



Universidad de Valladolid



ESCUELA DE INGENIERÍAS
INDUSTRIALES

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTRIALES

Grado en Ingeniería en Organización Industrial

Programación de proyectos mediante
cadena crítica. Dimensionamiento de
buffers con simulación de Montecarlo.

Autor:

Hortigüela Arozamena, Diego

Tutor:

Poza García, David Jesús

Valladolid, junio 2022.



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**



RESUMEN

La metodología CCPM (*Critical Chain Project Management*) aborda la incertidumbre asociada a la duración de las actividades del proyecto eliminando la protección implícita de cada actividad al mismo tiempo que añade *buffers* de protección que absorben los posibles retrasos que pueden sufrir las actividades del proyecto durante su ejecución. A pesar de que existen múltiples métodos para el dimensionamiento de estos *buffers*, los métodos tradicionales presentan ciertas limitaciones cuando se aplican en entornos de incertidumbre. En este contexto, en este Trabajo Fin de Grado (TFG) se comparan los métodos de dimensionamiento tradicionales con las técnicas más recientes, basadas en simulación de Montecarlo, y se ilustra el proceso de dimensionamiento de los *buffers* con ambos tipos de técnicas. Como resultado, ilustramos que mediante simulación de Montecarlo podemos obtener unos tamaños de *buffers* más próximos a lo esperable según la incertidumbre real asociada a la naturaleza aleatoria de cada actividad del proyecto.

PALABRAS CLAVE

Cadena Crítica, CCPM, simulación de Montecarlo, *buffer*, programación de proyectos.



ABSTRACT

The Critical Chain Project Management (CCPM) methodology addresses the uncertainty associated with the duration of project activities by eliminating the implicit protection of each activity while adding protection *buffers* that absorb the possible delays that project activities may suffer during their execution. Although there are multiple methods for sizing *buffers*, traditional methods present certain limitations when applied in uncertainty environments. In this context, in this Final Degree Project (TFG) we compare traditional sizing methods with the most recent techniques, based on Monte Carlo simulation, and illustrate the *buffer* sizing process with both types of techniques. As a result, we illustrate that by means of Monte Carlo simulation we can obtain *buffer* sizes closer to what can be expected according to the real uncertainty associated to the random nature of each project activity.

KEYWORDS

Critical Chain, CCPM, Montecarlo simulation, *buffer*, project scheduling.



ÍNDICE DE CONTENIDO

RESUMEN	3
PALABRAS CLAVE.....	3
ÍNDICE DE CONTENIDO.....	5
ÍNDICE DE ILUSTRACIONES	8
ÍNDICE DE TABLAS.....	10
I. INTRODUCCIÓN Y OBJETIVOS	13
1. ALCANCE DEL TRABAJO	13
2. OBJETIVO DEL TRABAJO	14
3. ESTRUCTURA DEL TRABAJO.....	14
II. DIRECCIÓN DE PROYECTOS.....	17
1. RESTRICCIONES	18
2. METODOLOGÍAS Y ESTÁNDARES PARA LA DIRECCIÓN DE PROYECTOS.....	20
2.1 <i>La Guía PMBOK</i>	20
2.2 <i>ICB</i>	21
2.3 <i>ISO21500</i>	22
2.4 <i>PM²</i>	22
2.5 <i>APM</i>	22
III. PROGRAMACIÓN DE PROYECTOS	25
1. MÉTODOS DETERMINÍSTICOS: GANTT, CPM Y ROY.	28
1.1 <i>GANTT</i>	29
1.2 <i>CPM</i>	30
1.3 <i>ROY</i>	32
2. MÉTODOS ESTOCÁSTICOS: PERT Y SIMULACIÓN DE MONTECARLO.	34
2.1 <i>PERT</i>	34
2.2 <i>Simulación de Montecarlo</i>	37
2.2.1 <i>Ejemplo</i>	38
IV. MÉTODO CCPM	45
1. BASES:	46
1.1 <i>Teoría de las restricciones</i>	46
1.1.1 <i>Método DBR (Drum Buffer Rope)</i>	46
a. <i>Drum (Tambor)</i>	46
b. <i>Buffer (Amortiguador)</i>	47
c. <i>Rope (Cuerda)</i>	47



2.	ORIGEN DE LOS RETRASOS.....	49
2.1	<i>Síndrome del estudiante</i>	49
2.2	<i>Ley de Parkinson</i>	50
2.3	<i>Ley de Murphy</i>	51
3.	<i>BUFFERS</i>	52
3.1	<i>Tipos de buffers</i>	54
3.1.1	<i>Buffers de proyecto</i>	54
3.1.2	<i>Buffers de alimentación</i>	56
3.1.3	<i>Buffers de recurso</i>	57
3.1.4	<i>Combinación de buffers de proyecto, alimentación, y recurso</i>	58
3.2	<i>Métodos de dimensionado de buffer de proyecto</i>	59
3.2.1	<i>La regla del 50% (Cut & Paste Method)</i>	60
3.2.2	<i>Método del tercio crítico</i>	62
3.2.3	<i>Método de la raíz cuadrada de la suma de cuadrados (SSQ)</i> ..	63
3.3	<i>Métodos de dimensionado de buffer de alimentación</i>	65
V. CASO PRÁCTICO: DIMENSIONAMIENTO DE <i>BUFFERS</i> UTILIZANDO SIMULACIÓN DE MONTECARLO FRENTE A LOS MÉTODOS TRADICIONALES .. 69		
1.	DEFINICIÓN DEL PROYECTO	69
2.	APLICACIÓN DE LA METODOLOGÍA DE LA CADENA CRÍTICA.....	75
3.	ESCENARIO 1: DIMENSIONAMIENTO DE LOS <i>BUFFERS</i> SEGÚN LOS MÉTODOS TRADICIONALES.	76
3.1	<i>Regla del 50%</i>	76
3.2	<i>Método del tercio crítico</i>	76
3.3	<i>Método de la raíz cuadrada de la suma de cuadrados (SSQ)</i>	76
3.4	<i>Cálculo de buffers</i>	77
A.	<i>Buffer de proyecto</i>	77
1)	<i>Regla del 50%</i>	77
2)	<i>Método del tercio crítico</i>	78
3)	<i>Método de la raíz cuadrada de la suma de cuadrados (SSQ)</i> ..	78
B.	<i>Buffer de alimentación 1</i>	79
1)	<i>Regla del 50%</i>	79
2)	<i>Método del tercio crítico</i>	80
3)	<i>Método de la raíz cuadrada de la suma de cuadrados (SSQ)</i> ..	80
C.	<i>Buffer de alimentación 2</i>	81
1)	<i>Regla del 50%</i>	81
2)	<i>Método del tercio crítico</i>	82
3)	<i>Método de la raíz cuadrada de la suma de cuadrados (SSQ)</i> ..	82
4.	ESCENARIO 2: DIMENSIONAMIENTO DE <i>BUFFERS</i> CON SIMULACIÓN DE MONTECARLO.	83
A.	<i>Buffer de proyecto</i>	88



B. <i>Buffer de alimentación 1</i>	90
C. <i>Buffer de alimentación 2</i>	93
5. ANÁLISIS DE LOS RESULTADOS.....	96
VI. CONCLUSIONES	99
VII. REFERENCIAS.....	101



