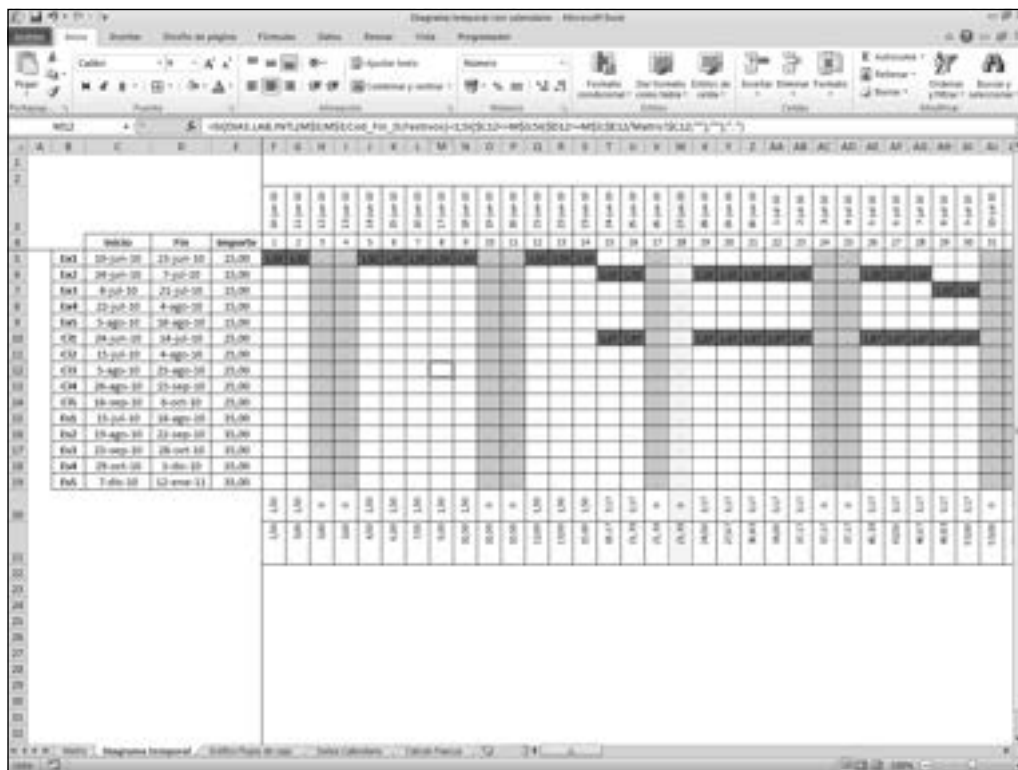


Figura 3.55. Diagrama temporal con fechas de calendario.

NOTA

Las hojas de cálculo completas se encuentran en Cap_03\Diagrama temporal con calendario natural y Cap_03\Diagrama temporal con calendario laboral, de los ejemplos del libro.

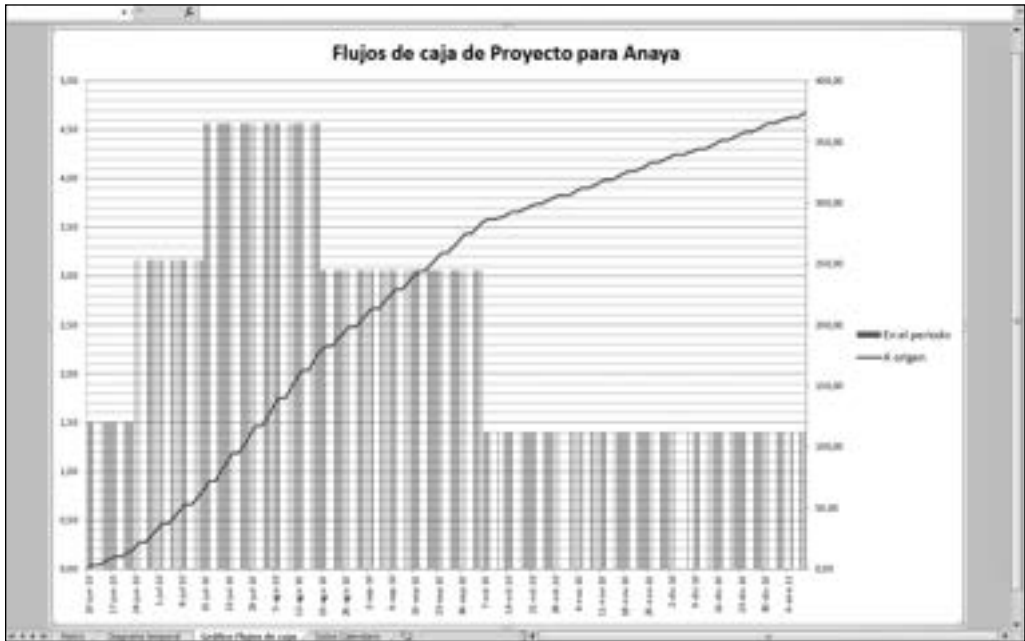


Figura 3.56. Gráfico de flujos de caja con fechas de calendario.

3.6.3.1. Anexo: El cálculo de la fecha de Pascua

Cuando acometemos un proyecto, una de las cosas que más dolores de cabeza nos produce y posiblemente la primera pregunta que nos hacemos es ésta:

¿Cuándo cae la fecha de Pascua este año?

Pero la pregunta que deberíamos de hacernos para poder contestar es:

¿Por qué la Semana Santa se mueve en el calendario, no debería de ser una fecha fija como la Navidad?

No hace falta decir que la respuesta tiene motivaciones religiosas, pero lo más curioso, y que muy poca gente conoce es que tiene motivaciones astronómicas, y más concretamente, que está basada en los ciclos lunares.

La fecha del Pascua o del domingo de resurrección, se obtendrá como *"el domingo posterior a la luna llena que sigue al primer equinoccio del año"*, pero no podía ser *"tan sencillo"*, además se ha de tener en cuenta que, *"ha de celebrarse en domingo, no ha de coincidir nunca con la Pascua judía, y nunca podrá celebrarse dos veces en el mismo año"*. Existen diferentes

formas de calcularla, pero la única fecha oficial (según estableció el concilio de Nicea en el año 325) es la calculada por la iglesia de Alejandría y comunicada por Roma, porque existían discrepancias entre las fechas del equinoccio de calendario calculadas por Roma y por Alejandría (el equinoccio se calcula según el calendario gregoriano y no tiene porque coincidir con el equinoccio astronómico).

Muchos matemáticos han desarrollado formulas para calcularla (aunque no están reconocidas como "oficiales" por la Iglesia Católica), y de entre todas, la más conocida es la de Gauss, que para fechas entre el año 1900 y el 2099 será:

$$a = \text{AÑO Mod } 19 \text{ (resto de la división entre el AÑO deseado y 19).}$$

$$b = \text{AÑO Mod } 4$$

$$c = \text{AÑO Mod } 7$$

$$d = (19a + 24) \text{ Mod } 30$$

$$e = (2b + 4c + 6d + 5) \text{ Mod } 7$$

Si $d + e < 10$, entonces la Pascua en AÑO será el día $(d + e + 22)$ de marzo, y en caso contrario será el día $(d + e - 9)$ de abril.

Existen dos excepciones a tener en cuenta que son si obtenemos como resultado el 26 o el 25 de Abril, en cuyo caso será el domingo inmediatamente anterior, es decir, el 19 o 18 de Abril.

Se ha incluido una hoja en el libro "Diagrama temporal con calendario" llamada "Cálculo Pascua" para facilitar estas operaciones.

La función de Excel que calcula el modulo o el resto de una división es =RESTO (número; núm_divisor).

4

El Control del Proyecto

4.1. Introducción al control del proyecto

Tradicionalmente, el control de proyectos se ha entendido como el análisis de la evolución de los plazos parciales de un proyecto a lo largo de su ejecución, con el objetivo de adoptar medidas que corrijan las posibles desviaciones con respecto a las previsiones realizadas.

En este enfoque tradicional, los valores contemplados correspondían tan solo a las fechas, y más concretamente a la diferencia existente entre las fechas de previstas y las fechas reales de las tareas o plazos parciales, obteniendo como resultado del control un informe que consistía en un valor que correspondía a un posible retraso o adelanto en la ejecución del proyecto sobre el plazo previsto en un instante dado.

Pero este enfoque tiene un problema, y es que no se tiene en cuenta la diferencia entre las unidades producidas y las previstas en un determinado

periodo de tiempo, y por supuesto tampoco son tenidos en cuenta los costes asociados a esa determinada producción, ni la evolución de estos a lo largo del tiempo con respecto a las previsiones.

No podemos afirmar que una tarea se haya retrasado si se ha aumentado su producción un cincuenta por ciento, demorando su finalización solo un 25 por ciento del plazo previsto, como tampoco podría considerarse un sobrecoste, que para ese aumento del 50 por ciento de producción haya necesitado incrementar mis costes un 40 por ciento del importe previsto, no siendo nada intuitivo determinar la bondad de mi gestión cuando esas diferencias se producen con decenas y cientos de tareas.



Figura 4.1. Gráfico de avance de proyecto.

Actualmente, la tendencia es controlar la producción, los plazos y los costes de una manera conjunta y simultánea mediante una novedosa técnica denominada Análisis del Valor Ganado o *Earned Value Analysis (EVA)*, que contempla esos tres aspectos de una forma asombrosamente sencilla y armoniosa, y que aunque a priori pueda parecer un poco compleja por los conceptos desconocidos que tiene aparejados, pronto nos hará sentir cómodos por las innumerables ventajas que presenta.

Aunque el Valor ganado es un avance importantísimo en la gestión de proyectos, estos aún siguen siendo informados en parámetros de plazo exclusivamente, pero es de destacar el notable aumento del número de empresas que empiezan a hacer uso del Valor Ganado, especialmente en el ámbito de las empresas con proyectos basados en procesos productivos, y muy especialmente las empresas constructoras.

4.2. El control tradicional de Proyectos

Tal y como se ha comentado anteriormente, este sistema de control es el más sencillo, y tan solo me ofrecerá información sobre los adelantos y retrasos en la ejecución del proyecto, pero presenta como ventaja que también es el más sencillo y rápido de implementar.

Vamos a exponer dos formas de afrontar este sistema de control con el uso de la hoja de cálculo Excel, uno rápido y sencillo, en el que de una manera visual podré analizar el avance en la ejecución de cada una de las tareas con respecto a la fecha actual pero sin ofrecer información cuantitativa sobre los adelantos y retrasos en el proyecto, y otro, en el que mediante la incorporación de una nueva matriz podré calcular los días de adelanto o retraso del proyecto respecto de la fecha prevista.

Ninguno de los dos sistemas es excluyente del otro, son dos maneras diferentes de representar lo mismo, pero con ciertas peculiaridades que pueden hacer que uno sea preferible al otro en determinadas circunstancias.

Mediante el primer sistema necesitaremos establecer para cada control, el porcentaje de tarea ejecutada hasta ese momento, visualizando en un solo informe como deberían de encontrarse los trabajos y como se encuentran realmente.

Gracias al segundo sistema, podremos determinar cuantitativamente el retraso o adelanto de un proyecto respecto de las previsiones iniciales. Dejaremos que Excel establezca automáticamente el día de control como el día actual y nosotros tan solo tendremos que indicar los días pendientes de ejecutar de cada una de las tareas. Este método presenta la ventaja de que podemos reprogramar nuevamente la matriz de control de las tareas en el caso de que alguna de las dependencias no haya sido establecida por criterios estrictos de dependencias (*hard precedence*) sino por otro tipo de condicionantes como la programación en base a la disponibilidad de recursos (*soft dependence*).

4.2.1. El control tradicional por avance

Para poder construir este sistema de control, deberemos o bien insertar una nueva hoja copiando contenido en ella o bien seleccionar la hoja "Diagrama temporal" y con el puntero del ratón sobre la pestaña activar el menú contextual y seleccionar Mover o copiar>(mover al final) y activar la casilla de verificación Crear una copia, procediendo a cambiarle el nombre

con el menú contextual **Cambiar nombre** y escribir **Diagrama de Control** teniendo un hoja exactamente igual a la original sobre la que solo nos quedará hacer una pocas y sencillas modificaciones.



Figura 4.2. Hacer copia de una Hoja.

Las modificaciones que haremos serán:

1. Insertar dos columnas nuevas antes de la columna Importe, una a la que llamaremos "Avance", donde introduciremos el avance real de la tarea en el momento de realizar el control como porcentaje sobre el total, y otra a la que llamaremos "Auxiliar" que utilizaremos exclusivamente como elemento auxiliar para almacenar valores y que posteriormente ocultaremos al carecer de relevancia su información.
2. Formatear y formular las dos nuevas columnas.
3. Reformular y añadir nuevos formatos condicionales a las celdas correspondientes al diagrama temporal para que muestren el avance del proyecto.
4. Eliminar la columna y filas correspondientes al Importe.

Para insertar las dos primeras columnas, seleccionaremos el encabezamiento de la columna correspondiente al Importe (columna E), y ejecutaremos dos veces **Menú contextual>Insertar**, debiendo de aparecer dos nuevas columnas en la hoja "Diagrama de Control" (véase la figura 4.3).

A estas dos nuevas columnas les pondré encabezamiento escribiendo **Avance** y **Auxiliar** en las celdas E5 y F5 respectivamente. Posteriormente seleccionaré el rango E5:E19 correspondiente a la columna "Avance" y le cambiaré el formato mediante **Menú contextual>Formato de Celdas>número>Porcentaje** y las posiciones decimales deseadas, aunque normalmente no se requieren

decimales (cero decimales). Haré lo mismo con los valores de la columna "Auxiliar", pero en este caso le daré el formato de fecha corta.

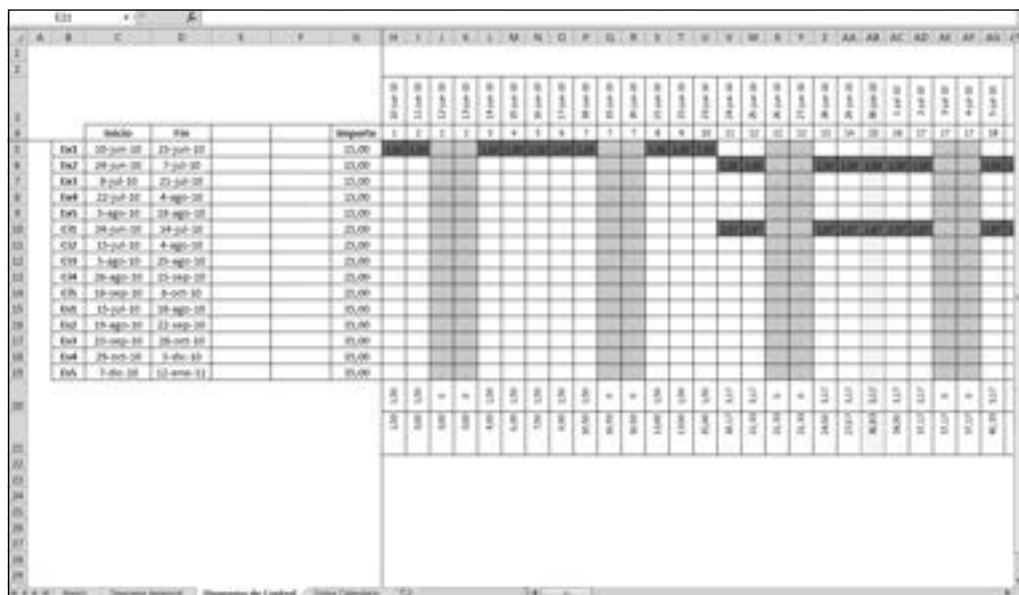


Figura 4.3. Dos columnas nuevas en gráfico de control.

En la columna "Avance", escribiremos el porcentaje de tarea ejecutado hasta el día del control, y en la columna "Auxiliar" Excel deberá mostrar la fecha que correspondería a ese avance, para lo que escribiré y copiare hacia abajo la siguiente fórmula en la celda F5:

```
=SI(E5<>"";DIA.LAB.INTL(Fecha_Inicio;Matriz!D5+Matriz!C5*E5-1;
Cod_Fin_D; festivos) ; "")
```

Las dos nuevas columnas están ya preparadas para enviar su información al diagrama temporal del control, pero antes tendré que preparar este diagrama para que interprete estos datos. La fórmula que escribiré en la celda H% y que después copiaré a lo largo de todo el diagrama será:

```
=SI (DIAS.LAB.INTL (H$3; H$3; Cod_Fin_D; festivos)=1; "", ".")
```

Mucho más sencilla que la fórmula original, porque en este caso tan solo me interesa que aparezca en las celdas los fines de semana y festivos.

Ya estamos en condiciones de eliminar la columna que no necesito correspondiente al "Importe", y que haré situándome con el ratón en el encabezamiento de la columna G y **Menú contextual>Eliminar**; para eliminar las

filas seleccionaré el encabezamiento de las filas correspondientes y operaré exactamente igual.

Mostrar el avance en el diagrama será sencillísimo con los conocimientos disponibles, tendremos que modificar los formatos condicionales para que la previsión aparezca en un tono más claro, y que el avance real se vea con un tono más oscuro.

La previsión la tenemos ya formulada, solo necesitaremos seleccionar la celda G5 y cambiar el color que tenemos a otro más claro mediante Inicio>Formato condicional>Administrar reglas. Para el avance, añadiremos una nueva regla, esta vez seleccionando todo el ámbito del diagrama (sin calendario) y hacer Inicio>Formato Condicional>Nueva regla>Dar formato a los valores donde esta fórmula sea verdadera y escribir =Y (\$C5<=G\$3; \$D5>=G\$3), dando ahora a las celdas el mismo color pero en un tomo más oscuro, teniendo la precaución de que las reglas estén en el siguiente orden, haciendo los cambios oportunos en caso que no sea así:



Figura 4.4. Reglas del formato condicional.

Llegados a este punto podemos considerar que ya hemos finalizado nuestra hoja de control, restando tan solo un último detalle, ocultar la columna "Auxiliar" seleccionando el encabezamiento de la columna F y Menú Contextual>Ocultar quedando:

NOTA

Al ocultar una columna no se borran sus valores, tan solo se ocultan a la vista del usuario. Para volverla a mostrar deberá seleccionar simultáneamente las columnas anterior y posterior a la columna oculta y después hacer Menú contextual>Mostrar.

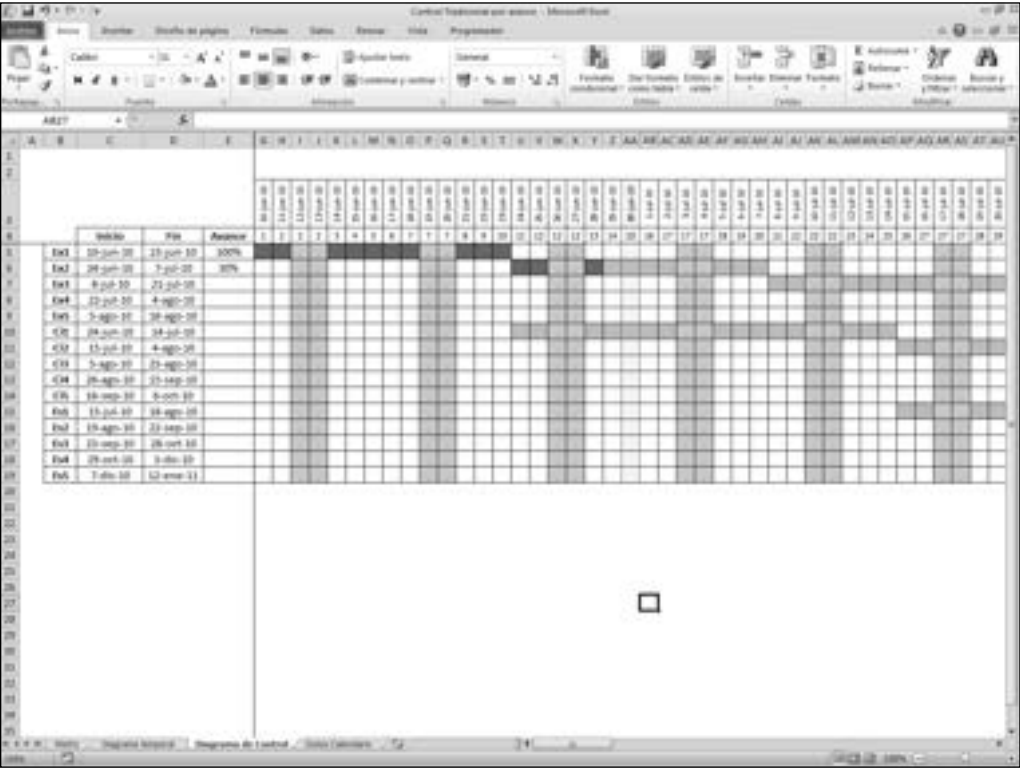


Figura 4.5. Diagrama temporal del control tradicional por avance.

NOTA

La hoja de cálculo completa se encuentra en Cap_04\Control Tradicional por avance, de los ejemplos del libro.

4.2.2. El control tradicional por matriz

Ya dijimos anteriormente que íbamos a explicar dos maneras diferentes de afrontar el control del proyecto por el sistema tradicional. En el sistema por avance, gracias a su representación, podemos apreciar la diferencia entre la planificación prevista y el avance de cada una de las tareas del proyecto, pero no me ofrece información acerca de los tiempos previstos de terminar de las tareas en ejecución, de los tiempos previstos comenzar y terminar de las tareas pendientes, y por supuesto tampoco sabré cual es la

fecha prevista de terminación del proyecto, y como de adelantado o atrasado me encuentro con respecto a las fechas previstas originalmente.

Además surge el problema de que puedo necesitar hacer una reprogramación (algo muy común por cierto) porque había previsto que tan solo disponía de una máquina para realizar diferentes trabajos, y para simplificar las he secuenciado linealmente (*soft dependences*), pero un vez empezados los trabajos, si que he podido disponer de dos máquinas, por lo que la secuenciación original no se cumplirá y se ha de modificar, algo que no está soportado en el sistema por avance.

NOTA

Es importante hacer hincapié en el hecho de que la Programación inicial no la cambiaré nunca salvo en muy contadas excepciones, pues es la que me servirá de base de comparación en los diferentes controles.

Este problema es más difícil de solucionar que el anterior, pero tan solo aparentemente, porque como todos los elementos nuevos ya los tenemos, tan solo necesitaremos copiarlos y hacerles unas pequeñas y sencillas modificaciones, así que vamos a hacer una relación de los pasos que hay que seguir:

1. Necesitaremos una nueva matriz donde establecer la cantidad de trabajo que queda pendiente de cada una de las tareas, así como las nuevas reglas de dependencias en caso de se produzcan.
2. Determinar el día de Control de Proyecto.
3. Necesitaremos un nuevo diagrama temporal en donde se reflejen las nuevas fechas previstas de comienzo y terminación de cada tarea.
4. Necesitaremos conocer la nueva fecha prevista de terminación del proyecto.

Para crear una nueva matriz de control para el cálculo de los tiempos de las tareas, copiare la hoja en donde se encuentra la hoja con la matriz inicial en una nueva hoja, a la que llamaré "Matriz de Control", situándome con el ratón en la pestaña de la hoja "Matriz" y **Menú contextual>Mover o copiar**, debiendo de crearla antes de la hoja "Datos Calendario", y con la precaución de activar el cuadro de selección **Crear una copia**. Y le cambiaremos el nombre con **Menú contextual>Cambiar nombre** escribiendo **Matriz de Control** como el nuevo nombre para esta hoja.

1. Cambiar las fórmulas de la columna "Inicio", escribiendo **=SI ('Matriz de Control'! C5>0; DIA.LAB.INTL (Fecha_Inicio; Matriz de Control'! D5; Cod_Fin_D; Festivos);"Finalizada")** en la celda C5, para posteriormente copiar y pegar al resto de la columna. Mediante esta formulación, en el caso de estar finalizada la tarea (duración en la matriz de control igual a cero), aparecerá el texto "Finalizada", y en caso contrario (no empezada o parcialmente ejecutada), presentará la fecha correspondiente al tiempo más pronto de empezar de la matriz.
2. Cambiar las fórmulas de la columna "Inicio", escribiendo **=SI ('Matriz de Control'! C5>0; DIA.LAB.INTL (Fecha_Inicio; 'Matriz de Control'! E5-1; Cod_Fin_D; Festivos);"Finalizada")** en la celda D5, para al igual que antes, copiar y pegar al resto de la columna. Con esta formulación, en el caso de estar finalizada la tarea aparecerá el texto "Finalizada", y en caso contrario la fecha correspondiente al tiempo más pronto de terminar de la matriz.
3. Eliminar las fórmulas del rango de celdas correspondiente al diagrama temporal, para escribir: **=SI (DIAS.LAB.INTL (E\$3; E\$3; Cod_Fin_D; Festivos)=1;"";".")**, porque como en el tipo de control que hacemos tan solo me interesa que aparezca en las celdas los fines de semana y festivos, y ninguna información sobre los importes, ya que no los puedo controlar.
4. Eliminar la columna "Importe" correspondiente a la columna E de Excel.

Ya está terminada la nueva hoja, solo cambia el ancho de las columnas del calendario a otro más acorde a la ausencia de valores en el interior de las celdas y tendrá que aparecer un resultado similar al siguiente, considerando que a Ex1 le queda 1 día y a Ex2 le queden 7 días (no hace falta que pulses "Formular", pues no se han realizado modificaciones en la estructura del grafo), véase la figura 4.7. Ahora regresa a la matriz de control, y haz estas modificaciones en las duraciones y dependencias de las tareas:

Tarea	Queda	Depende de...
Ex1	1 día	
Ex2	7 días	de nadie
Ex3	10 días	Ex2
Ex4	10 días	Ex1
Ex5	10 días	Ex4

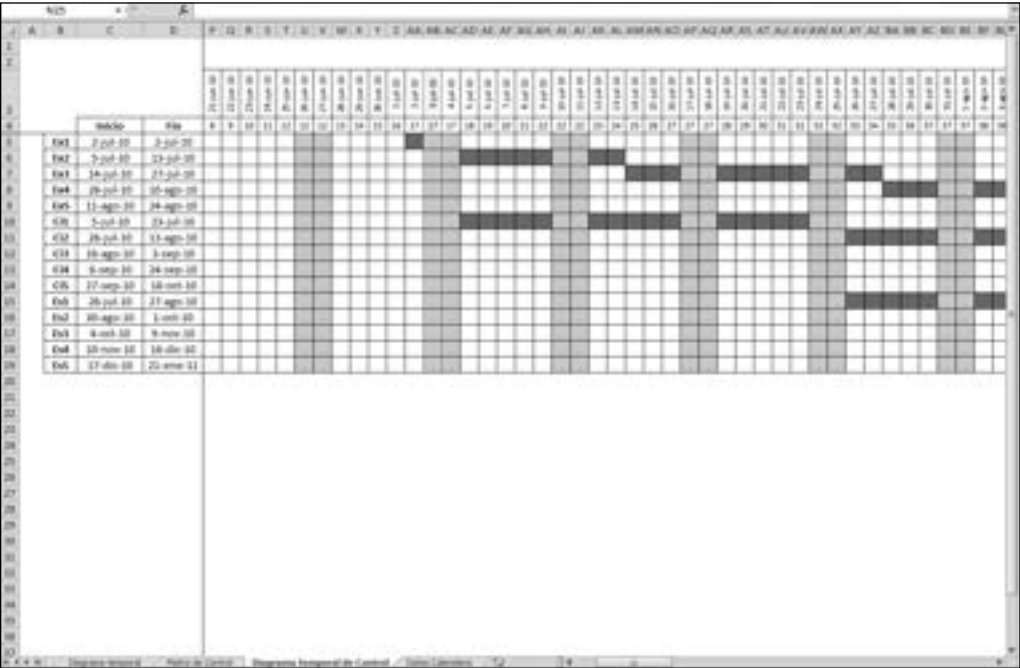


Figura 4.7. Diagrama temporal del control tradicional por matriz.



Figura 4.8. Matriz de control resultante.

Si pulsamos "Formular" con el botón del ratón tendrían que cambiar muchas cosas, pero no sucede así, y eso es porque estamos haciendo los cambios sobre la matriz original y no sobre la matriz de control. Vamos a arreglarlo:

1. Regresa a la hoja "Matriz" y pulsa sobre "Formular", para corregir la formulación inicial.
2. Programador>Visual Basic y haz doble clic sobre la hoja "Matriz de control" en el panel Proyecto ubicado en el lateral izquierdo.
3. Edición>Reemplazar y escribe **matriz** en "buscar" y **matriz de control** en "Reemplazar con". Ahora los cambios los afectará a la "Matriz de control" y no a la inicial.



Figura 4.9. Reemplazar un texto en Visual Basic.

4. Ya se puede cerrar la ventana del "Programador" y pulsar sobre el botón "Formular" para ver que le ha pasado a nuestro proyecto (véase la figura 4.10).

Perfecto, algunas de las tareas ya no se ejecutan de forma secuencial, y están comenzando más pronto de cuando estaba previsto, sería bueno que pudiera cuantificar ese adelanto en una cierta cantidad de días, para lo que iré a la hoja "Datos Calendario" e insertaré dos filas justo antes de la fila 5 (véase la figura 4.11).

Y escribiré:

Fórmula	Celda
=MAX ('Diagrama temporal de Control! D5:D19)	C5
=SI (B6=0;"En Plazo"; SI (B6<0;"días de retraso";"días de adelanto"))	C6
=MAX (Matriz! G5:G19)-MAX ('Matriz de Control! G5:G19)	B6

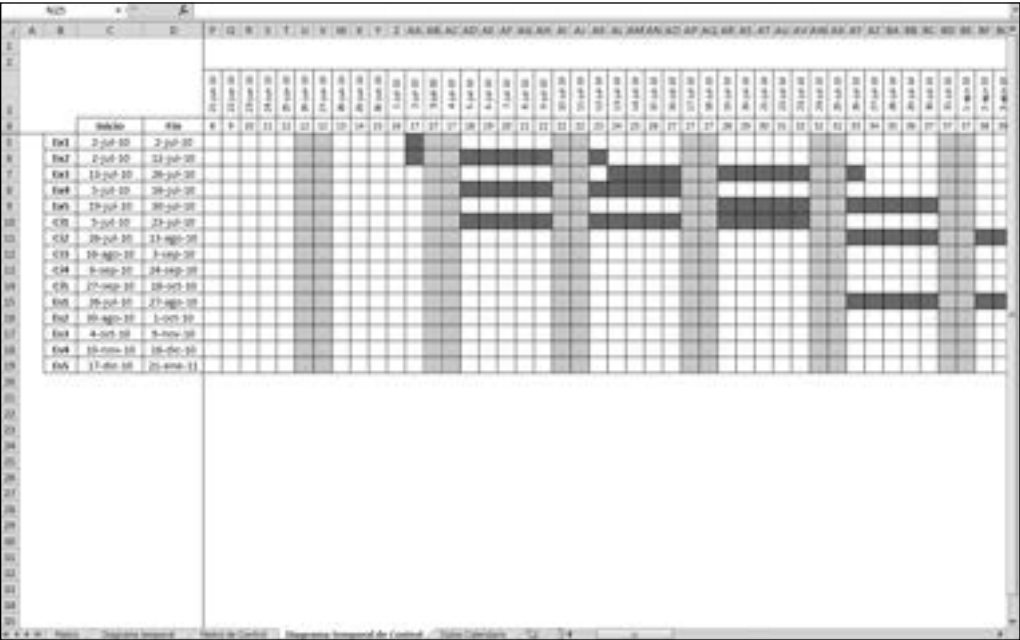


Figura 4.10. Diagrama temporal del control tradicional por matriz con cambios sobre las dependencias.

	A	B	C
1			
2		Inicio	10-jun-10
3		Hoy	sábado, 03 de julio de 2010
4		Fin	miércoles, 12 de enero de 2011
5			
6			

Figura 4.11. Dos nuevas filas en "Datos Calendario".

Obteniendo la fecha prevista de finalización, y los días (de trabajo) de adelanto o retraso con respecto a la fecha prevista:

	A	B	C
1			
2		Inicio	10-jun-10
3		Hoy	sábado, 03 de julio de 2010
4		Fin	miércoles, 12 de enero de 2011
5		Fin Act	viernes, 21 de enero de 2011
6		-7	días de retraso

Figura 4.12. Información de la evolución del plazo del proyecto.

Puede resultar paradójico que obtengamos una retraso cuando lo que se ha hecho ha sido adelantar muchas tareas y comenzarlas antes de lo necesario, pero si pensamos que estás tareas no eran críticas, y que cualquier adelanto nunca se reflejará en el plazo del proyecto empieza a estar claro.

Puede ser interesante por cuestiones legales o administrativas que el adelanto o atraso sobre el plazo sea expresado también en días naturales, pero eso será una simple diferencia de los valores de las celdas correspondiente al final previsto menos el final de control.

NOTA

La hoja de cálculo completa se encuentra en Cap_04\Control Tradicional por Matriz, de los ejemplos del libro.

Ya dijimos al principio que ninguno de los dos sistemas es excluyente del otro, por lo que lo más lógico sería que ambos pudieran coexistir en el mismo libro de Excel y para hacerlo tan solo se tendría que copiar la hoja del control de avance en el libro de control por matriz, siguiendo los criterios expuestos.

NOTA

La hoja de cálculo completa se encuentra en Cap_04\Control Tradicional por matriz y avance, de los ejemplos del libro.

4.3. El control de Proyectos por el Método del Valor Ganado

4.3.1. Introducción al control por Valor Ganado

Los dos métodos vistos anteriormente, son capaces de informar sobre el estado de un proyecto en un instante dado, pero la realidad de los proyectos es muy compleja, haciendo que informar en términos de adelanto o de atraso sin considerar otros factores hace que en ocasiones sea incompleto e incorrecto.

El objetivo principal de la programación de Proyectos consiste en producir un determinado bien en un plazo establecido y con una disponibilidad de limitados recursos al mínimo coste posible, por lo que olvidar los conceptos de coste y producción en el control hace que las conclusiones que obtengamos no sean útiles.

No solo hemos de considerar el tiempo empleado en ejecutar unas determinadas tareas, sino que hemos de tener en cuenta si las unidades producidas por estas tareas hasta ese instante son acordes con las previsiones realizadas, y al igual deberemos conocer si los costes en los que hemos incurrido son acordes con las producciones realizadas.

La respuesta a todo lo anterior es el Método del Valor Ganado o *Earned Value Analysis (EVA)*, que contempla todos esos aspectos de una forma asombrosamente sencilla y armoniosa, permitiéndome no solo informar en términos absolutos de plazo y costes, sino relativizados a las producciones realizadas en un determinado tiempo por todo el proyecto.

El MVG tiene su origen a principios del siglo veinte de la mano de Frank y Lillian Gilbreth y su industria de tejidos con la invención del concepto "tiempo ganado", aunque la mayoría de los autores lo llevan a la década de los años 60 del siglo pasado y concretamente atribuyen su autoría al Departamento de Defensa de los Estados Unidos de América.

Sin embargo, es cierto que el Departamento de Defensa de los Estados Unidos ha sido desde los años sesenta un gran impulsor de esta herramienta, y ha contribuido enormemente a su popularización exigiendo a los contratistas informar el avance de los proyectos adjudicados mediante el valor ganado.

El Método del Valor Ganado está sustentado en estos tres pilares fundamentales:

1. El coste presupuestado.
2. El coste real.
3. El coste presupuestado del trabajo realizado (Valor Ganado).

4.3.2. Los índices de primer orden

El Método del Valor Ganado, está sustentado por los tres anteriores pilares fundamentales, y que se denominan los índices de primer orden. Estos tres índices son especialmente importantes, porque a partir de ellos se construye toda esta novedosa metodología, surgiendo posteriormente

otros conceptos que no podríamos afrontar si antes no dedicamos un poco de tiempo a su explicación.

Estos índices de primer orden se representan mediante las denominadas curvas "S", que no son más que el gráfico de los valores a origen del índice que estoy analizando en cada momento, y que ya vimos en el capítulo anterior:

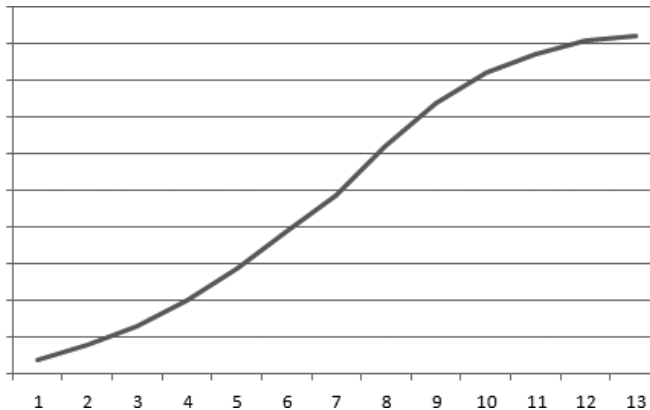


Figura 4.13. Curva "S".

4.3.2.1. El CPTP (Coste Presupuestado del Trabajo Programado)

También llamado *BCWS (Budgeted Cost of Work Scheduled)*.

El *Coste Presupuestado del Trabajo Programado*, es el importe de las unidades que tenía previsto producir al coste que tenía previsto producirlas y en el plazo que tenía previsto, referido a una determinada tarea o a la totalidad del proyecto.

Se podría resumir como: LO PREVISTO AL COSTE PREVISTO.

4.3.2.2. El CRTR (Coste Realizado del Trabajo Realizado)

También llamado *ACWP (Actual Cost of Work Performed)*.

El *Coste Realizado del Trabajo Realizado*, es el importe de las unidades que realmente he producido al coste que realmente me ha costado y en el plazo que realmente he necesitado para producirlas, referido a una determinada tarea o a la totalidad del proyecto.

Se podría resumir como: LO REAL AL COSTE REAL.

4.3.2.3. El CPTR (Coste Presupuestado del Trabajo Realizado)

También llamado *BCWP (Budgeted Cost of Work Performed)*.

El Coste Presupuestado del Trabajo Realizado, es el importe de las unidades que realmente he producido al coste que tenía previsto y en el plazo que realmente he necesitado para producirlas, referido a una determinada tarea o a la totalidad del proyecto.

Se podría resumir como: LO REAL AL COSTE PREVISTO.

A este concepto se le conoce como Valor Ganado, y es el más importante de todos los índices por la enorme trascendencia que tiene en el desarrollo de esta metodología. Mediante este valor, lo que hago es determinar el importe de los costes en los que debería de haber incurrido para la producción que realmente he ejecutado.

4.3.3. Los índices de segundo orden

A partir de los anteriores índices de primer orden, puedo establecer los llamados índices de segundo orden, que a mi entender, posiblemente sean los elementos más importantes del método del valor ganado por la gran consistencia del análisis que me permitirá realizar.

Estos índices de segundo orden nos informaran sobre la eficiencia de mi gestión y el verdadero avance de mi proyecto en términos de coste, producción y tiempo, y son:

1. La *desviación de coste (DC)*.
2. La *desviación de programa (DP)*.
3. La *programación ganada (PG)*.
4. La *adherencia a la programación (AP)*.

4.3.3.1. La desviación de coste

La *desviación de coste (DC)*, conocida como *Cost Variation (CV)* me informa sobre la eficiencia en la gestión del coste. Es la diferencia entre lo que debería de haber invertido en el proyecto según el trabajo realmente realizado y lo realmente invertido en un instante dado del proyecto.

Se calculará como:

$$DC = CPTR - CRTR$$

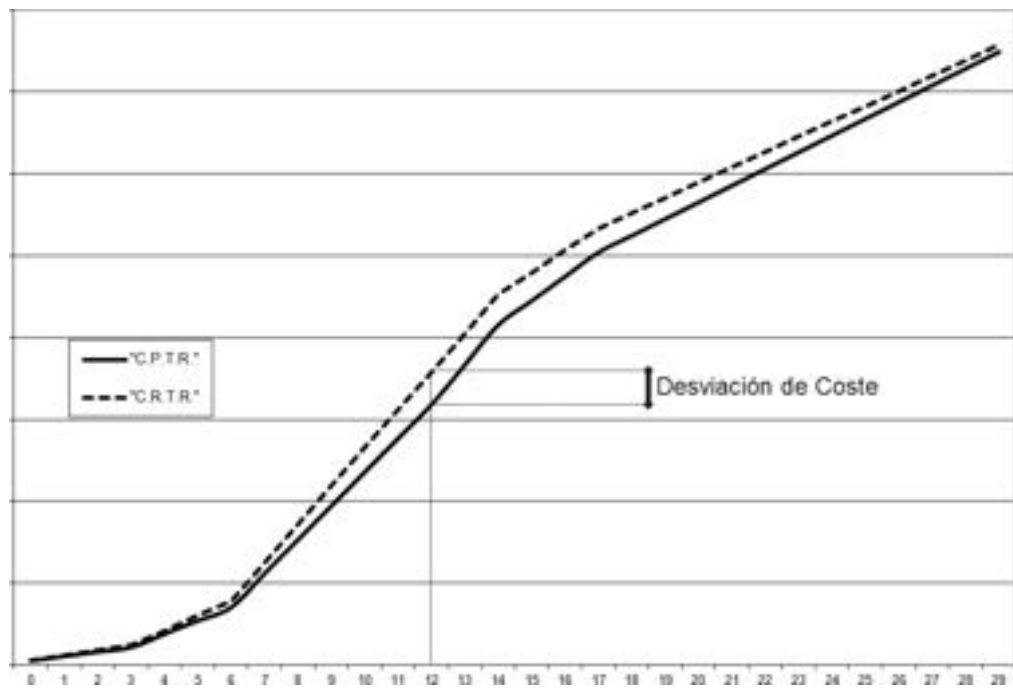


Figura 4.14. Desviación de coste.

El signo resultante de la anterior operación puede presentar tres tipos de valores diferentes:

Resultado de la DC	Conclusión
DC=0	Me gasto según lo previsto.
DC<0	Me gasto DC más de los previsto.
DC>0	Me gasto DC menos de lo previsto.

Pero este valor no es suficiente por sí mismo, porque puedo estar gastando menos, pero la producción la puedo llevar muy por debajo de las previsiones, por lo que tendré que complementar este índice con otro que me informe sobre la eficiencia en la gestión del programa.

4.3.3.2. La desviación de programa

La *desviación de programa (DP)*, conocida como *Schedule Variation (SV)* me informa sobre la eficiencia en el avance de la programación.

Compara el avance real del proyecto según los costes previstos con el avance previsto originalmente en un instante dado del proyecto.

Se calculará como:

$$DP = CPTR - CPTP$$

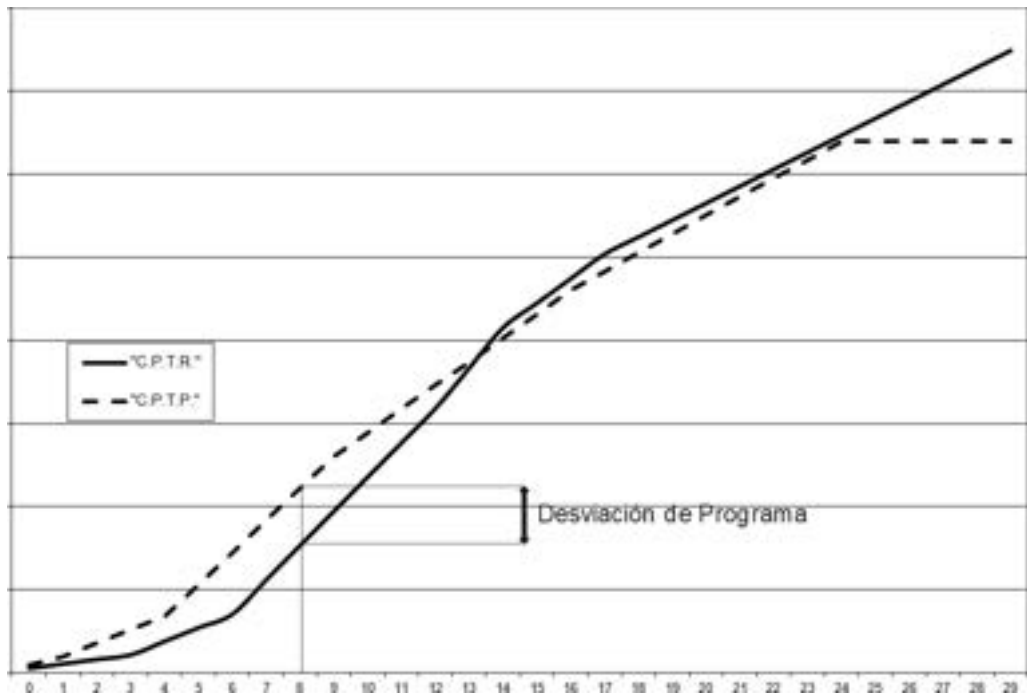


Figura 4.15. Desviación de programa.