ÍNDICE DE ILUSTRACIONES

Ilustración 1: Triángulo de la Calidad. Fuente: Elaboración propia.....	18
Ilustración 2: Triángulo de la Calidad ampliado. Fuente: Elaboración propia. 19	
Ilustración 3: Grupos de procesos. Fuente: (PMI, 2017)	21
Ilustración 4: Desarrollo Scrum. Fuente: (Staff HDC, 2021).....	24
Ilustración 5: Dependencia Inicio-Inicio. Fuente: Elaboración propia.	26
Ilustración 6: Dependencia Inicio-Fin. Fuente: Elaboración propia.	26
Ilustración 7: Dependencia Fin-Inicio. Fuente: Elaboración propia.	26
Ilustración 8: Dependencia Fin-Fin. Fuente: Elaboración propia.....	27
Ilustración 9: Actividades con holgura. Fuente: Elaboración propia.	28
Ilustración 10: Camino crítico del ejemplo. Fuente: Elaboración propia.	30
Ilustración 11: Notación en CPM. Fuente: (Posadas, 2018).	31
Ilustración 12: Grafo usando CPM. Fuente: (Poza, 2021).	32
Ilustración 13: Notación ROY. Fuente: (Marcos, 2021)	33
Ilustración 14: Grafo usando ROY. Fuente: (Poza, 2021).....	33
Ilustración 15 : Estimaciones para PERT. Fuente: (Josafat Gascón Busio).	35
Ilustración 16: Duración más probable PERT. Fuente: (Josafat Gascón Busio).	36
Ilustración 17: Grafo de las actividades del ejemplo. Fuente: (Poza, 2021). .	39
Ilustración 18: Método DBR. Fuente: (Roser, 2014).....	48
Ilustración 19: Síndrome del estudiante. Fuente: (Balyuk A. , s.f.)	50
Ilustración 20: Ley de Parkinson. Fuente: (Strom, 2022).....	51
Ilustración 21: Forma tradicional VS. CCPM. Fuente: Elaboración propia.....	53
Ilustración 22: Buffer de proyecto comparado con la forma tradicional. Fuente: Elaboración propia.	55
Ilustración 23: Cadenas paralelas a la crítica. Fuente: Elaboración propia. ...	56
Ilustración 24: Buffers de alimentación. Fuente: Elaboración propia.....	57
Ilustración 25: Buffers de recurso. Fuente: Elaboración propia.....	58
Ilustración 26: Combinación de buffers. Fuente: Elaboración propia.....	59
Ilustración 27: Regla del 50%. Fuente: Elaboración propia.....	61
Ilustración 28: Método del tercio crítico. Fuente: Elaboración propia.	63
Ilustración 29: Método de la raíz cuadrada de la suma de cuadrados. Fuente: Elaboración propia.	64
Ilustración 30: Cronograma con buffers incluidos. Fuente: Elaboración propia.	65
Ilustración 31: Grafo del proyecto. Fuente: Elaboración propia.	70
Ilustración 32: Percentiles 50, y 90. Fuente: (Vanhoucke, 2012)	71
Ilustración 33: Grafo del proyecto programando actividades sin las protecciones individuales (percentil 50). Fuente: Elaboración propia.....	73



Ilustración 34: Grafo del proyecto programando actividades con las protecciones individuales (percentil 90). Fuente: Elaboración propia.....	74
Ilustración 35: Funciones de distribución en Crystal Ball. Fuente: Crystal Ball.	85
Ilustración 36: Definición de la duración del camino crítico en Crystal Ball. Fuente: Crystal Ball.	86
Ilustración 37: Definición de la duración de la cadena no crítica B-E. Fuente: Crystal Ball.....	87
Ilustración 38: Definición de la duración de la cadena no crítica D. Fuente: Crystal Ball.....	87
Ilustración 39: Resultados de la simulación del camino crítico. Fuente: Elaboración propia mediante el software Crystal Ball.....	88
Ilustración 40: Resultados de la simulación de la cadena no-crítica B-E. Fuente: Elaboración propia mediante el software Crystal Ball.....	91
Ilustración 41: Resultados de la simulación del camino no crítico D. Fuente: Elaboración propia mediante el software Crystal Ball.....	93



ÍNDICE DE TABLAS

Tabla 1: Duraciones de las actividades del ejemplo. Fuente: Elaboración propia.	39
Tabla 2: Simulación de 10 iteraciones. Fuente: Elaboración propia.....	41
Tabla 3: Índice de criticidad. Fuente: Elaboración propia.....	42
Tabla 4: Duración media y Varianza del proyecto. Fuente: Elaboración propia.	43
Tabla 5: Actividades y sus precedentes. Fuente: Elaboración propia.....	70
Tabla 6: Duraciones de las actividades. Fuente: Elaboración propia.....	71
Tabla 7: Protecciones individuales. Fuente: Elaboración propia.....	72
Tabla 8: Regla del 50% al buffer de proyecto. Fuente: Elaboración propia.....	77
Tabla 9: Método del tercio crítico al buffer de proyecto. Fuente: Elaboración propia.....	78
Tabla 10: Método de la raíz cuadrada de la suma de cuadrados (SSQ) al buffer de proyecto. Fuente: Elaboración propia.....	78
Tabla 11: Regla del 50% al buffer de alimentación 1. Fuente: Elaboración propia.....	79
Tabla 12: Método del tercio crítico al buffer de alimentación 1. Fuente: Elaboración propia.....	80
Tabla 13: Método SSQ al buffer de alimentación 1. Fuente: Elaboración propia.....	80
Tabla 14: Regla del 50% al buffer de alimentación 2. Fuente: Elaboración propia.....	81
Tabla 15: Método del tercio crítico al buffer de alimentación 2. Fuente: Elaboración propia.....	82
Tabla 16: Método SSQ al buffer de alimentación 2. Fuente: Elaboración propia.....	82
Tabla 17: Resumen de buffers en escenario 1. Fuente: Elaboración propia.	83
Tabla 18: Estadísticos de la simulación del camino crítico. Fuente: Elaboración propia mediante el software Crystal Ball.....	89
Tabla 19: Percentiles de la simulación del camino crítico. Fuente: Elaboración propia mediante el software Crystal Ball.....	90
Tabla 20: Estadísticos de la simulación del camino no crítico B-E. Fuente: Elaboración propia mediante el software Crystal Ball.....	92
Tabla 21: Percentiles de la simulación del camino no crítico B-E. Fuente: Elaboración propia mediante el software Crystal Ball.....	92
Tabla 22: Estadísticos de la simulación del camino no crítico D. Fuente: Elaboración propia mediante el software Crystal Ball.....	94



Tabla 23: Percentiles de la simulación del camino no crítico D. Fuente: Elaboración propia mediante el software Crystal Ball.....	95
Tabla 24: Resumen de buffers en escenario 2. Fuente: Elaboración propia. .	95
Tabla 25: Comparación entre escenarios 1 y 2. Fuente: Elaboración propia..	96
Tabla 26: Resumen del tamaño de los buffers en ambos escenarios. Fuente: Elaboración propia.	97



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**



I. INTRODUCCIÓN Y OBJETIVOS

1. Alcance del trabajo

La Dirección de Proyectos se encarga de lograr los objetivos del proyecto utilizando metodologías y mejores prácticas para hacerlo dentro de los parámetros establecidos. Consta de dos partes, la primera, planificación y programación del proyecto, donde se identifican los objetivos, entregables, y actividades necesarias para conseguirlos, y se realiza un cronograma en el que se incluye el calendario de ejecución de dichas actividades, y, por otro lado, control y seguimiento del proyecto para asegurar el cumplimiento del plan diseñado para lograr los objetivos.

La metodología CCPM (*Critical Chain Project Management*), introducida en 1997 por Eliyahu M. Goldratt, aborda la incertidumbre asociada a la duración de las actividades del proyecto eliminando la protección implícita de cada actividad, al mismo tiempo que añade *buffers* (amortiguadores) de protección que absorben los posibles retrasos que pueden sufrir las actividades del proyecto durante su ejecución. El objetivo es conseguir que el proyecto en conjunto cumpla con la fecha de finalización, preocupando menos el cumplimiento de la fecha de finalización de cada actividad de manera individual, y centrándose en el proyecto de manera global.

La incertidumbre en la duración de las actividades es la causante de la mayoría de los retrasos de los proyectos. Goldratt identificó que el posible origen de estos retrasos tiene lugar en la Ley de Parkinson (si el operario sabe que tiene más tiempo para hacer una actividad que el mínimo necesario, la ejecución de esa actividad se va a alargar hasta cubrir todo el tiempo disponible), Síndrome del estudiante (la ejecución de la actividad se retrasa hasta aproximarse la fecha límite), multitarea de los trabajadores, etc. Para asegurar la fecha de finalización del proyecto, la metodología CCPM plantea planificar una duración de desarrollo de las actividades más justa, eliminando la protección implícita agregada a cada actividad individualmente, para mitigar los efectos de las mencionadas Ley de Parkinson o Síndrome del Estudiante, al mismo tiempo que añade protecciones en forma de *buffer* en puntos específicos del proyecto, de manera que absorban los posibles retrasos en la ejecución de las actividades. De esta manera, se consigue aumentar las probabilidades de cumplir con la fecha de finalización del proyecto.

El dimensionamiento de estos *buffers* ha supuesto el mayor desafío de la metodología CCPM desde su introducción, ya que su autor, Goldratt, no dejó



indicaciones concretas al respecto. A pesar de que a lo largo de los años se han propuesto múltiples métodos para el dimensionamiento de los *buffers* (entre los que destacamos la regla del 50%, el método del tercio crítico, o el método de la suma de los cuadrados, SSQ), los métodos tradicionales presentan ciertas limitaciones cuando se aplican en entornos de incertidumbre.

En este contexto, en este Trabajo Fin de Grado (TFG) se comparan los métodos de dimensionamiento tradicionales con las técnicas más recientes, basadas en simulación de Montecarlo, y se ilustra el proceso de dimensionamiento de los *buffers* con ambos tipos de técnicas. Mostraremos que cuando se trata de dimensionar estos *buffers* en entornos de incertidumbre, los métodos tradicionales no tienen en cuenta toda la incertidumbre de las actividades, pues la reducen a un simple parámetro fijo, que es la única característica de la actividad tenida en cuenta para el dimensionamiento de los *buffers*. Asimismo, mostraremos que el dimensionamiento de *buffers* mediante simulación de Montecarlo permite tener en cuenta toda la incertidumbre asociada a la naturaleza aleatoria de cada actividad, generando duraciones aleatorias consecuentes con las funciones de distribución que modelan la duración de cada actividad. Como resultado, ilustramos que mediante simulación de Montecarlo podemos obtener unos tamaños de *buffers* más próximos a lo esperable según la incertidumbre real asociada a la naturaleza aleatoria de cada actividad del proyecto.

2. Objetivo del trabajo

El objetivo de este TFG es, por tanto, ilustrar cómo los *buffers* pueden dimensionarse usando simulación de Montecarlo, y comparar los resultados de esta técnica con los obtenidos mediante los métodos tradicionales, destacando las ventajas de emplear simulación de Montecarlo para este propósito.

3. Estructura del trabajo

El resto del informe se estructurará como sigue, en el siguiente capítulo, el II, se introduce la Dirección de Proyectos, incluyendo las restricciones a las que están sujetas, y las distintas metodologías para afrontarlas. A continuación, el capítulo III, se centra en la programación de proyectos, una subdisciplina de la Dirección de Proyectos cuyo objetivo es la gestión del tiempo del proyecto. En este capítulo se describen distintos métodos para su gestión. Posteriormente,



en el capítulo IV, se explica detalladamente la metodología CCPM, incluyendo las bases y los orígenes de los retrasos que justifican la utilización de esta metodología, y también se exponen los tipos de *buffers* y métodos de dimensionamiento de estos. A continuación, en el capítulo V, se desarrolla un caso práctico donde se aplica la metodología CCPM en entornos de incertidumbre, comparando el uso de las técnicas tradicionales para el dimensionamiento de *buffers* con la simulación de Montecarlo. Por último, en el capítulo VI, se exponen las conclusiones de la investigación y se plantean posibles extensiones de este trabajo.



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**



II. DIRECCIÓN DE PROYECTOS

Un proyecto, según uno de los estándares más utilizados en la Dirección de Proyectos, la Guía PMBOK, es definido como “un esfuerzo temporal, único, multidisciplinar y organizado, que se lleva a cabo en entregables sujetos a requisitos y restricciones predefinidas” (PMI, 2017).

Los proyectos surgen a diario con el objetivo de satisfacer una necesidad, o solucionar un problema, y para cumplir con los objetivos de estos se necesita una dirección apropiada y una gestión consecuente.

Según el PMBOK, la Dirección de Proyectos es la aplicación del conjunto de técnicas, mejores prácticas y metodologías para lograr los objetivos del proyecto dentro de los parámetros establecidos (PMI, 2017). Es una herramienta utilizada para planificar, ejecutar y controlar proyectos de manera más efectiva y eficiente, aunque únicamente su uso no implica el éxito ya que existen muchos otros factores involucrados. Una investigación realizada por Roberts y Furlonger revela que el 85% - 90% de los proyectos no cumplen con el plazo planificado, terminan dentro del presupuesto o cumplen con los requisitos de calidad esperados, y que utilizando una metodología de dirección de proyectos la productividad aumenta en un 20% - 30%.

Por otro lado, a lo largo de las últimas décadas han ido surgiendo, dentro de la gestión de proyectos, metodologías para encontrar la solución a los problemas que aparezcan y alejen al proyecto de conseguir su objetivo.

La Dirección de Proyectos consta de una parte proactiva (planificación del proyecto), que tiene como objetivo dirigir y encaminar el proyecto hacia el cumplimiento de los objetivos, y de una parte reactiva (la monitorización y control de la ejecución del proyecto) pues se encarga de hacer frente a los problemas que surgen en el camino de conseguir dichos objetivos.

Para asegurar el éxito del proyecto es importante centrarse en una dirección acorde a los objetivos, garantizando su correcta gestión durante el desarrollo de este, es decir, a lo largo del ciclo de vida del proyecto, cuyas fases serán detalladas más adelante (sección III.3.1), lo cual permitirá a sus gestores:

- Mantener una organización acorde con la estrategia
- Monitorizar el progreso
- Mejor precisión en la planificación teniendo en cuenta posibles riesgos

1. Restricciones

La realización de todo proyecto está sujeto a restricciones (o limitaciones) que son independientes del mismo o de su tamaño, y que tienen un impacto en su desarrollo, siendo necesario estudiarlas con antelación para poder llegar a lograr los objetivos del proyecto.

Según la Guía PMBOK, las restricciones son las siguientes: “El estado, la calidad, o la sensación de ser restringido a un curso de acción o inacción determinado. Una limitación, ya sea interna o externa, afectará al desempeño del proyecto o proceso”. (PMI, 2017)

Estas limitaciones vienen recogidas en el Triángulo de Calidad (Ilustración 1) de un proyecto, donde se recoge la triple restricción en un triángulo equilátero, mostrando el equilibrio que debe existir entre las restricciones para garantizar el éxito del proyecto. Las tres variables son Costo – Tiempo – Alcance, y cualquier modificación de alguna de ellas, causará un impacto en el resto.



Ilustración 1: Triángulo de la Calidad. Fuente: Elaboración propia.

- 1- Tiempo – es un factor incontrolable, y es la razón por la que muchos proyectos fracasan. El tiempo que tarde la ejecución del proyecto y por tanto de sus actividades, dependerá de factores como el número de personas trabajando, su experiencia, sus habilidades etc. (iSolution). Eso se verá reflejado en la realización del cronograma (*Project scheduling*), que por tanto recoge la **programación** del proyecto.
- 2- Costo – se recoge en el presupuesto del proyecto, y limita la cantidad de recursos a utilizar (iSolution).

Si los plazos se reducen, tendrá un impacto directo en el coste del proyecto, ya que se tendrá que aumentar la cantidad de recursos destinados a realizar las tareas para poder finalizar a tiempo, incurriendo en costos adicionales (Team Asana, 2021).

- 3- Alcance – analiza el “tamaño” del proyecto, se encarga de elaborar una lista con los entregables del proyecto (iSolution), ósea recoge la **planificación** del proyecto. Cuanto mayor sea el tamaño del proyecto, mayor será el costo y el tiempo necesitado para finalizarlo (Team Asana, 2021).

La calidad, que no aparece en los vértices del triángulo, pero le da nombre, es el objetivo del proyecto, el resultado de equilibrar las tres limitaciones del triángulo.

En la actualidad, se han incluido otras variables al Triángulo de la Calidad (Ilustración 2), aparte del Costo – Tiempo – Alcance, y Calidad, se ha unido la satisfacción de los interesados o *stakeholders* del proyecto. Los *stakeholders* de un proyecto son todas aquellas personas, grupos u organizaciones que puedan afectar o ser afectados por el proyecto (Poza, 2020).



Ilustración 2: Triángulo de la Calidad ampliado. Fuente: Elaboración propia.

Por tanto, la **planificación** del proyecto recoge los objetivos del proyecto en función de su tamaño, así como las tareas requeridas para conseguirlo. Y es completada con la **programación** del proyecto, que se encarga del cronograma, es decir, de estimar la duración de las actividades y establecer una fecha de inicio y de fin del proyecto.



2. Metodologías y estándares para la Dirección de Proyectos

La gestión de proyectos permite planificar, programar y controlar las actividades a realizar durante el desarrollo del proyecto para conseguir los objetivos que se han planificado.

Hoy en día, los proyectos son más grandes y complejos, suponiendo un mayor desafío para el director del proyecto, que tiene que dar solución a problemas cada vez más difíciles para asegurar su éxito. Es por ello, que las metodologías de gestión de proyectos han ido evolucionando. A continuación, se hace una descripción general de algunas de las metodologías y estándares para la Dirección de Proyectos más extendidas.

2.1 La Guía PMBOK

La Guía PMBOK (*Project Management Body of Knowledge*), con su primera versión en 1990, es considerada una de las herramientas fundamentales en la gestión y dirección de proyectos basada en procesos (Wrike, 2021).

Fue desarrollada por la asociación profesional PMI (*Project Management Institute*), una organización sin ánimo de lucro, originada en Estados Unidos en 1969, diseñada para gestores de proyectos.

Es importante destacar que no se trata de una normativa, sino de un conjunto de buenas prácticas para la interpretación de proyectos, que se basa en identificar los procesos y áreas de conocimiento que integrarían su gestión. Esta guía permite distinguir 49 procesos repartidos en 10 áreas de conocimiento, y a su vez en 5 grupos de procesos interrelacionados y detallados con entradas, herramientas y salidas, que facilitan su ejecución.