El signo resultante de la anterior operación puede presentar tres tipos de valores diferentes:

Resultado de la <i>DP</i>	Conclusión
$DP=0$	El Proyecto avanza según lo previsto.
$DP<0$	El Proyecto se encuentra retrasado.
$DP>0$	El proyecto se encuentra adelantado.

Es importante destacar, que la *desviación de programa* al estar expresada en euros, me está informando realmente en términos de producción realizada frente a producción prevista, algo de enorme importancia para

las empresas con proyectos basados en procesos productivos como las empresas constructoras.

4.3.3.3. La programación ganada

Hasta ahora hemos visto la evolución del coste y la evolución del programa o mejor dicho, de la producción del programa, pero aún no hemos comentado nada sobre el plazo del proyecto. Este importantísimo elemento lo controlaremos mediante la *Programación Ganada (PG)* o *Earned Schedule (ES)*, que si bien es muy sencilla de entender no es tan fácil de calcular, necesitando echar mano de nuestros oxidados conocimientos de geometría elemental y semejanza de triángulos.

Mediante la programación ganada vamos a calcular la desviación en plazo, estableciendo el instante previsto en el cual nos deberíamos de encontrar (*CPTP*) para haber alcanzado el nivel de avance real del proyecto (*CPTR*):

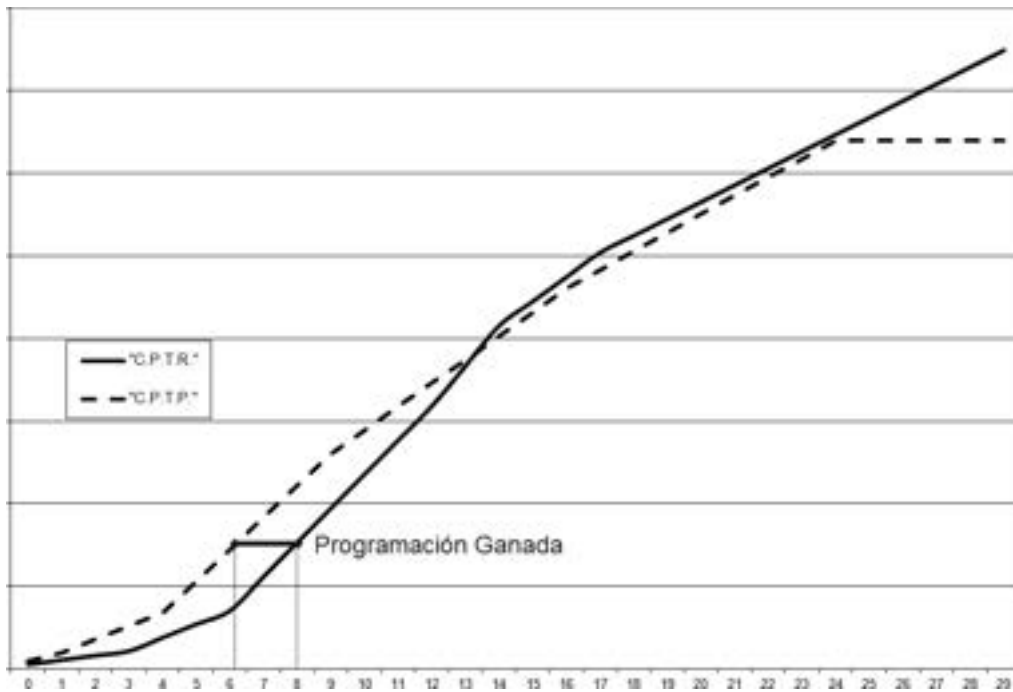


Figura 4.16. Programación ganada.

Y se calculará por semejanza de triángulos de la forma que se muestra a continuación:

era de tipo *soft* o de recurso. Pero veámoslo gráficamente según el siguiente avance previsto en un determinado instante de control:

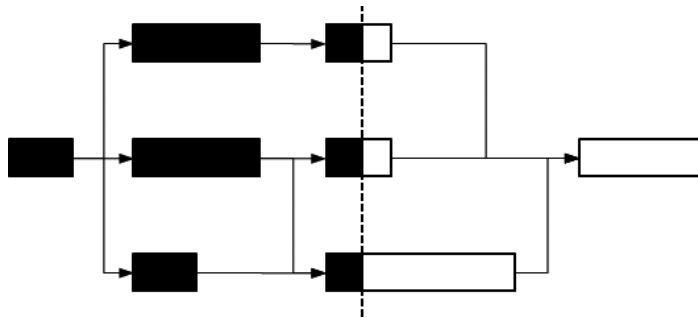


Figura 4.18. Avance previsto del proyecto.

Pero lo que realmente se ha ejecutado ha sido:

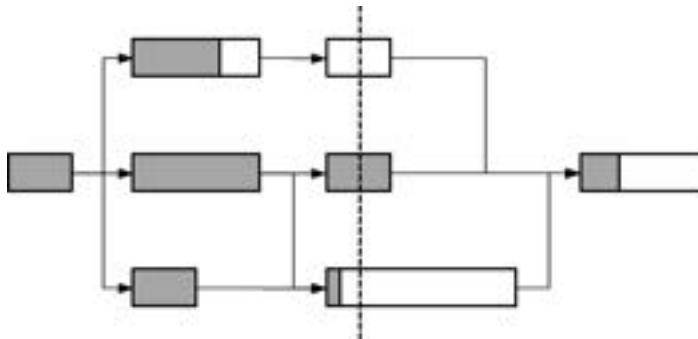


Figura 4.19. Avance real del proyecto.

Como podemos observar en las dos figuras anteriores, las tareas no se han ejecutado según lo previsto, vemos que algunas están adelantadas, otras sin embargo están retrasadas, y además no se están ejecutando según la estricta secuencia programada. Sería importante establecer un índice que nos indicará como de fiel está siendo la ejecución del proyecto a la secuencia previamente establecida, y ese índice no es más que la *adherencia a la programación* (véase la figura 4.20).

Cuantificar la adherencia a la programación consiste en establecer el porcentaje de trabajo realizado de acuerdo a la *programación ganada* (PG), tal y como se puede observar en la figura 4.21. Y la calcularemos como el cociente entre el *coste presupuestado del trabajo realizado* ($CPTR_{pg}$) y el *coste presupuestado del trabajo programado* de las tareas hasta el instante determinado por la *programación ganada* ($CPTP_{pg}$).

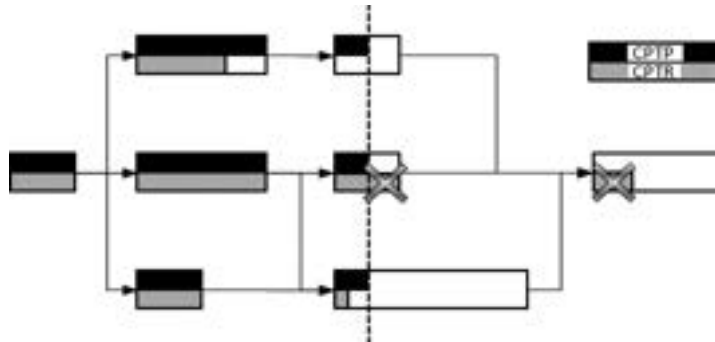


Figura 4.20. Avance real vs avance previsto.

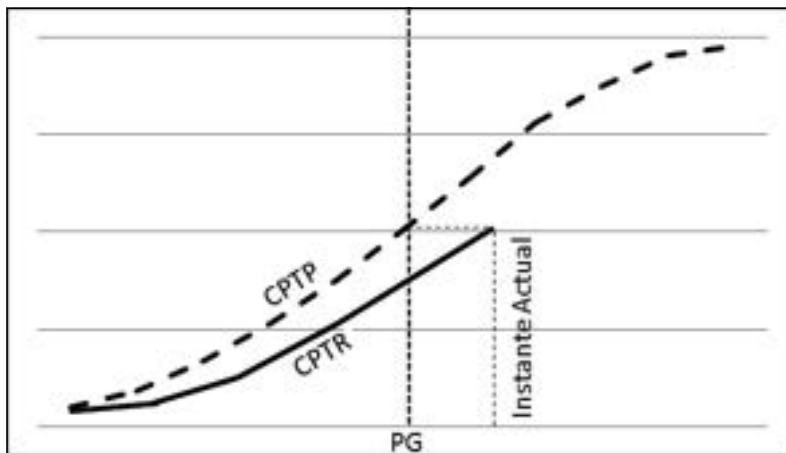


Figura 4.21. Adherencia a la programación.

Normalmente la *adherencia a la programación* ofrecerá valores menores o iguales a uno, salvo que estemos produciendo más unidades de las inicialmente previstas, en cuyo caso el *coste presupuestado del trabajo realizado* será mayor que el *coste presupuestado del trabajo programado*, dando valores mayores a la unidad.

4.3.4. Los índices de tercer orden

Los índices de tercer orden constituyen los valores de rendimiento o eficiencia y las proyecciones de programación. Los primeros son útiles cuando deseamos representar el avance del proyecto en función de índices, siendo los valores menores de uno representativos de una mala gestión del proyecto, valores superiores a la unidad corresponderán a una buena gestión y los

valores cercanos a uno serán indicativos de que el proyecto avanza según las previsiones realizadas.

Las proyecciones de programa las utilizaremos para hacer predicciones sobre el proyecto para su terminación y están basadas en los valores obtenidos a lo largo de su evolución.

4.3.4.1. Los Índices de eficiencia

4.3.4.1.1. Índice de rendimiento de coste

El índice de rendimiento de coste (IRC) se obtiene dividiendo el coste presupuestado del trabajo realizado (CPTR) entre el coste realizado del trabajo realizado (CRTR).

También es conocido como CPI o *Cost Perfomance Index*.

$$IRC = CPTR / CRTR$$

4.3.4.1.2. Índice de rendimiento de programa

El índice de rendimiento de programa (IRP) se obtiene dividiendo el coste presupuestado del trabajo realizado (CPTR) entre el coste presupuestado del trabajo programado (CPTP).

También es conocido como SPI o *Schedule Performance Index*.

$$IRP = CPTR / CPTP$$

4.3.4.1.3. Índice de rendimiento coste-programa

El índice de rendimiento de coste-programa (IRCP) se obtiene multiplicando el índice de rendimiento de coste por el índice de rendimiento de programa.

También es conocido como CSI o *Cost Schedule Performance Index*.

$$IRCP = IRC \cdot IRP$$

4.3.4.2. Las proyecciones

Tal y como se ha comentado anteriormente de forma somera, las proyecciones las utilizaremos para hacer predicciones sobre el proyecto para su terminación, y están basadas en los valores obtenidos a lo largo de su evolución.

Las estimaciones realizadas por las proyecciones son bastante discutibles, y actualmente está perdiendo adeptos a favor de la programación ganada y la adherencia a la programación, que me ofrecen valores ciertos y no estimaciones, que estarán muy afectadas por el momento en el que se hagan estas estimaciones.

Las proyecciones que se realizan al inicio de la ejecución del proyecto carecen de todo valor predictivo, y tan solo serán fiables aquellas que son realizadas en los últimos instantes del proyecto, y con carácter meramente orientativo.

4.3.4.2.1. Coste estimado para terminar

El *coste estimado para terminar (ETC)*, como su propio nombre indica, será utilizado para hacer predicciones sobre el coste restante del proyecto para su terminación, y se calculará de cuatro formas diferentes:

1. Haciendo una nueva estimación de los costes.
2. $ETC = \text{Coste Total Previsto} - CPTR$
3. $ETC = (\text{Coste Total Previsto} - CPTR) / IRC$
4. $ETC = (\text{Coste Total Previsto} - CPTR) / IRP$

El primer criterio será usado cuando las estimaciones han sido totalmente incorrectas, mientras que el segundo será usado en el caso contrario, es decir, cuando las predicciones y la evolución del proyecto coinciden.

Los otros dos sistemas son utilizados cuando se prevé que la evolución del proyecto seguirá las desviaciones que hemos observado hasta el momento, y diferentes autores recomiendan su predicción a partir del *índice de rendimiento de costes (IRC)* o bien del *índice de rendimiento de programa (IRP)*.

4.3.4.2.2. Coste estimado a la terminación

El *coste estimado a la terminación (EAC)*, se calculará sumando al coste realizado hasta el momento actual (*CRTR*) el *coste estimado para terminar (ETC)*, según alguno de los cuatro criterios mencionados anteriormente, que hayamos usado para su cálculo.

4.3.4.2.3. Índice de desempeño a la terminación

El *índice de desempeño a la terminación (TCPI)* se obtiene como el cociente entre la diferencia del coste total previsto y el coste presupuestado

del trabajo realizado (CPTR), y la diferencia entre el coste estimado a la terminación (EAC) y el coste realizado hasta el momento actual (CRTR).

$$TCPI = (\text{Coste Total Previsto} - CPTR) / (EAC - CRTR)$$

4.3.4.2.4. Variación a la terminación

La *variación a la terminación* (VAC) se calculará como la diferencia entre el *coste total previsto* y el *coste estimado a la terminación* (EAC).

$$VAC = \text{Coste Total Previsto} - EAC$$

Según su signo sea positivo o negativo podremos decir que terminaremos con costes menores a los previstos o bien mayores de los previstos respectivamente.

4.4. Implementación del control por el Valor Ganado en Excel 2010

La incorporación del control por valor ganado en un Libro de Excel no es una labor excesivamente compleja, pero sí que requiere que sigamos un cierto orden por la gran cantidad de nuevos términos que hemos introducido, para lo que partiremos de los elementos más sencillos para ir incrementando la dificultad a medida que avancemos.

Para su implementación de la manera más fácil, partiremos del libro expuesto en el capítulo anterior y que se encuentra en Cap_03\Diagrama temporal con calendario natural, sobre el cual haremos unas pequeñas modificaciones, además de tener la precaución de seguir un criterio de colores mediante el cual diferenciaremos los valores que introducimos en los diferentes controles, con los valores que son calculados para las celdas. De esta forma, usaremos un fondo crema con texto en rojo para las celdas con los valores que introduciremos, y textos en negro para las celdas calculadas.

4.4.1. El CPTP en Excel 2010

Para empezar nuestro nuevo libro de control de proyecto con el valor ganado con Excel 2010, empezaremos abriendo el libro llamado "Diagrama temporal con calendario natural", en el que tan solo dejaremos las hojas

denominadas "Matriz" y "Diagrama temporal", cambiando el nombre de esta última hoja a "CPTP". Una vez realizados estos cambios, tendremos que insertar una columna correspondiente a las unidades a producir previstas, colocar un encabezamiento que nos ayude a identificar rápidamente la hoja en donde nos encontramos, y dar formato a las celdas correspondientes al coste y a las unidades a producir como ya se ha comentado anteriormente, debiendo quedar de la siguiente manera:

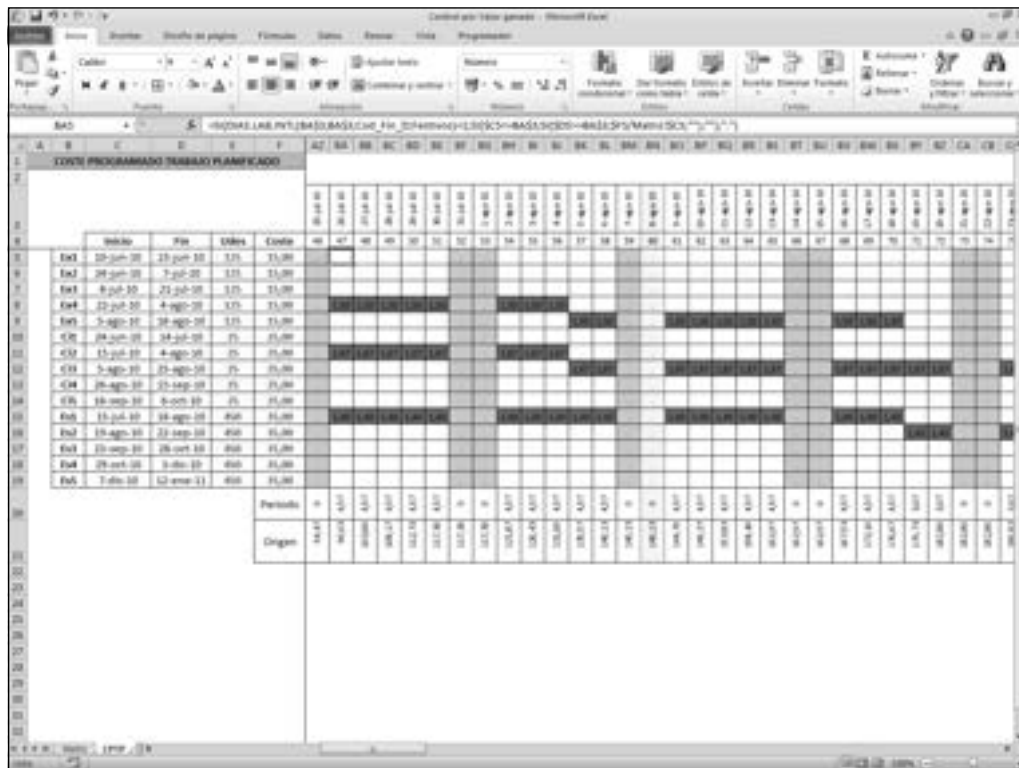


Figura 4.22. Coste presupuestado del trabajo programado.

NOTA

Además de cambiar el aspecto de las celdas es conveniente proteger las celdas que no van a ser modificadas, seleccionando el rango correspondiente a E5:F19 y mediante el botón derecho del ratón activar el menú contextual y seleccionar Formato de celdas>Proteger y deseleccionar Bloqueada, para posteriormente hacer clic con el botón derecho del ratón sobre la pestaña de la hoja "CPTP" y proteger la hoja, evitando así cambios no deseados o errores.

4.4.2. El coste realizado CR y el trabajo realizado TR en Excel 2010

Una vez construida la hoja correspondiente a las previsiones de mi proyecto, hemos de preparar las hojas en las que introduciremos los datos correspondientes a los costes en los que realmente hemos incurrido, y la cantidad de trabajo o de unidades realmente producidas hasta la fecha de control. Para ello procederemos exactamente igual que hemos visto en el capítulo anterior, creando dos nuevas hojas, una para el *coste realizado* (CR), y otra para el *trabajo realizado* (TR).

4.4.2.1. El coste realizado CR

En la hoja correspondiente al coste realizado (CR), no solo escribiremos los valores de coste reales del proyecto, sino que también estableceremos los datos principales y más relevantes del control, y que corresponden a:

1. Fecha de control.
2. Fecha de inicio real de la tarea.
3. Fecha de final real de la tarea. En el caso de que la tarea esté parcialmente ejecutada, es decir, sin finalizar, introduciremos la fecha de control como fecha de finalización para que la formulación de las celdas opere adecuadamente.
4. El coste en el que hemos incurrido para cada tarea y día en cada una de las celdas correspondientes del diagrama temporal.

Al igual que hemos en la hoja correspondiente al *CPTP*, daremos el formato de fondo crema y letra en rojo a las celdas correspondientes a los valores del día de control, fecha de inicio real y fin real de cada tarea, pero también cambiaremos la fórmula de las celdas del diagrama temporal, así como sus formatos condicionales, para lo cual procederemos de la siguiente forma:

1. Nos situaremos en la celda F5.
2. Como en este caso no deseamos que Excel calcule los valores que van a aparecer en las celdas del diagrama temporal, y tan solo deseamos que aparezcan resaltados los días festivos, cambiaremos la fórmula existente escribiendo la siguiente:

=SI (DIAS.LAB.INTL (F\$3; F\$3; Cod_Fin_D; Festivos)=1;"";".")

3. Cambiaremos el tercer formato condicional de tal manera que las celdas presenten un fondo anaranjado con las fuentes en color rojo.
4. Introduciremos un nuevo formato condicional que deberá quedar en segunda posición con la siguiente fórmula de condición: $=\$C\$3=F\$3$, que nos resaltará el día correspondiente a la fecha de control, que no tendrá porque coincidir con la fecha actual, y que podría ser por ejemplo de fondo rojo con una suave trama.
5. Copiaremos esta celda F5 a todo el diagrama temporal.

Una vez finalizado todo el proceso anterior, nos debería de quedar una hoja con el siguiente aspecto con algunos de los valores de coste realizado ya escritos en las celdas correspondientes:

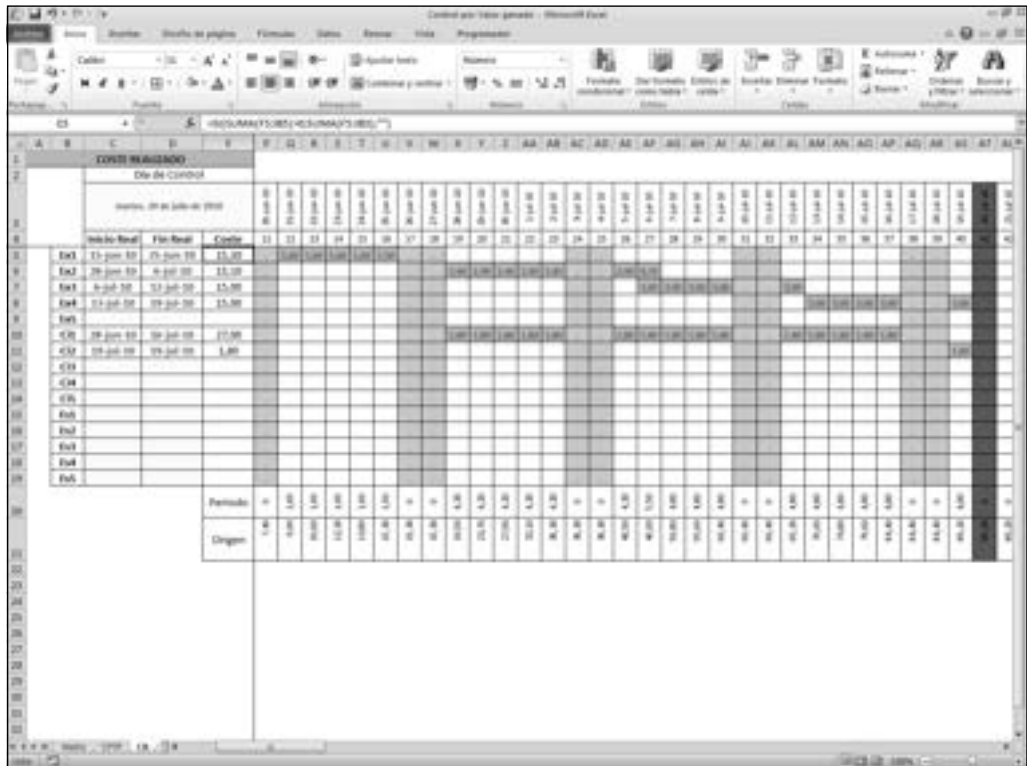


Figura 4.23. Coste realizado.

El lector habrá observado cómo ha desaparecido la columna correspondiente a las unidades previstas, y que la columna correspondiente al coste es automáticamente calculada por Excel sumando todos los valores

correspondientes a su fila mediante la fórmula SUMA, pero solo aparecen valores cuando existen datos escritos, gracias a la combinación de una condición en la suma, consiguiéndolo escribiendo la siguiente fórmula en la celda E5:

=SI(SUMA(F5:IB5)>0;SUMA(F5:IB5);"")

y que deberemos copiar en toda la columna.

NOTA

Cuando se introduce un valor en alguna celda del diagrama temporal, la fórmula existente es sustituida por ese valor, por lo que si se ha cometido algún error, se deberá de copiar nuevamente el contenido de alguna celda que contenga la fórmula a la celda que deseemos rectificar.

4.4.2.2. El trabajo realizado TR

Esta nueva hoja en la cual escribiremos los datos relativos al trabajo realizado, la crearemos copiando la hoja "CR" en una nueva hoja a la que posteriormente le cambiaremos el nombre por "TR", y sobre la que realizaremos los siguientes cambios, puesto que ya no necesitamos volver a escribir los valores que ya hemos introducido anteriormente:

1. En las celdas combinadas C3:E3, que contienen el día de control escribiremos la fórmula **=CR!C3**, y le cambiaremos el formato a fondo blanco y fuente en negro.
2. Las celdas de las fechas de inicio y final de las tareas serán las que tenemos escritas en la hoja "CR", por lo las formularemos nuevamente y les cambiaremos el formato exactamente igual que antes, pero en este caso para evitar que diferencia claramente los valores escritos en las celdas con valores vacíos escribiremos la siguiente fórmula en la celda C5 **=SI(CR!C5<>"";CR!C5;"")**, copiándola al resto de de las celdas de inicio y final.
3. Cambiaremos ahora en encabezamiento de la columna "Coste" por "Uds Ejec".
4. Cambiaremos el texto del encabezamiento por "Trabajo realizado".

Una vez finalizado, el aspecto de esta nueva hoja con algunos valores de unidades de trabajo realizado, deberá quedar como la siguiente figura:

Los valores de unidades ejecutadas los tomaremos directamente de la hoja "TR", escribiendo en la celda E5 la fórmula **=TR!E5**, que copiaremos a lo largo de todas las tareas, y para los valores de la columna "CPTR", no tendremos que hacer ninguna modificación.

En las celdas del diagrama temporal, tendremos que modificar la fórmula, porque en este caso queremos que Excel calcule sus valores correspondientes, en función de las previsiones de coste y producción, y las unidades realmente producidas.

Para este cálculo tendremos que determinar el coste previsto unitario de la tarea (coste previsto por unidad producida prevista), para posteriormente multiplicar este valor por las unidades realmente producidas, pero solo en el caso de que realmente se haya producido alguna unidad en ese día, y que además no sea festivo.

La condición de que exista alguna unidad producida en ese día y que además no sea festivo la resolveremos mediante la inclusión de la función lógica Y(condición 1; condición 1; ...), y el coste unitario previsto como (coste previsto / unidades previstas) y que se encuentran en la hoja "CPTP". Así la fórmula completa correspondiente a la celda G5 será:

```
=SI(DIAS.LAB.INTL(G$3;G$3;Cod_Fin_D;Festivos)=1;SI(Y(TR!F5<>"";
TR!F5<>".");(CPTP!$F5/CPTP!$E5)*TR!F5;"");".")
```

Que no es más que sustituir las comillas vacías y resaltadas en la fórmula original:

```
=SI (DIAS.LAB.INTL (F$3; F$3; Cod_Fin_D; Festivos)=1;"";".")
```

Por la expresión:

```
SI(Y(TR!F5<>"";TR!F5<>".");(CPTP!$F5/CPTP!$E5)*TR!F5;"")
```

Una vez finalizado todo lo anterior la hoja tendrá el aspecto que muestra la figura 4.25.

4.4.4. Los índices de segundo y tercer orden en Excel 2010

El cálculo de los índices de segundo y tercer orden en Excel 2010 no presenta ninguna dificultad, a excepción de la *programación ganada (PG)* y la *adherencia a la programación (AP)*, que requiere realizar unos pasos intermedios bastante confusos mediante la incorporación de funciones de

búsqueda que son relativamente complicadas de entender, por lo que alteraremos el orden que se ha seguido en la exposición de los términos conceptuales y dejaremos estos dos últimos para el final.

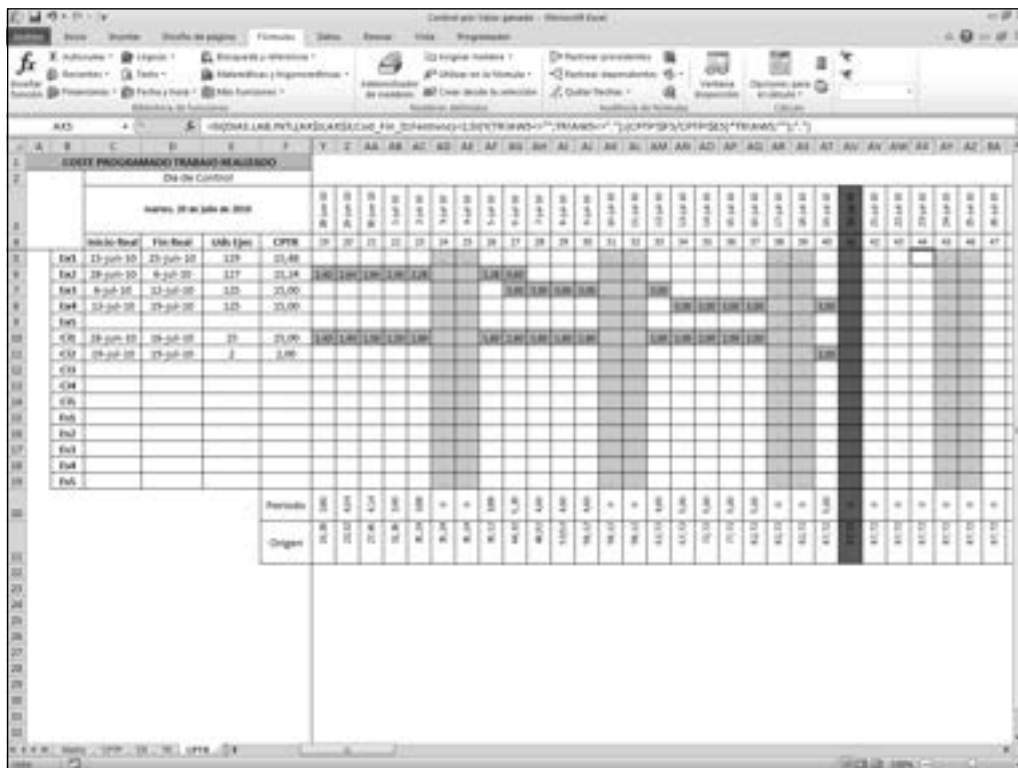


Figura 4.25. Coste presupuestado del trabajo realizado.

Antes de proceder con el cálculo de los índices, tendremos que crear una nueva hoja, que crearemos desde la hoja denominada "TR", haciendo el cambio habitual de escribir como cabecera de la hoja "índices de 2do y 3er orden", y cambiando el nombre de esta nueva hoja escribiendo **Índices**, encontrándonos ya en condición de incluir los primeros índices que reflejaran el avance de nuestro proyecto.

4.4.4.1. Las desviaciones de coste y programa en Excel 2010

La *desviación de coste (DC)* y la *desviación de programa (DP)*, son muy sencillas de calcular, solamente tendremos que cambiar el texto correspondiente a las cabeceras de las filas de "Período" y "Origen" y denominarlas

"DC" y "DP" respectivamente. Para formular adecuadamente las celdas recordaremos que $DC = CPTR - CRTR$, y que $DP = CPTR - CPTP$, y así las fórmulas resultantes controlando que solo calculen valores para los días anteriores al día de control establecido serán:

Celda	Fórmula
F20	=SI(F3<\$C\$3;CPTR!G21-CR!F21;"")
F21	=SI(F3<\$C\$3;CPTR!G21-CPTP!G21;"")

Las anteriores fórmulas las tendremos que copiarlas a lo largo de sus filas correspondientes, y para que sean todavía más representativas de la evolución del proyecto añadiremos un nuevo formato condicional que resalte las que tienen un valor menor que cero, y nos quedará de la siguiente forma:

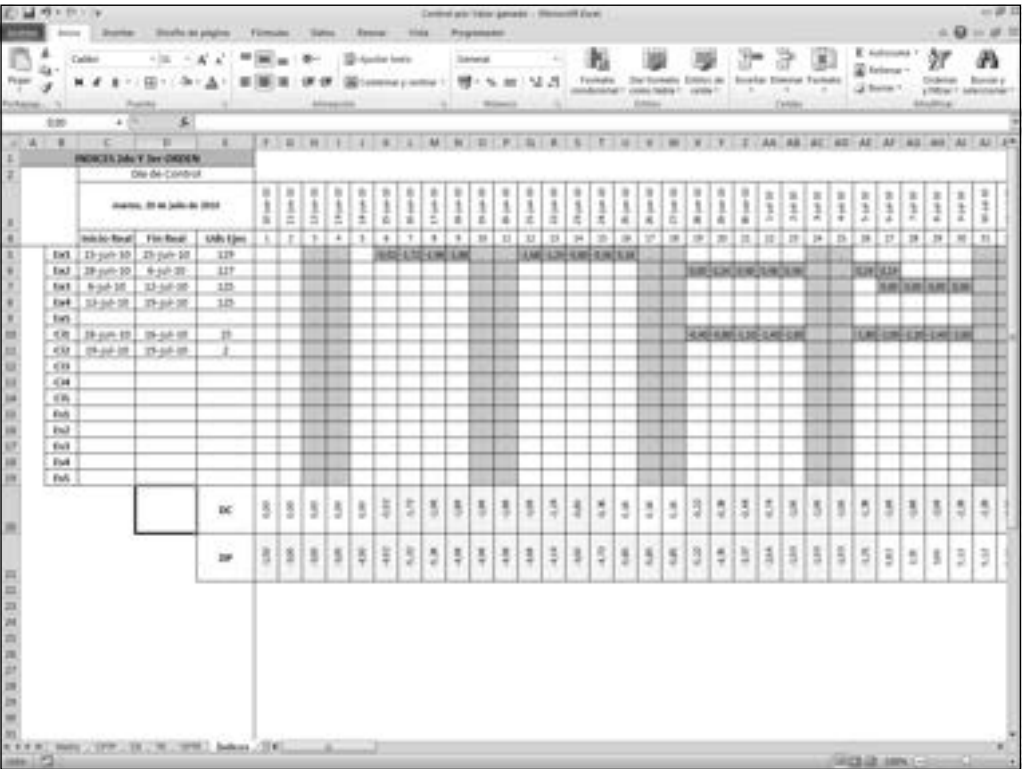


Figura 4.26. Desviación de coste y de programa.

4.4.4.2. Los índices de rendimiento de coste y de programa en Excel 2010

Los *índices de rendimiento de coste (IRC)* y *de programa (IRP)*, también son muy sencillos de calcular, y recordaremos que se obtenían mediante las fórmulas:

$$IRC = CPTR / CRTR$$

$$IRP = CPTR / CPTP$$

Para su incorporación a la hoja "Índices", utilizaremos las filas 23 y 24, escribiremos en el encabezamiento de la fila 23 IRC, y en la fila 24 IRP, y las siguientes formulas en la columna F que corresponde al primer día del calendario:

Celda	Fórmula
F23	=SI(F3<\$C\$3;SIERROR(CPTR!G21/CR!F21;"-");"")
F24	=SI(F3<\$C\$3;SIERROR(CPTR!G21/CPTP!G21;"-");"")

Las anteriores fórmulas no solo calculan valores para días anteriores al de control mediante una condición SI (condición menor que día de control; ...; ...), sino que presentan como novedad un control de error para el caso en que estemos dividiendo por cero, que se produce cuando un proyecto comienza con una posterior a la que teníamos prevista, en cuyo caso nos lo indicará mediante dos guiones en la celda. Evidentemente esta condición de error no es necesaria en la fila 24, aunque no está de más incluirla por si en la matriz de cálculo hemos puesto un instante de comienzo mayor que cero. Véase la figura 4.27.

4.4.4.3. La programación ganada en Excel 2010

La implementación de la *programación ganada (PG)* en Excel, es bastante más complejo de lo que hemos visto hasta ahora, porque como recordaremos, en su cálculo no hace uso de valores que se encuentran en su mismo periodo, sino que además de su periodo "*p*", requiere de valores que se encuentran en los periodos "*n*" y "*n+1*", valores que desconocemos y que dependen de los valores de "*CPTR*" en "*n*", lo que dificulta enormemente su tratamiento, pero como siempre Excel dispone de una formula, que aunque un poco compleja me ayuda a resolver este problema, teniendo solo

- Ordenado: este argumento es opcional y determina si la coincidencia en la primera fila a ser exacta o aproximada, y como en nuestro caso es aproximada que coincide es su opción por defecto, deberemos dejarlo vacío.

Ahora que conocemos como trabaja la función de búsqueda, deberemos crear una tabla que contenga los valores que deseamos encontrar, y lo haremos de la siguiente forma:

1. En la primera fila deseamos los valores correspondientes al *CPTP* en el instante actual "*p*", que serán los que deberán coincidir con el *CPTR* correspondiente, y concretamente para la celda F26 escribiremos **=CPTP!G21**.
2. En la segunda fila colocaremos los valores del *CPTP*, pero en este caso de un periodo posterior al anterior "*p+1*", es decir que para la celda F27 escribiremos **=CPTP!H21**.
3. En la tercera fila colocaremos los valores del *CPTR* en "*p*", que son los valores que deberemos de encontrar en la primera fila de la tabla.
4. En la cuarta fila colocaremos los valores de "*p*"

Una vez copiados los anteriores valores a lo largo de cada una de las filas correspondientes, debería de obtener como resultado la figura 4.29.

Los anteriores valores son los que constituyen los valores de la tabla de búsqueda, pero le tendré que indicar a Excel que esta es la tabla usada para calcular los valores de la programación ganada, para lo que seleccionaré el rango de celdas F26:IB29 y ejecutaré **Fórmulas>Asignar nombre>Definir nombre**, y escribiendo **tabla_PG** como nombre.

Ahora tan solo me queda determinar los valores de "*n*", y del *CPTP* en "*n*" y "*n+1*", para lo que operaré añadiendo unas nuevas filas para cada unos de estos valores de la siguiente forma:

1. El valor de "*n*" será el de la columna correspondiente a la cuarta fila ("*p*"), que coincida con F28 (*CPTR*) en la primera fila de la "tabla_PG".
=SIERROR(CONSULTAH(F28;tabla_PG;4);0).
2. El valor del $CPTP_n$ será el de la columna correspondiente a la primera fila que coincida con F28 (*CPTR*) en la primera fila de la "tabla_PG".
=SIERROR(CONSULTAH(F28;tabla_PG;1);0).
3. El valor del $CPTP_{n+1}$ será el de la columna correspondiente a la segunda fila que coincida con F28 (*CPTR*) en la primera fila de la "tabla_PG". **=SIERROR(CONSULTAH(F29;tabla_PG;2);0).**

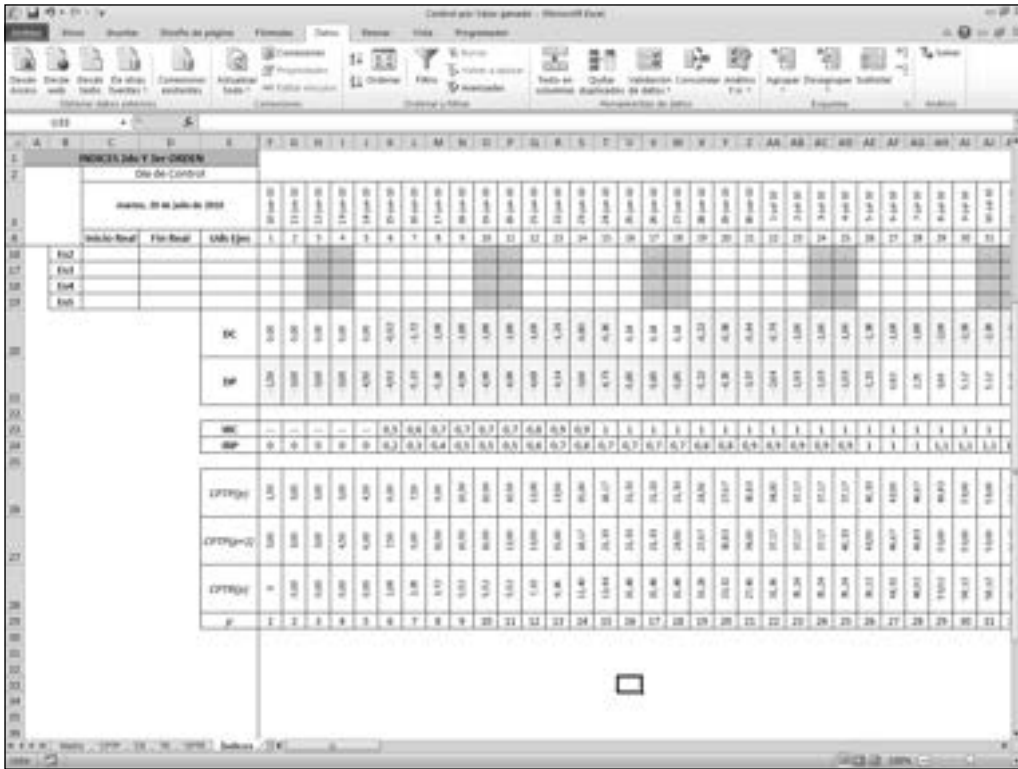


Figura 4.29. Tabla de programación ganada.

- El valor de la *programación ganada* (PG) será el de aplicar literalmente la fórmula ya expuesta, y concretamente para la celda F33 escribiremos $=SI(F3<\$C\$3;SIERROR(F30+((F28-F31)/(F32-F31));F30);"$

Además podemos añadir una nueva fila que nos calcule automáticamente los días de adelanto o atraso del proyecto para la producción que estamos realizando, simplemente restando a la *programación ganada* el valor de "p" correspondiente, y para F34 escribiremos:

$$=SI(F3<\$C\$3;F33-F4;")$$

Para finalizar, tan solo nos quedará copiar todas las formulas a lo largo de sus filas, pero podríamos ser un poco más exigentes y hacer la hoja un poco más amigable ocultando todos aquellos valores auxiliares que no me aportan información. Excel me permite ocultar o bien mostrar a voluntad las filas que desee agrupándolas, y para ello seleccionaré los encabezamientos de las filas 26 a 32 y ejecutaré **Datos>Agrupar**, obteniendo como resultado final:

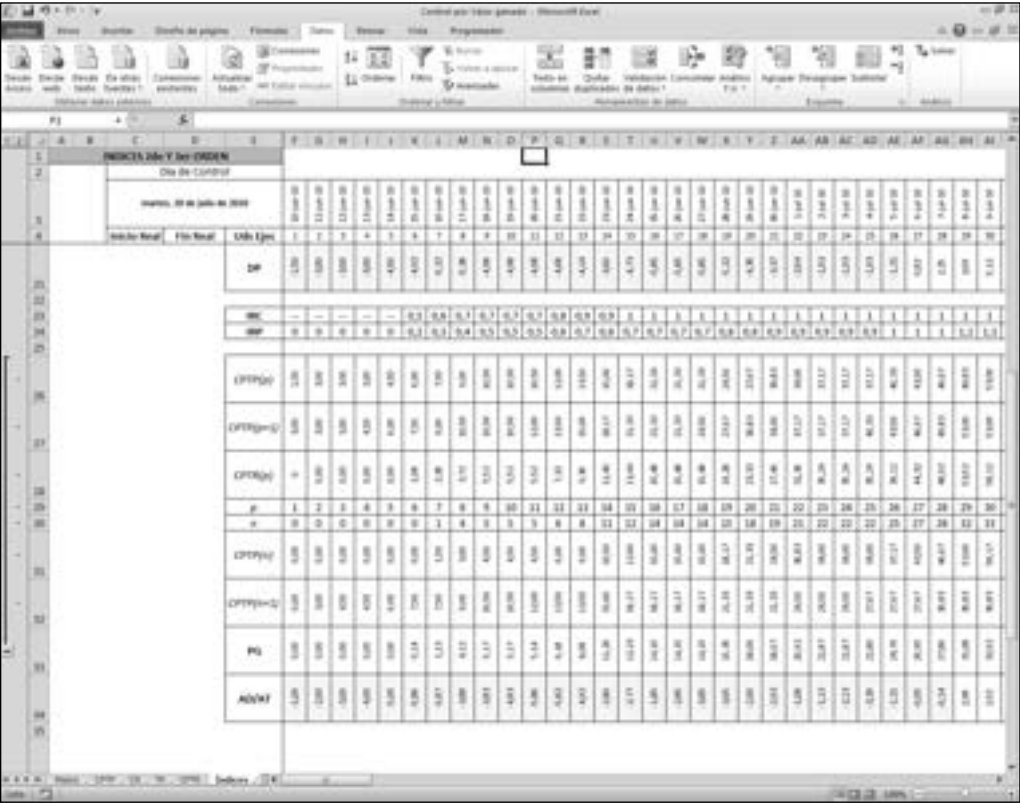


Figura 4.30. Programación ganada, adelantos y atrasos.

4.4.4.4. La adherencia a la programación en Excel 2010

La implementación de la *adherencia a la programación (AP)* en Excel, es muy similar a lo que hemos hecho con la programación ganada (PG), con la diferencia de que en este caso lo que deseamos encontrar de valor de PG para obtener el $CPTP_{PG}$ y $CPTR_{PG}$.

Tal y como hemos hecho anteriormente añadiremos unas nuevas filas, donde la fórmula para la columna F será:

Valor	Fila	Fórmula
p	36	=F29
$CPTP_p$	37	=F26
$CPTR_p$	38	=F28

Valor	Fila	Fórmula
CPTP _{PG}	39	=SIERROR(CONSULTAH(F33;tabla_AP;2);0)
CPTR _{PG}	40	=SIERROR(CONSULTAH(F33;tabla_AP;3);0)
AP	41	=SI(F3<\$C\$3;SIERROR(F40/F39;0);"")

Para la *adherencia a la programación (AP)*, además del agrupamiento de los valores de las filas auxiliares, vamos a introducir un nuevo refinamiento en el formato condicional, y que será que nosotros determinaremos el nivel a partir del cual las celdas de AP son resaltadas a partir de un determinado valor que regularemos mediante un control de formulario.

Empezaremos escribiendo en la celda D41 la fórmula =C41/10, y ejecutar: Programador>Insertar>Controles de formulario>Control de número, estableciendo los siguientes valores para el control:



Figura 4.31. Control de número de la AP.

Si ahora cambio el formato de la fuente de la celda C41 a blanco, y selecciono las celdas de la adherencia a la programación y además añado un nuevo formato condicional que resalte las celdas cuyo valor sea menor que D41 obtendré la siguiente hoja ya totalmente finalizada:

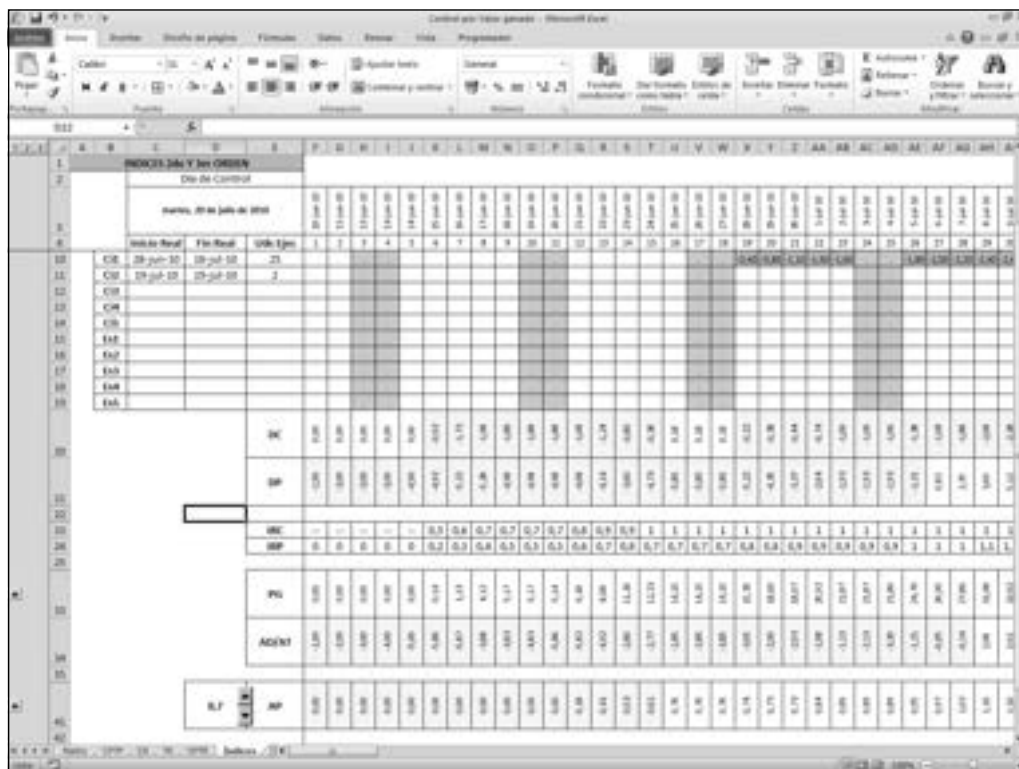


Figura 4.32. Hoja de índices de 2do y 3er orden.

4.4.5. Los gráficos de control por valor ganado en Excel 2010

Hasta ahora hemos visto como calcular los valores del control de proyecto mediante el valor ganado, pero estos valores deberemos de complementarlos con unos gráficos comparativos que permitan analizar rápidamente el avance y evolución de mi proyecto. No vamos a repetir el proceso de creación de gráficos que ya se expuso detalladamente en el capítulo anterior, procediendo exclusivamente a su enumeración y una breve explicación:

4.4.5.1. CPTP vs CRTR

Mediante este gráfico podremos analizar la evolución de los movimientos de caja reales *CRTR* comparándolo con las previsiones realizadas *CPTP*. Este gráfico no me aporta una gran información de la evolución real del

proyecto, pero puede ser útil para posibles análisis financieros de estos flujos de caja.

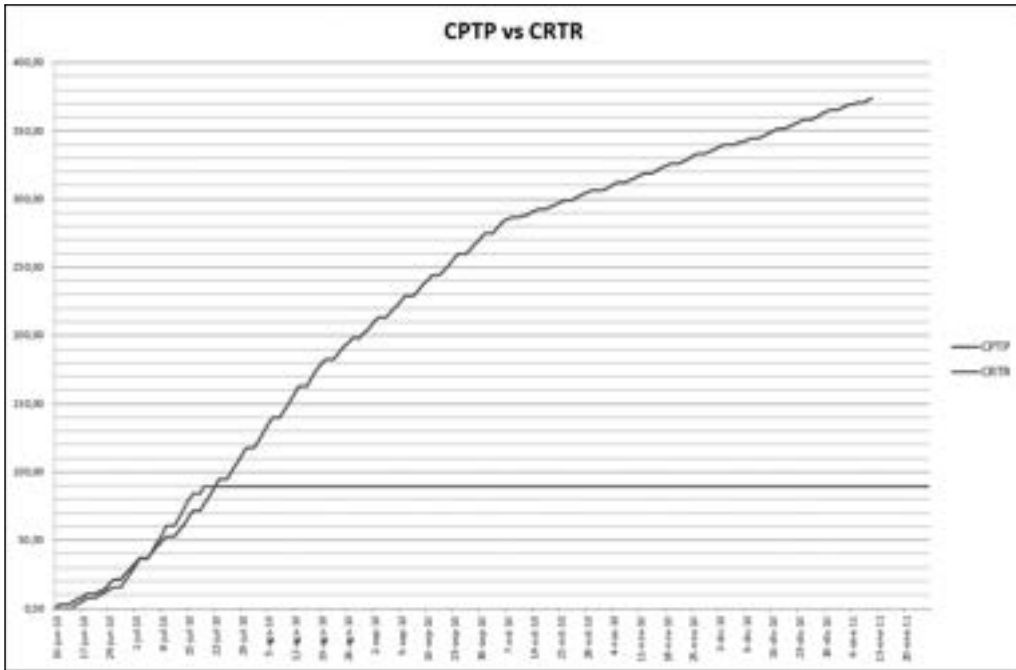


Figura 4.33. CPTP vs CRTR.

4.4.5.2. CPTP vs CPTR

Este gráfico comparativo del *CPTP* y del *CPTR* sí que ofrece gran información sobre la evolución del proyecto. Como se aprecia, las dos curvas evolucionan de modo diferente, estando la correspondiente al *CPTR* por debajo de la curva del *CPTP*, hasta que se cruzan en un punto, cambiando de posición a partir de entonces. Este punto de corte coincide con el punto de inflexión en la evolución de la *desviación de programa (DP)*, como puede ver en la hoja "Índices", indicándonos hasta donde el proyecto ha estado retrasado con respecto a las previsiones realizadas. Véase la figura 4.34.

4.4.5.3. Desviación de coste y programa

Este gráfico de *desviación de coste (DC)* y de *programa (DP)* tan solo es un reflejo de sus valores correspondientes en la hoja "Índices", no presentando ninguna dificultad su trazado. Véase la figura 4.35.

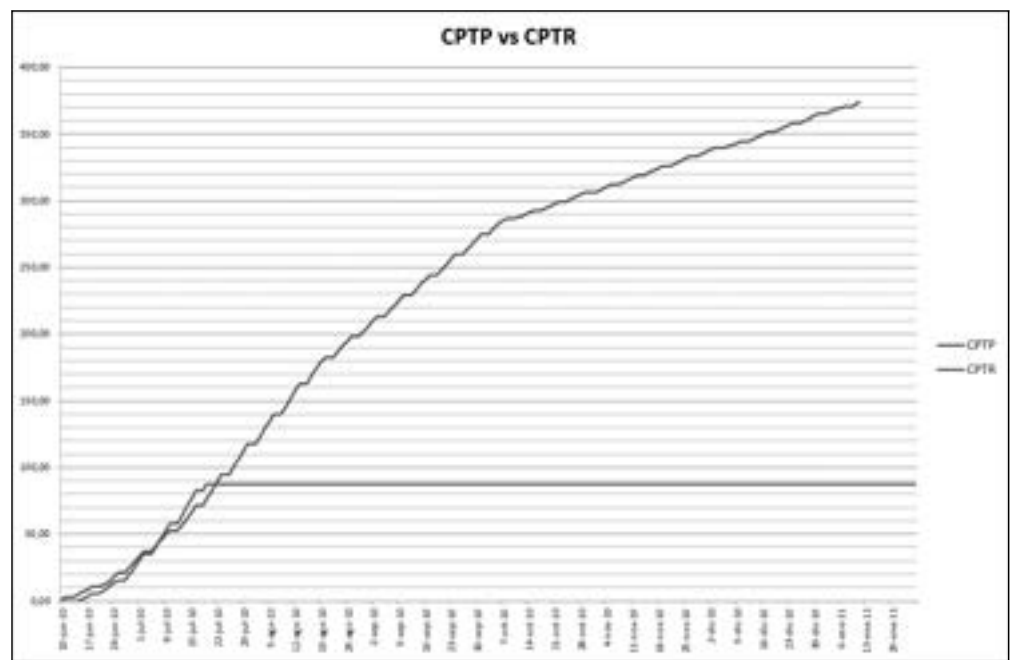


Figura 4.34. CPTP vs CPTR.



Figura 4.35. Desviación de coste y programa.

4.4.5.4. Adelantos y atrasos

Este gráfico, al igual que el anterior no presenta ninguna dificultad ni característica especial, tan solo recordar que los valores negativos corresponden a retrasos en el proyecto mientras que las positivos corresponden a adelantos.

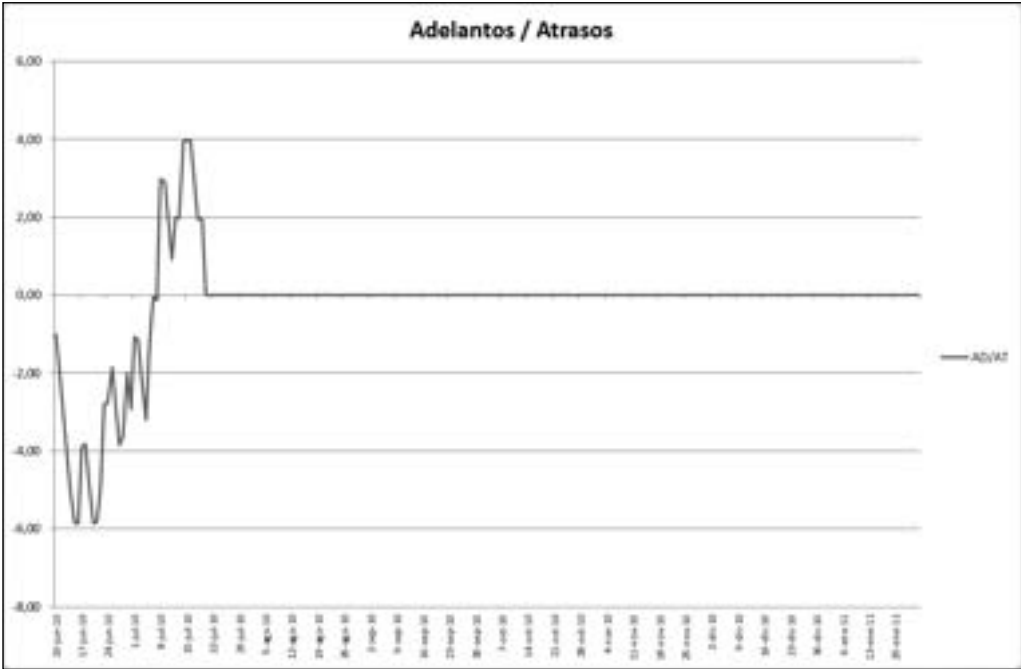


Figura 4.36. Adelantos y atrasos.

4.4.5.5. Índices de rendimiento de coste y de programa

En este gráfico de *índices de rendimiento de coste (IRC)* y de *programa (IRP)*, hemos de tener la precaución de que se ha de formato al eje vertical, indicándole que el horizontal lo cruza en valor de uno, que es la base de referencia de los índices.

NOTA

La hoja de cálculo completa se encuentra en Cap_04\Control por Valor ganado, de los ejemplos del libro.

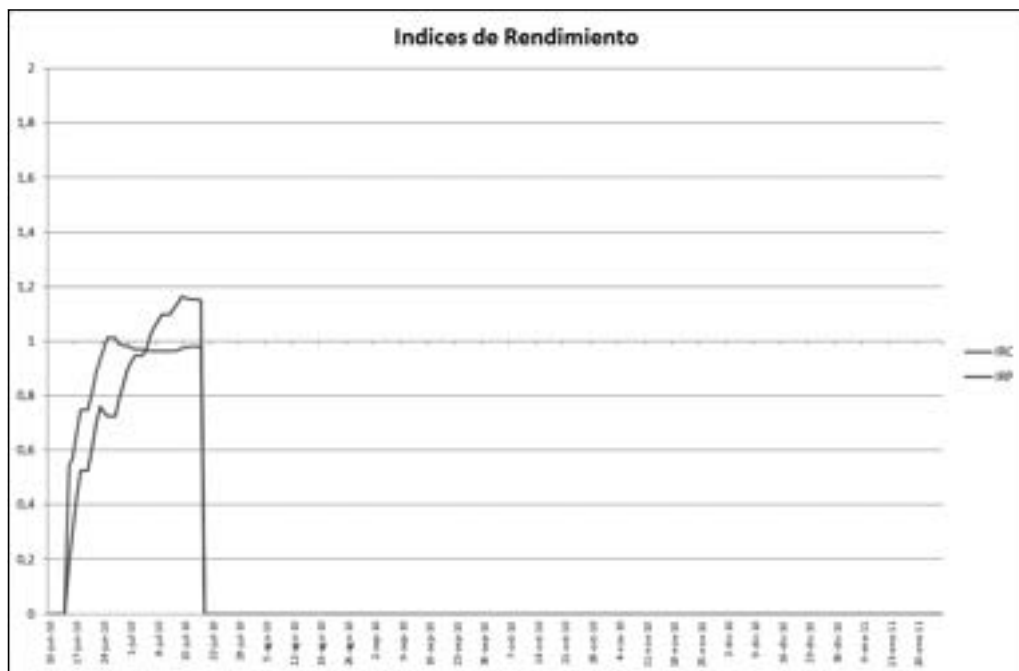


Figura 4.37. Índices de rendimiento de coste y de programa.

4.5. Anexo al control del proyecto

En este capítulo se han expuesto diferentes métodos de control, desde los más sencillos a los más evolucionados y que el *Project Management Institute*® califica como "*practica emergente*", por ser una metodología en constante evolución por parte de los profesionales e investigadores en gestión de proyectos.

Algunos lectores considerarán que las hojas que aparecen son demasiado impersonales, pero esa precisamente ha sido la intención, para hacerlas más legibles y fáciles de entender, además de evitar que las "manías" del que escribe estas líneas condicionen su personalización.

A partir de los próximos capítulos vamos a ver herramientas más complejas, basadas en modelos matemáticos y estadísticos más complejos que requerirán de un esfuerzo mayor por parte del lector, pero que seguro que serán de gran utilidad para comprender como funcionan (o no funcionan) los programas profesionales y así asomarnos a la profundidad en este inacabable mundo de la gestión de proyectos.

5

La gestión del riesgo en los proyectos

5.1. Breve historia de la gestión del riesgo de los proyectos

La gestión del riesgo en los proyectos es una preocupación tan antigua, que podríamos decir que nació junto con las modernas técnicas de gestión de proyectos que utilizamos actualmente.

Ya desde la década de los 50 del siglo XX, con el desarrollo del modelo *PERT* (*Program Evaluation and Review Technique*) por parte de Malcolm, Clark, Roseboom y Fazar, como miembros del equipo investigador de la marina de los EEUU en el proyecto Polaris, su preocupación era el determinar la probabilidad del cumplimiento de los plazos de entrega comprometidos.

Los trabajos de Malcon y Clark, supusieron un hito importantísimo, al utilizar herramientas estadísticas para determinar el proyecto entre unas fecha objetivo previamente establecidas. Estos trabajos han sido duramente criticados por las simplificaciones que se vieron obligados a asumir,

principalmente a causa de las limitaciones de cálculo de los ordenadores de la época. El método *PERT* estadístico ha evolucionado hacia metodologías más complejas, que en cierta medida solventan las simplificaciones que se vieron obligados a asumir sus creadores, pero siempre bajo la premisa de que los proyectos cumplen uno de los más famosos teoremas de la estadística y que se llama el "teorema del límite central".

De entre todas estas nuevas herramientas para el estudio y la gestión del riesgo en los proyectos, destacan de forma especialmente importante la simulación de Monte Carlo y la gestión de búferes mediante la *CCBM* (*Critical Chain-Buffer Management*) o "gestión de la cadena crítica" desarrollada por Goldratt en sus libros "La teoría de las restricciones" y "La cadena crítica", dos metodologías en principio diferentes, pero que unificaremos y desarrollaremos con detalle más adelante mediante su implementación en Excel.

Todas las metodologías anteriores están basadas en problemas estocásticos de incertidumbre, lo que lleva aparejado el uso de herramientas estadísticas con las ventajas e inconvenientes que esto supone. La gran ventaja es la enorme cantidad de información fiable que puedo obtener a partir de estas herramientas estadísticas, siempre que las distribuciones de probabilidad usadas en las simulaciones hayan sido rigurosamente estudiadas y analizadas, algo que en la práctica resulta enormemente difícil, por no decir que imposible.

Ante esta dificultad, está adquiriendo un gran auge la gestión del riesgo mediante la utilización de la lógica *fuzzy*, borrosa o difusa. Este nuevo enfoque iniciado por el ingeniero Norteamericano de origen iraní y profesor de la universidad de Berkeley, Lotfi A. Zadeh, ofrece como gran ventaja que ya no trabajamos en términos de incertidumbre, sino de imprecisión, no estando limitados por las rígidas reglas de la estadística, permitiendo realizar cualquier tipo de complejas operaciones con valores imprecisos, incluidos los evolucionados algoritmos de optimización metaheurística hasta hace poco limitados a problemas con valores ciertos y precisos.

5.2. El PERT tradicional

El método *PERT* fue la primera técnica pensada para evaluar el riesgo existente en el cumplimiento de los proyectos, y concretamente el de la construcción del submarino nuclear *Polaris*.

Sus creadores partieron de los siguientes principios:

1. La duración de las tareas del proyecto sigue la distribución de probabilidad beta, siendo su desviación típica (σ_i) un sexto de su intervalo a-b.
2. La duración del proyecto sigue una distribución normal cumpliéndose el "teorema del límite central".

Aunque el primero de los anteriores principios ha sido el elemento más controvertido, sus creadores lo defendieron, justificado en la singularidad del problema para el que había sido desarrollado, no siendo extrapolable a cualquier tipo de proyecto. El segundo de los puntos, sin embargo, es indiscutible (siempre que se cumplan ciertas condiciones), y nos permitirá determinar el riesgo de cumplimiento de nuestro proyecto aplicando la distribución normal para la media (μ_p) y desviación estándar (σ_p) obtenidas por medio de la simulación.

5.2.1. Las estimaciones del experto y la distribución beta

La duración de las tareas sigue una distribución de probabilidad beta, y su media (μ) será obtenida a partir de las denominadas estimaciones del experto:

- a. Duración optimista.
- b. Duración moda.
- c. Duración pesimista.

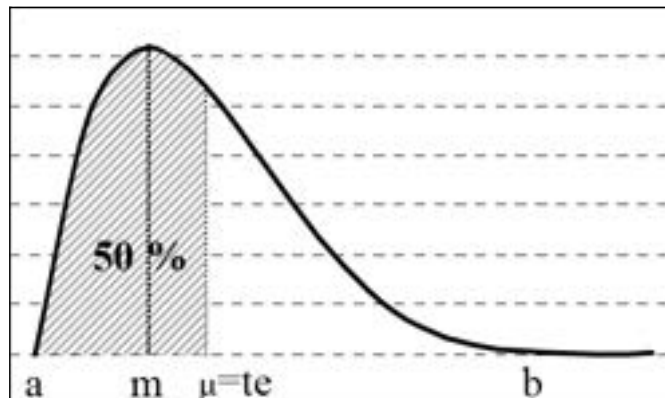


Figura 5.1. La distribución beta y las estimaciones del experto.

A partir de los anteriores valores estimados para cada una de las tareas, obtendremos la duración media (μ_i) o duración estimada (te_i) aplicando:

$$te = \mu = \frac{a + 4m + b}{6}$$

Figura 5.2. Duración media (μ_i) o estimada (te_i) en el PERT tradicional.

Y su varianza (σ_i^2):

$$\sigma^2 = \frac{(b - a)^2}{36}$$

Figura 5.3. Varianza de la tarea en el PERT tradicional.

Pero los anteriores valores no son los valores reales de la media (μ) y la varianza (σ^2) de la distribución beta, que como se puede ver a continuación es una función tetraparamétrica:

$$f(x) = \frac{(x - a)^{\alpha-1} (b - x)^{\beta-1}}{\beta(\alpha, \beta) (b - a)^{\alpha+\beta-1}} \quad \begin{array}{l} ; \alpha > 1; \\ ; \beta > 1 \\ ; a \leq m \leq b \end{array}$$

Figura 5.4. Función de densidad de la distribución beta.

Y los verdaderos para la media (μ) y la varianza (σ^2) son:

$$\mu = \frac{b \alpha + a \beta}{\alpha + \beta}$$

$$\sigma^2 = \frac{\alpha \beta (b - a)^2}{(\alpha + \beta)^2 (\alpha + \beta + 1)}$$

Figura 5.5. Media (μ) y varianza (σ^2) de la distribución beta.

El problema radica en que a partir de las tres estimaciones del experto existen infinitas funciones de distribución beta, y la utilización de los valores optimistas, moda y pesimista para su determinación es muy práctico

y está demasiado arraigado para su modificación. Como solución a este problema, el profesor R. Herrerías, de la Universidad de Valladolid propone el uso de un nuevo parámetro " k ", que denomina "*factor de confianza del experto*", que cumplirá:

$$k = \alpha + \beta - 2$$

Figura 5.6. Factor de confianza del experto.

De tal manera que siendo totalmente respetuoso con la tradición de las tres estimaciones del experto, se puede obtener el valor de " α " de la distribución beta en función de " k ":

$$\alpha = 1 + k \frac{m - a}{b - a}$$

Figura 5.7. Valor de " α " de la distribución beta.

Y el valor de " β ":

$$\beta = 1 + k \frac{b - m}{b - a}$$

Figura 5.8. Valor de " β " de la distribución beta.

A partir de " α " y " β ", el verdadero valor de la media (μ_i) o duración estimada (te_i) de la tarea con una distribución beta en función de " k ", será:

$$te = \mu = \frac{a + km + b}{k + 2}$$

Figura 5.9. Valor de la media (μ) o duración estimada (te) en función de " k ".

Y el de la varianza (σ_i^2) en función de " k ":

$$\sigma^2 = \frac{(\mu - a)(b - \mu)}{k + 3}$$

Figura 5.10. Valor de la varianza (σ_i^2) en función de " k ".

Para valores de " k " igual a cuatro, la media (μ_i) es la misma que la obtenida según el sistema tradicional, aunque la varianza (σ_i^2) será más

conservadora. El profesor R. Herrerías propone valores de " k " igual a dos, que *"proporciona distribuciones menos apuntaladas totalmente respetuosas con la información subjetiva del valor más probable proporcionado por el experto"*.

5.2.2. El teorema del límite central y la distribución Normal

A partir de los valores de las duraciones estimadas (te_i) de las tareas, Malcon y Clark calcularon nuevamente la duración del proyecto, que será precisamente la duración media del proyecto (μ_p) y cuya probabilidad de cumplimiento será del cincuenta por ciento, al obtenerse como la suma de las medias de las tareas (te_i). La varianza (σ_p^2) del proyecto, no será más que la suma de las varianzas (σ_i^2) de las tareas que conforman la ruta crítica, obteniendo su probabilidad de cumplimiento a partir de una distribución normal.

El anterior proceso, no es más que aplicar el *"teorema del límite central"*, que dice para nuestro caso en concreto:

"Independientemente de las distribuciones de probabilidad de las tareas, la duración de un proyecto, seguirá una distribución normal, y la varianza del proyecto (σ_p^2) será la suma de las varianzas (σ_i^2) de la ruta crítica, siempre que las actividades que forman la ruta crítica sea lo suficientemente grande".

La varianza (σ_p^2) del proyecto es la medida de dispersión por excelencia, y me sirve para determinar si los valores están más o menos dispersos con respecto a su media, y se define como *"la desviación cuadrática media de los valores de la muestra respecto a su media"*, siendo su formulación para una población de " n " valores " x_i " y una media de " μ ":

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2$$

Figura 5.11. Fórmula general para el cálculo de la varianza (σ^2).

El problema de la anterior formulación para el cálculo de la varianza (σ^2), radica en la necesidad de conservar los valores de las observaciones, algo que a priori no parece muy razonable por la gran cantidad de información que deberíamos de almacenar, por lo que haremos uso de la

siguiente formulación, equivalente a la anterior pero más práctica para nuestros fines:

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n x_i^2 - \mu^2$$

Figura 5.12. Fórmula específica para el cálculo de la varianza (σ^2).

La interpretación de la varianza (σ^2) no es muy intuitiva, porque se mide en las unidades métricas de la muestra elevadas al cuadrado, en nuestro caso días al cuadrado, por lo que haremos uso de la desviación típica o estándar (σ), que no es más que la raíz cuadrada de la varianza (σ^2):

$$\sigma = \sqrt{\sigma^2}$$

Figura 5.13. Cálculo de la desviación típica o estándar (σ).

Una vez obtenidas la duración media del proyecto (μ_p) y su desviación típica (σ_p), podemos hacer uso de los valores normalizados de la distribución normal estandarizada, determinando a cuantas desviaciones típicas (z) se encuentra la duración de proyecto de la que quiero determinar su probabilidad, con respecto a su duración media (μ_p), aplicando: $z = (T - \mu_p) / \sigma_p$.

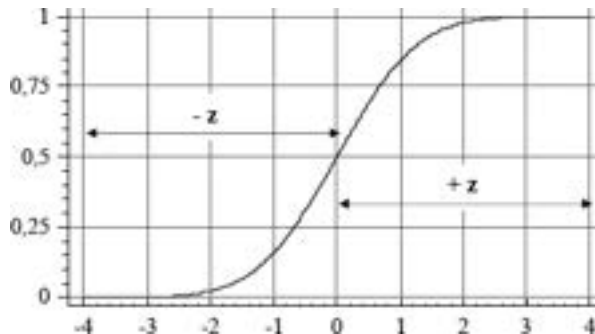


Figura 5.14. Curva de probabilidad normal estandarizada.

5.2.3. Los buffers de proyecto

En la distribución normal de probabilidad, si a la media (μ) le sumo una desviación estándar (σ) obtengo entonces como resultado un valor con una probabilidad aproximada del 84 por ciento, y si le sumo dos desviaciones

estándar (2σ) obtendré un nuevo valor con una probabilidad de más del 97 por ciento, de tal manera que para obtener el plazo del proyecto con una probabilidad de cumplimiento del 97 por ciento, tan solo tendré que sumarle dos desviaciones estándar ($2\sigma_p$) a la duración media simulada del proyecto (μ_p). Este valor aproximado de dos desviaciones estándar ($2\sigma_p$), lo utilizaremos para calcular la duración de una nueva tarea que añadiremos al final de nuestro proyecto y que denominaremos "*buffer de proyecto*", y que obtendremos de la siguiente forma:

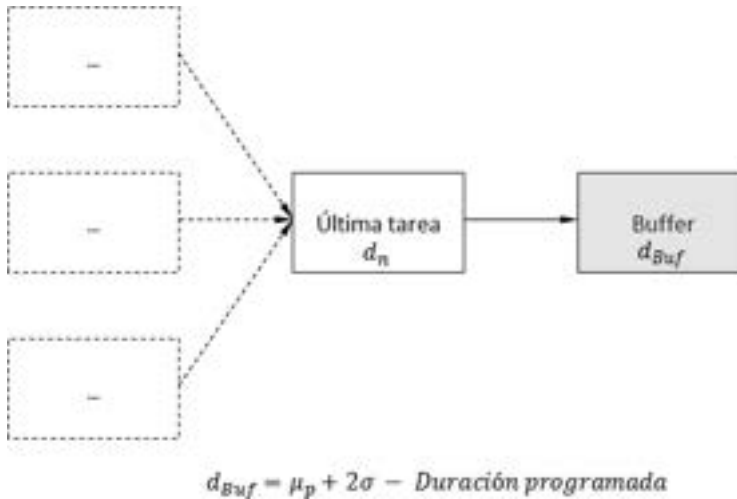


Figura 5.15. Cálculo del buffer de proyecto.

Esta nueva tarea, me servirá como un "amortiguador" que absorba las posibles desviaciones que se puedan producir durante la ejecución del proyecto, garantizando de esta manera que el plazo final de entrega al que nos comprometamos no se vea alterado.

5.2.4. Las limitaciones del PERT

Hemos visto como el *PERT* ha sido y es una valiosa herramienta para evaluar el riesgo existente en el cumplimiento del plazo de los proyectos, pero eso no significa que no haya sido objeto de numerosas críticas por las fuertes restricciones que se impone para su aplicación y análisis. Veamos cuales son las principales limitaciones y críticas que ha recibido:

1. Solo se considera la existencia de una ruta crítica (en la práctica, en caso de existir más una ruta crítica, se elegirá aquella que tenga una

varianza (σ_p^2) más desfavorable, para así estar del lado de la seguridad, aunque no es una solución aceptada por todos los autores).

2. No es justificable que las tareas solo puedan seguir una única distribución de probabilidad tipo β .
3. Subestima la duración total del proyecto por el "sesgo de unión de eventos" al ignorar las rutas cercanas a la criticidad en el cálculo del tiempo total de un proyecto.
4. Para aplicar el teorema del límite central, es requisito imprescindible que la cantidad de variables (exclusivamente las tareas de la única ruta crítica) sea muy grande. Algunos autores defienden que para poder aplicar el teorema del límite central se requiere como mínimo de 50 tareas en la ruta crítica, aunque llegan a rebajarlo a 30 en condiciones excepcionales.
5. No se tienen en cuenta las tareas no críticas, ni la posibilidad de que éstas puedan llegar a ser críticas en algún momento.

Como alternativa al *PERT*, y a sus limitaciones, disponemos de la simulación de Monte Carlo, técnica que mediante el muestreo aleatorio (generación de secuencias de números pseudoaleatorios), imita el comportamiento de un sistema (estático o dinámico) del mundo real, bien en un instante de tiempo determinado o mientras evoluciona a lo largo del tiempo.

5.3. La simulación de Monte Carlo

La simulación de Monte Carlo fue inventada por el estadístico británico Karl Pearson al descubrir la utilidad de los números aleatorios para resolver complejos problemas de probabilidad y estadística.

El nombre de Monte Carlo viene dado por la capital del principado de Mónaco y su famoso casino, al considerar a la ruleta como "el generador simple de números aleatorios". Karl Pearson dijo: *"El registro mensual del juego de la ruleta en Monte Carlo puede proporcionarnos material para el análisis de los fundamentos del conocimiento"*.

El proceso de la simulación es relativamente sencillo, consistiendo básicamente en calcular muchísimas veces un proyecto, obteniendo la duración media (μ_p) y desviación estándar (σ_p^2) de las simulaciones realizadas, generando aleatoriamente la duración de cada una de las tareas con diferentes distribuciones de probabilidad.

5.3.1. Las distribuciones de probabilidad

Las distribuciones de probabilidad que se pueden utilizar son muchísimas, estando limitados exclusivamente por las funciones disponibles en el lenguaje de programación utilizado, pero las distribuciones más comunes además de la ya conocida β , son la determinista, la uniforme y la triangular.

5.3.1.1. La distribución determinista

La distribución determinista no es una función de probabilidad propiamente dicha. Mediante el uso de esta función hacemos que la duración de la tarea sea un hecho cierto e invariable, no estando sujeta por lo tanto a incertidumbre.

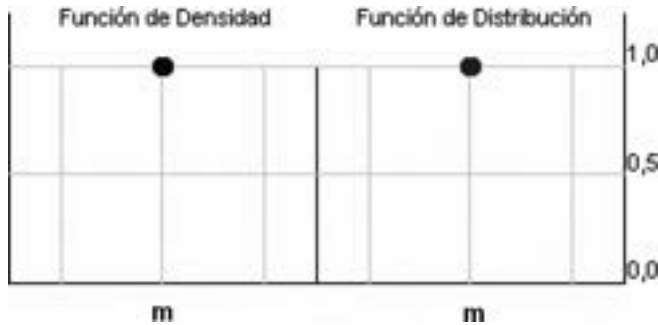


Figura 5.16. Distribución determinista.

Evidentemente la media (μ_i) será la propia duración moda (m), y su varianza (σ_i^2) será cero.

5.3.1.2. La distribución uniforme

La distribución uniforme proporciona la probabilidad de que la duración de la actividad esté dentro del intervalo $[a, b]$. Se emplea cuando no disponemos de ningún dato, y la probabilidad de la ocurrencia dentro de ese intervalo es directamente proporcional a la longitud del intervalo. Véase la figura 5.17.

Es una función constante sobre el intervalo $[a, b]$. Su media (μ_i) y su varianza (σ_i^2), a partir de su función de densidad serán las que se muestran en la figura 5.17.

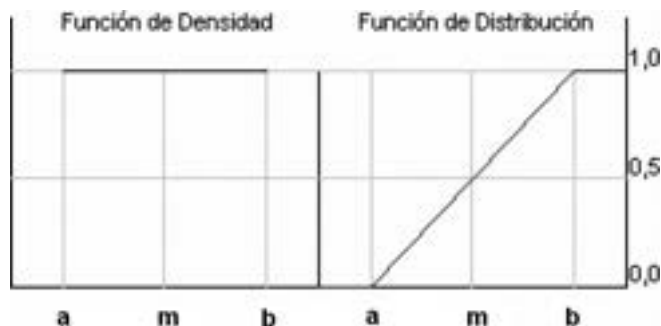


Figura 5.17. Distribución uniforme.

$$f(x) = \frac{1}{b - a}$$

$$\mu = \frac{a + b}{2} \quad \sigma^2 = \frac{(b - a)^2}{12}$$

Figura 5.18. Media (μ) y varianza (σ^2) de la distribución uniforme.

5.3.1.3. La distribución triangular

La distribución triangular se emplea cuando disponemos de pocos datos o como sustitutiva junto con la distribución trapezoidal (aunque esta última requiere de 4 parámetros) de la distribución β , por ser más sencilla de calcular. Al igual que la β , está definida por las tres informaciones subjetivas del experto de duración optimista (a), prevista o moda (m) y pesimista (b).

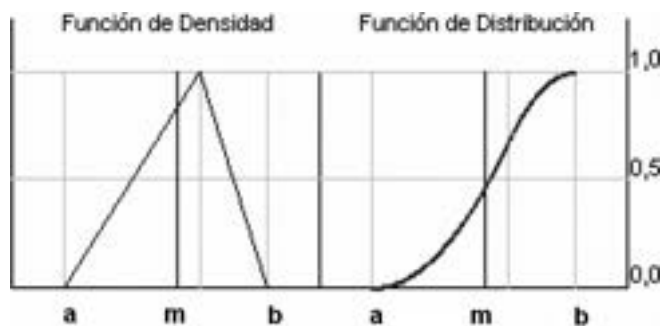


Figura 5.19. Distribución triangular.

La media (μ_i) y la varianza (σ_i^2), a partir de su función de densidad serán las siguientes:

$$\mu = \frac{a + b + m}{3}$$

$$\sigma^2 = \frac{a^2 + b^2 + m^2 + ab - am - mb}{18}$$

Figura 5.20. Media (μ_i) y varianza (σ_i^2) de la distribución triangular.

5.3.2. La generación de duraciones aleatorias

La simulación de las duraciones de las tareas se realiza a partir de la función de probabilidad acumulada y un generador de números aleatorios entre cero y uno, que correspondía a una probabilidad en tantos por uno. A partir de este número aleatorio, se determinaba el valor para el cual la probabilidad era precisamente ese número, de la siguiente forma:

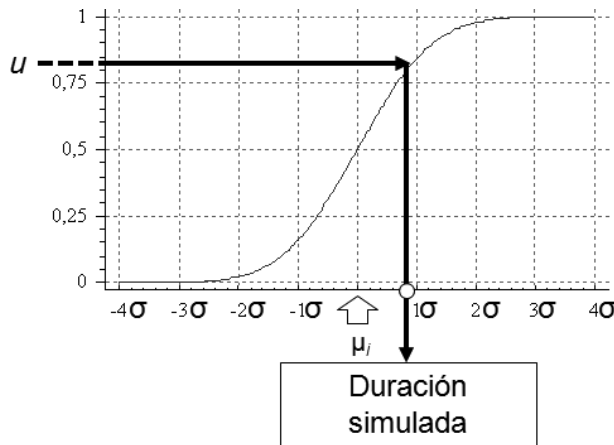


Figura 5.21. Generación de duraciones aleatorias.

La simulación de las duraciones con una distribución uniforme y triangular es relativamente sencillo, pero no así la de la distribución tipo β , pero en la nueva versión de Excel 2010, ya disponemos de la función inversa de esta distribución en Visual Basic, lo que facilita enormemente la creación del código.

5.3.2.1. La simulación de la distribución uniforme

Para simular la duración (ds) de una tarea según una distribución uniforme en un intervalo [a, b], a partir de un número aleatorio (u), escribiremos el siguiente código:

```
u = Rnd
ds = a + u * (b - a)
```

5.3.2.2. La simulación de la distribución triangular

Para simular la duración (ds) de una tarea según una distribución triangular en un intervalo [a, b] y una moda (m), a partir de un número aleatorio (u), escribiremos el siguiente código:

```
u = Rnd
Select Case (m - a) / (b - a)
Case Is >= u
    ds = a + Sqr(u * (b - a) * (m - a))
Case Else
    ds = a + Sqr((1 - u) * (b - a) * (b - m))
End Select
```

5.3.2.3. La simulación de la distribución β

La simulación de la duración (ds) de una tarea según una distribución β en un intervalo [a, b] y una moda (m), a partir de un número aleatorio (u), es más sencillo que los anteriores, y escribiremos el siguiente código:

```
u = Rnd
alfa = 1 + k * ((m - a) / (b - a))
beta = 1 + k * ((b - m) / (b - a))
ds = WorksheetFunction.Beta_Inv(u, alfa, beta, a, b)
```

Donde podemos apreciar que los valores alfa (α) y beta (β) de la forma, son obtenidos mediante la aplicación de la formulación de Herrerías y el coeficiente "k".

5.3.3. El índice de criticidad y el valor crítico

Con la simulación de Monte Carlo, ya hemos dejado atrás los conceptos de de tareas críticas y no críticas, así como los de rutas críticas y no críticas exclusivos de los problemas deterministas. Ahora podemos conocer cuantas veces ha sido crítica una tarea a lo largo de la simulación. Ya no nos interesan exclusivamente las tareas críticas, sino que gracias a la nueva

información obtenida derivamos nuestro interés a la probabilidad de que una tarea sea crítica y de que pertenezca a una ruta crítica. Disponemos de dos nuevos parámetros para medir el riesgo de nuestras tareas y de nuestro proyecto, y que son:

5.3.3.1. El índice de criticidad

Porcentaje que nos indica cuantas veces ha sido critica una tarea durante la simulación (en tantos por uno o tantos por cien).

5.3.3.2. El valor crítico

Es el índice de criticidad (en tantos por uno) multiplicado por la varianza de la duración de la tarea. Este valor nos indica la posibilidad que tiene una tarea de estar en el camino crítico, así como su grado de variabilidad en las duraciones posibles.

5.3.4. La simulación de Monte Carlo con Excel 2010

Antes de proceder con el código que se expone a continuación, deberemos de preparar nuestro libro de Excel, y como siempre partiremos de un libro que ya tenemos creado en un capitulo anterior (en este caso "Diagrama temporal con calendario laboral" del capítulo 3) y lo ajustaremos a nuestras necesidades con unas pocas modificaciones.

Una vez tenemos ya preparado el nuevo libro, insertaremos una hoja entre las hojas "Matriz" y "Diagrama temporal", a la que llamaremos "Simulador", y que deberá tener el aspecto que muestra la figura 5.22.

Para facilitar las operaciones en la hoja y en el código de Visual Basic, se han incorporado unos nuevos nombres a las celdas más importantes, y que son:

Nombre	Ubicación de la celda
Desviación_Simulada	=Simulador!\$G\$4
Duración_Inicial	=Simulador!\$G\$2
Duración_Simulada	=Simulador!\$G\$3

Nombre	Ubicación de la celda
factor_k	=Simulador!\$C\$2
Inicio_sim	=Simulador!\$B\$13
Sim	=Simulador!\$D\$3
Simulaciones	=Simulador!\$C\$3

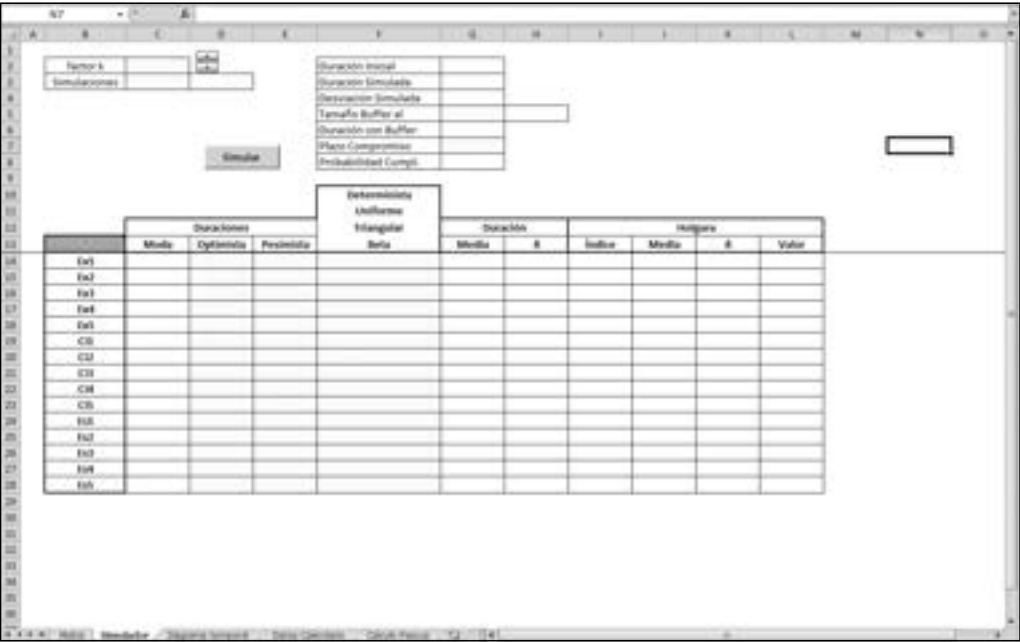


Figura 5.22. Hoja "Simulador".