Los procesos en la Dirección de Proyectos son las acciones y actividades que se realizan para alcanzar unos resultados previamente especificados (Sanz, 2017). El PMBOK clasifica los proyectos en 5 grupos de procesos (Roseke, 2017):

1. Procesos de Inicio
2. Procesos de planificación
3. Procesos de ejecución
4. Procesos de monitoreo y control
5. Procesos de cierre



Ilustración 3: Grupos de procesos. Fuente: (PMI, 2017)

2.2 ICB

La ICB (*IPMA Competence Baseline*) promueve el desarrollo de la profesión a través de competencias, proponiendo 29 de ellas. Las competencias sirven para describir los elementos fundamentales en la Gerencia de Proyectos (Sanz, 2017). Estas 29 competencias, están reflejadas en 3 áreas de competencia: perspectiva, personas y práctica, y quedan reflejadas en este estándar IPMA-ICB, que persigue estandarizar y reducir las tareas básicas necesarias para finalizar un proyecto de manera efectiva y eficiente (*IPMA Competence Baseline*).

La IPMA (*International Project Management Association*), es una asociación profesional internacional, creada en Austria, aunque establecida en Suiza, con más de 70 asociaciones nacionales (Haughey, 2019). Se dedica al desarrollo y promoción de la dirección de proyectos mediante la expedición de la Certificación IPMA en Project Management, siendo unas de las más reconocidas a nivel europeo en el ámbito de Dirección y Gestión de Proyectos, además fue la que desarrollo el estándar ICB.



2.3 ISO21500

Esta norma pretende dar una orientación a las empresas en su gestión, es por eso por lo que no posee una certificación ligada a ella, ya que no son requisitos sino pautas. Al igual que la Guía PMBOK, la ISO 21.500 también aborda la Dirección de Proyectos mediante procesos (Excellence, 2017).

Recoge las indicaciones sobre Dirección y Gestión de Proyectos que aconsejan como gestionar y ordenar las tareas de la organización, en relación con los proyectos, procesos, tiempos o riesgos. Además, está diseñada para permitir su aplicación a cualquier tipo de organización (pública privada, o comunitaria), y todo tipo de proyectos independientemente de la complejidad, tamaño, o duración.

2.4 PM²

Es una metodología de gestión de proyectos desarrollada por la Comisión Europea, desarrollada para cubrir las necesidades, limitaciones y cultura de las instituciones de la UE. Además, reúne elementos de referentes como el PMBOK, IPMA-ICB, Prince2, o Agile, cuyo objetivo es gestionar de manera efectiva a los equipos de proyecto para que desarrollen soluciones a los problemas en beneficio de la organización y cualquier *stakeholder* (Project Manager Soy, 2021).

PM² es *Open Source*, lo que significa que está disponible para todo tipo de proyectos, compañías, y profesionales, ya que permite la reutilización de documentos, disponiendo de plantillas personalizables para todo el público. Citando alguna de sus características, introduce la jerarquía del proyecto, con roles y responsabilidades, directrices de procesos, y guías y plantillas (Cáceres, 2021).

2.5 APM

Agile Project Management, recoge un conjunto de metodologías caracterizadas por fragmentar los proyectos en partes de tamaño reducido capaces de afrontarse y resolverse en poco tiempo. La idea es que uno debe emplear metodologías ágiles cuando los requisitos del resultado del proyecto no están claros en las primeras fases, por lo que se hacen una serie de iteraciones en el



desarrollo del producto final, en cada una de las cuales se revisan los requisitos del entregable final.

De entre todas las metodologías ágiles, destaca Scrum. Esta metodología promueve la gestión de proyectos dividiendo a los procesos en ciclos de menor tamaño a los que se conoce *sprints* o iteraciones, creando una gran ventaja para hacer frente a la solución de problemas durante el desarrollo del proyecto. Cada iteración debe aportar un entregable o resultado susceptible de ser entregado al cliente (Gillis).

El cronograma es dividido en sesiones de trabajo reducidas, los mencionados *sprints*, lo que permite reconocer si han sido exitosos o no, y poder corregir errores de inmediato (Gillis).

Se proponen tres roles principales: *Scrum master*, *Product owner*, y *Equipo de desarrollo* (Gómez, 2021).

1) *Scrum master*.

Encargado de asegurarse que se comprenden y cumplen las técnicas scrum, y del buen transcurso del proyecto (Staff HDC, 2021).

2) *Product owner (PO)*.

Es el encargado de representar al cliente en el equipo, y se hace cargo de la responsabilidad del resultado o producto obtenido. Hace todas las comunicaciones con el cliente y se encarga de transmitir sus necesidades al equipo mediante la creación de un listado de requisitos denominado *product backlog*.

3) *Equipo de desarrollo*.

Del inglés, *Development Team (DT)*, es el encargado de desarrollar el producto que cumpla con los requisitos definidos por el *product owner* en el *product backlog*. Además, estos equipos son autosuficientes y no tienen delegaciones para que la responsabilidad del *sprint* sea compartida (Staff HDC, 2021). Para cada *sprint*, el DT realiza la división de tareas a ejecutar basado en el *product backlog*, y este conjunto de tareas se incluyen en el *sprint backlog*.

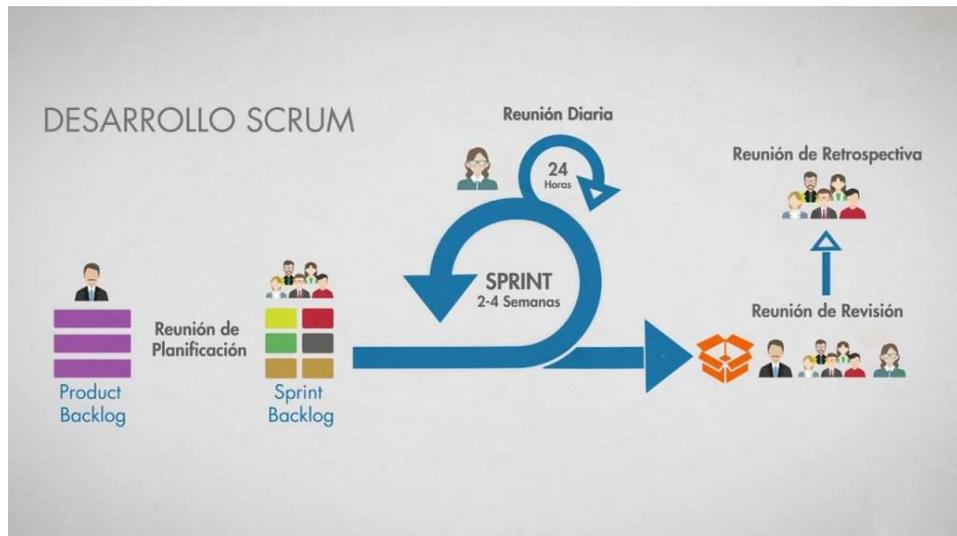


Ilustración 4: Desarrollo Scrum. Fuente: (Staff HDC, 2021)



III. PROGRAMACIÓN DE PROYECTOS

La planificación del proyecto recoge los objetivos del proyecto en función de su tamaño, así como las actividades requeridas para conseguirlo. Y es completada con la programación del proyecto, que se encarga del desarrollo del cronograma, es decir, de estimar la duración de esas actividades, el análisis de sus relaciones de precedencia y de la disponibilidad de los recursos necesarios para su ejecución para, con ello, establecer una fecha de inicio y de fin para cada una de las actividades del proyecto (Biblioteca de SAP Business, 2021).

La programación del proyecto culmina con la obtención del cronograma del proyecto, en el cual se especifica cuál debe ser la fecha de comienzo y fin de cada una de las actividades cumpliendo con las restricciones anteriores. Este cronograma, que servirá como referencia para medir el grado de avance del proyecto, recibe el nombre de línea base del cronograma (Poza, 2020).

El primer paso es estimar la duración de las actividades, que consiste en establecer la cantidad de periodos de trabajo necesarios para finalizar cada actividad con los recursos estimados. Existen dos tipos de actividades:

- Actividades de duración fija – su duración es independiente del número de recursos asignados.
- Actividades de duración variable – su duración puede disminuir al incrementar el número de recursos asignados.

Además, para poder saber la duración del proyecto total, es decir el número de días que pasan desde la fecha de inicio hasta la fecha de fin, es necesario definir las relaciones de precedencia entre las actividades, es decir una secuencia, lo que nos lleva al segundo paso (Neira, 2019).

En el segundo paso se detalla el orden que deben seguir las actividades del proyecto, esto es porque habrá ciertas actividades que necesiten esperar a la realización de una previa ya que necesitan el resultado de esa actividad para completarse. A continuación, se muestran los 4 tipos de dependencia que pueden darse entre dos actividades (Posadas, 2018):

- **Inicio-inicio** (Edraw, 2022) → la sucesora no comienza hasta que lo hace la predecesora, lo hacen de manera conjunta casi simultánea.
 - Diseño de la base de datos – Diseño pantalla. Ilustración 5.