El código está preparado para que solamente sea necesario escribir las duraciones optimista y pesimista de cada tarea, tomando las duraciones moda directamente de la hoja "Matriz", y por supuesto, necesitaremos los parámetros básicos de la simulación, como el valor del "factor de confianza" de β , el número de simulaciones que deseamos realizar, la probabilidad de cumplimiento para determinar el tamaño del Buffer que se ha llamado "Tamaño buffer al" y el plazo disponible o compromiso.

No se ha comentado nada sobre escribir el tipo de distribución de cada tarea, porque no va a hacer falta. Para evitar los errores tipográficos tan comunes en los que reiteradamente caemos, haremos uso de una función muy practica de Excel, ya disponible en versiones anteriores y que es la de

elegir valores de una lista. Para ejecutarla, nos ubicáremos en la columna de distribución de la primera tarea, abriremos el menú contextual con el botón derecho>Elegir de la lista desplegable..., apareciendo los valores elegibles y que son los que tenemos en la cabecera de la columna:

	Duraciones			Determinista Uniforme Triangular Beta
	Moda	Optimista	Pesimista	
Ex1	5	7	15	Uniforme
Ex2	10	7	15	Beta
Ex3	10	7	15	Determinista
Ex4	10	7	15	Triangular
				Uniforme

Figura 5.23. Elegir tipo de distribución de la tarea.

Tan solo tres celdas están formuladas, y son:

Celda	Fórmula
H5	=INV.NORM(G5;Duración_Simulada;Desviación_Simulada)-Duración_Inicial
G6	=H5+G2
G8	=DISTR.NORM.N(G7;Duración_Simulada;Desviación_Simulada;VERDADERO)

Y que no son más que aplicar las formulas que Excel no proporciona para aplicar los conceptos que se han visto a lo largo de este capítulo.

Ya solo quedará añadir el botón de control ActiveX y añadir el siguiente código:

```
Dim i As Integer
Dim fila_m As Integer
Dim columna_m As Integer
Dim fila_s As Integer
Dim columna_s As Integer
Dim Num_tareas As Integer
Dim sim As Integer
Dim Simulaciones As Integer
Dim Duración_Proy As Single
Dim Suma_Duración_Proy As Single
Dim Suma_Cuadrados_Duración_Proy As Single
Dim Duración_Media_Proy As Single
Dim Desviación_Proy As Single
Dim Dur_ini() As Single
Dim Sum_Dur() As Single
Dim Sum_cuad_Dur() As Single
```

```

Dim Criticidad() As Single
Dim Sum_Holg() As Single
Dim Sum_cuad_Holg() As Single
Dim d_o As Integer
Dim d_m As Integer
Dim d_p As Integer
Dim d_s As Single
Dim u As Single
Dim k As Integer
Dim alfa As Single
Dim beta As Single
'Guarda los valores iniciales
With Worksheets("Matriz")
    'Establecemos la posición de la celda Inicio
    fila_m = .Range("Inicio").Row
    columna_m = .Range("Inicio").Column
    'Calcula el número de tareas
    Do
        If .Cells(fila_m + Num_tareas + 1, columna_m).Value = "" Then Exit Do
        Num_tareas = Num_tareas + 1
    Loop
    ReDim Dur_ini(Num_tareas)
    ReDim Sum_Dur(Num_tareas)
    ReDim Sum_cuad_Dur(Num_tareas)
    ReDim Criticidad(Num_tareas)
    ReDim Sum_Holg(Num_tareas)
    ReDim Sum_cuad_Holg(Num_tareas)

    'Establece la duración inicial del proyecto
    'y guarda los valores iniciales de las duraciones
    For i = 1 To Num_tareas
        Dur_ini(i) = .Cells(fila_m + i, columna_m + 1).Value
        If .Cells(fila_m + i, columna_m + 5).Value > Duración_Proj Then
            Duración_Proj = .Cells(fila_m + i, columna_m + 5).Value
        End If
    Next
    Worksheets("Simulador").Range("Duración_Inicial").Value = Duración_Proj
End With

With Worksheets("Simulador")
    'Establecemos la posición de la celda Inicio
    fila_s = .Range("Inicio_sim").Row
    columna_s = .Range("Inicio_sim").Column
    'Establecemos el número de simulaciones a realizar
    Simulaciones = .Range("simulaciones").Value
    'Establecemos el valor de k
    k = .Range("factor_k").Value
    'Establece la duración moda
    For i = 1 To Num_tareas
        .Cells(fila_s + i, columna_s + 1).Value = Dur_ini(i)
    Next
    'Limpia los valores iniciales de la hoja
    For i = 1 To Num_tareas
        .Cells(fila_s + i, columna_s + 5).Value = ""
        .Cells(fila_s + i, columna_s + 6).Value = ""
        .Cells(fila_s + i, columna_s + 7).Value = ""
    Next
End With

```

```

.Cells(fila_s + i, columna_s + 8).Value = ""
.Cells(fila_s + i, columna_s + 9).Value = ""
.Cells(fila_s + i, columna_s + 10).Value = ""
Next
.Range("Desviación_Simulada").Value = 0
'*****
'Simulación de Mote Carlo
'*****
For sim = 1 To Simulaciones
    .Range("sim").Value = sim
    'Simula la duración de las areas
    For i = 1 To Num_tareas
        'Lee las duraciones moda, optimista y pesimista
        d_m = .Cells(fila_s + i, columna_s + 1).Value
        d_o = .Cells(fila_s + i, columna_s + 2).Value
        d_p = .Cells(fila_s + i, columna_s + 3).Value
        'Genera el nmero aleatório
        u = Rnd
        '*****
        'Simula las duraciones de las tareas
        '*****
        Select Case .Cells(fila_s + i, columna_s + 4).Value
            Case "Determinista"
                d_s = d_m
            Case "Uniforme"
                d_s = d_o + u * (d_p - d_o)
            Case "Triangular"
                Select Case (d_m - d_o) / (d_p - d_o)
                    Case Is >= u
                        d_s = d_o + Sqr(u * (d_p - d_o) * (d_m - d_o))
                    Case Else
                        d_s = d_o + Sqr((1 - u) * (d_p - d_o) *
                            (d_p - d_m))
                End Select
            Case "βeta"
                alfa = 1 + k * ((d_m - d_o) / (d_p - d_o))
                beta = 1 + k * ((d_p - d_m) / (d_p - d_o))
                d_s = WorksheetFunction.Beta_Inv(u, alfa, beta, d_o, d_p)
        End Select
        Worksheets("Matriz").Cells(fila_m + i, columna_m + 1).Value =
            Round(d_s, 2)
        'Sumatorios para calcular la media y desviación de la duración
        'de las tareas
        Sum_Dur(i) = Sum_Dur(i) + d_s
        Sum_cuad_Dur(i) = Sum_cuad_Dur(i) + d_s ^ 2
        'Sumatorios para calcular la criticidad, holgura media y desviación
        'de las tareas
        Sum_Holg(i) = Sum_Holg(i) + Worksheets("Matriz").Cells(fila_m + i,
            columna_m + 6).Value
        Sum_cuad_Holg(i) = Sum_cuad_Holg(i) + Worksheets("Matriz").Cells(
            fila_m + i, columna_m + 6).Value ^ 2
        If Worksheets("Matriz").Cells(fila_m + i,
            columna_m + 6).Value <= 0 Then
            Criticidad(i) = Criticidad(i) + 1
        End If
    End For
End For

```

```

Next
'Calcula la duración simulada y media del proyecto
Duración_Proj = 0: For i = 1 To Num_tareas
    If Worksheets("Matriz").Cells(fila_m + i, columna_m + 5).Value >
        Duración_Proj Then
        Duración_Proj = Worksheets("Matriz").Cells(fila_m + i,
            columna_m + 5).Value
    End If
Next
.Range("Duración_Simulada").Value = Duración_Proj
'Sumatorios para calcular la media y desviación del proyecto
Suma_Duración_Proj = Suma_Duración_Proj + Duración_Proj
Suma_Cuadrados_Duración_Proj = Suma_Cuadrados_Duración_Proj +
    Duración_Proj ^ 2
'Calcula la duración media y desviación del proyecto
If sim > 1 Then
    Duración_Media_Proj = Suma_Duración_Proj / sim
    Desviación_Proj = Sqr((Suma_Cuadrados_Duración_Proj / sim) -
        Duración_Media_Proj ^ 2)
    .Range("Duración_Simulada").Value = Duración_Media_Proj
    .Range("Desviación_Simulada").Value = Desviación_Proj
End If
Next
'Calcula la duración media y desviación de las tareas
For i = 1 To Num_tareas
    .Cells(fila_s + i, columna_s + 5).Value = Sum_Dur(i) / Simulaciones
    .Cells(fila_s + i, columna_s + 6).Value = Sqr((Sum_cuad_Dur(i) /
        Simulaciones) - (Sum_Dur(i) / Simulaciones) ^ 2)
Next
'Calcular la criticidad, holgura media y desviación de las tareas
For i = 1 To Num_tareas
    'Índice crítico
    .Cells(fila_s + i, columna_s + 7).Value = Criticidad(i) /
        Simulaciones
    'holgura media y desviación de las tareas
    .Cells(fila_s + i, columna_s + 8).Value = Sum_Holg(i) /
        Simulaciones
    .Cells(fila_s + i, columna_s + 9).Value = Sqr((Sum_cuad_Holg(i) /
        Simulaciones) - (Sum_Holg(i) / Simulaciones) ^ 2)
    'Valor crítico
    .Cells(fila_s + i, columna_s + 10).Value = (Criticidad(i) /
        Simulaciones) * .Cells(fila_s + i, columna_s + 6).Value ^ 2
Next
End With

'Establece las duraciones iniciales en las medias simuladas
For i = 1 To Num_tareas
    Worksheets("Matriz").Cells(fila_m + i, columna_m + 1).Value = Dur_ini(i)
Next

```

La pregunta más frecuente una vez que estamos preparados para ejecutar nuestra simulación es ¿Cuántas simulaciones se han de realizar?

Un sistema para determinar el número de simulaciones a realizar o el tamaño de la muestra en términos estadísticos, consiste en realizar una

simulación previa de por ejemplo mil iteraciones para establecer aproximadamente el tamaño de la varianza (σ_p^2), y aplicar:

$$n = \frac{z^2 \sigma^2}{e^2}$$

Figura 5.24. Número de simulaciones a realizar.

Donde "e" será el error máximo de la simulación (normalmente 0,05), y "z" obtenido a partir del nivel de confianza deseado, que podrá adoptar los siguientes valores:

Nivel de confianza	z
75	0,52
80	0,84
85	1,03
90	1,28
95	1,64
97	1,89

5.3.5. El diagrama temporal con buffer de proyecto

Una vez finalizada la simulación podemos terminar incluyendo una nueva tarea para el Buffer a nuestro diagrama temporal, insertando una nueva fila al final de todas las tareas, donde las fórmulas para las fechas de inicio y final serán respectivamente:

```
=DIA.LAB.INTL(Fecha_Inicio;MAX(Matriz!G5:G19);Cod_Fin_D;Festivos)

=DIA.LAB.INTL(Fecha_Inicio;Simulador!G6;Cod_Fin_D;Festivos)
```

Y para que destaque sobre el resto de tareas le cambiaremos el color verde del formato condicional por otro más llamativo, por ejemplo naranja, quedando de la siguiente forma:

y falsa, es decir, que tiene un cierto grado de verdad y simultáneamente y de forma complementaria un cierto grado de falsedad. Un día no solo es nublado o soleado, sino que puede ser parcialmente nuboso y parcialmente soleado.

Estas afirmaciones anteriores, que en nuestro lenguaje natural resultan tan comunes y cotidianas sin ninguna dificultad para su asimilación, eran todo un reto para la lógica clásica y su dualidad dicotómica, donde solo podían adoptar los valores de cero o uno como representación de la falsedad y la verdad, pero veamos cómo funciona, utilizando el ejemplo del se valía Zadeh para su explicación:

Supongamos que queremos clasificar a los hombres según su estatura, parece lógico afirmar que son hombres altos aquellos que miden más de 1,80 metros de altura, así que la lógica clásica representaría las estaturas de la siguiente forma:



Figura 5.26. Alturas según la lógica clásica.

Pero nosotros, haciendo uso de nuestra lógica natural, iríamos un poco más allá, y nos preguntaríamos ¿Qué pasa con aquellos hombres que miden 1,79 metros? ¿Son hombres bajos?

Evidentemente que no, pero la lógica clásica no es capaz de resolver este problema tan sencillo y decir con un lenguaje natural: "son un poco menos altos".

Podemos ampliar el problema y además afirmar que aquellos hombres que miden menos de 1,70 metros, no pertenecen al conjunto de hombre altos, de tal forma que aparece un subconjunto de valores que pertenece parcialmente al conjunto de hombres altos con ciertos "*grados de pertenencia*", y un hombre que mida 1,75 metros tendrá una pertenencia de 0,5 al conjunto de hombres altos, pero también y de forma complementaria una pertenencia de 0,5 al conjunto de hombres no altos. Véase la figura 5.27.

Esta diferente pertenecía es " α " o el *grado de pertenencia* a un determinado conjunto. Una vez definido el conjunto, será fácil determinar los valores que cumplen un determinado nivel o *grado de pertenencia*, que es lo que se llama " α -corte" del conjunto, de tal forma que un " α -corte = 0,5",

nos proporcionará el subconjunto {1.75, 1.76, 1.77, 1.78,...}, que son los que cumplen como mínimo un determinado grado de pertenencia.

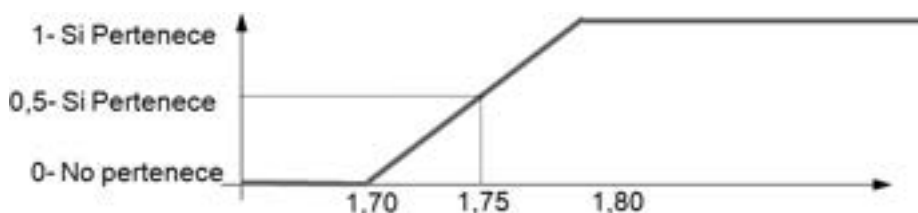


Figura 5.27. Alturas según la lógica fuzzy.

La función de pertenencia a un conjunto es lo que se denomina *número difuso*, *borroso* o *fuzzy*, y puede adoptar cualquier forma. Por simplicidad y similitud con las tres estimaciones del experto, no hemos centrado exclusivamente en números difusos del tipo triangular o *TFN* (*Triangular fuzzy number*). Se ha de dejar bien claro que la lógica difusa no es un problema estadístico, aunque al principio parece presentar aspectos comunes. La lógica difusa trabaja a niveles de imprecisión, no de incertidumbre, no utiliza probabilidades de cumplimiento, sino grados de cumplimiento, simplificándolo mucho podríamos afirmar que trabaja con grados de error en las estimaciones.

5.4.1. El número difuso o fuzzy

Número difuso es un conjunto difuso, convexo y normal con función de pertenencia continua a trozos.

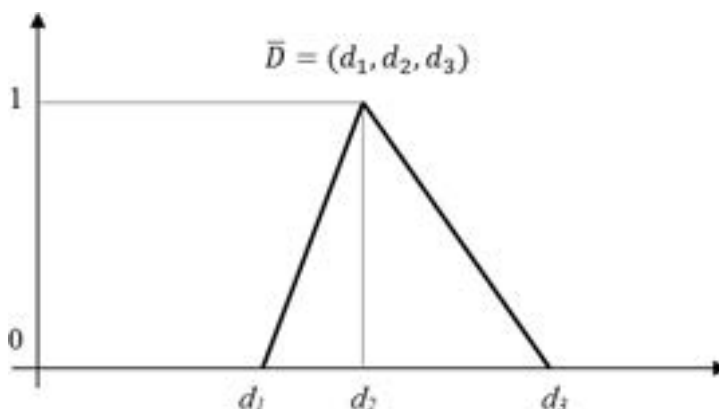


Figura 5.28. Número difuso o fuzzy.

5.4.1.1. El núcleo del número difuso o fuzzy

Núcleo del número difuso (d_2) es el valor para el cual la función de pertenencia de los elementos del conjunto difuso es la unidad.

5.4.1.2. El soporte inferior del número difuso o fuzzy

Soporte inferior del número difuso (d_1) es el valor inferior para el cual la función de pertenencia de los elementos del conjunto difuso es cero.

5.4.1.3. El soporte superior del número difuso o fuzzy

Soporte superior del número difuso (d_3) es el valor superior para el cual la función de pertenencia de los elementos del conjunto difuso es cero.

5.4.2. El alfa-corte $S_\alpha()$

Es el subconjunto del conjunto difuso en el que sus elementos tienen una función de pertenencia mayor o igual que α .

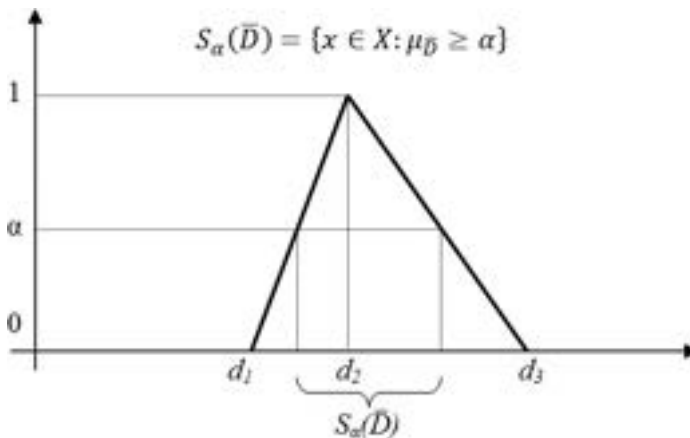


Figura 5.29. El *alfa-corte* $S_\alpha()$ de un número difuso o fuzzy.

El *alfa-corte* $S_\alpha()$, es quizás el concepto más importante de los números difusos, porque mediante el ajuste del valor α , podremos determinar el rango o conjunto de valores que satisfacen un determinado *grado de pertenencia*, o expresado de otra manera nivel de satisfacción, precisión del resultado o robustez del modelo.

El ajuste del El *alfa-corte* $S_\alpha()$ nos permitirá determinar entre qué valores se encontrarán los tiempos de ejecución de las tareas, ajustando consecuentemente la robustez del programa del proyecto.

$$\bar{D} = \{d_1, d_2, d_3\}$$

$$S_\alpha(\bar{D}) = \{d_1 + \alpha \cdot (d_2 - d_1), d_3 - \alpha \cdot (d_3 - d_2)\}$$

Figura 5.30. Cálculo del subconjunto del *alfa-corte* $S_\alpha()$.

5.4.3. La aritmética difusa o fuzzy

Al igual que con los números reales, con los números difusos también podemos realizar las operaciones aritméticas necesarias para resolver cualquier tipo de problema o ecuaciones que se puedan plantear.

Dados dos números difusos, podemos necesitar sumarlos, restarlos o bien dividirlos, para lo cual haremos uso del *alfa-corte* $S_\alpha()$, de la siguiente forma:

$$S_\alpha(\bar{A} + \bar{B}) = [S_\alpha(a_1 + b_1), a_2 + b_2, S_\alpha(a_3 + b_3)]$$

$$S_\alpha(\bar{A} - \bar{B}) = [S_\alpha(a_1 - b_3), a_2 - b_2, S_\alpha(a_3 - b_1)]$$

$$S_\alpha(\bar{A} * \bar{B}) = [S_\alpha(a_1 * b_1), a_2 * b_2, S_\alpha(a_3 * b_3)]$$

$$S_\alpha(\bar{A} / \bar{B}) = [S_\alpha(a_1 / b_3), a_2 / b_2, S_\alpha(a_3 / b_1)]$$

Figura 5.31. Operaciones aritméticas difusas o fuzzy.

La anterior formulación es una simplificación de la formulación general totalmente correcta si los valores del núcleo y de los soportes de ambos números difusos son mayor o igual a cero, circunstancia que se cumple con las duraciones de las tareas.

5.4.4. La ordenación de números difusos

Cuando aplicamos los algoritmos de cálculo de los tiempos de las tareas, y tenemos dos valores de tiempo difusos y queremos saber cuál es el mayor y cuál es el menor, y así adoptar el mayor en el *forward pass* y el menor en el *backward pass*, el problema puede ser muy sencillo o bien ligeramente más difícil.

Dados dos números difusos $A(a_1, a_2, a_3)$ y $B(b_1, b_2, b_3)$:

Sí, $a_1 < b_1, a_2 < b_2, a_3 < b_3 \rightarrow A < B$

Pero cuando nos encontramos con dos números difusos como los siguientes, la solución no es tan inmediata:

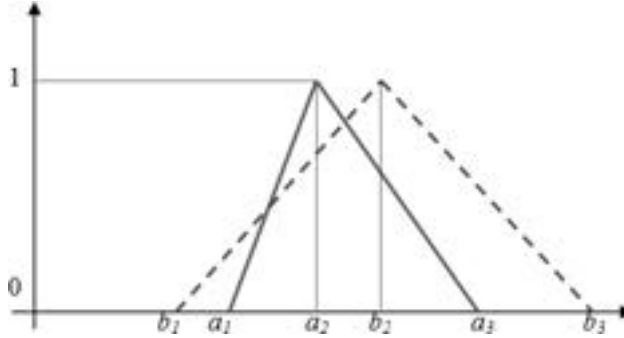


Figura 5.32. Ordenación de números difusos.

Para su solución existen muy diversas técnicas, pero la que más me gusta es la de ordenación baricéntrica, que ordena los números en función del valor de la ordenada a la que esté su baricentro, y su formulación será:

$$\text{Baricentro}(\bar{A}) = \frac{a_1 + 2 \cdot a_2 + a_3}{4}$$

$$\text{Baricentro}(\bar{A}) = \frac{a_1 + 2 \cdot a_2 + a_3}{4}$$

$$\text{si} \begin{cases} \text{Baricentro}(\bar{A}) < \text{Baricentro}(\bar{B}) \leftrightarrow \bar{A} < \bar{B} \\ \text{Baricentro}(\bar{A}) > \text{Baricentro}(\bar{B}) \leftrightarrow \bar{A} > \bar{B} \\ \text{Baricentro}(\bar{A}) = \text{Baricentro}(\bar{B}) \begin{cases} a_3 > b_3 & \leftrightarrow \bar{A} > \bar{B} \\ a_3 = b_3; a_1 < b_1 & \leftrightarrow \bar{A} < \bar{B} \\ a_3 < b_3 & \leftrightarrow \bar{A} < \bar{B} \end{cases} \end{cases}$$

Figura 5.33. Ordenación baricéntrica.

5.4.5. La criticidad difusa

Recordaremos que los valores ya no pertenecen con carácter de exclusividad a un determinado conjunto, sino que tienen grados de pertenencia a diferentes conjuntos. Lo mismo se puede decir de las tareas, estas ya no

pertenecen al conjunto de tareas críticas o no críticas, sino que pertenecen simultáneamente a ambos conjuntos, con un cierto grado de pertenencia a cada uno de ellos y complementarios entre sí.

Para determinar la criticidad de una tarea aparece un nuevo concepto que es el *grado de criticidad* (CD_i) de la tarea, y que no es más que el valor del *alfa-corte* $S_\alpha()$ a partir del cual la tarea comienza a ser crítica, y se determina fácilmente mediante semejanza de triángulos de la siguiente forma:

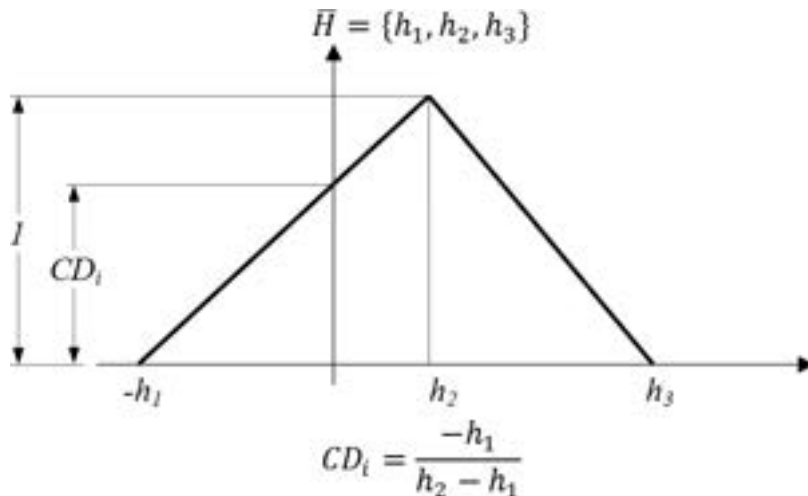


Figura 5.34. Grado de criticidad.

Si el núcleo de la holgura es menor o igual a cero, el grado de criticidad será la unidad, y si el soporte inferior es mayor que cero el grado de pertenencia será de cero. En el caso de que el soporte inferior sea menor que cero y su núcleo mayor que cero, el grado de criticidad lo obtendremos mediante el cociente entre el soporte inferior y la diferencia entre el núcleo y el soporte inferior cambiado de signo.

5.4.6. Implementación de la lógica difusa en Excel

La implementación que se propone a continuación de la gestión de riesgos de proyectos con duraciones difusas, es una solución simplificada con el fin de no modificar los códigos que expusimos en el capítulo 3 para el cálculo de los tiempos de las tareas.

La simplificación consiste en calcular tres veces el proyecto, una vez para los valores del núcleo de las tareas, y otras dos veces para los valores del soporte, obteniendo los valores núcleo y soporte de los tiempos de las tareas, y aplicando posteriormente el *alfa-corte* $S_\alpha()$ a estos valores, en vez de operar directamente con los valores difusos.

Las diferencias que pueden existir siempre serán del lado de la seguridad, y tenemos la ventaja de que la hoja "Matriz" y su código asociado no ha sufrido ninguna modificación.

Tal y como hicimos con la simulación de Monte Carlo, empezaremos con el mismo libro "Diagrama temporal con calendario laboral" del capítulo 3, y haremos unas pocas modificaciones.

El primer cambio que haremos, será insertar una hoja que llamaremos "fuzzy" entre las hojas "Matriz" y "Diagrama temporal", y crearemos los siguientes nombres:

Nombre	Ubicación de la celda
alfa_corte	=fuzzy!\$D\$2
Inicio_fuzzy	=fuzzy!\$B\$8
mk_inf	=fuzzy!\$H\$2
mk_nuc	=fuzzy!\$H\$3
mk_sup	=fuzzy!\$H\$4

La hoja "fuzzy" es muy sencilla, y no contiene fórmulas en sus celdas, porque el código proporcionado se encarga de formular adecuadamente cada una de ellas, y tiene el aspecto que muestra la figura 5.35.

Tal como se puede observar, todas las fórmulas que crea el código están referenciadas a la celda que hemos llamado "alfa_corte", que a su vez se ajusta con un control de formulario. Esto nos permite obtener los tiempos de inicio y finalización para cualquier *alfa-corte* $S_\alpha()$ sin necesidad de volver a calcular el proyecto.

Otro detalle es, que en este caso, los valores de las duraciones de las tareas son totalmente independientes de los que contiene la matriz, pudiendo ajustar tanto el núcleo de la duración como los soportes, que a tal fin aparecen como sombreados de la forma ya habitual (recordemos que en la simulación de Montecarlo la duración modo se obtenía de la Matriz).

Y el código que tendremos que escribir tras crear el botón de comando ActiveX, será:

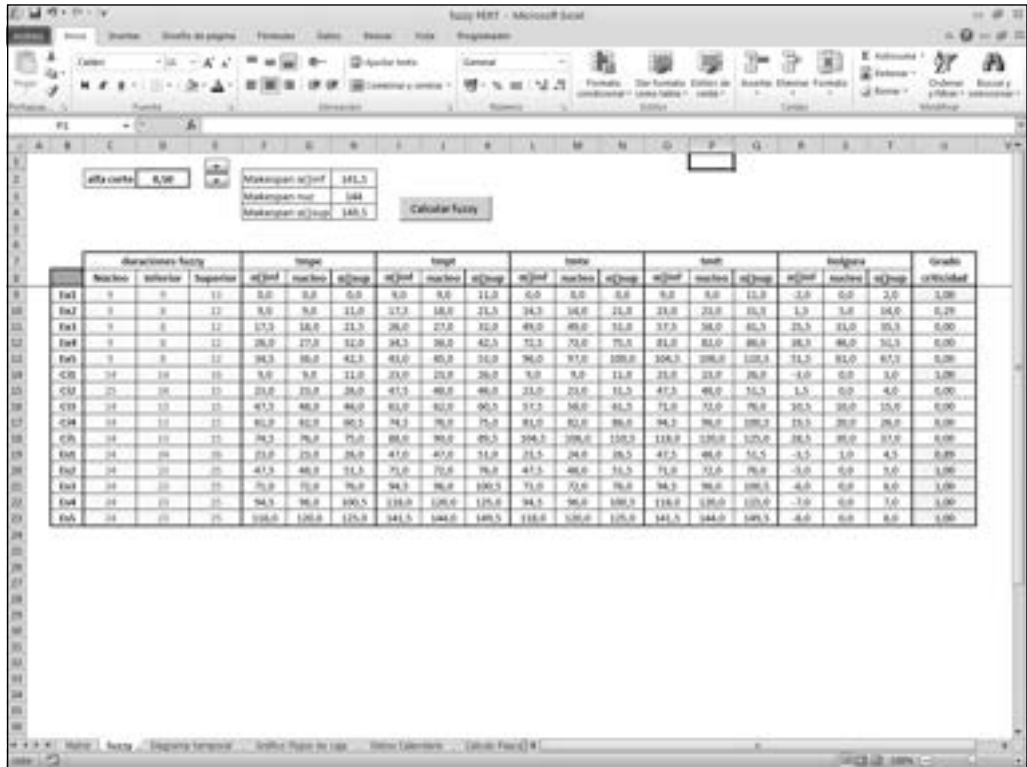


Figura 5.35. Hoja fuzzy.

```

Dim i As Integer
Dim j As Integer
Dim texto As String

Dim alfa_corte As Single
Dim Num_tareas As Single
Dim mk_inf As Integer
Dim mk_nuc As Integer
Dim mk_sup As Integer

Dim fila_m As Integer
Dim columna_m As Integer
Dim fila_f As Integer
Dim columna_f As Integer

Dim Dur_ini() As Integer
Dim Dur_nucleo() As Integer
Dim Dur_sop_inf() As Integer
Dim Dur_sop_sup() As Integer

Dim ESn() As Integer
Dim ESi() As Single
Dim ESS() As Single

```

```

Dim EFn() As Integer
Dim EFi() As Single
Dim EFs() As Single

Dim LSn() As Integer
Dim LSi() As Single
Dim LSS() As Single

Dim LFn() As Integer
Dim LFi() As Single
Dim LFs() As Single

Dim Hi() As Integer
Dim Hn() As Integer
Dim Hs() As Integer

'Dimensiona las matrices
With Worksheets("Matriz")
    'Establecemos la posición de la celda Inicio
    fila_m = .Range("Inicio").Row
    column_m = .Range("Inicio").Column
    'Calcula el número de tareas
    Do
        If .Cells(fila_m + Num_tareas + 1, column_m).Value =
            "" Then Exit Do
        Num_tareas = Num_tareas + 1
    Loop

    ReDim Dur_ini(Num_tareas)
    ReDim Dur_nucleo(Num_tareas)
    ReDim Dur_sop_inf(Num_tareas)
    ReDim Dur_sop_sup(Num_tareas)

    ReDim ESn(Num_tareas)
    ReDim ESi(Num_tareas)
    ReDim ESS(Num_tareas)

    ReDim EFn(Num_tareas)
    ReDim EFi(Num_tareas)
    ReDim EFs(Num_tareas)

    ReDim LSn(Num_tareas)
    ReDim LSi(Num_tareas)
    ReDim LSS(Num_tareas)

    ReDim LFn(Num_tareas)
    ReDim LFi(Num_tareas)
    ReDim LFs(Num_tareas)

    ReDim Hi(Num_tareas)
    ReDim Hn(Num_tareas)
    ReDim Hs(Num_tareas)
End With

'Lee las duraciones fuzzy
With Worksheets("fuzzy")

```

```

'Lee la posición de la celda Inicio
fila_f = .Range("Inicio_fuzzy").Row
columna_f = .Range("Inicio_fuzzy").Column
'Lee el valor del alfa corte
alfa_corte = .Range("alfa_corte").Value
'Lee los valores fuzzy de las duraciones
For i = 1 To Num_tareas
    Dur_nucleo(i) = .Cells(fila_f + i, columna_f + 1).Value
    Dur_sop_inf(i) = .Cells(fila_f + i, columna_f + 2).Value
    Dur_sop_sup(i) = .Cells(fila_f + i, columna_f + 3).Value
Next
'Limpiar los valores iniciales de la hoja
For i = 1 To Num_tareas
    For j = 4 To 19
        .Cells(fila_f + i, columna_f + j).Value = ""
    Next
Next
End With
'#####
'Calcula los soportes y el núcleo del proyecto fuzzy
'#####
With Worksheets("Matriz")
    'Lee los valores iniciales de la duración de la matriz
    For i = 1 To Num_tareas
        Dur_ini(i) = .Cells(fila_m + i, columna_m + 1).Value
    Next
    '*****
    'Calcula los valores nucleo
    '*****
    'Copia la duración nucleo a la matriz
    For i = 1 To Num_tareas
        .Cells(fila_m + i, columna_m + 1).Value = Dur_nucleo(i)
    Next
    'Lee los valores nucleo de la matriz
    For i = 1 To Num_tareas
        ESn(i) = .Cells(fila_m + i, columna_m + 2).Value
        EFn(i) = .Cells(fila_m + i, columna_m + 3).Value
        LSn(i) = .Cells(fila_m + i, columna_m + 4).Value
        LFn(i) = .Cells(fila_m + i, columna_m + 5).Value
        If mk_nuc < LFn(i) Then mk_nuc = LFn(i)
    Next
    '*****
    'Calcula los valores soporte inferior
    '*****
    'Copia la duración soporte inferior a la matriz
    For i = 1 To Num_tareas
        .Cells(fila_m + i, columna_m + 1).Value = Dur_sop_inf(i)
    Next
    'Lee los valores soporte inferior de la matriz
    For i = 1 To Num_tareas
        ESi(i) = .Cells(fila_m + i, columna_m + 2).Value
        EFi(i) = .Cells(fila_m + i, columna_m + 3).Value
        LSi(i) = .Cells(fila_m + i, columna_m + 4).Value
        LFi(i) = .Cells(fila_m + i, columna_m + 5).Value
        If mk_inf < LFn(i) Then mk_inf = LFi(i)
    Next

```

```

'*****
'Calcula los valores soporte superior
'*****
'Copia la duración soporte superior a la matriz
For i = 1 To Num_tareas
    .Cells(fila_m + i, columna_m + 1).Value = Dur_sop_sup(i)
Next
'Lee los valores soporte superior de la matriz
For i = 1 To Num_tareas
    ESs(i) = .Cells(fila_m + i, columna_m + 2).Value
    EFs(i) = .Cells(fila_m + i, columna_m + 3).Value
    LSs(i) = .Cells(fila_m + i, columna_m + 4).Value
    LFs(i) = .Cells(fila_m + i, columna_m + 5).Value
    If mk_sup < LFs(i) Then mk_sup = LFs(i)
Next

'*****
'Calcula la Holgura
'*****
For i = 1 To Num_tareas
    Hn(i) = LFn(i) - ESn(i) - Dur_nucleo(i)
    Hi(i) = LFi(i) - ESs(i) - Dur_sop_sup(i)
    Hs(i) = LFs(i) - ESi(i) - Dur_sop_inf(i)
Next

'*****
'restaura los valores iniciales de la duración de la matriz
'*****
For i = 1 To Num_tareas
    .Cells(fila_m + i, columna_m + 1).Value = Dur_ini(i)
Next
End With

'#####
'representa los valores del alfa corte de las tareas
'#####
With Worksheets("fuzzy")
    '*****
    'Presenta los valores núcleo
    '*****
    For i = 1 To Num_tareas
        .Cells(fila_f + i, columna_f + 5).Value = ESn(i)
        .Cells(fila_f + i, columna_f + 8).Value = EFn(i)
        .Cells(fila_f + i, columna_f + 11).Value = LSn(i)
        .Cells(fila_f + i, columna_f + 14).Value = LFn(i)
    Next
    '*****
    'Presenta los valores alfa corte inferior
    '*****
    For i = 1 To Num_tareas
        texto = "=" & Trim(Str(ESi(i))) & "+ alfa_corte *(" & Trim(Str(ESn(i)))
            & "-" & Trim(Str(ESi(i))) & ")"
        .Cells(fila_f + i, columna_f + 4).formula = texto
        texto = "=" & Trim(Str(EFi(i))) & "+ alfa_corte *(" & Trim(Str(EFn(i)))
            & "-" & Trim(Str(EFi(i))) & ")"
        .Cells(fila_f + i, columna_f + 7).formula = texto
    Next
End With

```

```

        texto = "=" & Trim(Str(LSi(i))) & "+ alfa_corte *(" & Trim(Str(LSn(i)))
            & "-" & Trim(Str(LSi(i))) & ")"
        .Cells(fila_f + i, columna_f + 10).formula = texto
        texto = "=" & Trim(Str(LFi(i))) & "+ alfa_corte *(" & Trim(Str(LFn(i)))
            & "-" & Trim(Str(LFi(i))) & ")"
        .Cells(fila_f + i, columna_f + 13).formula = texto
    Next

    '*****
    'Presenta los valores alfa corte superior
    '*****
    For i = 1 To Num_tareas
        texto = "=" & Trim(Str(ESs(i))) & "+ alfa_corte *(" & Trim(Str(ESn(i)))
            & "-" & Trim(Str(ESs(i))) & ")"
        .Cells(fila_f + i, columna_f + 6).formula = texto
        texto = "=" & Trim(Str(EFs(i))) & "+ alfa_corte *(" & Trim(Str(EFn(i)))
            & "-" & Trim(Str(EFs(i))) & ")"
        .Cells(fila_f + i, columna_f + 9).formula = texto
        texto = "=" & Trim(Str(LSs(i))) & "+ alfa_corte *(" & Trim(Str(LSn(i)))
            & "-" & Trim(Str(LSs(i))) & ")"
        .Cells(fila_f + i, columna_f + 12).formula = texto
        texto = "=" & Trim(Str(LFs(i))) & "+ alfa_corte *(" & Trim(Str(LFn(i)))
            & "-" & Trim(Str(LFs(i))) & ")"
        .Cells(fila_f + i, columna_f + 15).formula = texto
    Next

    '*****
    'Presenta los valores de la Holgura
    '*****
    For i = 1 To Num_tareas
        texto = "=" & Trim(Str(Hi(i))) & "+ alfa_corte *(" & Trim(Str(Hn(i)))
            & "-" & Trim(Str(Hi(i))) & ")"
        .Cells(fila_f + i, columna_f + 16).formula = texto

        .Cells(fila_f + i, columna_f + 17).Value = Hn(i)

        texto = "=" & Trim(Str(Hs(i))) & "+ alfa_corte *(" & Trim(Str(Hn(i)))
            & "-" & Trim(Str(Hs(i))) & ")"
        .Cells(fila_f + i, columna_f + 18).formula = texto
    Next

    '*****
    'Presenta los Grados de criticidad
    '*****
    For i = 1 To Num_tareas
        If Hi(i) >= 0 Then .Cells(fila_f + i, columna_f + 19).Value = 0
        If Hs(i) <= 0 Then .Cells(fila_f + i, columna_f + 19).Value = 1
        If Hi(i) < 0 And Hs(i) > 0 Then .Cells(fila_f + i, columna_f +
            19).Value = -Hi(i) / (Hn(i) - Hi(i))
    Next

    '*****
    'Presenta la duración del proyecto
    '*****
    texto = "=" & Trim(Str(mk_inf)) & "+ alfa_corte *(" & Trim(Str(mk_nuc))
        & "-" & Trim(Str(mk_inf)) & ")"

```

```
.Range("mk_inf").formula = texto
.Range("mk_nuc").Value = mk_nuc
texto = "=" & Trim(Str(mk_sup)) & "+ alfa_corte * (" & Trim(Str(mk_nuc))
        & "-" & Trim(Str(mk_sup)) & ")"
.Range("mk_sup").formula = texto
End With
```

5.4.7. El diagrama temporal difuso

El diagrama temporal también ha sufrido una pequeña modificación, para reflejar con una suave trama los valores difusos de los comienzos y finales de las tareas, para lo que se han insertado dos nuevas columnas y modificado la formulas que teníamos para el inicio y final de las tareas:

Celda	Fórmula
C5	=DIA.LAB.INTL(Fecha_Inicio;fuzzy!G9;Cod_Fin_D;Festivos)
D5	=DIA.LAB.INTL(Fecha_Inicio;fuzzy!J9-1;Cod_Fin_D;Festivos)
E5	=DIA.LAB.INTL(Fecha_Inicio;fuzzy!F9;Cod_Fin_D;Festivos)
F5	=DIA.LAB.INTL(Fecha_Inicio;fuzzy!K9-1;Cod_Fin_D;Festivos)

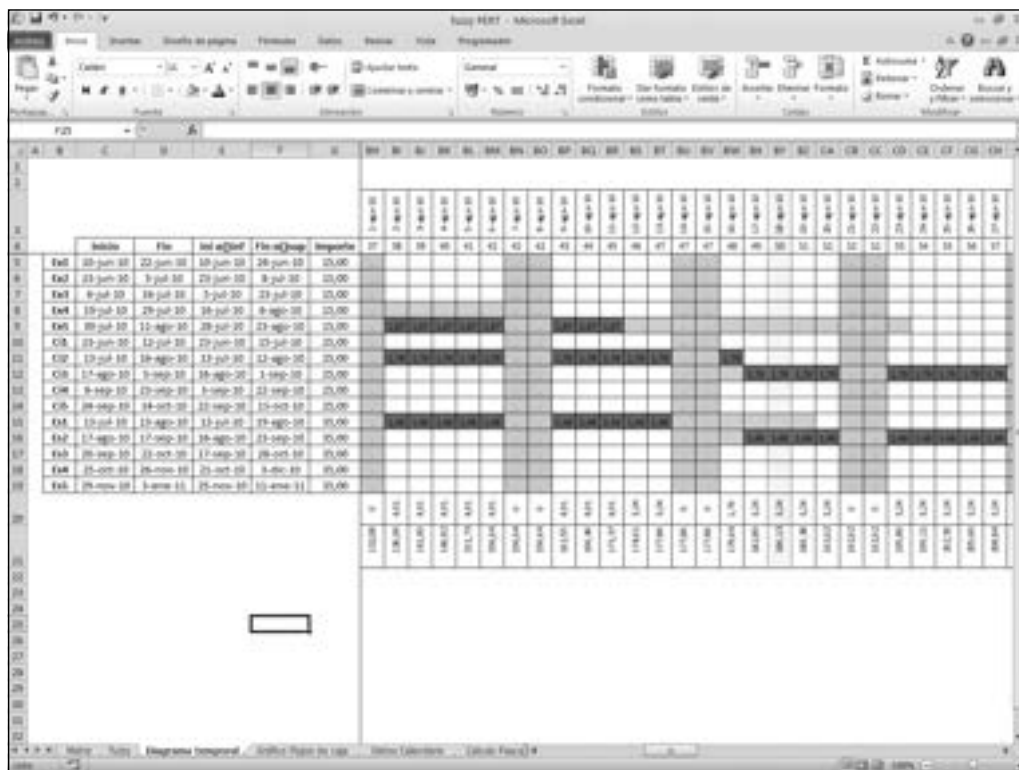
Y para terminar hemos añadido un nuevo formato condicional similar al que ya teníamos, pero referenciado a los tiempos difusos, que deberá quedar en tercera posición, obteniendo finalmente el diagrama que muestra la figura 5.36.

NOTA

La hoja de cálculo completa se encuentra en Cap_05\fuzzy PERT, de los ejemplos del libro.

5.5. Unas reflexiones finales

En este capítulo se han expuesto dos modelos diferentes para gestionar el riesgo de cumplimiento de los proyectos, uno basado en simulaciones estocásticas y otro en lógica difusa, ambos en primera línea de investigación en el campo de la gestión de proyectos.



6

El problema de la optimización de costes

6.1. Introducción al TCTP

La optimización de los costes de un proyecto, es el problema de optimización más antiguo de cuantos se pueden aplicar a los grafos de proyectos, y también por supuesto uno de los más estudiados, pero sin embargo, y aunque parezca una contradicción, no está implementado en ningún programa comercial de gestión de proyectos.

El problema se conoce en la literatura científica como *TCTP* (*Time Cost Trade off Problem*), y consiste en determinar las duraciones de las tareas de tal manera que el coste del proyecto sea mínimo, o dicho de manera más correcta, en minimizar (optimizar) una función de costes, con ciertas condiciones que llamaremos restricciones al problema.

Desde el punto de vista matemático, el problema clásico es un "problema fácil" de optimización combinatoria, y existen distintos algoritmos para su resolución, desde algoritmos llamados heurísticos que proporcionan soluciones factibles al problema aunque no siempre garantizan el óptimo, hasta algoritmos exactos de enumeración implícita y explícita.

Como he comentado anteriormente, la formulación clásica es un "problema fácil" porque asume ciertas restricciones que lo hacen poco realista y muy difícil de aplicar en la práctica.

En los problemas de optimización entera, al contrario de lo que sucede en los problemas lineales, añadir restricciones (condiciones) al problema reduciendo su universo de soluciones, hace que estos sean más fáciles de solucionar, técnica que se conoce como "relajación del problema".

La dificultad surge, porque cuando eliminamos esas restricciones al problema, esas relajaciones, lo convertimos en un problema no lineal y discreto, en un "problema difícil" para el que no existen algoritmos exactos, necesitando recurrir a algoritmos metaheurísticos o de enumeración explícita.

Hasta la versión de Excel 2010, para solucionar un problema no lineal y discreto de forma eficiente, era necesario diseñar algoritmos específicos con Visual Basic, pero ahora contamos con la nueva versión del "Solver" de Excel, que incorpora la posibilidad de aplicar algoritmos evolutivos, y así afrontar eficientemente su aplicación.

Para plantear y solucionar este problema con Excel, vamos a empezar exponiendo la formulación del modelo clásico lineal convexo con las relajaciones que lleva aparejadas y que limitan su aplicación práctica, para terminar con la formulación del modelado no lineal y discreto, que podremos utilizar para optimizar los costes de cualquier proyecto con condicionantes y variables reales.

6.2. El problema lineal del TCTP

El problema clásico del *TCTP*, supone que existen dos tipos de coste en un proyecto, los costes de producción de las tareas que son inversamente proporcional a la duración de las mismas, los costes de estructura del proyecto que son directamente proporcional a su duración, y la suma de estos dos constituyen los costes totales del proyecto, que tienen la particularidad de su forma convexa, con un único punto de inflexión que determina su duración óptima.

6.2.1. Los costes de producción del proyecto

Los costes de producción de un proyecto son la suma de todos los costes de producción de las tareas, y están formados por:

- Coste de materiales y productos.
- Coste de subcontratistas.
- Coste de amortizaciones.
- Coste de mano de obra.
- Coste de transportes y fletes directamente imputables a la tarea.
- Coste de alquileres directamente imputable a la tarea.
- Costes de calidad y no calidad.

Estos costes, están condicionados por el consumo que hacemos de los recursos, es decir, que si ejecutamos la tarea en unas condiciones normales tendremos un coste normal asociado, y a medida que aceleramos su ejecución, aumentará el consumo de recursos y su coste asociado, hasta un cierto límite de duración:

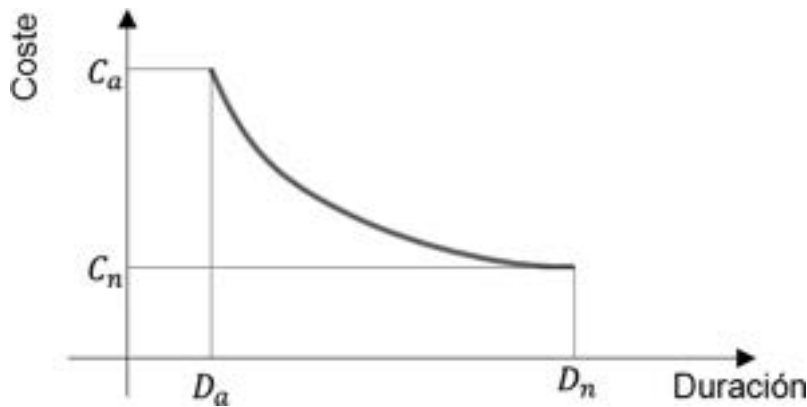


Figura 6.1. Costes de producción normal y acelerado de la tarea.

Es decir, que para una duración normal (D_n) tendremos un coste de producción normal (C_n) asociado, y para una duración acelerada (D_a) un coste de producción acelerado (C_a), de manera que para cualquier posible duración en t (D_t), tendremos un coste de producción asociado en t (C_t), véase la figura 6.2.

La anterior función es relativamente compleja, por lo que el modelado clásico la simplifica de la forma que muestra la figura 6.3.

Pudiendo determinar fácilmente a partir de la anterior simplificación, el coste (C_t) para cualquier valor de t , a partir de los costes y duraciones normal y acelerada, véase la figura 6.4.

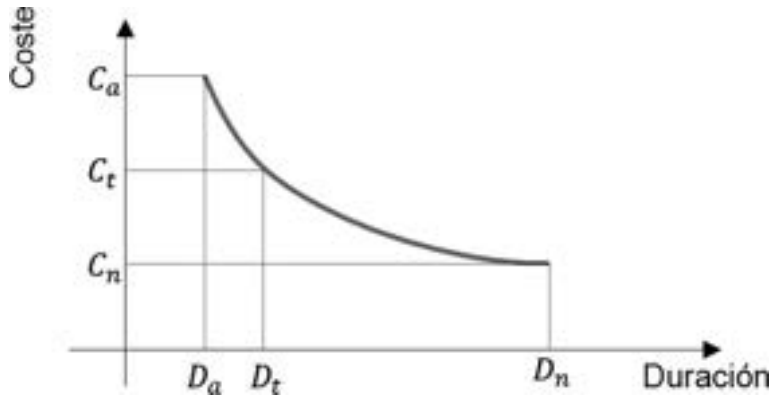


Figura 6.2. Coste de producción en t de la tarea.

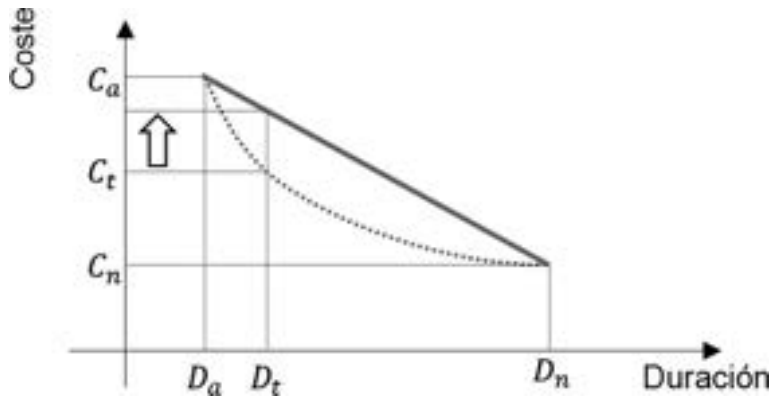


Figura 6.3. Simplificación de la función de coste de producción de la tarea.

$$C_{t_i} = C_{a_i} + \frac{C_{n_i} - C_{a_i}}{D_{n_i} - D_{a_i}} \cdot (D_{t_i} - D_{a_i})$$

Figura 6.4. Cálculo del coste de producción de la tarea en t .

Esta simplificación, es una importante alteración de la verdadera función de costes de producción de la tarea, por los siguientes motivos:

1. Se supone que la función de coste de costes de producción de la tarea es una función lineal, algo que como se ha visto no es cierto.
2. Se supone que la función de costes de producción de la tarea es continua en el tiempo, obviando el hecho de que una tarea no tiene porque adoptar cualquier posible valor en t , es decir, es una función discreta siendo factible solo para ciertos valores de t .

3. Se supone que la función de costes de producción es inversamente proporcional a su duración algo que tampoco tiene porque ser cierto en todos los casos.

Así, la verdadera función de costes de producción de una tarea se asemejará más a:

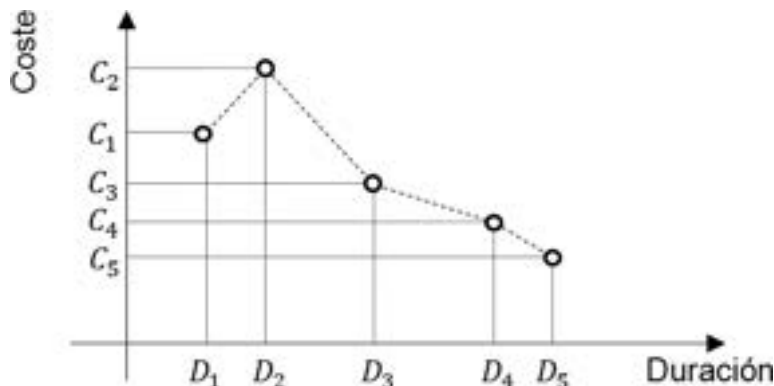


Figura 6.5. Función discreta de costes de producción de la tarea.

Que como se puede observar, tan solo puede adoptar cinco posibles valores en t (D_1, D_2, D_3, D_4, D_5), y los costes asociados de producción a cada una de estas posiciones (C_1, C_2, C_3, C_4, C_5) no siguen ninguna función concreta. Estos posibles valores para t , es lo que se conoce como "*modos de ejecución*", y puede adoptar cualquier valor entre uno e infinito, aunque en la práctica es muy difícil encontrarnos con más de cinco posibles modos de ejecución, siendo lo común disponer solamente de dos o tres alternativas diferentes.

La suma de todos los costes de producción de las tareas en su modo de ejecución correspondiente, constituirá el coste de producción del proyecto, y lo habitual es que sea inversamente proporcional a su plazo de ejecución, aunque no necesariamente, véase la figura 6.6.

6.2.2. Los costes de estructura del proyecto

Son aquellos que dependen de la estructura de los sistemas de producción, pero no de la cantidad de unidades producidas.

La anterior definición es muy difícil de llevar a la práctica, pues evidentemente a medida que aumenta el sistema productivo se requerirán mayores medios de estructura y consecuentemente se generarán mayores costes,

pero para su aplicación práctica, supondremos que el volumen de la producción está previamente determinado y que son fijados con anterioridad al inicio de los trabajos.

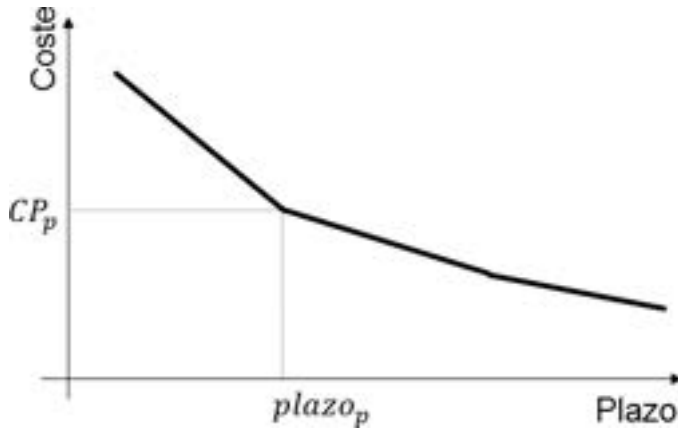


Figura 6.6. Función de costes de producción del proyecto.

A su vez los costes de estructura del proyecto los clasificaremos en:

- Costes Fijos (C_f): aquellos que se producen con independencia de la duración del proyecto, también conocidos como coste de implantación.
- Costes Variables (C_v): aquellos que son función exclusivamente del tiempo, siendo (C_v) el coste en el que se incurre por cada uno de los días de duración del proyecto.

Así, los costes de estructura del proyecto seguirá esta función lineal:

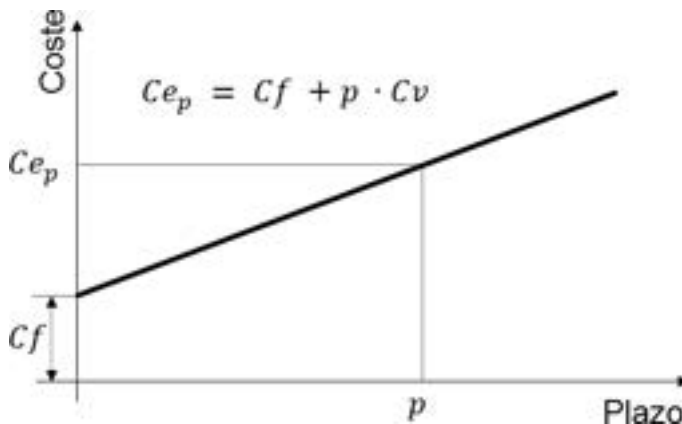


Figura 6.7. Función de costes de estructura del proyecto.

De tal forma que para cualquier posible plazo programado (p), se incurrirá en un coste de estructura (Ce_p) que será el coste fijo (Cf) más el producto del plazo (p) por el coste variable (Cv) por día transcurrido.

6.2.3. Los costes totales del proyecto y su convexidad

El coste total del proyecto (CTP_p) para un determinado plazo p , será la suma de los costes de producción de las tareas ($\sum Ct_i$) y el coste de estructura (Ce_p) del proyecto para ese plazo p . Para cada posible plazo p , existe una determinada combinación de modos de ejecución de las tareas que determinan el coste total mínimo para ese plazo p , y de entre todos los posibles plazos del proyecto, existe una combinación que establece el plazo óptimo del proyecto con un coste total óptimo del proyecto ($CTP_{\text{óptimo}}$).

La función de costes totales de proyecto en la formulación clásica (sin modos de ejecución discretos), es una función convexa muy robusta, y se puede determinar de una forma relativamente sencilla.

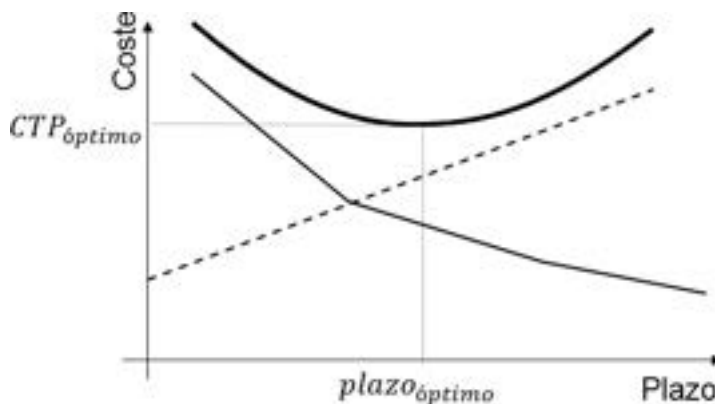


Figura 6.8. Función de costes totales del proyecto.

6.2.4. La formulación del modelo matemático del problema lineal

El problema consiste en determinar cuál es la duración (Ct_i) de las tareas que haga que el coste total del proyecto sea óptimo, estableciendo además ese plazo p óptimo. Y la formulación del modelo matemático es:

Minimizar:

$$CTP_p = Cf + p \cdot Cv + \sum_{i=1}^{\text{total tareas}} \left[Ca_i + (Dt_i - Da_i) \cdot \frac{Cn_i - Ca_i}{Dn_i - Da_i} \right]$$

Sujeto a las siguientes restricciones:

$$Ca_i \leq Ct_i \leq Cn_i$$

$$\{Ct_i\} \geq 0 \text{ y enteras}$$

Figura 6.9. Formulación del modelo matemático del problema lineal.

La formulación clásica tiene como función objetivo a minimizar el coste total del proyecto (CTP), con dos únicas restricciones, la primera establece que las duraciones programadas para las tareas han de estar entre las dos duraciones cota acelerada y normal, y la segunda, que obliga a que estas sean enteras y mayores que cero, constituyendo un problema de programación lineal entera con un único óptimo.

6.2.5. El Solver de Excel

Para resolver los problemas de optimización, Excel ofrece la herramienta Solver, que en la nueva versión 2010, presenta muchas novedades.

Antes de trabajar con Solver, deberemos de cargar su complemento, para lo cual ejecutaremos: Archivo>Opciones>Complementos, y en la ventana de Opciones: Administrar: Complementos de Excel>Ir..., seleccionando la casilla correspondiente a Solver:



Figura 6.10. Cargar complemento Solver.

Una vez cargado el complemento de Solver, debe aparecer al final de la pestaña de datos, pudiendo ya acceder a sus funciones, con este aspecto:



Figura 6.11. Ventana de Solver.

La ventana de Solver, presenta diferentes parámetros que deberemos rellenar, y que corresponden a:

- Establecer objetivo: Aquí seleccionaremos la celda cuyo valor deseamos optimizar, y que será resultado de la formulación de la función objetivo de nuestro modelo.
- Para:
 - Mín: Esta casilla la seleccionaremos si el objetivo es minimizar el valor de una función como por ejemplo en el caso de los costes de un proyecto como es el nuestro.
 - Máx: Esta casilla la activaremos si lo que deseamos es maximizar la función objetivo, como es el caso en los que deseamos maximizar los beneficios de una inversión.
 - Valor de: Si queremos que la función objetivo adopte un valor concreto.

- Cambiando las celdas variables: Aquí estableceremos el rango de celdas de las variables de decisión de nuestra formulación. Si las celdas de las variables de decisión no son adyacentes, podemos entonces separar los diferentes rangos mediante comas. La única limitación que todavía presenta Solver, es que el número máximo de variables de decisión se sigue manteniendo en 200, cantidad que aunque pueda parecer grande, nos obligará a realizar ciertos equilibrios en las formulaciones del próximo capítulo.
- Sujeto a las restricciones: Mediante los botones Agregar, Cambiar, Eliminar, estableceremos las restricciones de la formulación en diferentes líneas, que tienen la siguiente estructura en tres zonas:

Conjunto de Celdas	Condición	Restricción
--------------------	-----------	-------------

- Conjunto de Celdas: Aquí estableceremos la celda o rangos de celdas sobre las que queremos crear una restricción.
- Condición: Las condiciones pueden ser menor o igual que ($\leq$), igual que ($=$) o mayor o igual que ($\geq$) la restricción. Además podemos aquí determinar si la celda o rango de celdas han de ser enteras (ent), binarias (bin) o diferentes (dif), apareciendo integer, binary o alldifferent en la zona reservada a la restricción.
- Restricción: aquí introduciremos las celdas que restringen o limitan los valores de las variables de decisión que hemos seleccionado en la primera zona. Es importante que cuando trabajemos con un modelo lineal, la linealidad se mantenga también en las restricciones, así, una restricción no lineal del tipo $x / y \geq 3$, podemos linealizarla transformándola en $x \geq 3 y$.
- Convertir variable sin restricciones en no negativas: Esta casilla de verificación es útil cuando tenemos una condición de no negatividad en las variables de decisión, no teniendo necesidad de añadir restricciones tipo ≥ 0 para cada una de ellas.
- Método de resolución: Aquí es donde el nuevo Excel 2010 nos ofrece una de las mejoras más importantes en sus algoritmos de resolución:
 - Simplex LP: Es el método de resolución diseñado para formulaciones lineales, aplicando el algoritmo del Simplex, pero requiere que no existan fórmulas de Excel en su formulación, porque en este caso lo considera un problema no lineal.
 - GRG Nonlinear: Este método usa un algoritmo GRG (*Generalized Reduced Gradient*), al igual que la versión anterior de Excel y es

útil para modelos no lineales en los que las funciones son muy suaves (robustas), en los que no existe el peligro de que el algoritmo se encierre en un óptimo local.

- **Evolutionary:** A mí parecer, esta es la mejora más trascendental de todas cuantas aporta el nuevo Excel, y que es la posibilidad de utilizar algoritmos evolutivos en la optimización. La trascendencia no solo estriba en los propios algoritmos utilizados, sino que nos permite usar formulas de tipo condicional en la función objetivo y en las restricciones. Es importante hacer constar, que el hecho de usar algoritmos evolutivos, no me garantiza que la solución obtenida sea la óptima, algo que solo podré establecer mediante algoritmos de enumeración explícita, pero sí que será lo suficiente buena como para estar muy satisfecho.

Una vez conocida la formulación de un modelo de optimización y el funcionamiento del Solver de Excel, estamos en condiciones de optimizar los costes de nuestro primer proyecto.

6.2.6. Implementación del modelo lineal en Excel 2010

Para que resulte más fácil de comprender el funcionamiento del Solver de Excel, esta primera hoja que vamos a construir será muy sencilla.

Constará de tres columnas para las duraciones y tres para los costes, correspondiendo la primera columna a la duración y coste programado, y las otras dos para las duraciones y los costes acelerados respectivamente, véase la figura 6.12. Una vez creada la estructura de la hoja, tendremos que escribir los valores de las duraciones aceleradas y normales, así como los costes acelerado y normal para cada tarea, teniendo la precaución de que la duración normal siempre ha de ser mayor que la acelerada y que el coste normal siempre será menor que el acelerado para el problema lineal.

Tarea	Duración acelerada	Duración normal	Coste Acelerado	Coste normal
Ex1	5	10	20	15
Ex2	5	10	20	15
Ex3	5	10	20	15

Tarea	Duración acelerada	Duración normal	Coste Acelerado	Coste normal
Ex4	5	10	20	15
Ex5	5	10	20	15
Ci1	10	15	35	25
Ci2	10	15	35	25
Ci3	10	15	35	25
Ci4	10	15	35	25
Ci5	10	15	35	25
Es1	15	25	50	35
Es2	15	25	50	35
Es3	15	25	50	35
Es4	15	25	50	35
Es5	15	25	50	35

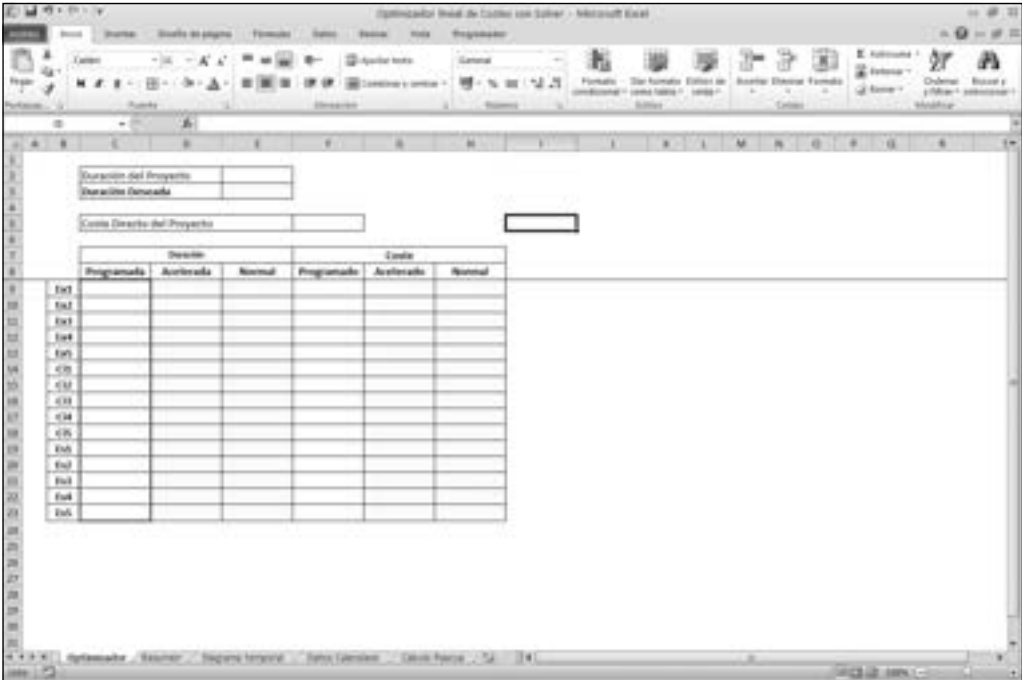


Figura 6.12. Hoja optimizador lineal.

La celda E2 corresponde a la duración del proyecto, tomada directamente desde la hoja "Matriz", y se formula escribiendo: =MAX (Matriz! G5:G19).

La celda E3, es la de la duración del proyecto para la cual queremos determinar la duración de las tareas, que haga que el proyecto tenga el coste mínimo para ese plazo, y será determinado para cada ocasión.

La celda F5, es la correspondiente al coste de producción del proyecto, y será la suma de los costes programados de todas las tareas, y su fórmula: =SUMA(F9:F23).

Para formular la columna del coste programado, que ya vimos anteriormente, escribiremos en la primera celda de la columna la siguiente fórmula: =G9+((H9-G9)/(E9-D9))*(C9-D9), que tendremos que copiar a lo largo de todas las tareas.

Para comprobar que la anterior formula es correcta, escribe en la duración programada de una tarea su duración normal, deberá aparecer en su coste programado el coste normal, si escribes la duración acelerada, aparecerá su coste acelerado, y si escribes cualquier otra, se obtendrá el coste programado para esa duración programada.

Pero aún no está todo terminado, aún tenemos pendiente hacer que Excel calcule para cualquier posible duración programada de las tareas, la duración del proyecto, para lo que entraremos en la hoja "Matriz" y formularemos la columna de la duración, para que obtenga el valor de la duración programada de la hoja "Optimizador", así, por ejemplo, en la celda C5 de la hoja "Matriz" escribiremos =Optimizador!C9.

Ya estamos en condiciones de optimizar el proyecto, para lo que escribiremos 145 como la duración del proyecto deseado, y ejecutaremos: Datos> Solver, con los siguientes parámetros (véase la figura 6.13):

Parámetro	Escribir
Función objetivo	\$F\$5
Variables de decisión	\$C\$9:\$C\$23
Restricciones	\$C\$9:\$C\$23 <= \$E\$9:\$E\$23
	\$C\$9:\$C\$23 >= \$D\$9:\$D\$23
	\$C\$9:\$C\$23 = entero
	\$E\$2 = \$E\$3
Método	GRG non linear



Figura 6.13. Parámetros Solver del modelo lineal.

Y tras hacer clic sobre Resolver y esperar unos instantes aparecerá la siguiente ventana:

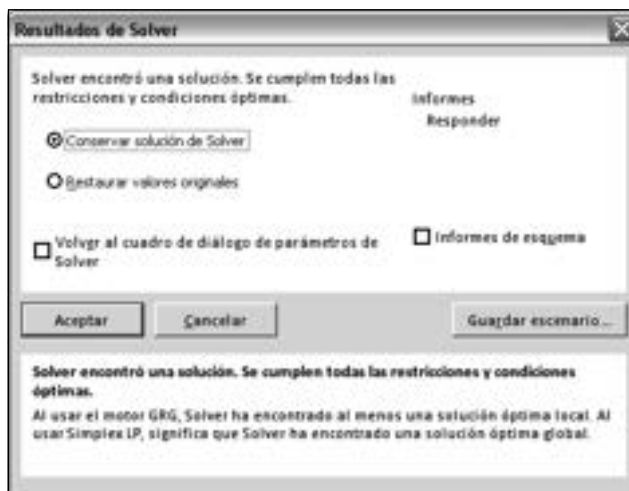


Figura 6.14. Resultado de Solver lineal.

Si ahora analizamos las duraciones programadas, veremos que las tareas Ex1 y Ex2 han cambiado su duración programada a 5 y 10 respectivamente, y que el coste del proyecto óptimo para esa duración es de 380 unidades.

Si realizamos el mismo proceso para todas las posibles duraciones del proyecto hasta 90 días obtendremos las siguientes funciones de costes, donde los costes totales del proyecto son muy suaves, muy robustos:



Figura 6.15. Función de costes del modelo lineal.

NOTA

La hoja de cálculo completa se encuentra en Cap_06\Optimizador lineal de Costes con Solver, de los ejemplos del libro.

6.3. El problema no lineal del TCTP

El problema no lineal del *TCTP* elimina la condición de linealidad de la función de costes de la tarea, y adquiere su forma discreta basada en múltiples modos de ejecución, o "multi-modo" como se conoce en la literatura científica, estableciendo que una tarea solo puede ejecutarse con determinadas duraciones preestablecidas, y que los costes no tienen porque seguir una función decreciente, pudiendo adoptar cualquier carácter.

El problema de la no linealidad estriba en que además del óptimo total de la función a optimizar, existen otros óptimos locales propios de su no total convexidad, lo que dificulta enormemente su resolución mediante las herramientas tradicionales de optimización por enumeración explícita (calculando todas las posibles soluciones al problema), necesitando recurrir a algoritmos más complejos.

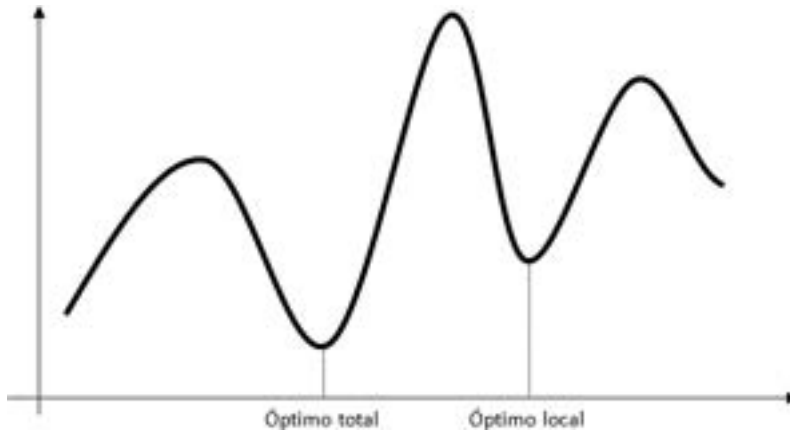


Figura 6.16. Óptimos locales del problema no lineal.

6.3.1. Las penalizaciones y bonificaciones

Es habitual, que los proyectos, además del plazo de entrega comprometido, establezcan un sistema de bonus-malus, de tal manera que si es finalizado y entregado antes de la fecha comprometida seremos bonificados en un determinado importe por cada día de adelanto, e igualmente penalizados si incurrimos en retrasos.

6.3.2. El modelado del problema no lineal

La formulación del modelo no lineal es aparentemente más sencilla que la formulación clásica, y está basada en unas variables binarias x_{ik} , que tan solo pueden adoptar valores cero o uno, y la única restricción que tiene, es que una tarea tan solo puede programarse en un único modo de ejecución, que se garantiza haciendo que el sumatorio de valores x_{ik} de la tarea sea la unidad, para todo el conjunto N de tareas, siendo $r(i)$ el número de modos en que puede ser ejecutada la tarea i .

$$x_{ik} = \begin{cases} 1 & \text{si la tarea } i \text{ se ejecuta en modo } k \\ 0 & \text{si la tarea } i \text{ no se ejecuta en modo } k \end{cases}$$

C_{ik} Coste de ejecutar la tarea i en modo k

$r(i)$ Numero de modos en que se puede ejecutar la tarea i

N Número de tareas del proyecto

Minimizar:

$$CTP_p = Cf + p \cdot Cv + \sum_{i=1}^N \sum_{k=1}^{r(i)} [x_{ik} \cdot C_{ik}]$$

Sujeto a las siguientes restricciones:

$$\sum_{k=1}^{r(i)} x_{ik} = 1 \quad \forall i \in N$$

$$x_{ik} \in \{0, 1\} \quad \forall i \in N, k \in \{1, 2, \dots, r(i)\}$$

Figura 6.17. Formulación del modelo multi-modo.

6.3.3. La complejidad algorítmica del problema no lineal

La anterior formulación no lineal, es llamada de complejidad no polinómica, es decir, que la complejidad del problema o universo de soluciones, no crece proporcionalmente al número de tareas y modos de ejecución, sino que su crecimiento es exponencial.

La complejidad de esta formulación es conocida como $O(n^q)$, y más concretamente, que el universo de soluciones es el producto de los modos de ejecución de todas las tareas.

Puede parecer no demasiado complejo, y ciertamente no lo es. Entre los problemas difíciles es quizás de los más fáciles, pero si analizamos detalladamente lo que representa su complejidad, veremos que es enorme.

Supongamos nuestro proyecto de quince tareas con cinco posibles modos de ejecución para cada una, estaríamos hablando de cinco elevados a quince, 30.517.578.125, más de treinta mil quinientos millones de soluciones posibles. Si disponemos de un ordenador capaz de procesar diez mil soluciones por segundo, necesitaríamos más de 35 días de cálculo para analizar todas las soluciones y determinar su plazo y coste óptimo.

Evidentemente, no parece muy eficiente tener que esperar más de un mes para poder optimizar un sencillísimo proyecto de quince tareas, lo que ha provocado que esta metodología no esté muy extendida, y que su aplicación práctica se vea circunscrita exclusivamente a las publicaciones de investigación.

Los investigadores en el ámbito de la gestión de proyectos, están constantemente proponiendo nuevos algoritmos para afrontar este tipo de problemas con algoritmos metaheurísticos, de los que los algoritmos evolutivos forman parte, y aunque la programación *ad hoc* de algoritmos evolutivos no es nada complicada, requiere de la exposición de determinados conceptos que escapen a las pretensiones de este libro.

Hasta la aparición de Excel 2010, era necesario adquirir costosos programas de optimización con algoritmos evolutivos de propósito general, sin embargo ahora ya están incluidos en su Solver, y su aplicación es realmente sencilla.

6.3.4. Implementación del modelo no lineal en Excel 2010

La implementación que se va a proponer, no sigue la formulación expuesta para modelos no lineales, pero su filosofía es exactamente la misma, diferenciándose de la formulación matemática en que vamos a hacer uso de funciones específicas de Excel que van a ser muy útiles y sencillas en este caso, y que además ya hemos usado anteriormente.

La hoja propuesta tiene la estructura que se puede observar en la figura 6.18.

Como en casos anteriores, partimos del libro "Diagrama temporal con calendario laboral", a la cual le insertamos una nueva hoja que llamaremos "Optimizador", y que tiene dos aéreas diferenciadas, la superior relativa a los datos de costes y duraciones del proyecto, y la inferior de las tareas, que a su vez contiene en su zona izquierda las variables de decisión con sus valores de coste y duración asociados, y la información relativa a los modos de ejecución.

Este modelado está preparado para cinco posibles modos de ejecución, con cinco columnas de duración y cinco de costes, y el modo de ejecución correspondiente figura en su encabezamiento.

Así, definiremos los siguientes nombres a los rangos correspondientes a las duraciones y a los costes respectivamente:

Nombre del Rango	Rango de celdas
Duraciones_Modo	=Optimizador!\$H\$12:\$L\$27
Costes_Modo	=Optimizador!\$M\$12:\$Q\$27

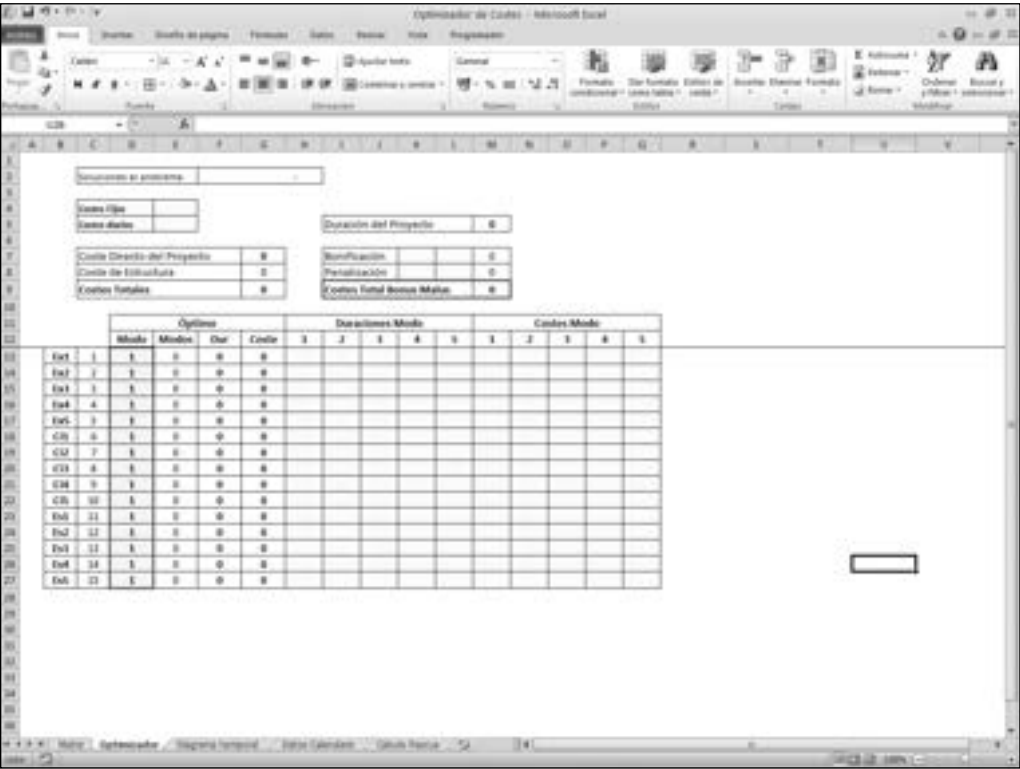


Figura 6.18. Hoja optimizador no lineal.

Respecto a la zona izquierda, vamos a ver detalladamente cada una de las columnas:

- Primera columna: corresponde al nombre de la tarea, y su formulación para la primera fila es: =Matriz!B5.
- Segunda columna: es el número de orden de la fila correspondiente, y se puede crear fácilmente escribiendo 1, en la primera fila y arrastrando la celda a lo largo de toda la columna, seleccionando luego **serie de relleno**. Es muy importante que los valores de esta columna sean consecutivos, para que la formulación funciones correctamente.



Figura 6.19. Rellenar serie.

- Tercera columna: Es la columna con las variables de decisión, y deberá contener unos antes de ejecutar el complemento Solver.
- Cuarta columna: Es el número total de modos de ejecución de una tarea, utilizado para reducir el universo de soluciones posibles del modelo, y su fórmula para la primera fila será: =CONTAR(H13:L13).
- Quinta columna: Es la duración correspondiente al modo de ejecución establecido, y se formulará para la primera fila por medio de la ya conocida función de búsqueda: =CONSULTAH (ENTERO (D13); Duraciones_Modo; C13+1).
- Sexta columna: Es el coste correspondiente al modo de ejecución establecido, y se formula igual que antes: =CONSULTAH (ENTERO (D13); Costes_Modo; C13+1).

En la zona superior de la hoja, es donde se encuentran los datos de costes y duraciones del proyecto, y para su formulación tan solo se requiere seguir las indicaciones que se han expuesto en los puntos anteriores.

Especial mención requiere el sistema de bonificaciones y penalizaciones, donde se ha de establecer el plazo a partir del cual se aplican las bonificaciones diarias, y el plazo a partir del cual se aplican las penalizaciones con su importe correspondiente, calculándose a partir de un condicional.

Para que el modelo funcione, es necesario que en cada iteración obtengamos la duración de proyecto correcta, en función del modo de ejecución establecido, para lo que modificaremos la columna "Duraciones" de la hoja "Matriz", y así la celda C5 se formulará escribiendo =Optimizador!F13, replicando la formula a lo largo de toda la columna.

Para optimizar el modelo, tendremos que ejecutar Datos>Solver, y establecer los siguientes parámetros:

Parámetro	Escribir
Función objetivo	\$M\$9
Variables de decisión	\$D\$13:\$D\$27

Parámetro	Escribir
Restricciones	$\$D\$13:\$D\$27 \leq \$E\$13:\$E\27 $\$D\$13:\$D\$27 \geq 1$ $\$D\$13:\$D\$27 = \text{entero}$
Método	Evolutionary



Figura 6.20. Parámetros Solver del modelo no lineal.

Y tras hacer clic sobre Resolver obtendremos el resultado que podemos observar en la figura 6.21.

Que nos indica que Solver ha encontrado una solución que no puede mejorar, no significando necesariamente que la solución ofrecida sea la óptima, dándonos la posibilidad de conservar su solución o bien adoptar los valores originales.



Figura 6.21. Resultado de Solver.

NOTA

La hoja de cálculo completa se encuentra en Cap_06\Optimizador No lineal de Costes con Solver, de los ejemplos del libro.

6.4. Anexo: Implementación del TCTP con enumeración explícita

Para los lectores que deseen verificar la bondad ofrecida por el Solver de Excel, les ofrezco la posibilidad de probar esta aplicación de Visual Basic, la cual mediante en un algoritmo recursivo es capaz de enumerar y evaluar todas las posibles soluciones de cualquier instancia del problema.

Esta pequeña pero atroz aplicación tan solo requiere dotarse de un rápido ordenador y una buena dosis de paciencia.

El código está formado por dos rutinas, la primera calcula el número de tareas, establece el coste mínimo del proyecto como un número muy grande ($100 \cdot 10^6$), y guarda la mejor solución como la solución actual, ejecutando por primera vez la rutina `itera()`. Esta segunda rutina, llamada `Itera()`, aunque muy corta en líneas de código, es potentísima, llamándose a sí misma de forma recursiva, y es la que genera las combinaciones de modos de ejecución, pudiéndose aplicar a infinidad de problemas de optimización no lineales, siempre que estos no sean muy grandes.

```

Dim fila As Integer
Dim columna As Integer
Dim Num_tareas As Integer
Dim mejor() As Single
Dim Minimo As Single

Private Sub CommandButton1_Click()
    With Worksheets("Optimizador")
        'Establecemos la posición de la celda Inicio
        fila = .Range("Inicio_Opt").Row
        columna = .Range("Inicio_Opt").Column
        'Calcula el número de tareas
        Do
            If .Cells(fila + Num_tareas + 1, columna).Value = "" Then Exit Do
            Num_tareas = Num_tareas + 1
        Loop
    End With
    ReDim mejor(Num_tareas)

    tarea = 0
    Minimo = 100 * 10 ^ 6
    Worksheets("Optimizador").Range("solución") = 1
    Call itera(tarea)
    For z = 1 To Num_tareas: Cells(fila + z, columna + 2) = mejor(z): Next z
End Sub

Sub itera(ByVal tarea As Integer)
    Do While tarea < Num_tareas
        tarea = tarea + 1
        For i = 1 To Cells(fila + tarea, columna + 3).Value
            Cells(fila + tarea, columna + 2) = i
            If i < Cells(fila + tarea, columna + 3).Value Then
                Call itera(tarea)
                With Worksheets("Optimizador")
                    .Range("solución") = .Range("solución") + 1
                End With
            End If
            If Minimo > Worksheets("Optimizador").Range("Coste_total") Then
                Minimo = Worksheets("Optimizador").Range("Coste_total").Value
                For z = 1 To Num_tareas: mejor(z) = Cells(fila + z, columna + 2): Next z
            End If
        Next i
    Loop
End Sub

```

NOTA

La hoja de cálculo completa se encuentra en Cap_06\Optimizador de Costes por enumeración, de los ejemplos del libro.

7

El problema de la optimización de recursos

7.1. Introducción al RCPSP

Los proyectos requieren del consumo de cierta cantidad de diaria de recursos para que estos puedan ser realizados, y de hecho, una buena programación no solo requiere que las dependencias entre tareas sean procedimentalmente correctas, sino que además han de poder ser ejecutados sin sobrepasar la disponibilidad de los recursos disponibles. Este tipo de problema se denomina *RCPSP (Resource Constrained Project Scheduling Problem)*, o problema de secuenciación de proyectos con recursos limitados, y consiste en secuenciar adecuadamente las tareas para que no se sobrepasen los recursos disponibles a lo largo de la ejecución del proyecto.

Además del problema de secuenciación de proyectos, existe otro problema donde lo que prima no es la disponibilidad de recursos, que se considera en un ilimitada en principio, sino la eficiencia que hacemos de su consumo, y que llamaremos de nivelación de proyectos.

Ambos problemas son considerados difíciles, más difíciles incluso que el *TCPT* visto en el capítulo anterior, y no disponemos de ningún tipo de algoritmo o procedimiento que me ofrezca una solución óptima de cualquiera

de ellos, disponiendo tan solo de algoritmos constrictivos para diseñar secuencias factibles y algoritmos de mejora sobre las anteriores secuencias.

Al contrario de lo que sucedía con el problema del *TCTP* y la optimización de los costes, el problema de los recursos sí que está implementado en el software profesional, aunque muy poca gente sabe, ni siquiera los profesionales más expertos, que las soluciones que ofrecen este tipo de programas son soluciones factibles aplicando algoritmos constructivos, que muy difícilmente nos proporcionarán buenas secuencias.