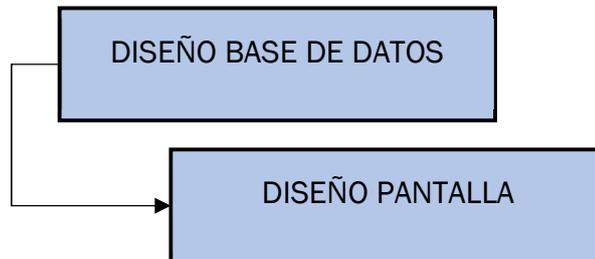


Ilustración 5: Dependencia Inicio-Inicio. Fuente: Elaboración propia.

- **Inicio-fin** → la sucesora no termina hasta que no comienza la predecesora
 - Casa nueva lista – Mudarse casa nueva (Desalojar casa antigua). Ilustración 6.



Ilustración 6: Dependencia Inicio-Fin. Fuente: Elaboración propia.

- **Fin-inicio** → la sucesora no comienza hasta que no termine la predecesora. Es el tipo más frecuente de relación de dependencia entre dos actividades. Ilustración 7.
 - Comprar lienzo – Pintar cuadro

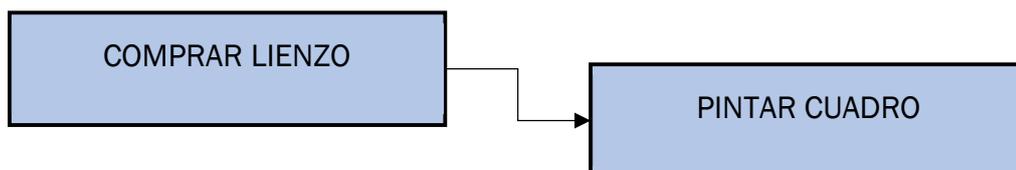


Ilustración 7: Dependencia Fin-Inicio. Fuente: Elaboración propia.

- **Fin-fin** → la sucesora no termina hasta que la predecesora no haya terminado, finalizan de manera conjunta.
 - En un examen la revisión y la corrección de errores termina a la vez. Ilustración 8.

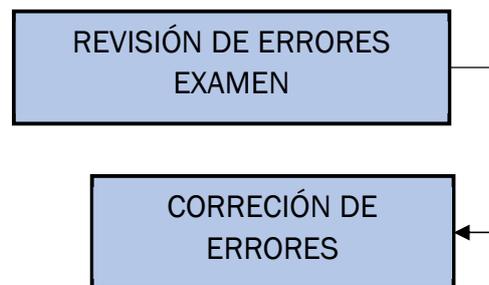


Ilustración 8: Dependencia Fin-Fin. Fuente: Elaboración propia.

Cuando ya se han calculado las duraciones de las actividades y la relación de dependencia entre ellas, se está en disposición de representarlo gráficamente en un calendario o cronograma y así saber la fecha de inicio o de finalización del proyecto. Pueden darse dos sucesos:

- Fecha de inicio conocida – en este caso la programación del proyecto se hace de manera progresiva, hacia adelante, partiendo de la fecha de comienzo conocida y utilizando las duraciones y relaciones de precedencia, se halla la fecha de finalización más temprana posible.
- Fecha de fin conocida – este tipo de programación se llama regresiva o hacia atrás, cuando se sabe la fecha límite, la programación regresiva nos dará la fecha más tardía en la que comenzar el proyecto para cumplir con los plazos establecidos.

También, hay que diferenciar entre dos tipos de actividades, las que tienen holgura y las que no. La holgura se define como el tiempo que se puede retrasar el comienzo de una actividad sin que eso cause un retraso en la fecha finalización del proyecto. Por otro lado, las actividades con holgura 0, son aquellas llamadas actividades críticas, que no tienen margen para retrasarse, ya que, si lo hacen, este retraso afectaría al proyecto retrasando su fecha de finalización. El conjunto de actividades críticas de un proyecto forma el camino/ruta crítica, y va desde el comienzo al final del proyecto. En algún caso puede que haya más de una ruta crítica, pero siempre al menos una, y marcará la duración total del proyecto, que será la duración de las actividades que componen la ruta crítica.

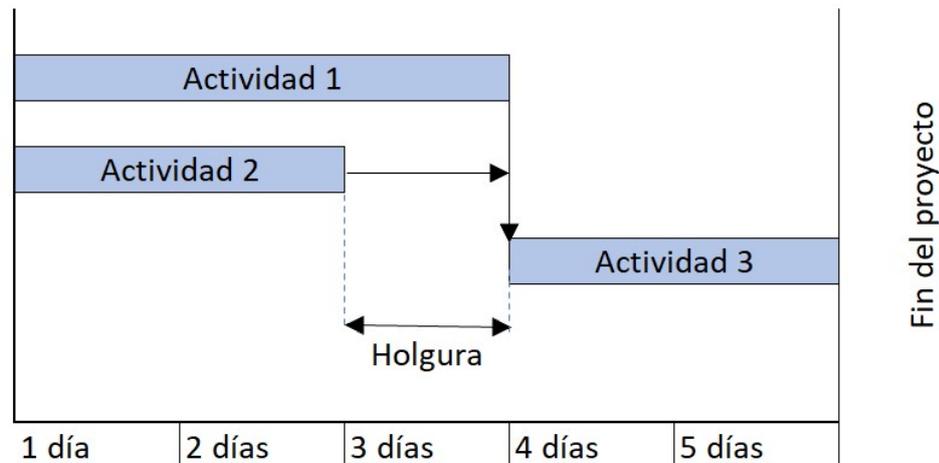


Ilustración 9: Actividades con holgura. Fuente: Elaboración propia.

Como se observa en la Ilustración 9, la actividad 2 dura dos días y posee una holgura de un día, es decir, esa actividad tiene ese margen extra para su ejecución, sin que ese retraso llegue a afectar a la duración del proyecto. Sin embargo, la actividad 1 dura tres días, y es crítica (tiene holgura 0). Por tanto, si se produjera un retraso en su ejecución, éste tendría un impacto directo en la fecha de finalización del proyecto, porque retrasaría el inicio de la actividad 3, y por tanto también su finalización. En este ejemplo las actividades 1 y 3 son críticas, formando la ruta crítica.

A continuación, se exponen dos tipos de métodos empleados para la realización del cronograma de un proyecto. Se distinguen entre los métodos determinísticos donde la duración de las actividades es conocida, como Gantt, CPM, o ROY, y los métodos estocásticos donde la duración de las actividades está sujeta a variables aleatorias, como PERT, o simulación de Montecarlo.

1. Métodos determinísticos: Gantt, CPM y Roy.

La realización del cronograma nos va a permitir visualizar la duración de las actividades, así como las holguras, y por tanto las fechas de inicio y finalización programada del proyecto.

A continuación, se detallan las herramientas más extendidas para encontrar las fechas. Se distinguen las que están basadas en gráficos temporales, como



es el diagrama de Gantt, y las que se basan en grafos, entre las que destacan CPM (*Critical Path Method*), y ROY.

1.1 GANTT

Es la forma de representar el cronograma de un proyecto más utilizada. Es un gráfico temporal donde las tareas se representan con rectángulos de longitud proporcional a la duración (Poza, 2021).

Henry Gantt, un ingeniero mecánico norteamericano, desarrollo esta metodología en 1910 con el objetivo de ayudar a sus supervisores a comprender el progreso de los proyectos, garantizando que las actividades se realizaban siguiendo la planificación. Se trata por tanto de una herramienta visual que permite programar, gestionar, y monitorizar las actividades de un proyecto (Kopp, 2022).

Típicamente es un gráfico de barras colocadas horizontalmente, donde en el eje X se representa el tiempo, es decir la barra será tan larga como la duración de la actividad que se esté representando, y en el eje Y las distintas tareas que componen el proyecto.

Permite ver la secuencia de actividades que deben realizarse, la duración de cada una de ellas, las holguras, y las fechas de inicio y fin del proyecto. Además, permite distinguir claramente la ruta crítica del proyecto.

Se distinguen dos tipos de diagramas de Gantt. En primer lugar, el Gantt de Actividades, que es el más utilizado y el que se ha descrito, y en segundo lugar el Gantt de Recursos, que permite programar teniendo en cuenta la limitación de los recursos (PMI, 2017).