En este capítulo, se van a proponer diferentes alternativas para encontrar la solución óptima a los dos problemas, el problema de nivelación y el problema de secuenciación de proyectos.

7.2. El problema de la Nivelación

El problema de la nivelación consiste en distribuir adecuadamente las tareas del proyecto de tal manera que sin aumentar su plazo de ejecución, su distribución sea lo más uniforme posible.

7.2.1. El histograma de recursos

Supongamos que tenemos la siguiente distribución de consumos de recursos de mano de obra de un proyecto a lo largo de su ejecución, y que en adelante llamaremos histograma:

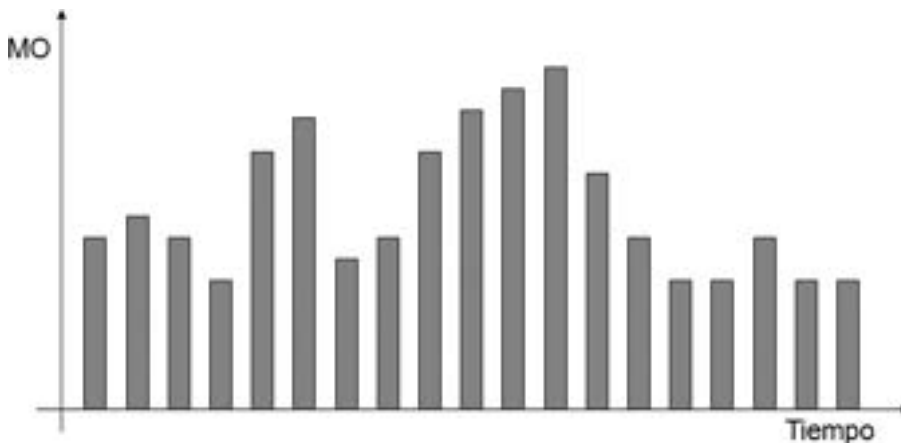


Figura 7.1. Histograma de recursos.

En el anterior histograma se puede observar claramente que el consumo de recursos no es constante a lo largo del proyecto, existen días con unas necesidades pequeñas, junto a otros con grandes necesidades, obligándonos a contrataciones, despidos o subcontrataciones.

Lo ideal sería que el consumo de recursos fuera constante a lo largo de todo el proyecto, es decir, que el consumo de mano de obra de cada día coincidiera precisamente con el consumo medio a lo largo del proyecto, sin necesidad de tener que aumentar el plazo de ejecución del proyecto, pero en la vida real, en el mejor de los casos tan solo podremos llegar a reducir las diferencias entre las crestas y los valles, obteniendo un histograma como el siguiente:

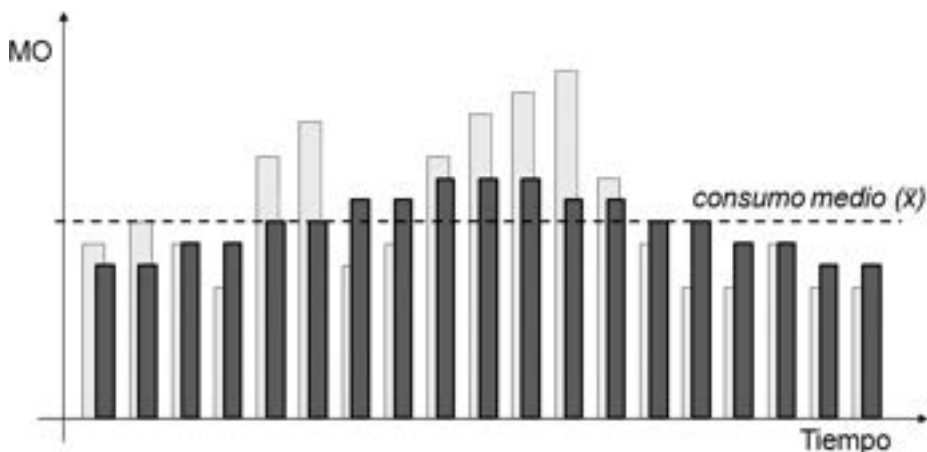


Figura 7.2. Histograma nivelado de recursos.

7.2.2. La optimización por mínimos cuadrados

Lo que realmente se ha conseguido al reducir la diferencia entre las crestas y los valles del histogramas ha sido reducir la varianza de los valores del histograma, es decir, que si mi objetivo es nivelar los recursos de un proyecto, bastará con formular un modelo de optimización donde la función objetivo que hay que minimizar sea la propia varianza de los valores del histograma.

La minimización del valor de la varianza del histograma, es equivalente a la minimización de la suma del cuadrado de los consumos diarios de los recursos del proyecto, llamándose este sistema de optimización como de "Mínimos Cuadrados".

El problema consiste en determinar la posición de las tareas, que me ofrezcan el menor valor posible de sumatorio de mínimos cuadrados, pero como no podemos aumentar la duración del proyecto, tan solo intervendrán en la optimización las tareas cuya holgura total sea mayor de cero, y su desplazamiento máximo será precisamente el valor de la holgura total de cada tarea, puesto que retrasar el comienzo de tareas críticas, recordaremos que lleva aparejado un retraso en la finalización del proyecto.

Tal y como expresaron Burgess y Killebrew:

"La eficacia en la asignación de un recurso determinado, en función de una distribución ideal, varía en sentido inverso a la suma obtenida en cada unidad de tiempo, del principio al fin del proyecto de los cuadrados de las diferencias entre las cargas totales que corresponden a las dos asignaciones."

7.2.3. La formulación matemática de la nivelación

La formulación tradicional del modelo matemático está basada en unas variables binarias que determinan si una tarea se ejecuta o no en un determinado periodo, debiendo de cumplir evidentemente que la suma de variables binarias de una tareas coincida con su duración, y una función a minimizar que corresponde con la suma de los cuadrados de la carga de todos los periodos.

$$x_{it} = \begin{cases} 1 & \text{si la tarea } i \text{ se ejecuta en el periodo } t \\ 0 & \text{si la tarea } i \text{ no se ejecuta en el periodo } t \end{cases}$$

N Número de tareas del proyecto

T Duración del proyecto

R_i Necesidad diaria del recurso por la tarea i

d_i Duración de la tarea i

Minimizar:

$$Z = \sum_{t=1}^T \left(\sum_{i=1}^N [x_{it} \cdot R_i] \right)^2$$

Sujeto a las siguientes restricciones:

$$\sum_{t=1}^T x_{it} = d_i \quad \forall i \in N$$

$$x_{it} \in \{0, 1\} \quad \forall i \in N$$

Figura 7.3. La formulación de la nivelación.

7.2.4. La complejidad algorítmica de la nivelación

Este problema de optimización, al igual que el TCTP, también es de complejidad no polinómica, y también de orden $O(n^q)$, y más concretamente podemos decir que el universo de soluciones al problema es el producto de la holgura total más uno de todas las tareas del proyecto.

Así pues, si realizamos la anterior operación en nuestro proyecto, obtendremos entonces que el universo de soluciones es de 408.626.825.376, casi medio billón de soluciones, o bien quinientos billones en la numeración anglosajona, algo verdaderamente monstruoso e inabordable, puesto que a medida que crece el universo de soluciones, menos probabilidades tengo de que el algoritmo evolutivo de Excel encuentre la solución óptima del problema.

Un proyecto muy pequeño de tan solo quince tareas donde solamente ocho tareas no ficticias afectan a su complejidad, y no sé si podré obtener una buena solución en un tiempo razonable.

No nos pongamos nerviosos, este problema tan enorme, e inabordable computacionalmente en la práctica, lo podemos convertir en un problema relativamente fácil, y todo pasa por reducir el tamaño de las holguras de las tareas.

Si en vez de programar el proyecto en días lo programo en semanas, estaré reduciendo de forma indirecta el tamaño de las holguras, es decir las duraciones de 10 días las transformaré en 2 semanas, las de 15 días en 3 semanas, y así sucesivamente, quedando el universo de soluciones a tan solo 3.439.800, un tamaño de problema muy pequeño comparado con el original, solucionable en muy poco tiempo, y por supuesto que la solución obtenida, será exactamente la misma que la original.

Otra posible solución para reducir el tamaño del problema, y acumulable a la anterior, consiste en fragmentar el proyecto en proyectos más pequeños (siempre que esto sea posible), la solución global quizás no sea óptima, pero será muchísimo mejor que la solución original.

Otro problema con el que me encontraré cuando modele la formulación en Excel es que necesitaré un número de variables binarias que será el número de periodos multiplicado por el número de tareas, y que suele sobrepasar el límite de 200 variables de decisión que tiene el Solver, pero le daremos solución, modificándola de tal manera que tan solo use como variables de decisión el traslado que haremos de las tareas.

7.2.5. La implementación de la nivelación en Excel 2010

El modelado de este problema en Excel es bastante más complejo que los vistos hasta ahora, y por eso se ha dejado este capítulo para el final. Se va a proponer tres modelados diferentes, crecientes en su nivel de dificultad, siendo el primero de ellos el de nivelación por mínimos cuadrados.

7.2.5.1. El modelado de nivelación por mínimos cuadrados

Para ejecutar este nuevo libro, partiremos como ya es habitual de uno ya existente para modificarlo, eligiendo en este caso el llamado "Diagrama temporal con formatos y valores" del capítulo 3, porque carece de calendarios, que en este caso al trabajar en semanas no nos va a hacer falta.

La hoja "Gráfico de los flujos de caja" la renombraremos como "Histograma inicial", puesto que los valores que vamos a representar son los correspondientes a los consumos de mano de obra. Eliminaremos la línea correspondiente a los valores acumulados, seleccionándola con el botón derecho del ratón activando el menú contextual y haciendo clic en **Eliminar**.

En la hoja "Diagrama temporal" realizaremos estas modificaciones:

- Eliminaremos las columnas correspondientes a la duración, el Tmte y el Tmtt, cambiando el encabezamiento de la columna "Importe" por "Rec", donde escribiremos las necesidades diarias de mano de obra de cada una de las tareas.
- Formularemos de nuevo las celdas del diagrama temporal, para que en vez de calcular el importe diario nos refleje el consumo de recursos, así, la primera celda del diagrama temporal será: =SI (\$C5+1<=F\$4; SI (\$D5>=F\$4-1; \$E5;"");""), que deberemos de copiar a lo largo de toda su extensión, y que dejaremos en 50 semanas.
- Eliminaremos la fila donde se reflejaban los importes acumulados, al no ser necesarios para este análisis de recursos.
- Modificaremos los formatos condicionales del diagrama temporal, dejando tan solo el primero de ellos.

Una vez realizadas las anteriores modificaciones, la hoja del diagrama temporal se tendrá que ver de la forma que muestra la figura 7.4.

Y la del histograma de recursos como muestra la figura 7.5.

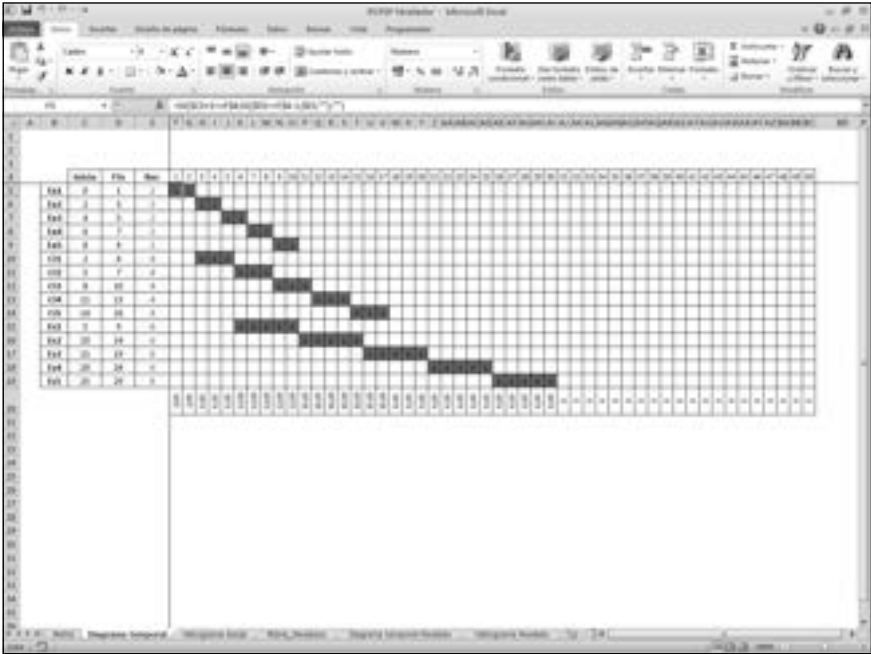


Figura 7.4. Diagrama temporal inicial.

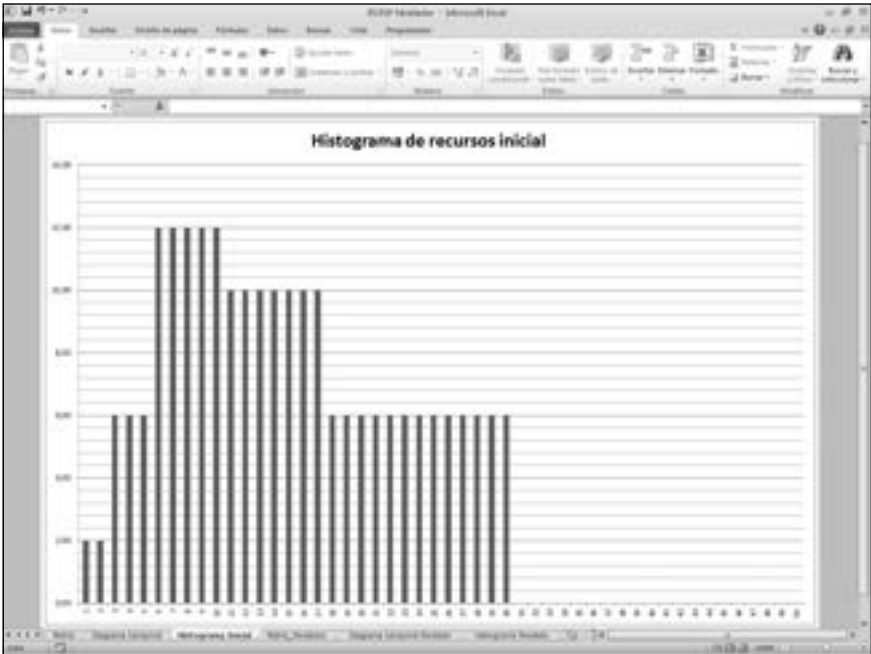


Figura 7.5. Histograma de recursos inicial.

Ahora que tenemos preparado el libro de Excel, ha llegado el momento de incorporar el modelado, para lo que necesitaremos añadir tres nuevas hojas:

- Hoja "Matriz nivelador"
- Hoja "Diagrama temporal nivelado"
- Hoja "Histograma nivelado"

La creación de estas nuevas hojas sigue el proceso que hemos realizado a lo largo de este libro, añadiendo copias de las hojas existentes y cambiándoles el nombre a continuación, para a continuación realizar los cambios necesarios.

La hoja "Matriz nivelador" tendrá este aspecto cuando esté finalizada:

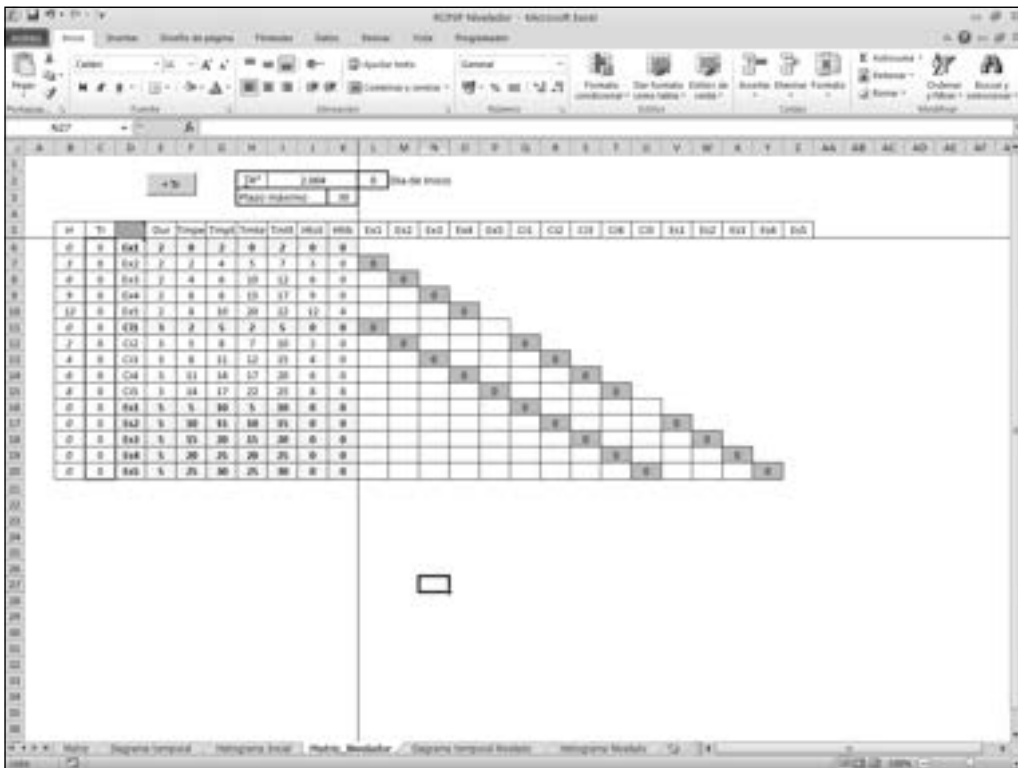


Figura 7.6. Hoja "Matriz nivelador".

Y a continuación vamos a ver cuáles son las transformaciones que hemos realizado:

- Hemos insertado dos nuevas columnas justo a la izquierda de la columna de la celda inicio. La primera columna la llamaremos "H", y corresponderá a la holgura total, de las tareas, pero con una peculiaridad, serán los valores de la holgura total y no las formulas. Una forma rápida de rellenarla será seleccionar el rango de celdas de la columna donde está la formulación de la holgura total y copiarla en el portapapeles, para posteriormente situándonos en la primera celda de la nueva columna "H", pegarla mediante **Botón derecho>Pegado especial>Valores**. La segunda columna llamada "Tr" es la que contiene las variables de decisión y que corresponden con los traslados de las tareas.
- Las celdas I2, J2 y K2, las hemos combinado en una sola celda, y será la que contenga la función objetivo del sumatorio de los cuadrados de los recursos, y la fórmula que escribiremos será ésta: **=SUMA.CUADRADOS ('Diagrama temporal Nivelado'! F20:BC20)**.
- La celda K3, es muy poco ortodoxa, al igual que la primera columna que creamos anteriormente, pero es necesaria a causa de las peculiaridades del algoritmo evolutivo usado por el Solver Excel, y en ella deberemos de escribir la duración del proyecto, pero como valor, es nuestro caso escribiremos 30.
- Añadiremos un botón de comando que llamaremos "+Tr", para que los valores que introduzcamos en los traslados se vean automáticamente reflejados en los tiempos de comienzo y final de las tareas, y tendrá el siguiente código asociado:

```
Option Explicit
Function Letra_Col(columna)
    Dim pasos As Integer
    pasos = Int(columna / 26)
    If pasos > 0 Then
        Letra_Col = Chr(pasos + 64)
    End If
    Letra_Col = Letra_Col + Chr((columna Mod 26) + 65)
End Function

Private Sub CommandButton4_Click()
    Dim i As Integer
    Dim fila As Integer
    Dim columna As Integer
    Dim Num_tareas As Integer
    Dim texto As String
    With Worksheets("Matriz_nivelador")

        'Establecemos la posición de la celda Inicio
        fila = Range("Inicio").Row
```

```

columna = Range("Inicio").Column
'Calcula el número de tareas
Do
    If .Cells(fila + Num_tareas + 1, columna).Value =
        "" Then Exit Do
    Num_tareas = Num_tareas + 1
Loop
For i = 1 To Num_tareas
    texto = .Cells(fila + i, columna + 2).formula
    .Cells(fila + i, columna + 2).formula = texto & "+" &
        Letra_Col(columna - 2) & Trim(Str(fila + i))
Next
End With
End Sub

```

Mediante este código, lo único que hacemos es añadir a la fórmula existente el valor existente en el traslado y sumarlos, así la celda F6 que tenía la siguiente fórmula: =MAX (Dia_Inicio), pasará entonces a tener =MAX (Dia_Inicio)+C6.

NOTA

Es muy importante que el código del botón "+ Tr" sea ejecutado una sola vez. En caso de equivocarnos y ejecutar el código dos veces, tan solo deberemos copiar las formulas de la matriz original en la matriz nivelada, y volver a ejecutar el código "+Tr", sin necesidad de volver a crear toda la hoja.

Esta modificación puede realizarse manualmente, aunque cuando hay muchas tareas puede ser realmente engorroso, y para evitar errores no deseados, se han eliminado los botones originales.

Ahora, si escribes cualquier valor en la columna de traslados, podremos ver como los valores son automáticamente reajustados contemplando la afección que tiene el traslado introducido en el proyecto.

La hoja "Diagrama temporal nivelado" tan solo tiene tres modificaciones, puesto que queremos que los valores que refleje sean los de la hoja "Matriz nivelador", y son:

Celda	Fórmula
C5	=ENTERO (Matriz_Nivelador! F6)
D5	=ENTERO (Matriz_Nivelador! G6)
E5	=Diagrama temporal! E5

En la hoja "Histograma nivelado" tan solo tendremos que tener la precaución de modificar los valores del gráfico para que sean tomados de la hoja "Diagrama temporal nivelado". Ya estamos en condiciones de aplicar el complemento Solver a nuestro modelado, tenemos definida la función objetiva, y preparada la hoja para establecer las restricciones al problema, tan solo nos falta incorporarlo en los parámetros de Solver:



Figura 7.7. Parámetros Solver de nivelación por mínimos cuadrados.

Parámetro	Valor
Función objetivo	\$I\$2
Para	Minimizar
Cambiando las celdas	\$C\$6:\$C\$20
Restricciones	\$C\$6:\$C\$20 <= \$B\$6:\$B\$20 \$C\$6:\$C\$20 = entero \$I\$6:\$I\$20 >= \$K\$3
Método	Evolutionary

Y tras pulsar en Resolver, y esperar unos instantes aparecerá la siguiente ventana:



Figura 7.8. Final de Solver.

Solver nos propone una solución con un valor de mínimos cuadrados de 1.924, frente a los 2004 iniciales, que es la solución óptima de problema, y los diagramas temporales inicial y nivelado comparados son:

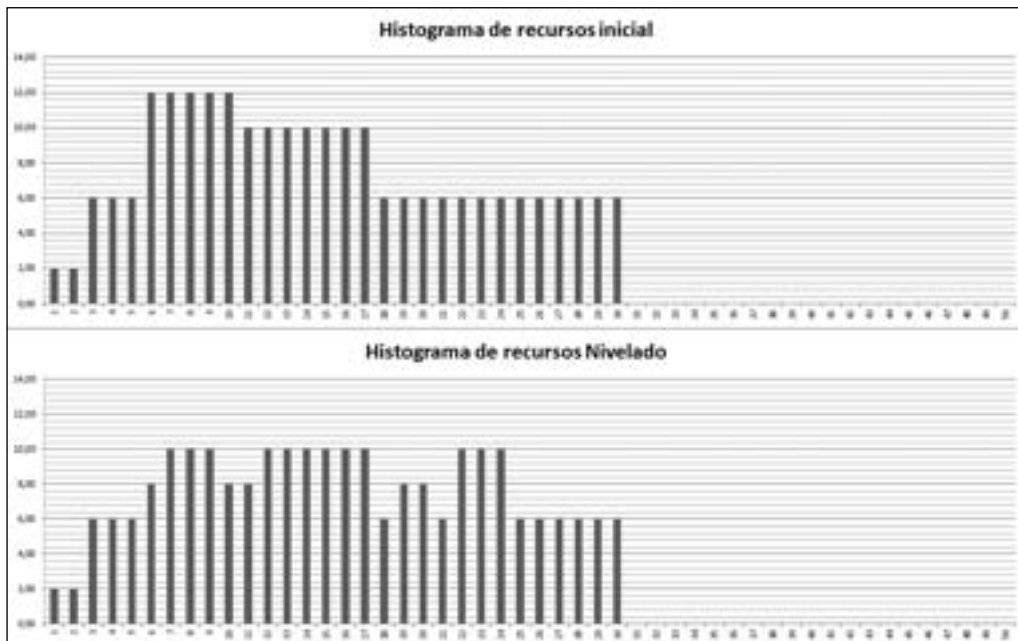


Figura 7.9. Histograma inicial vs histograma nivelado.

Hemos nivelado el histograma de recursos, teníamos unas necesidades puntuales de 12 operarios que han desaparecido, pasando a unas necesidades máximas de 10 operarios, pero todavía nos quedan preguntas por hacernos:

El valor de mínimos cuadrados de 1924 es el óptimo, pero ¿existen otros óptimos que me puedan interesar más?

NOTA

La hoja de cálculo completa se encuentra en Cap_07\RCPSP Nivelador, de los ejemplos del libro.

7.2.5.2. El modelado de nivelación por mínimos cuadrados con recuperación de holguras

Si analizamos la solución ofrecida por Solver, vemos que algunas tareas han consumido prácticamente su holgura total, y no parece muy razonable conformarnos con esta solución haciendo que el proyecto pase a ser prácticamente crítico.

Una solución a este problema consiste en castigar el valor de los mínimos cuadrados en la función objetivo con la suma de los traslados, pero de tal manera que no afecten al objetivo principal mediante la aplicación de un coeficiente "K" de minoración muy pequeño, que puede oscilar de manera orientativa entre una milésima y una centésima.

Para contemplar esta mejora del modelo, añadiremos dos nuevas filas, justo debajo de la fila 3, y entonces asignaremos nombre a las siguientes celdas:

Celda	Nombre
K4	Suma_tr
K5	Coef

Y en la celda I2, de la función objetivo escribiré esta nueva fórmula:

=SUMA.CUADRADOS ('Diagrama temporal Nivelado'! F20:BC20) +
Coef * Suma_tr

[illegible]

Y tras ejecutar el Solver de Excel, obtendré una nueva solución con el siguiente histograma comparado con el inicial (véase la figura 7.11).

Esta nueva solución parece ser mejor que la anterior, se ha obtenido un histograma óptimo, con un mínimo traslado de las tareas, y consecuentemente un proyecto menos crítico, pero además hemos conseguido que los valores grandes estén más agrupados, menos dispersos.

La hoja de cálculo completa se encuentra en Cap_07\RCPSP Nivelador plus, de los ejemplos del libro.

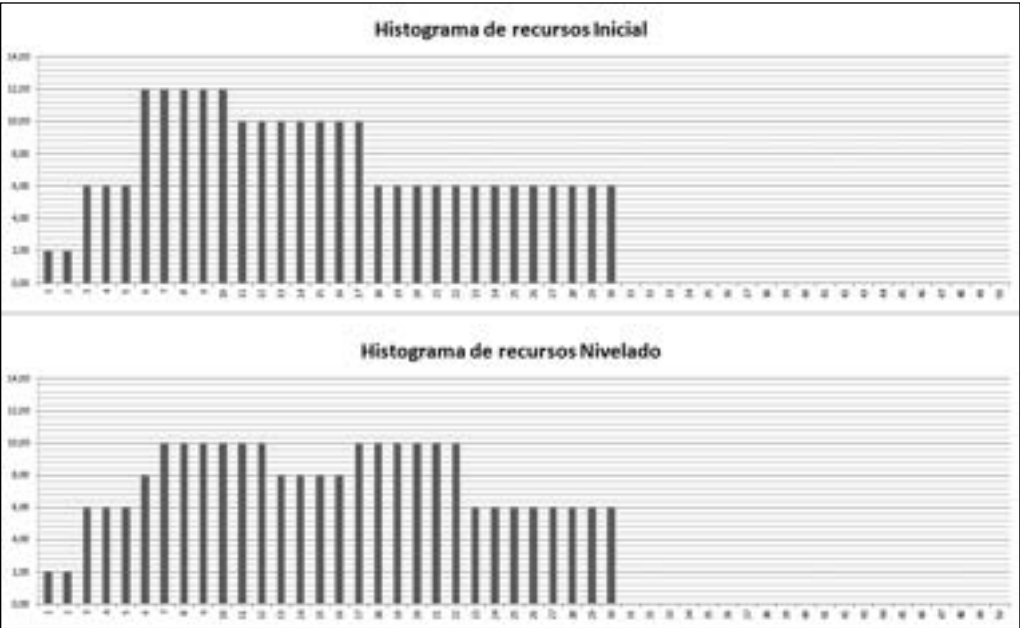


Figura 7.11. Histograma inicial vs histograma nivelado con recuperación de holguras.

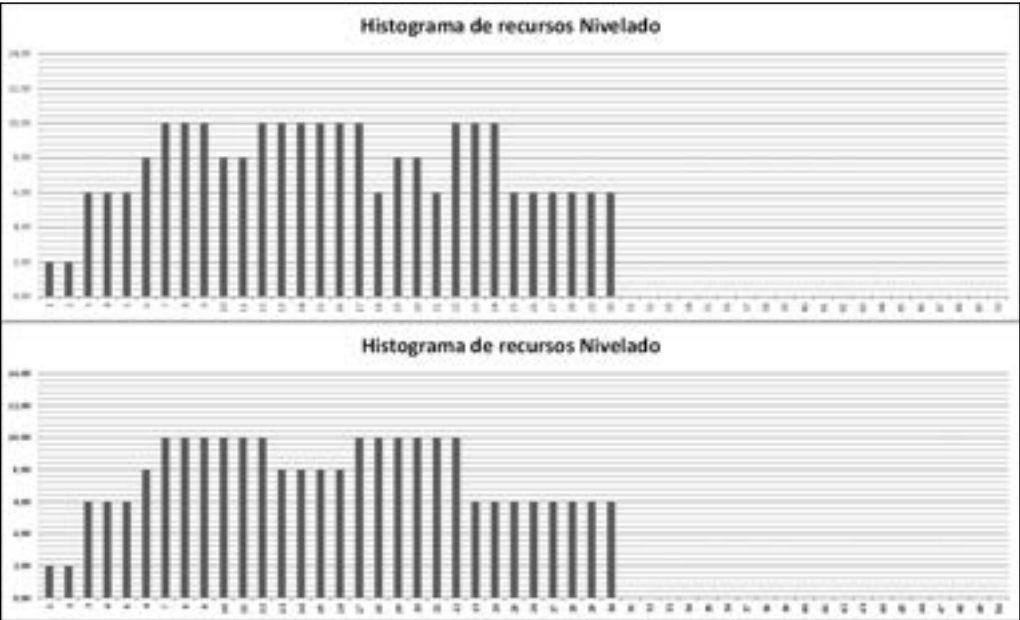


Figura 7.12. Histograma nivelado por mínimos cuadrados vs histograma nivelado con recuperación de holguras.

Prácticamente sin darnos cuenta y de una forma relativamente sencilla, hemos formulado, modelado y resuelto de manera óptima problemas muy complejos en estos dos últimos capítulos que no están contemplados en el software profesional, más caro y prestigioso disponible en el mercado, pero aún podemos ir más allá y mejorar las soluciones que tenemos hasta ahora.

En el capítulo anterior contemplamos el hecho de que las tareas no tienen porque ejecutarse de un único modo, y planteamos una formulación multi-modo. Con los recursos podemos hacer algo similar. Si cambiamos la duración de algunas de las tareas, también cambiarán las necesidades en los recursos que consumen, y aunque algunas de ellas consuman puntualmente una mayor cantidad de recursos, pero también duraran menos y dejaran huecos para ubicar otras tareas, y es posible que exista una combinación de traslados y modos de ejecución que nos ofrezca una nivelación mejor que las obtenidas hasta ahora.

7.2.5.3. El modelado de nivelación multi-modo por mínimos cuadrados con recuperación de holguras

El modelo que se propone en este nuevo libro de Excel, es una combinación del modelo del *TCTP* multi-modo y el de la nivelación con recuperación de holguras, y la hoja de la "Matriz nivelador" tendrá el aspecto que muestra la figura 7.13.

No vamos a exponer la formulación del modelo matemático, porque es una evolución de los modelos ya expuestos anteriormente, pasando a describir directamente el modelado de la hoja.

Como podemos observar, en la hoja hemos añadido nueve columnas, justo antes de la columna correspondiente a los valores de la holgura total y que denominamos "H", contemplando 3 posibles modos de ejecución para cada tarea, simplemente insertando nuevas columnas podemos contemplar tantos modos de ejecución como deseamos.

Las columnas insertadas son las que muestran la figura 7.14.

Y que corresponden a:

- Columna B: Número de orden de la tarea. Este valor servirá para identificar la fila en la función de búsqueda.
- Columna C: Modo de ejecución elegido para la tarea que encabezaremos como "Mdo". Contendrá las variables de decisión del modo de ejecución.

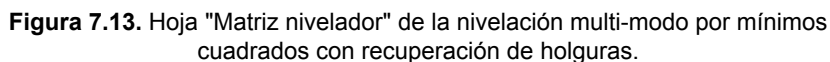


Figura 7.14. Nuevas columnas para el multi-modo de la nivelación.

- Columna D: Número de modos de ejecución de cada tarea que encabezaremos como "Mdos". Gracias a esta columna podremos reducir el universo de soluciones del problema ajustándolo a los modos ejecución realmente previstos de los tres posibles para cada tarea. La formulación para la primera celda será: =CONTAR (E8:G8). Recuerda que esta fórmula has de copiarla a lo largo de toda la columna.
- Columna E, F, G: Las duraciones correspondientes a cada modo de ejecución. Contendrán en su encabezamiento el número correspondiente a cada modo, es decir 1, 2 y 3. Para facilitar las siguientes formulaciones, a todo el rango de celdas E7:G22 le asignaremos el nombre **Duraciones_modo**.
- Columna H, I, J: Las necesidades de recursos correspondientes a cada modo de ejecución, con el mismo encabezamiento anterior. Al todo el rango de celdas H7:J22 le asignaremos el nombre **Recursos_modo**.

Una vez finalizadas las nuevas columnas, tendremos que hacer que la columna donde se encuentra la duración de las tareas en la matriz de cálculo, adopte la duración del modo de ejecución, para lo que escribiremos en la celda N8 la fórmula:

=CONSULTAH (ENTERO (C8); Duraciones_modo; B8+1)

y la copiaremos en todas las celdas de la duración de la matriz.

Tan solo nos quedará modificar en la hoja "Diagrama temporal nivelado" las necesidades de recurso elegido según el modo de ejecución, así, en la celda E5 escribiremos la fórmula:

=CONSULTAH (ENTERO (Matriz_Nivelador! C8); Recursos_modo; Matriz_Nivelador! B8+1)

y la copiaremos para todas las tareas.

Las duraciones que escribiremos para cada modo de ejecución serán:

Tarea	Duraciones en Modo		
	1	2	3
Ex1	2	1	--
Ex2	2	1	--
Ex3	2	1	--
Ex4	2	1	--
Ex5	2	1	--

Tarea	Duraciones en Modo		
Ci1	3	2	1
Ci2	3	2	1
Ci3	3	2	1
Ci4	3	2	1
Ci5	5	4	3
Es1	5	4	3
Es2	5	4	3
Es3	5	4	3
Es4	5	4	3
Es5	5	4	3

Y en las necesidades de recurso para cada una de las anteriores duraciones escribiremos:

Tarea	Recursos en Modo		
	1	2	3
Ex1	2	3	--
Ex2	2	3	--
Ex3	2	3	--
Ex4	2	3	--
Ex5	2	3	--
Ci1	4	6	8
Ci2	4	6	8
Ci3	4	6	8
Ci4	4	6	8
Ci5	4	6	8
Es1	6	8	10
Es2	6	8	10
Es3	6	8	10
Es4	6	8	10
Es5	6	8	10

Ya estamos en condición de ejecutar el complemento Solver de Excel, para el que estableceremos los siguientes parámetros:

Parámetro	Valor
Función objetivo	\$R\$2
Para	Minimizar
Cambiando las celdas	\$L\$8:\$L\$22; \$C\$8:\$C\$22
Restricciones	\$C\$8:\$C\$22 <= \$D\$8:\$D\$22 \$C\$8:\$C\$22 >= 1 \$L\$8:\$L\$22 <= \$K\$8:\$K\$22 \$C\$8:\$C\$22 = entero \$L\$8:\$L\$22 = entero \$R\$8:\$R\$22 <= \$T\$3
Método	Evolutionary

Y tras pulsar en Resolver, y esperar unos instantes aparecerá la solución óptima con un valor de mínimos cuadrados de 1.508,24, incluido el castigo por desplazamientos. El nuevo diagrama temporal nivelado será:

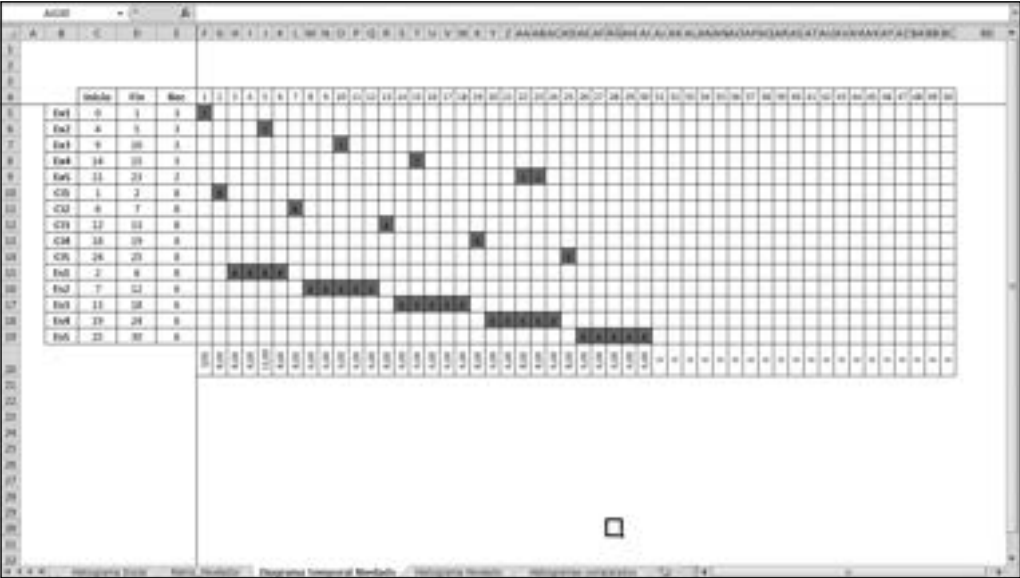


Figura 7.15. Diagrama temporal nivelado por multi-modo.

Y el histograma comparado con el original será:

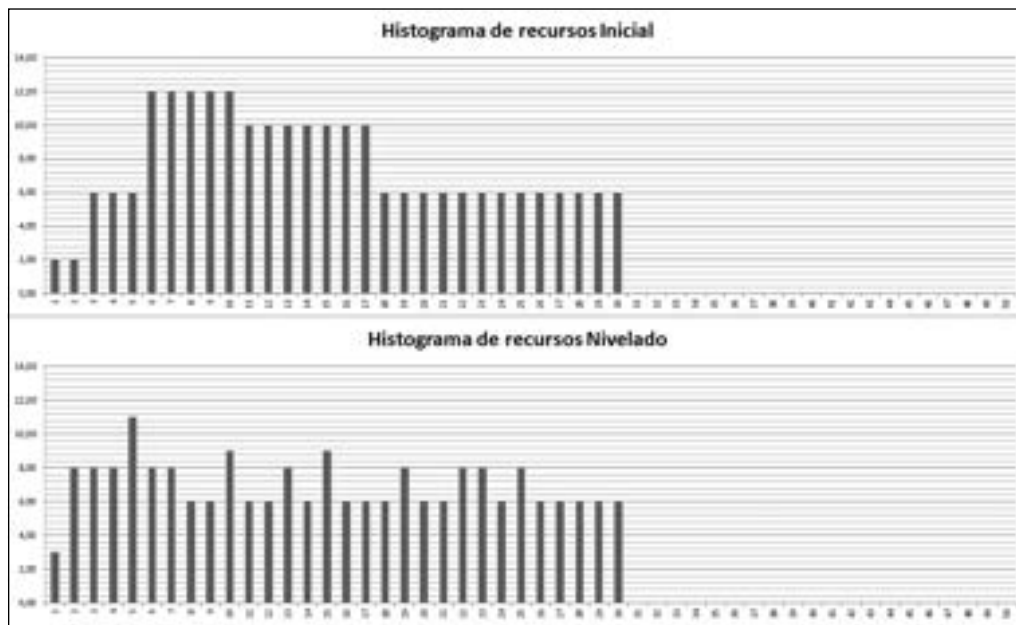


Figura 7.16. Histograma inicial vs histograma nivelado con multi-modo.

NOTA

La hoja de cálculo completa se encuentra en Cap_07\RCPSP Nivelador Multimodo, de los ejemplos del libro.

7.3. El problema de la Secuenciación

El problema de la secuenciación de proyectos es el problema realmente conocido como *RCPSP*, o bien problema de secuenciación de proyectos con recursos limitados, y es con diferencia el problema más difícil desde el punto de vista algorítmico de cuantos se han expuesto a lo largo de todo este libro.

Recordemos el histograma inicial de necesidades de mano de obra de nuestro proyecto, que requería de unas necesidades máximas de mano de obra de 12 operarios diarios, y después de ejecutar los métodos expuestos anteriormente de nivelación por mínimos cuadrados, con recuperación de

holguras y con un modelado multi-modo obtuvimos los siguientes histogramas nivelados:

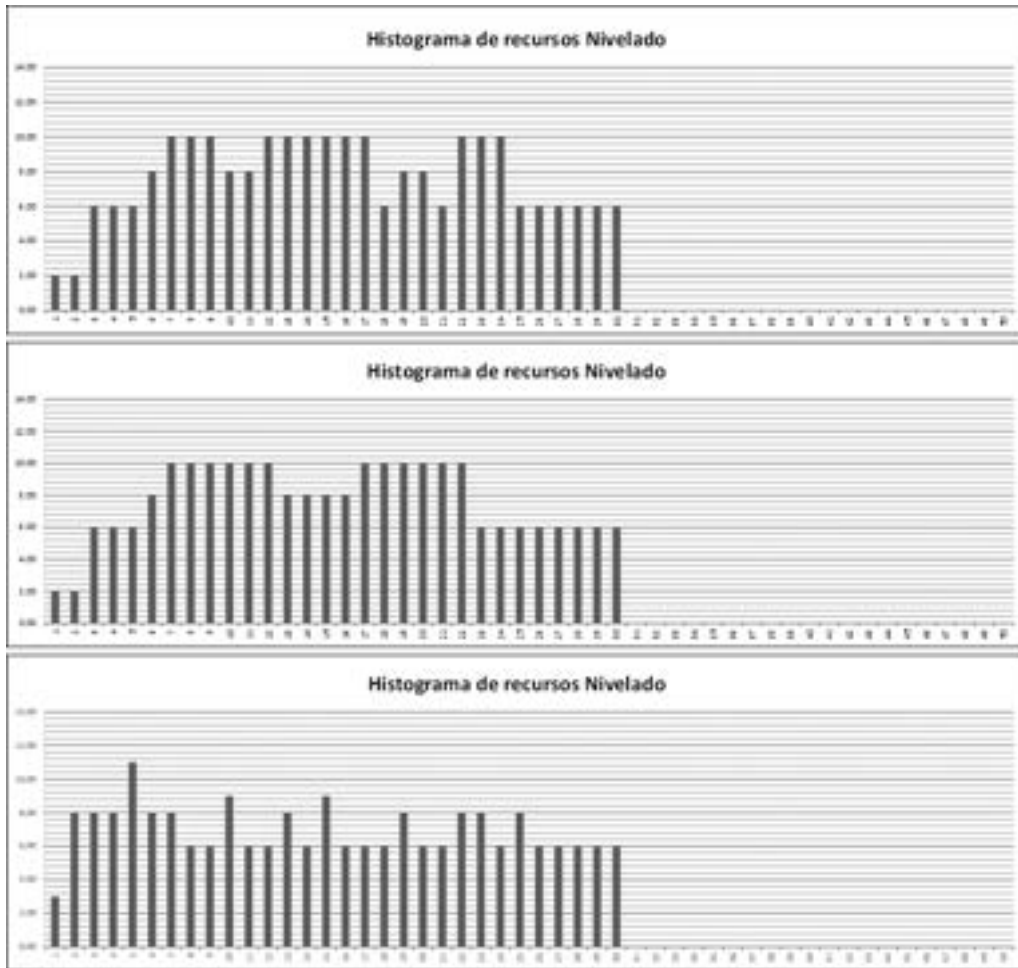


Figura 7.17. Histogramas nivelados con los tres métodos.

Como podemos observar, los histogramas han mejorado muchísimo a medida que hemos aumentados la complejidad de la formulación aplicada, con la primera solución tenemos 12 días "punta" de 10 operarios bastante dispersos a lo largo del proyecto, mediante la segunda formulación con recuperación de holguras, hemos conseguido las mismas puntas de 10 operarios, pero más agrupadas que lo hacen más eficiente, y gracias a la tercera formulación tenemos una "punta" de 11 operarios y 2 de 9 operarios, que es todavía más eficiente.

El problema con el que nos encontramos ahora, es que aunque hemos mejorado mucho el histograma inicial, no podemos ejecutar el proyecto porque nuestra disponibilidad de mano de obra es de tan solo 8 operarios.

Necesitamos encontrar la correcta combinación de traslados de las tareas, que proporcione la menor duración de proyecto posible sin sobrepasar la disponibilidad de recursos que dispongo.

Los usuarios de Microsoft Project conocen perfectamente bien este problema, puesto que lo tiene implementado en sus "Herramientas" como "Redistribuir recursos", pero lo que la gente no sabe, y que tan solo los expertos pueden intuir leyendo sus manuales, es que las soluciones ofrecidas son soluciones "factibles", no necesariamente soluciones óptimas, utilizando un algoritmo constructivo, que según las pruebas realizadas por este autor es del tipo paralelo con regla de prioridad LFT (*Latest Finish Time*).

Nosotros vamos a afrontar el problema para la búsqueda de la solución óptima, con dos tipos de modelados, el modelado tradicional con un solo modo de ejecución y multi-modo.

7.3.1. La formulación matemática del RCPSP

Esta es una de las muchas formulaciones existentes del modelo matemático para resolver el problema de secuenciación de proyecto con recursos limitados y un único recurso.

Está basada en unas variables binarias que determinan si una tarea se comienza o no en un determinado periodo, debiendo de cumplir evidentemente que la suma de variables binarias de una tarea sea la unidad, porque solo puede comenzar una vez, y una función a minimizar que corresponde con terminación de una tarea ficticia denominada "*fin*". Véase la figura 7.18.

7.3.2. La complejidad algorítmica de la secuenciación

Este problema también es de complejidad no polinómica, pero de mayor complejidad si cabe que los anteriores, porque su universo de soluciones es muchísimo mayor. Es también de orden $O(n^q)$, y más concretamente podemos decir que el universo de soluciones al problema es el producto de la holgura total más uno y la diferencia entre una cota conocida del problema

y la solución inicial de todas las tareas del proyecto. Si realizamos la anterior operación en nuestro proyecto, suponiendo una cota conocida del problema de 50 semanas (que corresponde con la suma de las duraciones de todas las tareas), obtendremos que el universo de soluciones es de $5,20 \cdot 10^{20}$, y concretamente 520.205.417.092.229.000.000 soluciones, algo más monstruoso e inabordable si cabe que los problemas anteriores. Recordaremos que a medida que crece el universo de soluciones, menos probabilidades tenemos de que el algoritmo evolutivo de Excel encuentre la solución óptima del problema.

El secreto de mi éxito radica en conocer una cota del problema que cumpla con las restricciones del problema, y cuanto menor sea esa cota, mayores probabilidades tendré de obtener una buena solución.

N Número de tareas del proyecto
 $\bar{T}$ Cota conocida de una solución del problema
 R_i Necesidad diaria del recurso por la tarea i
 $Disp$ Disponibilidad del recurso
 d_i Duración de la tarea i

$$X_{it} = \begin{cases} 1, & \text{si la tarea } i \text{ comienza en el periodo } t \\ 0, & \text{en caso contrario} \end{cases} \quad \forall i \in N$$

$$\text{Min} \sum_{t=1}^{\bar{T}} t \cdot X_{fin t}$$

Sujeto a las siguientes restricciones:

$$\sum_{t=1}^{\bar{T}} X_{it} = 1 \quad \forall i \in N$$

$$\sum_{t=1}^{\bar{T}} (t + d_i) \cdot X_{it} \leq \sum_{t=1}^{\bar{T}} (t + d_j) \cdot X_{jt} \quad \forall i \in N, j \in P_i$$

$$\sum_{i \in V} R_i \sum_{q=t-d_i}^{t-1} X_{iq} \leq Disp \quad \forall t \in \{1, 2, \dots, \bar{T}\}$$

$$X_{it} \in \{0, 1\} \quad \forall i \in V, t \in \{1, \dots, \bar{T}\}$$

Figura 7.18. La formulación del RCPSp.

7.3.3. La implementación de la secuenciación en Excel 2010

Para la implementación del problema de secuenciación, partiremos del libro "RCPSp Nivelador", porque las modificaciones que tendremos que realizar serán mínimas, afectando a unas pocas celdas y a los parámetros de Solver.

Cambiaremos el nombre de las hojas sustituyendo los conceptos "Nivelador" por los de "Secuenciador", aunque no es imprescindible, quedando como "Matriz Secuenciador", "Diagrama temporal secuenciado" e "Histograma Secuenciado".

En caso de que optemos por cambiar los nombres de las hojas, tan solo deberemos de tener la precaución de cambiar el código de los traslados, debiendo de ser:

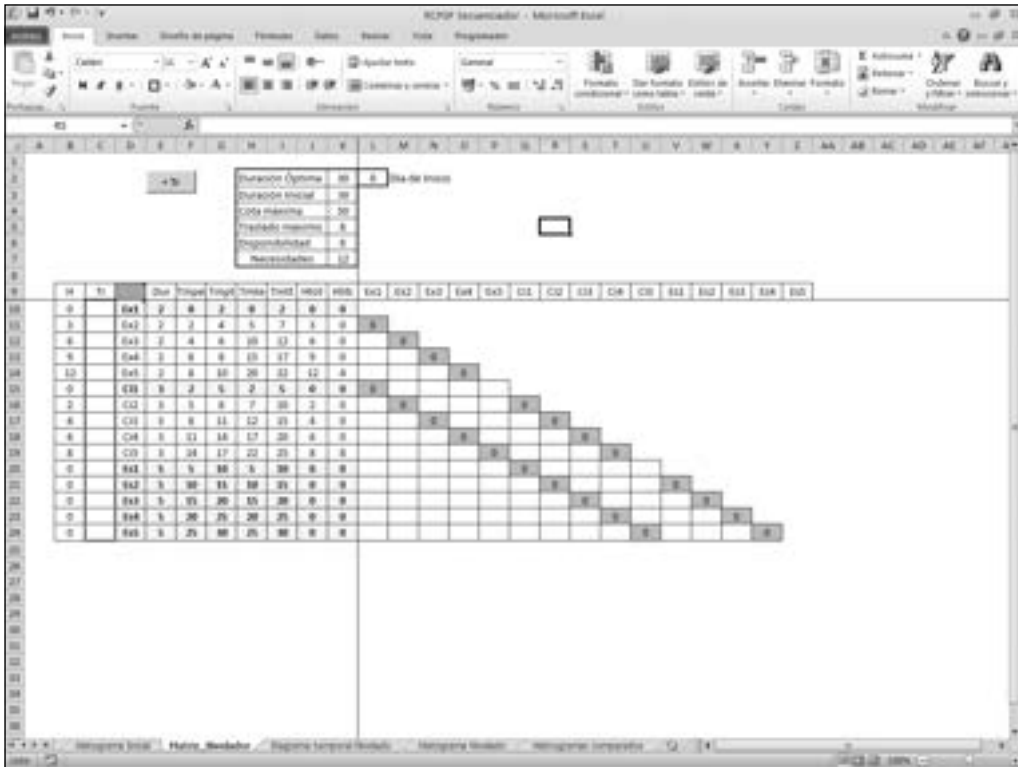
```
Option Explicit
Function Letra_Col(columna)
    Dim pasos As Integer
    pasos = Int(columna / 26)
    If pasos > 0 Then
        Letra_Col = Chr(pasos + 64)
    End If
    Letra_Col = Letra_Col + Chr((columna Mod 26) + 65)
End Function

Private Sub CommandButton4_Click()
    Dim i As Integer
    Dim fila As Integer
    Dim columna As Integer
    Dim Num_tareas As Integer
    Dim texto As String
    With Worksheets("Matriz Secuenciador")
        'Establecemos la posición de la celda Inicio
        fila = Range("Inicio").Row
        columna = Range("Inicio").Column
        'Calcula el número de tareas
        Do
            If .Cells(fila + Num_tareas + 1, columna).Value = "" Then Exit Do
            Num_tareas = Num_tareas + 1
        Loop
        For i = 1 To Num_tareas
            texto = .Cells(fila + i, columna + 2).formula
            .Cells(fila + i, columna + 2).formula = texto & "+" &
                Letra_Col(columna - 2) & Trim(Str(fila + i))
        Next
    End With
End Sub
```

7.3.3.1. La implementación de la secuenciación con un único modo

La hoja de la "Matriz Secuenciador", deberá tener el siguiente aspecto, que como se puede ver cambia muy poco de la hoja original (véase la figura 7.19).

El significado y la formulación de las celdas de la parte superior de la hoja es:



tareas individualmente cumplen la disponibilidad de recursos, por lo que la mayor cota inferior conocida será la obtenida tras colocar todas las tareas secuenciadas una detrás de otra, siendo su valor la suma de las duraciones de todas las tareas.

- Celda K5. Es de las más importantes, y contiene la restricción de los traslados de las tareas. La formularemos escribiendo $=K4-K3+MAX(B10:B24)$, que es la diferencia entre la cota inferior conocida y la duración del proyecto original, más la máxima holgura del problema inicial. Esta formulación es bastante conservadora, puesto que la verdadera restricción debería de ser para cada tarea, y no común para todas las tareas, pero requeriría añadir una nueva columna, cambiando la estructura de las columnas de la hoja utilizada con el nivelador. La formula de la primera celda de esta nueva columna insertada sería $=L\$4-L\$3+B10$, recordando que también se deberá de cambiar la restricción en Solver.

NOTA

Se ha incluido una hoja de cálculo con esta diferente alternativa que está en Cap_07\RCPSP Secuenciador alternativa, de los ejemplos del libro.

- Celda K6. Escribiremos la disponibilidad que tenemos del recurso, que en este caso es de mano de obra, y escribiremos 8 que es menor que el mejor valor obtenido gracias a la nivelación por mínimos cuadrados en cualquiera de sus tres versiones.
- Celda K7. Escribiremos la siguiente fórmula $=MAX('Diagrama temporal Secuenciado'! F20:BM20)$, que es el mayor valor de necesidades diarias del recurso que figura en el diagrama temporal.

Una vez finalizadas las formulaciones de las celdas y conocido su significado, nos encontramos en condiciones de ejecutar el complemento de Solver para encontrar esa duración óptima del proyecto, que tendrá los siguientes parámetros (véase la figura 7.20):

Parámetro	Valor
Función objetivo	\$K\$2
Para	Minimizar
Cambiando las celdas	\$C\$10:\$C\$24

Parámetro	Valor
Restricciones	\$C\$10:\$C\$24 <= \$K\$5
	\$C\$10:\$C\$24 = entero
	\$K\$7 <= \$K\$6
Método	Evolutionary



Figura 7.20. Parámetros de Solver para secuenciador.

Además de los anteriores parámetros, se ha modificado una de las opciones del complemento Solver Evolutionary, que es el tamaño de la población a su tamaño máximo permitido y que es de 200 individuos, véase la figura 7.21.

Pulsaremos en Resolver y esperaremos unos segundos más que en los problemas anteriores, obteniendo una solución de 42 días sin sobrepasar la limitación de 8 operarios que impusimos en la celda K6 de "Disponibilidad". Véanse las figuras 7.22 y 7.23.



Figura 7.21. Opciones de Solver Evolutionary.

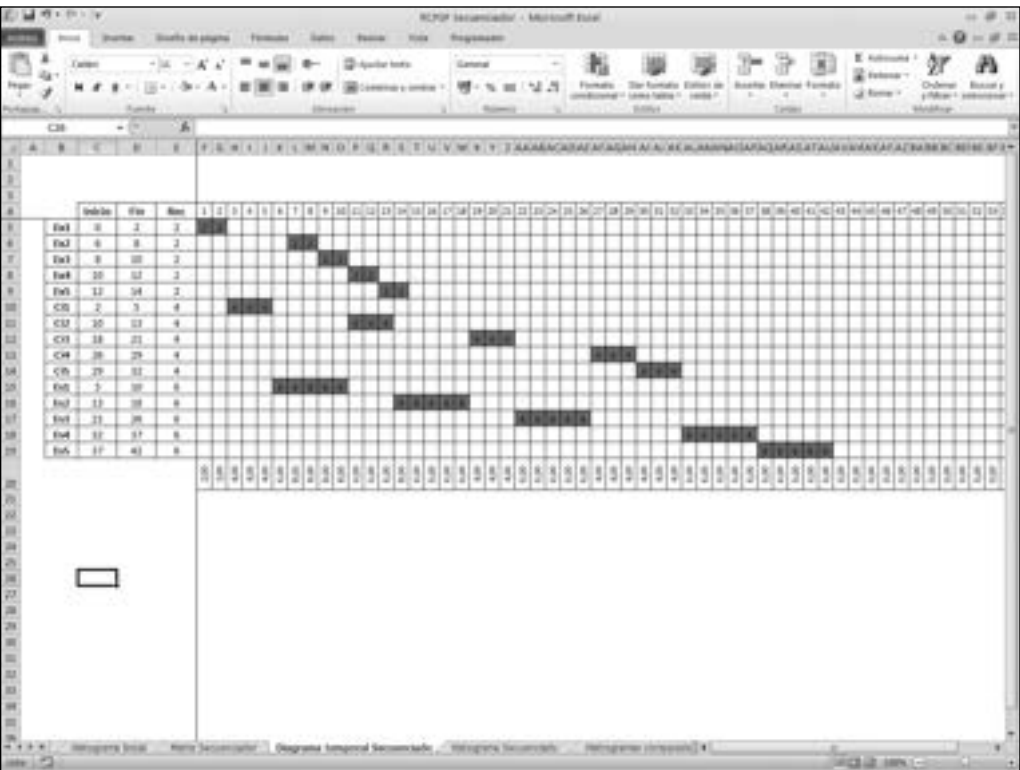


Figura 7.22. Diagrama temporal secuenciado.



Figura 7.23. Histograma de recursos limitados.

NOTA

La hoja de cálculo completa se encuentra en Cap_07\RCPSP Secuenciador, de los ejemplos del libro.

7.3.3.2. La implementación de la secuenciación multi-modo

Tal y como sucede en todos los casos anteriores, somos conscientes de que las tareas pueden ser ejecutadas de diferentes formas que se llaman "modos de ejecución" que tenían carácter discreto, y que aumentaban sus necesidades de recursos inversamente a la reducción de la duración de las tareas.

¿Puede ser posible que obtengamos una solución mejor que las anteriores a pesar de aumentar las necesidades diarias del recurso mano de obra en algunas tareas?

En casos anteriores así ha sido, pero siempre a costa de hacer más complejo el modelo, aunque en este caso las modificaciones que se hacen están prácticamente limitadas al complemento Solver, puesto que la hoja "Matriz Secuenciador" es una fusión de la hoja del nivelador multi-modo con la del secuenciador, sin ninguna característica especial, utilizando incluso las duraciones modo así como las necesidades de recurso empleadas en el nivelador con el fin de que la comparación de resultados sea más significativa, por lo que tan solo vamos a comentar los parámetros del complemento Solver.

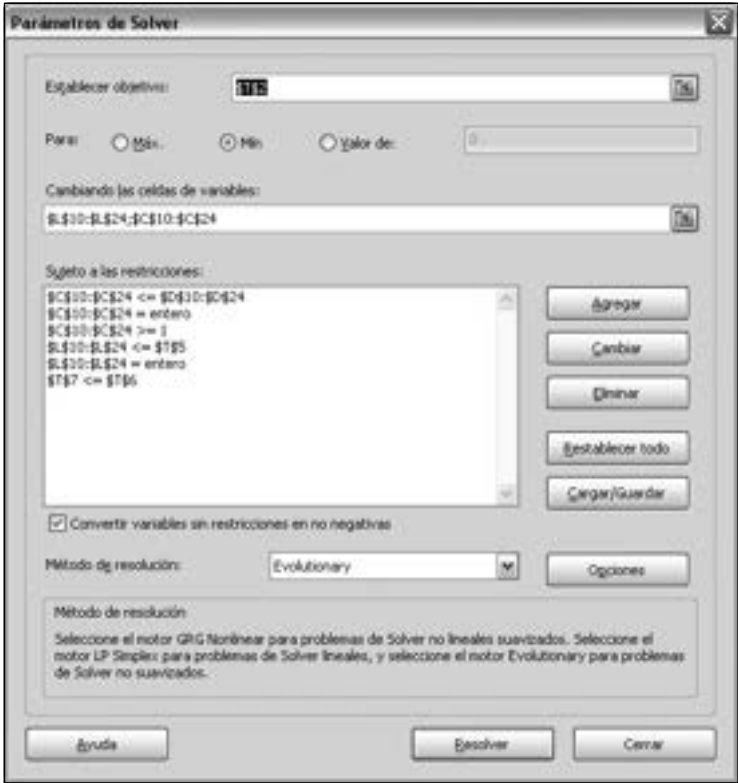


Figura 7.24. Parámetros de Solver para secuenciador multi-modo.

Parámetro	Valor
Función objetivo	\$T\$2
Para	Minimizar
Cambiando las celdas	\$L\$10:\$L\$24; \$C\$10:\$C\$24

Parámetro	Valor
Restricciones	$\$C\$10:\$C\$24 \leq \$D\$10:\$D\24 $\$C\$10:\$C\$24 = \text{entero}$ $\$C\$10:\$C\$24 \geq 1$ $\$L\$10:\$L\$24 \leq \$T\5 $\$L\$10:\$L\$24 = \text{entero}$ $\$T\$7 \leq \$T\6
Método	Evolutionary

Pulsaremos en Resolver, y transcurrido un rato bastante más largo que los anteriores obtendremos una impresionante solución de 31 días.

Recordaremos que la solución inicial era de 30 días con unas necesidades de 12 operarios, y aumentando tan solo un día la duración del proyecto, hemos conseguido un histograma con 8 operarios y prácticamente uniforme, con el siguiente diagrama temporal asociado:

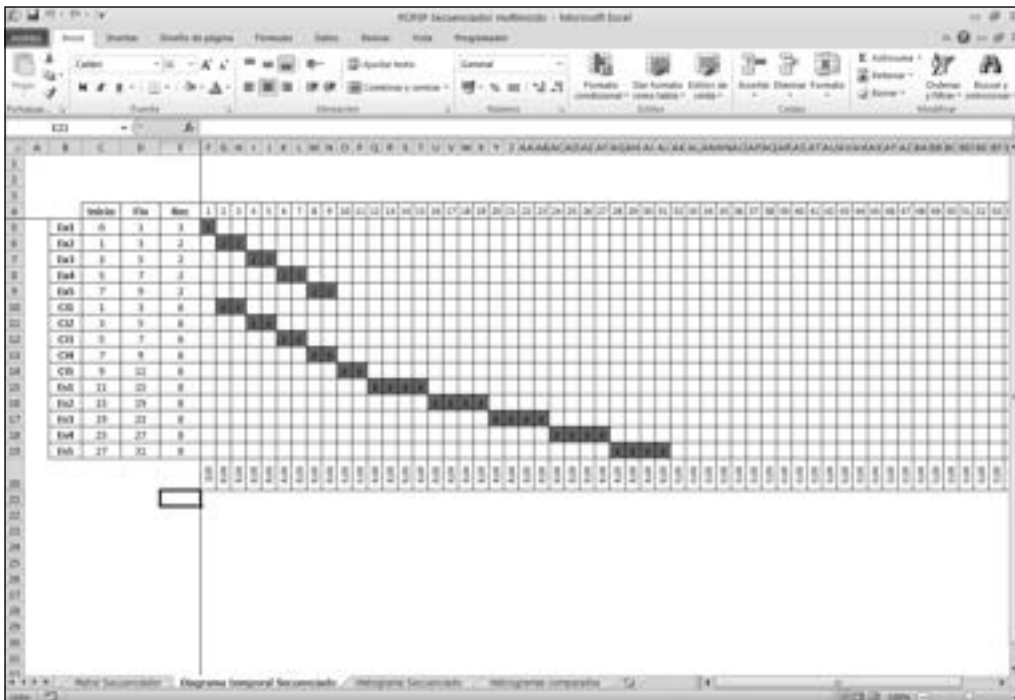


Figura 7.25. Diagrama temporal secuenciado multi-modo.

Y el Histograma de recursos comparado con el histograma original será el siguiente:

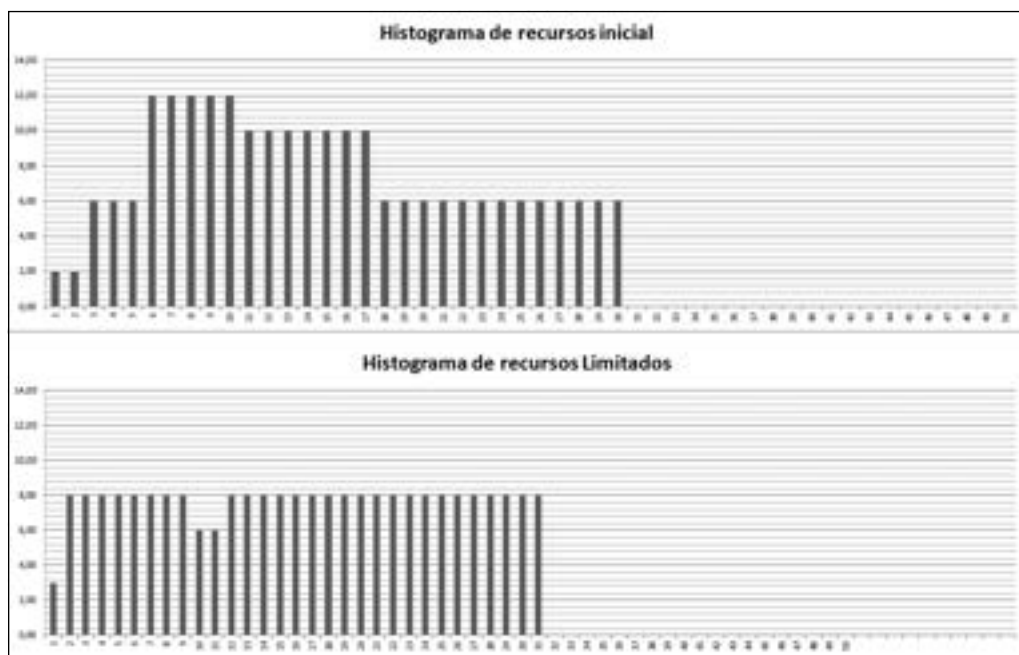


Figura 7.26. Histograma inicial vs histograma secuenciado multi-modo.

NOTA

La hoja de cálculo completa se encuentra en Cap_07\RCPSP Secuenciador multi-modo, de los ejemplos del libro.

7.4. Anexo

En este capítulo se han expuesto unos modelos para solucionar el problema de la optimización de los recursos mediante traslados en las tareas y diferentes modos de ejecución, pero estos modelos no solo pueden ser aplicados a los problemas con recursos.

Existe un problema de optimización financiera que puede ser resuelto aplicando las directrices expuestas, y que consiste en la maximización del VAN o *Valor Actual Neto*. Las tareas tienen diferentes valores de flujos de

caja, y posibilidad de ser trasladadas aquellas con valores negativos o adelantadas las de valores positivos, con diferentes posibilidades o modos de ejecución, pudiendo establecer la secuencia óptima que maximice el valor descontado de los flujos de caja.

Otro refinamiento consiste en combinar entre si los problemas como un TCTP con nivelación, o con recursos limitados, donde la única dificultad estriba en establecer la función objetivo cuando los valores no se miden en las mismas magnitudes o sus óptimos no convergen en la misma dirección debiéndolos ponderar.

NOTA

En este tipo de problemas, a veces resulta verdaderamente difícil obtener una solución óptima o buena con la utilización del motor evolutivo del complemento Solver en el primer intento. Este algoritmo evolutivo hace uso de un componente aleatorio muy importante, y ese es precisamente el secreto de sus extraordinarias soluciones, por lo que tendremos que tantear con diferentes alternativas alterando sus opciones de tasa de mutación (con valores entre 0 y 1), tamaño de la población (hasta un máximo de 200 individuos) y valor de inicialización aleatorio. Es muy importante así mismo, que siempre que se pueda no se desactive la opción requerir límites en variables, puesto que tal y como se ha comentado a lo largo del libro es muy importante acotar lo más posible el rango de las variables para reducir el universo de soluciones para lograr buenos resultados.

8

El Proyecto como inversión

8.1. Introducción

Hasta ahora hemos estudiado los proyectos como una secuencia procedimental de tareas destinadas a lograr un determinado objetivo, que normalmente suele ser la prestación de un determinado servicio, o la creación de un bien prototípico como la construcción de un edificio o una carretera, teniendo como limitaciones el plazo, la disponibilidad de recursos, y ejecutarlo al menor coste posible.

En el mundo de la gestión óptima de proyectos, se suele decir que los matemáticos e ingenieros tienen como objetivo la minimización del coste, y los economistas la maximización de los beneficios, y desde este punto de vista vamos a afrontar este nuevo capítulo, maximizar el beneficio que puedo obtener de la ejecución de mi proyecto.

Muchos proyectos suelen ser inversiones económicas muy importantes a medio-largo plazo, con una determinada corriente de cobros y pagos, que tienen como objetivo el de obtener un beneficio económico sobre una inversión inicial realizada, pero no necesariamente han de ser proyectos de inversión, pues la ejecución de grandes proyectos de infraestructura tienen tal dimensión económica y temporal que se hace necesario su estudio

y análisis desde un enfoque financiero, que maximice los beneficios que puedo obtener de su desarrollo.

Para la elaboración de este capítulo, he contado con la colaboración de la M.Eng. D^a Doria Gil Senabre, alumna de doctorado que está realizando sus investigaciones en el ámbito de las inversiones inmobiliarias en entornos de imprecisión, gran conocedora de las implicaciones del factor financiero en la gestión de proyectos.

8.2. El análisis de viabilidad de un proyecto

Como se ha comentado anteriormente, algunos proyectos, por su dimensión económica y temporal, han de ser analizados económica y financieramente, a partir de las corrientes de pagos y cobros que se producen a lo largo de su desarrollo y ejecución.

Estas corrientes de pagos y cobros, son conocidas como flujo de caja.

Los flujos de caja los definiremos como la diferencia entre todas las entradas de dinero y todas las salidas de dinero, es decir, la diferencia entre los cobros y los pagos resultantes durante la duración de un proyecto de inversión.

Es importante destacar que los flujos de caja contemplan el instante en que el cobro es hecho efectivo y de igual manera el pago, puesto que la facturación no tiene por qué coincidir con la entrada o salida efectiva de dinero, soliendo estar estos movimientos diferidos en el tiempo.

8.2.1. Medidas estáticas y dinámicas de viabilidad

El beneficio de un proyecto es la diferencia entre todos los ingresos y todos los pagos a lo largo de su evolución, pero existen dos maneras de considerar este beneficio, como un valor estático sin tener en consideración otros factores que no sean los pagos y los cobros, o como un valor dinámico que tenga en cuenta el efecto que paso del tiempo produce sobre el valor real del dinero, considerando incluso la diferencia entre tener una tesorería positiva, que nos proporcione rentabilidades financieras positivas, o una tesorería negativa que nos obligue a recurrir a fuentes de financiación con el coste que tienen aparejadas.

8.2.2. El Valor Actual Neto

El sistema dinámico más utilizado para cuantificar la bondad de un proyecto de inversión es el conocido como *Valor Actual Neto* (VAN, NPV o *Net Present Value*), y consiste en traer al momento presente (descontar) una corriente de flujos futuros, para ver si la inversión puede incrementar o no su valor, actualizar, según se aprecia en la figura 8.1.

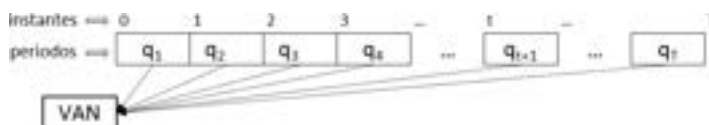


Figura 8.1. Descontar valores al instante inicial.

Descontar el valor de caja en un instante dado t al instante cero es relativamente sencillo, consistiendo en despejar el valor del capital inicial C_0 en la fórmula del capital compuesto y aplicarlo para todos los instantes (desde cero hasta T), según se ve en la figura 8.2.

$$\begin{aligned}
 C_t &= C_{0t} \cdot (1+i)^t \\
 \Downarrow \\
 C_{0t} &= \frac{C_t}{(1+i)^t} \\
 \Downarrow \\
 VAN &= \sum_{t=0}^T C_{0t} \\
 \Downarrow \\
 VAN &= \sum_{t=0}^T \frac{C_t}{(1+i)^t}
 \end{aligned}$$

Figura 8.2. Cálculo del VAN.

La anterior formulación también puede encontrarse con el valor de la inversión en el instante cero fuera del sumatorio de los flujos de caja descontados, denominándola como inversión inicial y considerando que los movimientos de dinero se producen al final de cada periodo, quedando tal y como se muestra en la figura 8.3.

$$VAN = C_0 + \sum_{t=1}^T \frac{C_t}{(1+i)^t}$$

Figura 8.3. Formulación equivalente del VAN.

8.2.3. Las tasas de interés y de inflación

Hemos visto de qué forma los flujos de caja son descontados al instante inicial del proyecto afectándolos por la inversa de una tasa de tal manera que los valores más lejanos en el tiempo tendrán un menor valor descontado, y esa tasa tiene dos afecciones diferentes, la producida por el coste del dinero, y la de la depreciación del valor del dinero causada por la inflación.

8.2.3.1. Las tasas de interés

La tasa de interés, se suele definir como el precio del dinero en el mercado financiero, aunque no solo hemos de considerar en nuestro análisis la tasa de interés del mercado, sino que deberíamos diferenciar entre la tasa de interés activa (tipo de interés que pagamos al realizar un servicio de crédito), la tasa de interés pasiva (tipo de interés que cobramos al realizar un depósito) y el coste de oportunidad o valor de la mejor opción no realizada o interés que se obtendría invirtiendo el dinero en el mercado financiero con un riesgo similar al del proyecto.

También sería interesante hacer mención a que al igual que el flujo de caja varía según el periodo que analizamos en función de las tasas, hemos de tener en cuenta que estas varían en el tiempo según el mercado, por lo que deberíamos primarlas con un factor que absorba las posibles fluctuaciones que puedan sufrir a lo largo de la vida del proyecto, llamada prima de riesgo, penalizando los flujos de caja más lejanos en el tiempo, aminorando el valor de dichos flujos y por consiguiente su influencia en la rentabilidad de la inversión.

Como hemos visto, existen diferentes tipos de interés, así que para poder efectuar los análisis lo más aproximados posibles pero de una forma simplificada, utilizaremos un único valor que pondere los diferentes tipos de interés intervinientes, llamado "*Coste de Capital Promedio Ponderado*", y que calcularemos en función del origen de los fondos que necesitaremos para mi inversión y su peso proporcional sobre esta.

Además deberemos de tener en cuenta, que todo proyecto de inversión ha de cubrir el coste de la deuda y proporcionar un retorno para los socios, que esperan obtener una tasa de interés sobre su inversión superior a la de mercado.

Esta tasa de retorno a los socios la llamaremos *Tasa Mínima de Retorno (TMR)*.

8.2.3.2. La tasa de inflación

La tasa de inflación la definimos como la pérdida del poder adquisitivo de una moneda en una economía específica.

Para poder simplificar la operativa, algunos tratados consideran la tasa de inflación en el propio valor de la tasa de interés, pero no es totalmente correcto, puesto que realmente es una depreciación del valor descontado, o dicho de otra manera una descuento sobre el descuento, tal y como se puede apreciar en la figura 8.4, donde i es la tasa de interés y g la tasa de inflación.

$$VAN = \sum_{t=0}^T \frac{C_t}{(1+i)^t} = \sum_{t=0}^T \frac{C_t}{(1+i)^t \cdot (1+g)^t} = \sum_{t=0}^T \frac{C_t}{(1+i+g+i \cdot g)^t}$$

Figura 8.4. Formulación del VAN con inflación.

8.2.4. Diferentes modos de coste pero un solo modo de cobro

Tal y como se ha visto en los capítulos anteriores cuando tratamos los problemas de la optimización de los costes y de los recursos, las tareas pueden ser ejecutadas de diferentes modos, requiriendo para cada uno de los diferentes modos de ejecución una cantidad de recursos diferente, y consecuentemente un coste diferente.

Cuando tratamos el problema de la maximización del beneficio, nos encontramos en una situación semejante, pero una particularidad, y es que aunque podamos ejecutar las tareas de diferentes modos con unos costes diferentes para cada uno de ellos, el importe parcial que cobraremos por cada una de ellas será siempre el mismo, encontrándonos con tareas que puedan ofrecer beneficios o pérdidas en función del modo en que se establezca su ejecución.

El problema surge cuando deseamos establecer el modo en que han de ser ejecutadas las tareas del proyecto.

Esto es así ya que se puede dar la circunstancia de que al acelerar la ejecución de una tarea, aunque genere pérdidas parciales, posibilite la ejecución más temprana de otras tareas, que por el beneficio que puedan producir, sea conveniente financieramente su adelanto, con el fin de generar valores de flujo de caja positivos que mejoren el comportamiento global del proyecto.

8.2.5. Adelantar o retrasar la ejecución de las tareas

Además, de cambiar el modo de ejecución de las tareas, también podemos modificar las fechas en que estas han de ser ejecutadas, de tal manera que intentaremos que las tareas que generan beneficios se ejecuten lo antes posible, retrasando aquellas que producen flujos de caja negativos.

Pero lo anterior, no es tan sencillo, porque nos podemos encontrar con tareas que por su forma de secuenciarse no sea nada evidente si han de ser adelantadas o retrasadas, debiendo encontrar el punto justo, de tal manera que podamos maximizar el beneficio del proyecto.

8.3. La implementación de la optimización de la rentabilidad con Excel 2010

Una vez expuestos los principios de la maximización del valor de un proyecto de inversión, vamos a implementarlo en Excel para su resolución mediante el complemento Solver, con dos enfoques diferentes. El primero de ellos será considerando tan solo los diferentes modos de ejecución, pero en un entorno lo más realista posible, y el segundo será considerando además los traslados en las tareas, que a causa de su mayor complejidad algorítmica será afrontado de forma más conceptual.

8.3.1. La optimización del VAN con múltiples modos de ejecución

La implementación que se va a proponer a continuación supone un planteamiento lo más realista posible, con la dificultad que este planteamiento tiene aparejado, puesto que cuanto más fiel a la realidad es un modelo, mayor es la dificultad que encontraremos para determinar el óptimo.

La pantalla principal del optimizador del VAN es muy similar a las que se expusieron cuando tratamos el problema de los costes en el capítulo 6, pero incorporando en este caso los valores de certificación (cobro) de las tareas, así como las tasas de interés y de inflación estimadas durante la ejecución del proyecto, tal y como se muestra en la figura 8.5, conservando

en el modelo las posibles bonificaciones y penalizaciones con las que contractualmente nos podemos encontrar en el caso de que el bien o servicio sea entregado antes o después de la fecha comprometida.



Figura 8.5. Optimizador del VAN.

La relativa complejidad la encontramos en la hoja "Diagrama temporal", donde se desarrolla la corriente de cobros y pagos de nuestro proyecto, y que se puede ver en la figura 8.6, pero esta dificultad estriba exclusivamente en el cálculo de los costes de estructura, puesto que estos son calculados a partir de las fechas reales de calendario, al igual que las penalizaciones o bonificaciones en las que incurriríamos.

Este planteamiento, muy cercano a la realidad, ocasiona que el complemento Solver de Excel 2010, a pesar de ofrecer unas soluciones excelentes y muchísimo mejores que las soluciones de partida, difícilmente puedan coincidir con el óptimo global, viéndonos obligados a realizar tanteos sucesivos con los diferentes parámetros del Solver con el fin de ir mejorando iterativamente la solución definitiva.

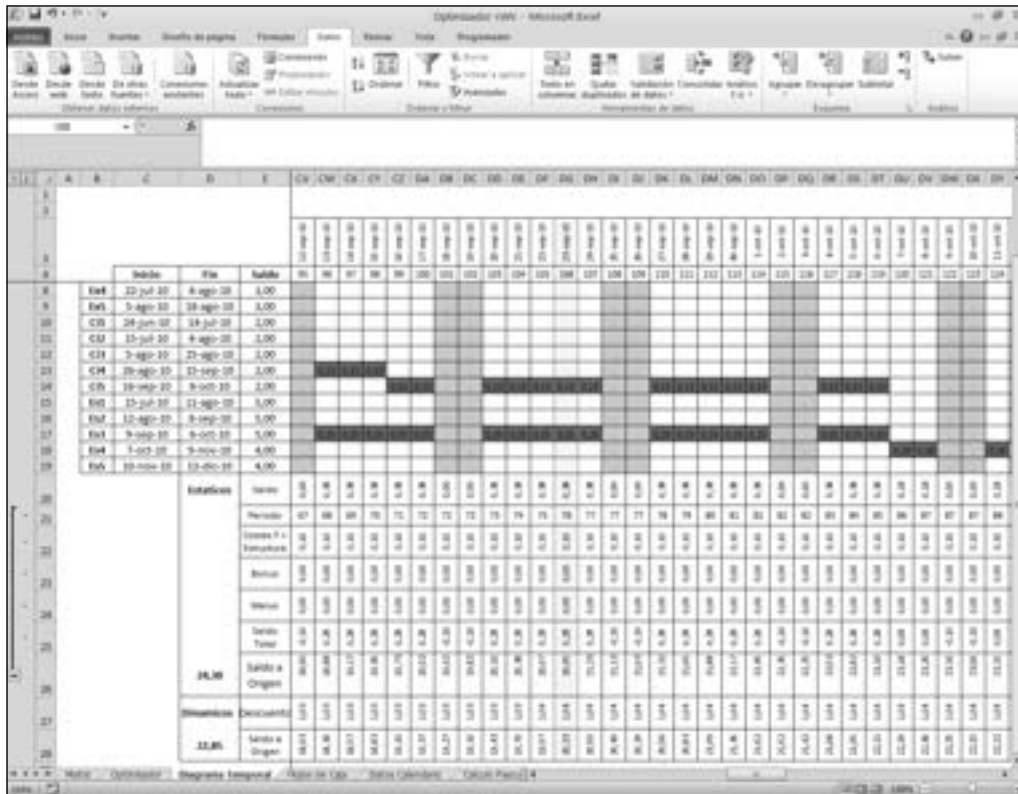


Figura 8.6. Diagrama temporal del optimizador del VAN.

La hoja "Optimizador" está estructurada en dos zonas diferentes, la superior con los datos económicos relativos al proyecto, y la inferior con los datos temporales y económicos relativos a las tareas. Cada uno de los valores económicos del proyecto que podremos modificar, y las celdas correspondientes son:

Valor	Celda	Descripción
Costes fijos	E2	Contiene los costes fijos del proyectos.
Costes diarios	E3	Contiene los costes por día del proyecto.
Bonificación	L5	Bonificación por cada día de adelanto.
Penalización	L6	Penalización por cada día de retraso.
	M5:N5	Plazo a partir de la cual se aplican las bonificaciones.
	M6:N6	Plazo a partir de la cual se aplican las penalizaciones.

Valor	Celda	Descripción
Tasa inflación anual	R4	Introduciremos la tasa de inflación anual.
Tasa interés anual	R4	Introduciremos la tasa de interés anual.

A partir de los valores anteriores, calcularemos la tasa de inflación diaria y el tipo de interés diario que son los que vamos a necesitar para calcular los valores descontados, para lo cual solamente tendremos que dividir por 365, escribiendo en la celda S4 la fórmula `=R4/365`, y en la celda S5 `=R4/365`.

La fecha de terminación del proyecto se encuentra en la celda combinada M3:N3, y se establece escribiendo la fórmula `=DIA.LAB.INTL (Fecha_Inicio; L3-1; Cod_Fin_D; Festivos)` y que ya es explicamos en el capítulo 3.

Los valores económicos de las tareas se encuentran organizados por grupos de columnas, las dos primeras columnas corresponden al código de la tarea y al número de orden que ocupan, necesaria está segunda para poder determinar la fila correspondiente a cada modo de ejecución, y que recordaremos la importancia de que sean valores estrictamente consecutivos.

El segundo grupo de columnas corresponde a los valores de la optimización, siendo la primera de ellas el modo de ejecución establecido para cada tarea y que serán las variables de decisión del modelo, la segunda columna corresponde al número de modos en que puede ser ejecutada la tarea, y que para la celda E12 escribiremos la fórmula `=CONTAR(H12:L12)`, que recordaremos que tendremos que copiar a lo largo de todas las tareas del proyecto.

Las dos siguientes columnas contienen la duración que corresponde al modo de ejecución seleccionado, y la segunda al coste correspondiente a ese modo de ejecución.

La fórmula que escribiremos en la celda F12 es ya sobradamente conocida y se ha explicado con detalle en los capítulos anteriores, y será ésta: `=CONSULTAH (ENTERO (D12); Duraciones_Modo; C12+1)`, que buscará el modo en el rango de celdas `Duraciones_Modo`, y en la celda G12 escribiremos la fórmula `=CONSULTAH (ENTERO (D12); Costes_Modo; C12+1)` que realiza la misma función pero en el rango `Costes_Modo`.

Los dos bloques de columnas siguientes, son los que contienen las duraciones y los costes de cada modo de ejecución, y en ellos estableceremos los rangos `Duraciones_Modo` para las celdas H11:L26 y `Costes_Modo` para las celdas M11:Q26.

En el último grupo de columnas escribiremos los valores que cobraremos para cada una de las tareas y que se ha encabezado como "Certificación", además de calcular el saldo resultante como la diferencia entre el importe de la certificación y el coste programado que tenemos en la columna G, de tal forma que su formulación para la celda S12 será =R12-G12.

La hoja "Matriz" es como en todos los casos anteriores, la que contiene la formulación para el cálculo de los tiempos de las tareas, pero necesitará una pequeña modificación para que adopte como duración de las tareas, la correspondiente al modo de ejecución programado. Así para la celda C5 escribiremos la fórmula =Optimizador! F12.

La hoja "Diagrama temporal" es muy similar a la misma hoja del libro "Diagrama temporal con calendario natural" que vimos en el capítulo 3, con la diferencia de que los valores que diferimos a lo largo de la ejecución de las tareas son los saldos, por lo que cambiaremos el encabezamiento de la columna E que contenía el texto "Importe" escribiendo Saldo. Para el cálculo del VAN, necesitamos calcular primeramente los diferentes valores estáticos del flujo de caja del proyecto, por lo que para trabajar de una manera más organizada se han estructurado las filas 20 a 28 en dos grupos, la primera correspondiente a los valores estáticos de la inversión, y la segunda correspondiente a los valores descontados o dinámicos.

La primera fila de los valores estáticos es la fila 20, y en ella reflejamos los valores de saldo de las tareas distribuido temporalmente y agrupado por periodos de un día cada uno, y la fórmula de la celda F20 será ésta: =SI (ESERROR (SUMA (F5:F19)); 0; SUMA (F5:F19)), que contiene un control de error para el caso de que nos encontremos en un día festivo sin valores que nos produciría un error.

La fila 21 contiene los días de trabajo del proyecto, y es necesaria como fila auxiliar para cálculos posteriores, y la fórmula para la celda F20 será =DIAS.LAB (Fecha_Inicio; F3; Festivos), y recordaremos nuevamente que tanto esta fórmula como la anterior deberemos copiarla a lo largo de todas las columnas del diagrama temporal.