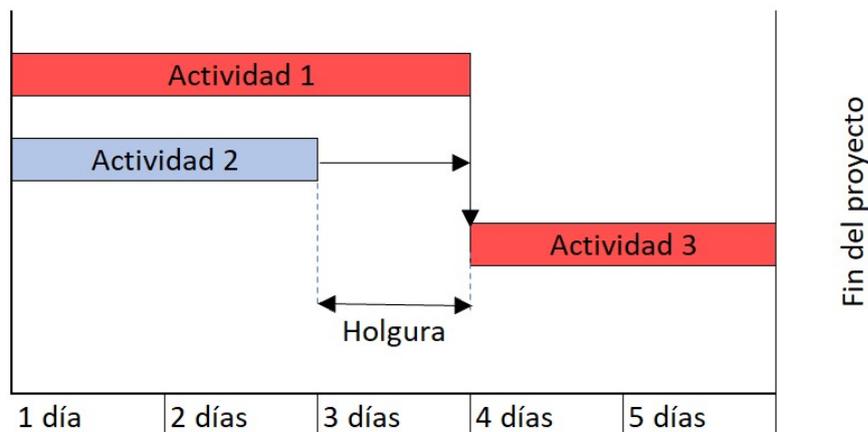


Ilustración 10: Camino crítico del ejemplo. Fuente: Elaboración propia.

En la Ilustración 10, se muestra el Gantt de actividades para un proyecto formado por tres, y se aprecia en color rojo el camino crítico, formado por las actividades 1 y 3.

1.2 CPM

Critical Path Method, es una herramienta de programación de proyectos que tiene su origen entre el año 1956 y el 1959. Este sistema de gestión de tiempos fue desarrollado por James Kelley de Remington, y Morgan Walker Dupont, con el objetivo de supervisar los tiempos de realización de los proyectos.

La principal característica de CPM es que calcula el mínimo tiempo posible de duración del proyecto, formado por la secuencia de actividades más larga necesarias para finalizar el proyecto (Redbooth Team, 2018).

Las actividades se representan en los arcos y los nodos (círculo) representan eventos temporales, incluyendo uno al principio del proyecto que representa el inicio, y otro al final. Es frecuente para el uso de esta herramienta, la necesidad de recurrir a incorporar actividades ficticias (sin duración), para cuadrar la estructura del proyecto (Kenton, 2021).

También permite la identificación de la ruta crítica, que es aquella secuencia de actividades con holgura 0, es decir, aquella secuencia formada por actividades que, de sufrir algún retraso en su fecha de finalización, tendría un impacto directo en la fecha de finalización del proyecto entero. Para ello hay que tener en cuenta que requiere 3 parámetros que se incluyen en los nodos.

Estos son la estimación de la fecha más temprana y más tardía de comienzo de cada tarea, y la holgura de cada actividad, y su representación queda plasmada en la Ilustración 11.

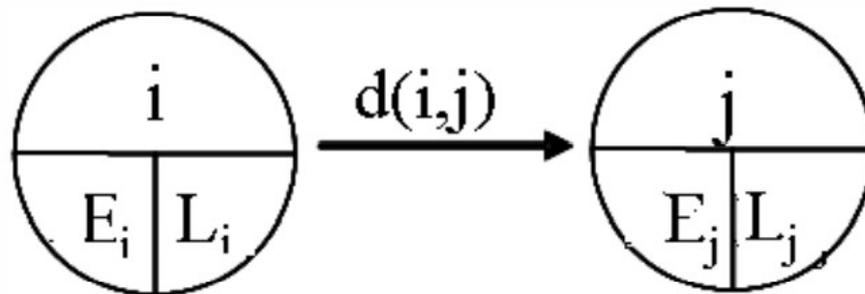


Ilustración 11: Notación en CPM. Fuente: (Posadas, 2018).

- **E_i**
 - Es la fecha más temprana de comienzo de la actividad i.
 - Se calcula recorriendo la red de izquierda a derecha, aplicando $E_i = \max(E_i + d_{i,j})$
Donde: - E_i del primer nodo es 0
- Las demás E_i se calculan según el orden de numeración de los nodos.
- **L_i**
 - Es la fecha más tardía de comienzo de la actividad i.
 - Se calcula recorriendo la red de derecha a izquierda, aplicando $L_i = \min(L_i - d_{i,j})$
Donde: L_i del primer nodo de la derecha coincide con su E_i .
- Las demás L_i se calculan según el orden inverso de numeración de los nodos.
- **$H_{i,j}$**
 - La holgura, o margen de que dispone cada actividad para retrasarse sin afectar a la fecha de finalización del proyecto.
 - Se calcula como la diferencia entre la fecha más tardía y la fecha más temprana de comienzo de las actividades. $H_{i,j} = L_i - E_i$
 - Aquellas actividades cuya holgura sea 0, se las denomina críticas, y el conjunto de todas las actividades críticas forma el camino crítico, el cual como se ha visto, no debe sufrir retrasos ya de que tendría un impacto en la fecha de finalización del proyecto.

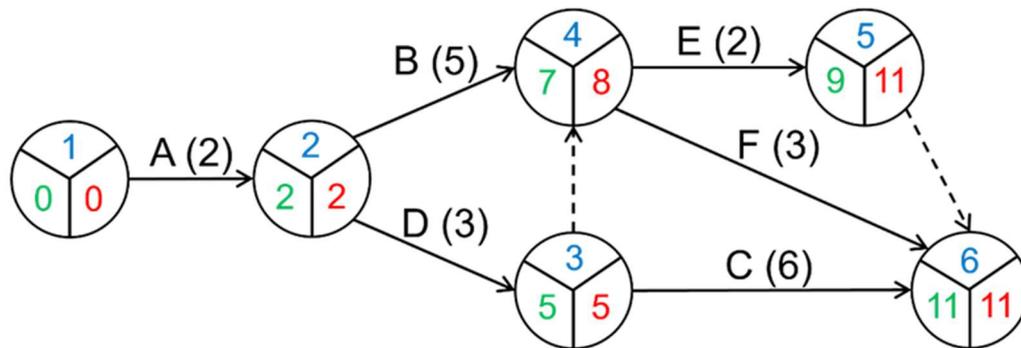


Ilustración 12: Grafo usando CPM. Fuente: (Poza, 2021).

En la Ilustración 12 se muestra un ejemplo de un grafo realizado usando la herramienta CPM. Como se aprecia, las actividades están en los arcos, y en los nodos con forma circular se muestran los eventos temporales con los tres parámetros de los que se ha hablado. A partir de este ejemplo se pueden calcular las holguras:

$$H_A = L_2 - E_1 - d_{1,2} = 2 - 0 - 2 = 0$$

$$H_B = L_4 - E_2 - d_{2,4} = 8 - 2 - 5 = 1$$

$$H_C = L_6 - E_3 - d_{3,6} = 11 - 5 - 6 = 0$$

$$H_D = L_3 - E_2 - d_{2,3} = 5 - 2 - 3 = 0$$

El camino crítico es el formado por A-D-C.

Por último, las ventajas de esta herramienta incluyen la facilidad de visualización y la flexibilidad para hacer cambios, aunque si el proyecto es complejo y tiene muchas actividades, el grafo se complica y es mejor escoger otra de las herramientas que se describirán a continuación.

1.3 ROY

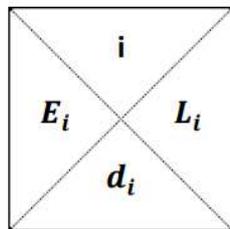
Originado en Europa entre 1958 y 1961, por los ingenieros B. Roy y M. Simonard. Es similar a PERT y CPM en que también permite establecer redes, pero sin necesidad de utilizar actividades ficticias (solo al inicio y final).

El método de ROY también es un método que sirve para encontrar fechas para las actividades y está basado en grafos, pero se diferencia del PERT/CPM porque su representación es la inversa. En ROY las actividades se representan

en los nodos que tienen forma de cuadrado, y los arcos que unen los nodos representan las relaciones de precedencia.

El grafo tendrá solamente dos actividades ficticias, una al inicio del grafo para reflejar el comienzo del proyecto, y otra al final, representando el final. El nodo inicial no tiene ninguna flecha incidente puesto que es el comienzo, y de él salen tantas como actividades haya sin una precedente. Por otra parte, del nodo final no saldrá ninguna flecha, pero a él llegan tantas como actividades no tienen sucesoras (Posadas, 2018).

Cálculo de fechas (Marcos, 2021), representado en la Ilustración 13:



- i: actividad i.
- E_i : fecha más temprana en que puede comenzar la actividad i.
- d_i : duración de la actividad i.
- L_i : fecha más tardía en que puede comenzar la actividad i.