La fila 22 contiene los costes fijos y de estructura del proyecto, y en este caso tendremos en cuenta que si bien los costes de estructura son iguales para cada uno de los días del proyecto, los fijos se aplican al primer día de ejecución, por lo que la formulación de este primer día será ligeramente diferente del resto. Así la fórmula que escribiremos en la celda F22 será =SI (F\$21<=Optimizador!\$L\$3; Optimizador!\$E\$3; 0)+Optimizador! E2, y en la celda G22 =SI (G\$21<=Optimizador!\$L\$3; Optimizador!\$E\$3; 0), que comprueban si el periodo de trabajo (fila 21) es menor que la duración del proyecto (\$L\$3 de la hoja "Optimizador"), para en cuyo caso añadir el valor correspondiente

al coste de estructura diario del proyecto, y que se encuentra en \$E\$3 de la misma hoja "Optimizador".

El cálculo de la bonificación, que se ha denominado "Bonus", se encuentra en la fila 23, y se calcula determinando si la fecha de terminación del proyecto que se encuentra en la celda \$M\$3, es anterior a la fecha a partir de la cual se aplican las bonificaciones (\$M\$5), para en su caso aplicar la operación de los días de adelanto por la bonificación ((Optimizador!\$M\$5-Optimizador!\$M\$3)*Optimizador!\$L\$5), en el día de finalización del proyecto ('Diagrama temporal'!F3=Optimizador!\$M\$3). La fórmula completa correspondiente a la celda F23 será: =SI (Optimizador!\$M\$3<Optimizador!\$M\$5; SI ('Diagrama temporal'!F3 = Optimizador!\$M\$3; (Optimizador!\$M\$5-Optimizador!\$M\$3)*Optimizador!\$L\$5; 0); 0).

En la fila 24 encontramos el cálculo de la penalización, y consiste en añadir, para cada día posterior a la fecha de penalización (F3>=Optimizador!\$M\$6) y hasta la fecha de finalización (F3<=Optimizador!\$M\$3), el importe correspondiente a cada día de retraso en la finalización del proyecto (Optimizador!\$L\$6), y la fórmula completa para la celda F24 será =SI (F3 >= Optimizador!\$M\$6; SI (F3<=Optimizador!\$M\$3; Optimizador!\$L\$6; 0); 0).

El saldo de cada periodo lo obtendremos restando al saldo bruto de cada periodo los costes de estructura, sumándole la bonificación y restando la penalización:

Saldo = Saldo bruto - Costes de estructura + Bonificación - Penalización, y la fórmula correspondiente para la celda F25 será =F20-F22+F23-F24.

En la fila 26 ubicaremos el saldo a origen, y será el resultado de sumar al saldo acumulado anterior el valor del saldo actual del periodo. Así, en la celda F26, que corresponde al primer periodo, escribiremos =F25 ya que no posee valores anteriores, y en la celda G26 escribiremos =G25+F26, que será la que copiaremos a lo largo de toda la fila.

Para establecer el Saldo estático del proyecto de inversión, y que se encuentra en la celda D26, tan solo tendremos que buscar la fecha de terminación del proyecto (Optimizador!L3), en el rango de valores anteriores (F21:IB26), que coincida con el número del periodo (fila 21 de la hoja), y devolver la correspondiente al saldo acumulado (fila 6 del rango).

La fórmula completa será: =CONSULTAH (Optimizador!L3; F21:IB26; 6).

Para calcular el valor descontado de cada periodo, primero tendremos que calcular el factor de descuento, que expusimos anteriormente y podemos ver en la figura 8.4.

Para que la fórmula del factor de descuento quede más clara, en la hoja "Optimizador" se han incluido los siguientes nombres de celda:

Celda	Nombre
S4	inflación
S5	tasa_int

Y la fórmula que escribiremos en la celda F27 quedará de la siguiente manera:

$$= (1 + \text{tasa_int} + \text{inflación} + \text{tasa_int} * \text{inflación}) ^ \text{'Diagrama temporal'! F4}$$

Una vez establecido el valor del descuento, tan solo tendremos que dividir el saldo a origen de cada periodo por su descuento correspondiente, y la fórmula de la celda G27 será: =SI (F21 <= Optimizador!\$L\$3; F26/F27; \$D\$28).

Esta fórmula contiene un condicional para evitar que se siga descontado el valor de la inversión una vez finalizado el proyecto (F21<=Optimizador!\$L\$3), y en caso de que así sea presentar el valor descontado del periodo de finalización del proyecto (\$D\$28).

Y por fin el valor del Saldo dinámico del proyecto de inversión, y que se encuentra en la celda D28 lo obtendremos de una forma similar a como se ha hecho para el estático, pero en el rango (F21:IB28), y devolviendo el valor correspondiente a la fila 8 del rango, para lo que escribiremos la siguiente fórmula:

=CONSULTAH (Optimizador!L3; F21:IB28; 8).

Una vez ya totalmente finalizado el modelo, tan solo nos restará ejecutar el complemento de Solver, que tendrá estos valores (véase la figura 8.7):

Parámetro	Valor
Establecer objetivo	\$M\$8
Para	Maximizar
Sujeto a las restricciones	\$D\$12:\$D\$26 <= \$E\$12:\$E\$26 \$D\$12:\$D\$26 >= 1 \$D\$12:\$D\$26 = entero
Método	Evolutionary



Figura 8.7. Complemento Solver para optimización del VAN.

La primera de las restricciones garantiza que el modo programado es menor que el número de modos disponibles, al igual que la segunda para que sea mayor o igual que el modo 1, y la tercera que este modo de ejecución sea un valor entero.

Antes de ejecutar el complemento Solver, el valor descontado de la inversión es de 8,19 unidades monetarias, y tras ejecutarlo tan solo una primera vez conseguimos mejorarlo hasta 22,59 unidades monetarias, que es casi tres veces más beneficio del programado inicialmente, aunque para lograr el óptimo que es de 28,73 unidades monetarias, tendremos que hacer varias iteraciones conservando en cada una los valores obtenidos en la anterior.

Si se analiza detenidamente la solución óptima se puede observar cómo algunas tareas son ejecutadas en un modo acelerado a pesar de aumentar el coste y no ofrecer beneficios parciales, por el beneficio global que producen en el proyecto de inversión. El gráfico en el que podemos ver los flujos de caja de cada periodo, con la punta característica que produce la bonificación en la entrega del proyecto, y el efecto que tienen las tasas de

interés y de descuento, se encuentra en la hoja "Flujos de Caja", y tendrá el aspecto de la figura 8.8.

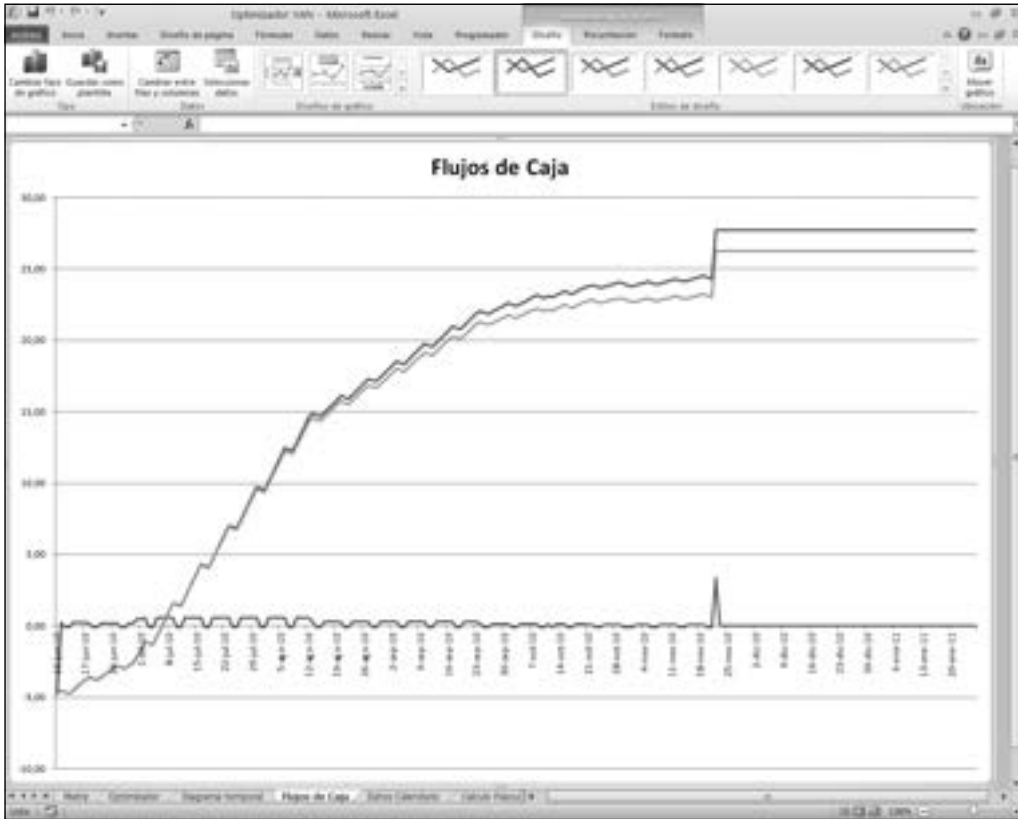


Figura 8.8. Flujos de caja e inversión descontada.

NOTA

La hoja de cálculo completa se encuentra en CDROM\08\Optimizador VAN.

8.3.2. La optimización del VAN con múltiples modos de ejecución y traslados

En la implementación anterior, hemos visto cómo acelerar algunas tareas a costa de que puedan entrar en pérdidas puede mejorar el comportamiento financiero del proyecto en su globalidad, al permitir que otras tareas se

ejecuten más pronto y así incrementar el VAN o *Valor Actual Neto*. Pero no siempre es interesante adelantar la ejecución de las tareas, porque si estas producen perdidas parciales parece que lo más lógico sería retrasarlas lo más posible en el tiempo, pero teniendo la precaución de que no se retrasen aquellas que mejorarían los flujos de caja y consecuentemente su valor descontado, debiendo de encontrar el punto más favorable, el óptimo, entre los traslados y los adelantos de todas las tareas del proyecto.

Para implementar este planteamiento, algorítmicamente bastante más difícil que el anterior, se va a proponer un modelo más sencillo, más conceptual, para facilitar que el complemento Solver encuentre más fácilmente la solución optima. El hecho de que hagamos un planteamiento diferente, no significa por supuesto que esta sea la única o la mejor forma de resolverlo, tan solo es una propuesta diferente para plantear y resolver diferentes problemas, debiendo el lector establecer la forma más adecuada para cada concreto, e incluso probar con diferentes planteamientos para adoptar aquella que mejores soluciones le proporcione.

Como en los casos anteriores, cuando tratamos el problema de los recursos, el modelo se encuentra en la hoja "Matriz Secuenciador", que tendrá el aspecto que se puede ver en la figura 8.9, hoja que se ha construido tal y como comentamos en el capítulo 7.

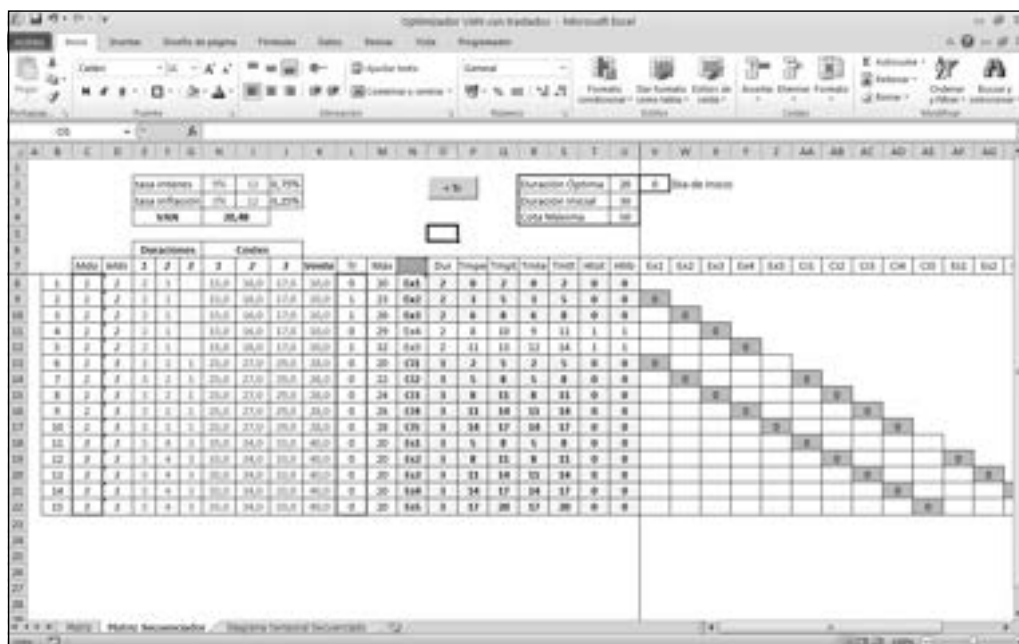


Figura 8.9. Optimizador del VAN con traslados.

La hoja "Matriz Secuenciador" está preparada para tres diferentes modos de ejecución para las tareas, aunque por supuesto puede ser ajustada para contemplar tantos diferentes como se deseen, aunque siempre tendrá como en la implementación anterior un único modo para el cobro o la certificación parcial de las tareas del proyecto.

Para hacer el modelo más fácil algorítmicamente, se ha propuesto un planteamiento para los periodos en meses, aunque por supuesto éste es un valor totalmente ajustable y a discreción del responsable del proyecto, y el diagrama temporal resultante lo podemos encontrar en la hoja "Diagrama temporal Secuenciado" (véase la figura 8.10).

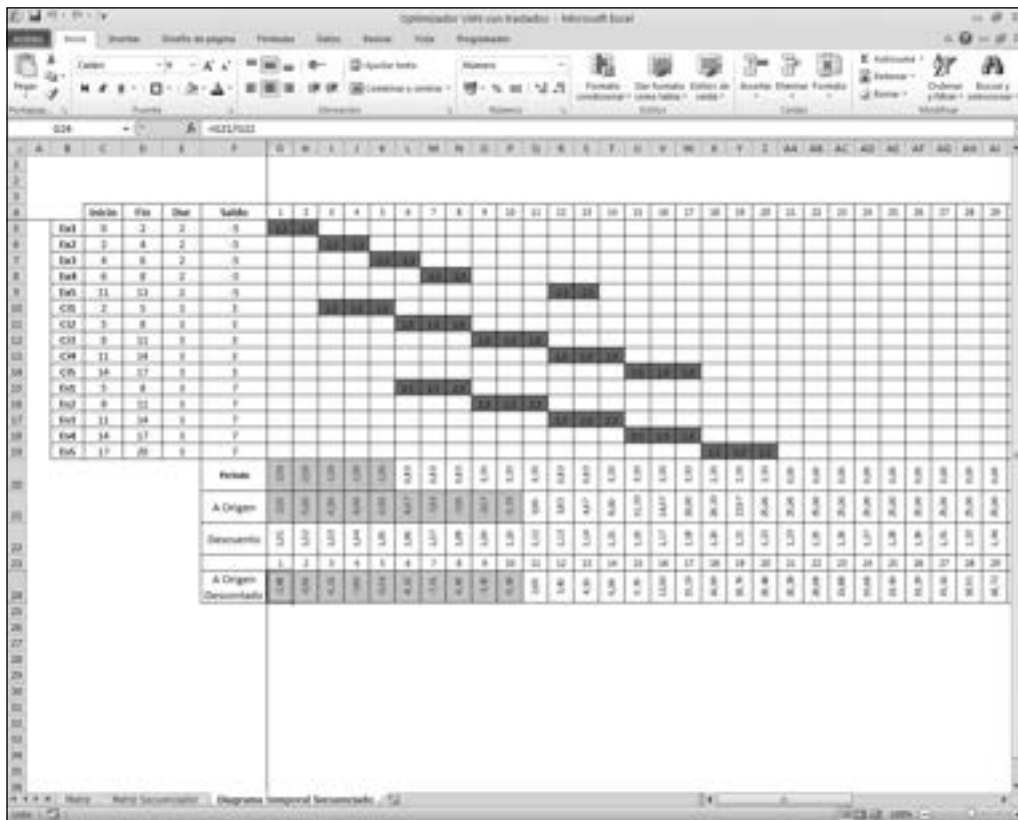


Figura 8.10. Diagrama temporal del optimizador del VAN con traslados.

Pero veamos detenidamente cómo se ha construido el modelo.

Los valores económicos del proyecto se han dispuesto en la zona superior de la hoja, y cada una de las celdas utilizadas, así como su significado es el siguiente:

Celda	Descripción
H2	Tasa interés nominal anual.
H3	Tasa de inflación nominal anual.
I2	Número de periodos a considerar para la tasa de interés.
I3	Número de periodos a considerar para la tasa de inflación.
J2	Tasa interés nominal para cada periodo.
J3	Tasa de inflación nominal para cada periodo.
H4:I4	Valor Actual Neto del proyecto.
U2	Duración óptima del proyecto.
U3	Duración inicial del proyecto.
U4	Cota máxima.

Para facilitar las formulaciones, a las celdas J2 y J3 se les ha nombrado "interés" e inflación respectivamente, resultando una fórmula para la celda combinada H4:I4 que será ésta: =CONSULTAH (U2; 'Diagrama temporal Secuenciado'! G23:BN24; 2), muy similar a la utilizada en el modelado anterior, que lo que hace es buscar el valor de la duración del proyecto en el rango G23:BN24 de la hoja "Diagrama temporal Secuenciado", y devolver el valor correspondiente en la fila 2 del rango.

Las celdas U3 y U4 que corresponden a la duración inicial del proyecto y a la cota máxima, son utilizadas para limitar el universo de soluciones del modelo, acotando los valores que pueden adoptar los traslados de las tareas, y así facilitar su resolución.

Recordaremos que el botón de comando "+Tr", reformula la columna del tiempo más pronto de empezar de las tareas, y que tan solo debe ser ejecutado una vez. En caso de equivocarnos y ejecutar el código dos veces, tan solo deberemos copiar las fórmulas de la matriz original en la matriz secuenciador, y volver a ejecutar el código "+Tr", sin necesidad de volver a crear toda la hoja.

Con respecto a la zona inferior, se encuentran los valores económicos y de duraciones de las tareas del proyecto, correspondiendo las tres primeras columnas al número de orden de las tareas, la segunda columna al modo de ejecución programado y que será parte de las variables de decisión del complemento Solver, y la tercera columna que contiene los modos de ejecución en que pueden ser ejecutadas cada una de las tareas, y que para la celda D8 contendrá la fórmula =CONTAR (E8:G8).

Las siguientes columnas contienen las duraciones modos y los costes modo para cada tarea, y se han agrupado en los nombres *Duraciones_modo* para el rango de celdas E7:G22, y *Costes_modo* para el rango H7:J22, que tiene como encabezamiento el modo de ejecución correspondiente.

La siguiente columna es la columna correspondiente al valor que cobraremos por cada una de las tareas, y que como ya se ha comentado en varias ocasiones anteriores, tan solo adopta un posible modo.

Las dos columnas siguientes (L y M) corresponden a los traslados programados para las tareas y el traslado máximo permitido para cada una de ellas. El traslado programado, que serán el resto de las variables de decisión, se ha encabezado como "Tr", y el traslado máximo como "Máx", siendo esta última formulada a partir de la holgura total original más la cota máxima (\$U\$4) menos la duración inicial del proyecto (\$U\$3), para lo que escribiremos en la celda M8 la fórmula `=Matriz!H5 + $U$4 - $U$3`.

La hoja del "Diagrama temporal Secuenciado", es bastante similar a la del modelado anterior, pero muchísimo más sencilla, presentando como prácticamente único elemento a destacar la formulación correspondiente a la columna del saldo que será para la celda F5 la siguiente: `=Matriz Secuenciador!K8 -CONSULTAH ('Matriz Secuenciador'!C8; Costes_modo; 'Matriz Secuenciador'!B8 + 1)`, que lo que hace es buscar el coste del modo de ejecución programado ('Matriz Secuenciador'!C8), en el rango *Costes_modo*, y restarlo del importe que cobraremos de cada tarea ('Matriz Secuenciador'!K8).

Una vez creado el modelo, tan solo nos restará ejecutar el complemento Solver, tal y como se muestra en la figura 8.11, adoptando los siguientes valores para sus parámetros:

Parámetro	Valor
Establecer objetivo	\$H\$4
Para	Maximizar
Sujeto a las restricciones	\$C\$8:\$C\$22 <= \$D\$8:\$D\$22
	\$C\$8:\$C\$22 >= 1
	\$C\$8:\$C\$22 = entero
	\$L\$8:\$L\$22 <= \$M\$8:\$M\$22
	\$L\$8:\$L\$22 = entero
Método	Evolutionary

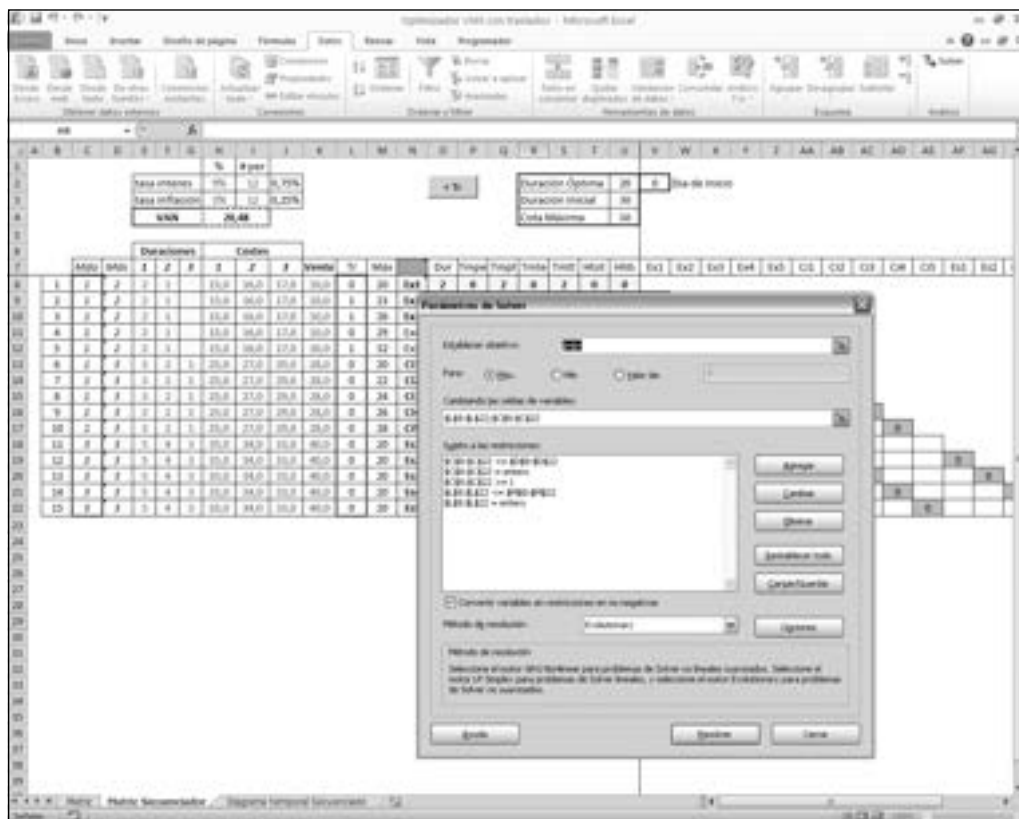


Figura 8.11. Complemento Solver para optimización del VAN con traslados.

La primera y segunda de las restricciones garantizan que las tareas se ejecutan en un modo de ejecución mayor o igual a 1, pero nunca mayor que los modos disponibles, y la tercera restricción que estos modos serán valores enteros.

Las dos últimas restricciones obligan a que los traslados sean menores que los traslados máximos establecidos y que además estos sean valores enteros.

Una vez ejecutado el complemento, obtenemos un valor para el VAN de 20,48 unidades monetarias, trasladando tres de las cimentaciones un mes en su ejecución, valor que mejora considerablemente hasta casi duplicar las 11,12 unidades monetarias del proyecto inicial, quedando evidenciado la necesidad de retrasar aquellas tareas que arrojan saldos negativos, y de adelantar aquellas con un saldo parcial positivo, encargándose Solver de encontrar el justo punto de equilibrio que maximiza el valor descontado del proyecto de inversión.

NOTA

La hoja de cálculo completa se encuentra en Cap_08\ Optimizador VAN con traslados, de los ejemplos del libro.

A

Glosario de símbolos y abreviaturas

α : Factor de forma de la distribución Beta.

β : Factor de forma de la distribución Beta.

σ_i : Desviación típica de la tarea i .

σ_i^2 : Varianza de la tarea i .

σ_p : Desviación típica del proyecto.

σ_p^2 : Varianza del proyecto.

$\sum C t_i$: Suma de los costes de producción de las tareas ejecutándose en t_i .

μ_p : Duración media del proyecto.

μ_i : Duración media de la tarea i .

A

a: Duración optimista.

ACWP: *Actual cost of work performed* o coste realizado del trabajo tealizado (CRTR).

AoA: *Activity on Arrow* o tarea en flecha.

AoN: *Activity on Node* o tarea en Nodo.

AP: *Adherencia a la Programación*.

AVG: Análisis del valor ganado.

B

b: Duración pesimista.

BCWP: *Budgeted cost of work performed* o coste presupuestado del trabajo realizado (CPTR).

BCWS: *Budgeted cost of work scheduled* o coste presupuestado del trabajo programado (CPTP).

C

CC(z) ij : Relación de comienzo-comienzo entre i y j con un desfase de z lapsos de tiempo.

CCBM: *Critical Chain-Buffer Management* o bien "gestión de la cadena crítica".

CD _{i} : Grado de criticidad de la tarea i .

Ce _{p} : Coste de estructura del proyecto.

CF(z) ij : Relación de comienzo-final entre i y j con un desfase de z lapsos de tiempo.

Cf: Costes Fijos del proyecto.

C _{ik} : Coste de ejecutar la tarea i en modo k .

Cn _{i} : Coste de producción normal de la tarea i .

CPI: *Cost performance index* o índice de rendimiento de coste (IRC).

CPM: Critical path method.

CP _{p} : Coste de producción del proyecto a un plazo p .

CPTP: Coste presupuestado del trabajo programado o *Budgeted cost of work scheduled* (BCWS).

CPTR: Coste presupuestado del trabajo realizado o *budgeted cost of work performed* (BCWP).

CRTR: Coste realizado del trabajo realizado o *actual cost of work performed* (ACWP).

CSI: *Cost schedule performance index* o índice de rendimiento de coste-programa (IRCP).

Ct_i : Coste de producción en t de la tarea i .

$CTP_{\text{óptimo}}$: Coste total óptimo del proyecto.

CTP_p : Coste total del proyecto para un plazo p .

CV: *Cost variation* o desviación de coste (DC).

Cv : Costes Variables del proyecto.

D

d_1 : Soporte inferior del número difuso.

d_2 : Núcleo del número difuso.

d_3 : Soporte superior del número difuso de la holgura total.

Da_i : Duración acelerada de la tarea i .

D_{Buf} : Duración de la tarea *Buffer*.

DC: Desviación de coste o *cost variation* (CV).

d_i : Duración de la tarea i .

Dn_i : Duración normal de la tarea i .

DP: *Desviación de programa* o *schedule variation* (SV).

Dt_i : Duración en t de la tarea i .

E

e : Error máximo de la simulación.

EAC: *Coste estimado a la terminación*.

EF_i : *Early finish* o tiempo más pronto de terminar ($Tmpt_i$).

ES: *Earned schedule* o *programación ganada* (PG).

ES_i : *Early start* o tiempo más pronto de empezar ($Tmpe_i$).

ETC: Coste estimado para terminar.

EVA: *Earned value analysis* o análisis del valor ganado.

F

$FC(z)$ ij : Relación de final-comienzo entre i y j con un desfase de z lapsos de tiempo.

$FF(z)$ ij : Relación de final-final entre i y j con un desfase de z lapsos de tiempo.

G

GRG : Generalized Reduced Gradient.

H

h_1 : Soporte inferior del número difuso de la holgura total.

h_2 : Núcleo del número difuso de la holgura total.

h_3 : Soporte superior del número difuso.

Hc_i : Holgura condicional.

Hi_i : Holgura independiente.

Hl_i : Holgura libre.

Ht_i : Holgura total.

I

IRC : Índice de rendimiento de coste o también *cost perfomance index* (CPI).

$IRCP$: Índice de rendimiento de coste-programa o *cost schedule performance index* (CSI).

IRP : Índice de rendimiento de programa o *schedule performance index* (SPI).

K

K : Coeficiente de minoración de los traslados.

k : Factor de confianza de las estimaciones del experto.

L

LF_i : *Latest finish* o tiempo más tarde de terminar (T_{mtt_i}).

LFT : *Latest Finish Time* o tiempo de terminación más tardío.

LS_i : *Latest start* o tiempo más tarde de empezar (T_{mte_i}).

M

m: Duración moda.

MVG: Método del valor ganado.

O

$O(n^q)$: Complejidad algorítmica exponencial.

P

PDM: *Precedence Diagramming Method* o bien método de diagrama de precedencias.

PERT: *Program Evaluation and Review Technique*.

PG: *Programación ganada* o *earned schedule (ES)*.

PMI: *Project Management Institute*®.

R

$r(i)$: Número de modos en que puede ejecutarse la tarea i .

RCPSP: *Resource Constrained Project Scheduling Problem* o problema de secuenciación de proyectos con recursos limitados.

R_i : Necesidad diaria de recurso por la tarea i .

S

SPI: *Schedule performance index* o índice de rendimiento de programa (IRP).

SV: *Schedule variation* o desviación de programa (DP).

S_α : alfa-corte de un número difuso.

T

T : Duración del Proyecto.

$TCPI$: Índice de desempeño a la terminación.

$TCTP$: *Time cost trade off problem* o problema de optimización de costes.

te_i : Duración estimada de la tarea i .

TFN : *Triangular fuzzy number* o número difuso de tipo triangular.

$Tmpe_i$: Tiempo más pronto de empezar o *early start* (ES_i).

$Tmpt_i$: Tiempo más pronto de terminar o *early finish* (EF_i).

$Tmte_i$: Tiempo más tarde de empezar o *latest start* (LS_i).

$Tmtt_i$: Tiempo más tarde de terminar o *latest finish* (LF_i).

U

u : Número aleatorio.

V

VAC : Variación a la terminación.

VAN : Valor Actual Neto.

VBA : *Visual Basic for applications*.

X

x_{it} : Variable de decisión binaria de la tarea i en el periodo t .

Z

z_{ij} : Desfase entre i y j .

z : Valor de la desviación típica de la distribución normal estandarizada.

B

Bibliografía

Álvarez - Valdés, R., & Tamarit Goerlich, J. M. (1989). Heuristic Algorithms for Resource-Constrained Project Scheduling: A review and Empirical Analysis. En Advances in Project Scheduling. R. Skovinski and J. Weglarz. Elsevier Sc.

Badiru, A. B. (1995). Comprehensive Project Management: Integrating Optimization Models, Management principles and computers. Prentice hall PTR. ISBN 01303092257.

Badiru, A. B. (1996). Project Management in Manufacturing and high Technology operations. New York: John Wiley & Sons. ISBN 0471127213.

Badiru, A. (2008). Industrial Project management: Concepts, Tools and Techniques. Boca Raton: CRC. Taylor & Francis. ISBN 0849387736.

Bertelsen, S. (2003). Complexity-Construction in a New Perspective. Proceeding 1th annual meeting of the international group for Lean Construction, Blacksburg, Virginia USA.

Blazewicz, J., & Lenstra, J. a. (1983). Scheduling subject to resource constraints: classification and complexity. Discrete Applied Mathematics 5, 11-24.

Bonnal, P., & Didier Gourc, G. L. (2004). Where Do We Stand with Fuzzy Project Scheduling? *Journal of Construction Engineering and Management*, 114-123.

Buckley, J. (1989). "Fuzzy PERT", "Application of Fuzzy Set Methodologies in Industrial Engineering", Pp 103-114. Amsterdam: Elsevier Science.

Buitrago, L. E. (2006). Tesis doctoral "Análisis de una red de actividades con actividades de duración aleatorias e inciertas". Universidad Pontificia.

BURGESS, KILLEBREW. (2001). "Variation in Activity Level on a cyclic Arrow Diagram". *Journal of Industrial Engineering*", vol. 13, núm. 2, Marzo-Abril, 1962.

Chanas, S. a. (2001). Critical path analysis in a network with fuzzy activity times. *Fuzzy Sets Syst.* 122(2), 195-204.

Chanas, S. a. (2002). The computational complexity of the interval critical path method. *European Journal of Operations Research.* 136(3), 541-550.

Chanas, S. a. (1981). The use of Fuzzy variables in PERT. *Fuzzy Sets Syst.*, 5, 11-19.

Chanas, S., & Zielinski, P. I. (2003). On the hardness of evaluating criticality of activities in a planar network with duration intervals. *Operations Research Letters* 31, 53 – 59.

Chang, I. S., Tsujimura, Y., Gen, M., & Tozawa, T. (1993). An efficient approach for large project planning based on fuzzy Delphi method. *Fuzzy Sets and Systems* 76, 227-288.

Charte, Francisco. (2010). "Excel 2010". Anaya Multimedia. ISBN 978-84-415-2788-1.

Chen-Tung Chen, S.-F. H. (2007). Applying fuzzy method for measuring criticality in project network. *Information Sciences* 177, 2448-2458.

Chen-Tung Chen, S.-F. H. (2007). Applying fuzzy method for measuring criticality in Project Network. *Information Sciences.* 177, 2448-2458.

Luis Yu Chuen-Tao. (1967). "Aplicaciones practicas del PERT y CPM. Nuevos métodos de dirección para planificación, programación y control de proyectos". Ediciones Deusto.

Crandall, K. (1973). Project Planning with Precedence Lead/Lag Factors. *Project Management Quarterly* 4, 18-27.

Cui, W., Yue, C., Yin, Z., & Yao, S. (2006, June 21 - 23). Criticality Analysis in Activity-on-Arc Networks with Generalized Precedence Relations

(GPRs). 6th World Congress on Intelligent Control and Automation. Dalian, China.

Demeulemeester, E. L. (2002). Project scheduling: a research handbook. International series in operations research & management science, Volume 49. Springer. ISBN 10207051, 9781402070518. 685 páginas.

Dobois, D., & Prade, H. (1988). Theorie des possibilities. 2nd. Paris: Ed. Masson.

Fondahl, J. W. (1961). A Non-Computer Approach to the Critical Path Method for the Construction Industry. Department of Civil Engineering, Stanford University.

Eliyahu M. Goldratt. (1997). "Critical Chain" Ediciones Castillo.

Hajdu, M. (1997). Networks Scheduling Techniques for Construction Project Management. Nonconvex Optimization and its applications. Kluwer Academic Publishers.

Hapke, M. a. (1996). Fuzzy priority heuristic for project scheduling. Fuzzy Sets Syst. 83, 291-299.

Hartman, S. (1999). Project scheduling under limited Resources: Models, Methods and Applications. Springer ISBN 3540663924, 9783540663928. 221 Páginas.

Harris. (1978). "Precedence and Arrow Networking Techniques for Construction". John Wiley & Sons.

Harris. (1989). "The Packing Method of Resources leveling (PACK)". Journal of Construction Engineering and Management, Fellow, ASCE, 116, 331-350.

Thomas L. Healy. (1961). "Activity subdivision and PERT Probability statements, Institute for Operations Research.

Herrerias, R. (1989). "Modelos probabilísticos alternativos para el método PERT, Aplicación al análisis de inversiones". Estudios de economía aplicada Pág. 89-112. Secretariado de Publicaciones de la Universidad de Valladolid.

Herrerias, R. (1995). "Un nuevo uso de las tres estimaciones subjetivas del PERT". IX Reunión Anual ASEPELT-España. Vol. IV Pág. 411-416.

Herroelen, W. (1999). The multi-mode resource-constrained project scheduling problem with generalized precedence relations. European Journal of Operations Research, 119, 538-556.

- Herroelen, W., & Leus, R. (2005).** "Project scheduling under uncertainty: Survey and research potentials". *European Journal of Operational Research* 165, 289–306.
- Huang, I.-H. (2003).** Comments on "Fuzzy Project Scheduling. Submitted on June 11, 2003 as a course project report for Fuzzy Theory and Applications (EC-591).
- IBM. (1964).** Users Manual for IBM 1440 Project Control System (PCS). IBM.
- JASSBI, J., & Khanmohammadi, S. (2008).** A New Approach for Predicting Project Duration Using Beta Shape Membership Functions and Simulation.
- Kamburowki, J. C. (1981).** The use of fuzzy variables in PERT. *Fuzzy Sets Syst*, 5, 11-19.
- Kamburowski, J. (1996).** "New validations of PERT times". *Omega Int. Journal of Management Science* Vol 25, n° 3 Pág. 323-328.
- Kleim, R. (1999).** Scheduling of Resource-Constrained Projects. Kluwer Academic Publishers.
- Koteswara. (2008).** Optimal selection of PERT for large complex and distributed Projects. *IJCSNS International Journal of Computer Science and Network Security*, VOL.8 No.6, 7-17.
- Kutcha, D. (2001).** Use of fuzzy numbers in Project risk (criticality) assessment. *International Journal of Project Management* 19, 305-310.
- Leachman, R. (1983).** Multiple resource leveling in Construction Systems through Variations Intensities. *Naval Research Logistics Quarterly*". 30 (3), 187-198.
- Leachman, R., Dincerler, A., & Kim, S. (1990).** Resource- Constrained Scheduling of Projects with Variable-Intensity Activities. *IIE Transactions*, Volume 22, Number 1.
- LIN, F.-T., & YAO, J.-S. (2003).** Fuzzy Critical Path Method Based on Signed-Distance Ranking and Statistical Confidence-Interval Estimates. *The Journal of Supercomputing*, 24, 305–325.
- Long, L. D., & Ohsato, A. (2008).** Fuzzy critical chain method for project scheduling under resource constraints and uncertainty. *International Journal of Project Management* 26, 688–698.
- Lootsma. (1989).** Stochastic and fuzzy PERT. *European Journal Operations research*, 43, 174-183.

- Lootsma, F. A. (1997).** Fuzzy logic for planning and decision making. Springer 195 pages.
- Malcolm, C. R. (1965).** Applications of a Technique for Research and development Program Evaluation". Operations Research Vol. 7 pag, 646-670.
- Markus, A., Váncza, J., Kis, T., & Kovács. (2003).** Project scheduling approach for production planning. Annals of the CIRP, 52, 259-362.
- Moder, J. C. (1983).** Project Management with CPM, PERT and Precedence Diagramming. New York: Van Nostrand Reinhold.
- Moghadam, A., & Ghoseiri, J. (2008).** Continuous Fuzzy Longest Path Problem in Project Networks. Journal of Applied Sciences, Volume: 8, Issue: 22, 4061-4069.
- Moitra, S.D. (1990).** "Skinless and the Beta Distribution". Journal of Operations Research Society Vol. 41-10 Pág. 953-961.
- Mon, D.-L., Ching-Hsue, & Lu, H.-C. (1995).** Application of fuzzy distributions on Project management. Fuzzy Sets and Systems, 73, 227-234.
- Mubarak, S. A. (2005).** "Construction Project Scheduling and Control. Upper Saddle River, NJ: Pearson Prentice Hall. ISBN 0130973149.
- Nasution. (1994).** Fuzzy critical path method. IEEE Trans. Syst. Man Cybern. 24(1), 48-57.
- Neuman, K. (1993).** Operations Research. München: Carl Hanser.
- Nicholas, J. M. (2008).** Project Management for Business, Engineering and Technology: Principles and Practice. Oxford: Butterworth-Heinemann. ISBN 9780750683999.
- Philips, J. (2010).** PMP Project management Professional Study Guide. McGraw-Hill/Osborne. ISBN 9780071626736.
- Plotnick, F. L. (2005).** CPM in Construction Management". 6th Ed. New York: McGraw-Hill. ISBN 0071457690.
- Plotnick, F. L. (2010).** CPM in Construction Management". 7th Ed. New York: McGraw-Hill. ISBN 9780071636643.
- Ponz, J. L. (2009).** Project management con Redes PERT. ISBN: 978-84-8363-345-8. Editorial de la Universidad Politécnica de Valencia. 278 páginas.
- Prisker, A., & L.J. Watters, a. P. (1969).** Multiproject Scheduling with Limited Resources: A Zero-One Programming Approach. Management Science 16, 95-108.

Project Management Institute. (2003). Construction Extension to a Guide for a Project Management body of Knowledge. Newtown Square, Pa: Project management Institute. ISBN 1930699409.

Project Management Institute. (2008). PMBOK. Project Management Institute.

Rambaud, S. C., & Perez, J. G. (2005). The expert confidence versus fuzzy probability in the mean activity times computation. *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems*. Vol. 13, no. 4, 437-452.

Ravindran, A. (2008). Operations Research and Management Science Handbook. Boca Raton. CRC. Taylor & Francis. ISBN 0849397219.

Ravindran, A. (2009). Operations Research Applications. Boca Raton. CRC. Taylor & Francis. ISBN 9781420091861.

Roy, B. (1962). Graphes et ordonnancements. *Revue Francaise de Recherche Operatinelle*, 323-333.

Roy, B. (1959). Théorie des graphes: Contribution de la théorie des graphes à l'étude de certains problèmes linéaires. *Comptes rendus des Séances de L'Académie des Sciences*, séance du Avril, 2437-2449.

Sasieni M. W. (1968). "Fundamentals of Operations Research", John Wiley & Sons.

Sasieni. (1986). "A Note PERT Times". *Management Science* 32, Pág. 1652-1653.

Slyeptsov, A. I., & Tyshchuk, T. A. (2003). Fuzzy temporal characteristics of operations for project management on the network models basis. *European Journal of Operational Research* 147, 253-265.

Stanford Research Institute. (1949). An Improved Rational and Mathematical Explanation of the Progress Curve in Airframe Production. Stanford, California, August, 5.

Trofin, J. (2005). Impact of uncertainty on construction project performance using linear scheduling. University of Florida: A thesis presented to the graduate school of the University of Florida in partial fulfillment of the requirements for the degree of Master of Science.

Tulio, T., Urgo, M., & Alferi, A. (2008). Project Scheduling with Precedence Relations: An Application to Production Planning. 9th Biennial ASME Conference on Engineering System Design and Analysis. Esda 2008. Haifa, Israel.

- Valls, V., & Lino, P. (2001).** Criticality analysis in Activity-on-Node Networks with Minimal Time Lags. *Annals of Operations research* 102, 17-37.
- Walkenbach, John. (2010).** "Excel 2010. Programación VBA". Anaya Multimedia. ISBN 978-84-415-2828-4.
- Walker, A. (2007).** Project Management in Construction. Oxford UK: Malden, MA USA: Blackwell. ISBN 1405158247.
- Wang, J. (2002).** A fuzzy project scheduling approach to minimize schedule risk for product development. *Fuzzy Sets and Systems* 127, 99-116.
- Wiest, J. a. (1977).** A Management Guide to PERT/PDM/DCPM and other Networks. Englewood Cliffs, NJ: Prentice-Hall.
- Fred E. Williams.** "PERT Completion Times Revisited". *Transactions on Education* 6:1(21-34).
- Winston, W. I. (2001).** Practical Management Science. Pacific Grove, CA Duxbury, cop.
- Yakhchali, S. H., & Ghodsypour, S. H. (2009).** Computing latest starting times of activities in interval-valued networks with minimal time lags. *European Journal of Operational Research*, doi:10.1016/j.ejor.2009.01.051.
- Zadeh, L. (1965).** Fuzzy Sets. *Information and Control*, 8, 338-353.
- Zaderenko, S. G. (1968).** Sistemas de Programación por camino crítico: PERT-CPM-MAN SCHEDULING-RAMPS y otros métodos de elaboración y control de programas. Buenos Aires. Argentina: 001.424 Z16. Librería Mitre. 197 páginas.
- Zammori, F. A., Braglia, M., & Frosolini, M. (2009).** A fuzzy multi-criteria approach for critical path definition. *International Journal of Project Management* 27, 278-291.
- Zhang, H., Tam, C., & Li, H. (2005).** Modeling uncertain activity duration by fuzzy number and discrete-event simulation. *European Journal of Operational Research* 164, 715-729.
- Zielinski, P. (2003).** Latest starting times and floats of activities in networks with uncertain durations. *European Journal of Operations Research*, Volume 147, 266-280.