Ilustración 13: Notación ROY.
Fuente: (Marcos, 2021)

Cálculo de Holguras: $H_i = L_i - E_i$

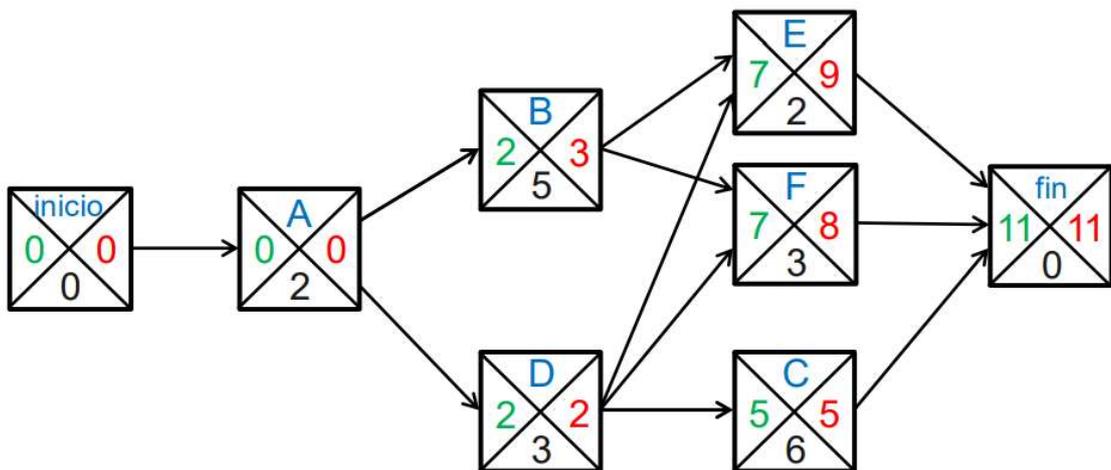


Ilustración 14: Grafo usando ROY. Fuente: (Poza, 2021)

En la Ilustración 14 se muestra un grafo ROY, en el que se aprecia la forma cuadrada de los nodos, donde se representan las actividades (A, B, C...),



contando como hemos dicho con un único nodo inicial, y uno final, donde se ve la duración del proyecto. Véase como ejemplo la actividad “B”. Esta actividad como pronto podrá comenzar en 2, porque “A” comienza en 0 y tiene una duración de dos unidades de tiempo, y es la única antecesora. Sin embargo, como tarde “B” debe comenzar en 3, esto es porque tiene una duración de 5 días, y sus predecesoras “E” y “F” como tarde pueden comenzar en 9 y 8 respectivamente, es decir, la relación entre “F” y “B” está limitando a la actividad “B” a haber finalizado en 8, y como dura 5 unidades de tiempo, para cumplirlo tiene que empezar como tarde en 3.

Las cuatro metodologías de las que se ha hablado solo tienen en cuenta las relaciones de precedencia como restricción para realizar el cronograma del proyecto. En la realidad, también hay que tener en cuenta los recursos necesarios por cada actividad para desarrollarse, puesto que no son ilimitados. Como se describía en la sección I.1, estas restricciones están implícitas en todos los proyectos, y el coste del proyecto, o el tiempo que se tiene para llevarlo a cabo, son limitaciones a tener en cuenta, pero también la cantidad de trabajadores o la utilización de las máquinas tiene un límite. Es por ello por lo que es importante controlar y planificar también la utilización de los recursos, y con esa idea surge CCPM (Critical Chain Project Management) que será detallada en el siguiente capítulo.

2. Métodos estocásticos: PERT y simulación de Montecarlo.

Para el cálculo del cronograma, estos métodos modelan la duración de las actividades mediante variables aleatorias, lo que provoca que la predicción sea más fiable, y tenga más probabilidades de ajustarse a la realidad. PERT considera aleatoriedad únicamente en las duraciones de las actividades del camino crítico, y simulación de Montecarlo, para más fiabilidad, también tiene en cuenta la aleatoriedad de las duraciones de las actividades que forman parte de las cadenas no-críticas.

2.1 PERT

Program Evaluation Review Technique, en español significa Técnica de Revisión de Evaluación de Programas, y fue desarrollado por la marina de EEUU

en 1957. Se desarrolló en paralelo a la ya mencionada herramienta CPM, al ver la necesidad de tener un sistema de planificación y control para su programa de misiles.

Es una herramienta similar al CPM, se asemejan en que ambos son una herramienta de programación de proyectos (ayuda a encontrar una fecha para la ejecución de las actividades), y ambos métodos emplean grafos para la realización del cronograma, con la diferencia de que en PERT las duraciones de las actividades son aleatorias, y en CPM la duración es conocida. PERT permite obtener la función de distribución de probabilidad de la duración del proyecto, ayudando así a predecir la probabilidad de que el proyecto termine antes de cierta fecha.

Debido a la incertidumbre generada por la duración aleatoria de las actividades, para predecir dicha duración se basa en tres estimaciones: la estimación optimista, la pesimista, y la más probable.

- Tiempo optimista (T_o) – estimación de tiempo más corta (mejor valor del histórico).
- Tiempo pesimista (T_p) – estimación más larga (peor valor del histórico).
- Tiempo más probable (T_m) – el valor más probable.

Se asume que las duraciones de las actividades, que son variables aleatorias, se ajustan a una distribución de probabilidad de tipo β (Beta).

Distribución Beta

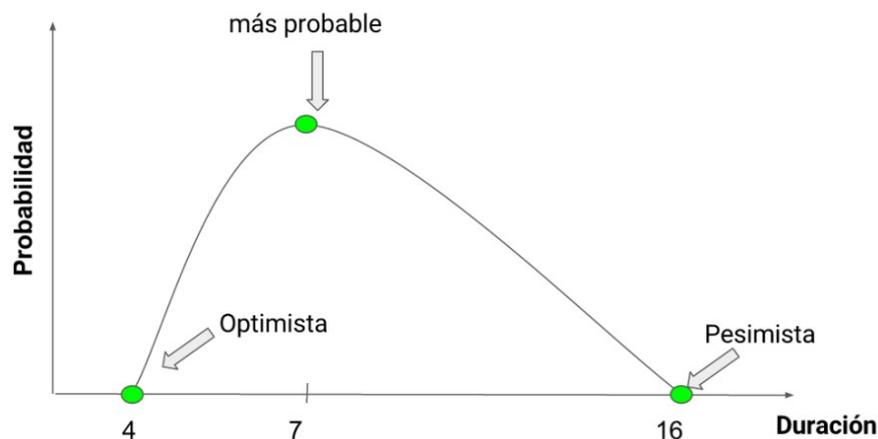


Ilustración 15 : Estimaciones para PERT. Fuente: (Josafat Gascón Busio).

En el ejemplo de la Ilustración 15, se observan las tres estimaciones calculadas para cierta actividad: T_o es cuatro días, T_p es dieciséis días, y T_m es siete días.

Para la duración de las actividades se calcula la duración “media” de las tres estimaciones mencionadas anteriormente mediante la fórmula:

$$E(d) = \frac{T_o + 4T_m + T_p}{6}$$

Distribución Beta

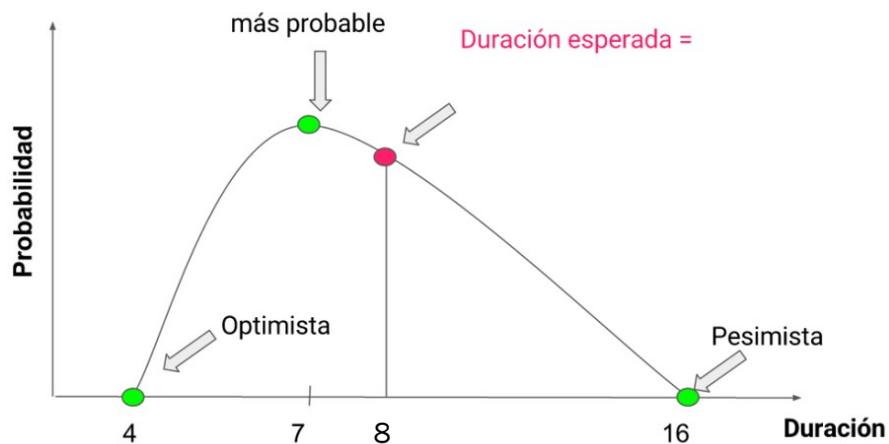


Ilustración 16: Duración más probable PERT. Fuente: (Josafat Gascón Busio).

Aplicando la fórmula para obtener la duración esperada en el ejemplo, obtenemos:

$$E(d) = \frac{4 + 4 \times 7 + 16}{6} = 8 \text{ días}$$

Y para calcular la varianza en la duración del proyecto:

$$Var_i = \left(\frac{T_p - T_o}{6}\right)^2$$

Volviendo al ejemplo, la varianza sería:

$$Var_i = \left(\frac{16 - 4}{6}\right)^2 = 4$$



El camino crítico, de igual manera que las actividades, tendrá una duración aleatoria modelada como una función gaussiana cuya media se obtiene como la suma de las duraciones de las actividades que componen el camino crítico; y cuya varianza es la suma de las varianzas de las actividades que componen el camino crítico.

El grafo de este método es similar al del CPM, pues en los nodos circulares se representan las relaciones de precedencia, mientras que las actividades se representan en los arcos.

2.2 Simulación de Montecarlo

Ya se ha detallado la técnica PERT, utilizada para estimar las fechas de las actividades de un proyecto cuando tienen una duración aleatoria, y con ello estimar la duración total del proyecto en términos probabilísticos. Esta técnica asumía para la duración aleatoria de las actividades una distribución beta, basada en tres parámetros (tiempo optimista, pesimista, y más probable) con los cuales hallaba el tiempo esperado (IBM, 2020). La duración del proyecto se calcula como la suma de las duraciones medias (tiempo esperado) de las actividades del camino crítico, sin tener en cuenta la posible influencia de las actividades no críticas (Poza, 2019).

El hecho de ignorar el impacto que podría ocasionar la aleatoriedad en la duración de las actividades no críticas, en la duración del proyecto, hace que el método PERT sea considerado un método demasiado optimista, pues subestima la duración del proyecto.

Para obtener un escenario más próximo a la realidad deben tenerse en cuenta más variables, lo que, sin embargo, dificulta su planificación y la hace más compleja. Una alternativa muy extendida para realizar un pronóstico es la simulación, realizada con ordenador, que permita tener en cuenta todas las variables y realizar iteraciones experimentando con distintas situaciones para saber qué resultados esperar ante determinadas causas.

La simulación busca replicar características y comportamientos del sistema real mediante métodos matemáticos. Permite la introducción de nuevas variables, modificar sus valores, y analizar las consecuencias de estas modificaciones. La simulación Montecarlo es un conjunto de métodos estadísticos que combina aleatoriedad y determinismo y que permite resolver problemas matemáticos utilizando números aleatorios como base para realizar simulaciones. Mayoritariamente se utiliza para hacer frente a problemas para



los cuales no es posible (porque hay demasiada incertidumbre) o no es práctico (gran complejidad) darles solución mediante algoritmos deterministas, por tanto, ayuda a obtener una aproximación numérica a un problema cuya solución analítica es demasiado compleja (Orobio & Gómez, 2015).

John von Neumann y Stanislaw Ulam, diseñaron esta idea (Martínez Pérez, 2013), hacia 1940, para facilitar la toma de decisiones en ambientes de incertidumbre ya que permite ver todos los resultados posibles y evaluar el impacto del riesgo (Palisade).

Este método permite predecir un conjunto de resultados, utilizando unos valores comprendidos entre un número mínimo y un número máximo, a partir de un conjunto de valores de entrada fijos. Para que los resultados sean significativos se debe generar una cantidad de datos aleatorios lo suficientemente grande, y la obtención de estos números aleatorios se hace con programas informáticos.

El nombre de la metodología viene de la ciudad de Montecarlo, ya que la aleatoriedad de la simulación recuerda al juego más popular en su casino, la ruleta.

Conviene remarcar que los resultados obtenidos con esta metodología son estimaciones que permiten hacerse a la idea de cómo el sistema se comportaría ante determinadas condiciones iniciales y con la interacción de distintas variables, pero no es un cálculo exacto ni debe entenderse así.

Los pasos a seguir para utilizar los métodos de Montecarlo son (IBM, 2020):

- 1- Configurar el modelo, identificando tanto la variable dependiente que se debe predecir, como las variables independientes o variables de entrada que van a permitir su predicción.
- 2- Especificar las distribuciones de probabilidad de las variables independientes, y usando la experiencia o datos históricos, definir un rango de posibles valores y su probabilidad.
- 3- Realizar simulaciones repetidamente para conseguir los valores aleatorios de las variables independientes hasta que haya suficientes.

2.2.1 Ejemplo.

A continuación, a modo de ejemplo, se representa un proyecto de seis actividades al que se aplica PERT y simulación de Montecarlo para hallar las duraciones de las actividades mediante una simulación de 1.000 iteraciones y

extrayendo la duración media. Este ejemplo se corresponde con el grafo utilizado en la sección 3.1.2.

PERT asume que la duración del proyecto se puede modelar como una variable aleatoria gaussiana de media la suma de las duraciones medias de las actividades críticas, y varianza la suma de las varianzas de las actividades críticas. Las seis actividades, (A-B-C-D-E-F) se considera que tienen duración aleatoria con una función de distribución Beta. En la Tabla 1 se indica la duración optimista, pesimista, y más probable para cada actividad.

Actividad	A	B	C	D	E	F
T_o (Duración Optimista)	1	4	4	1,5	1	2
T_m (Duración más probable)	1,5	4,5	5,5	2,5	2	3
T_p (Duración Pesimista)	5	8	10	6,5	3	4
Duración Media	2	5	6	3	2	3

Tabla 1: Duraciones de las actividades del ejemplo. Fuente: Elaboración propia.

Resolución aplicando la herramienta PERT

Una vez sabemos las duraciones de las actividades, se pueden calcular, utilizando la técnica PERT, el resto de las fechas (E_i y L_j). En la Ilustración 17 se ve representado:

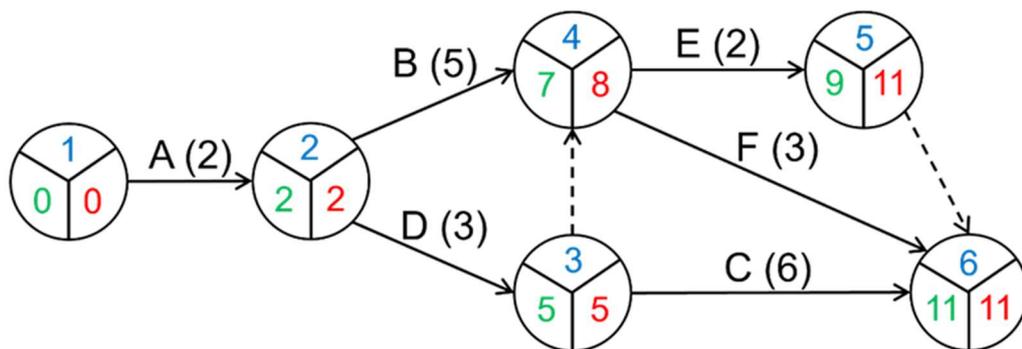


Ilustración 17: Grafo de las actividades del ejemplo. Fuente: (Poza, 2021).



A continuación, se buscan todos los caminos que conectan el nodo inicial con el nodo final, y su duración. Estos son:

- A-B-E $\rightarrow 2+5+2 = 9$
- A-B-F $\rightarrow 2+5+3 = 10$
- A-D-E $\rightarrow 2+3+2 = 7$
- A-D-F $\rightarrow 2+3+3 = 8$
- A-D-C $\rightarrow 2+3+6 = \underline{11}$

Identificamos el camino más largo, el ADC, que es el camino crítico y permanecerá invariable, porque como se ha expuesto, la herramienta PERT no tiene en cuenta ni el impacto de las actividades no críticas, ni los posibles valores aleatorios que pueden tomar las duraciones de las actividades (sólo se tiene en cuenta la duración media de las actividades), por lo que, para el método PERT, el camino crítico será siempre el mismo.

La varianza del proyecto es la suma de las varianzas de las actividades que forman el camino crítico, A, D, y C en este caso.

$$Var_i = \left(\frac{T_p - T_o}{6}\right)^2 = \left(\frac{5 - 1}{6}\right)^2 + \left(\frac{6,5 - 1,5}{6}\right)^2 + \left(\frac{10 - 4}{6}\right)^2 = 2,138$$

Resumiendo, según PERT, obtenemos que el camino crítico del proyecto está formado por las actividades A-D-C, con una duración esperada de 11 unidades de tiempo, y una varianza de 2,138.

Resolución aplicando simulación de Montecarlo

En primer lugar, ya tenemos la tabla con las estimaciones de las duraciones de las actividades, así como el grafo del proyecto, y la lista de caminos que unen el nodo inicial con el nodo de fin del proyecto.

- A-B-E; A-B-F; A-D-E; A-D-F; A-D-C

Para la realización de este ejercicio se ha utilizado Microsoft Excel para simular 1.000 rondas, y poder estimar la media y la varianza del proyecto a partir de las duraciones obtenidas en cada ronda de la simulación, así como estimar la criticidad de sus actividades.

Para generar la duración aleatoria de las actividades se utiliza la función BETA.INV (), a la que se le pasa como argumentos la probabilidad, los parámetros α y β , y la duración optimista y la pesimista.

- Como probabilidad se establece un número aleatorio entre 0 y 1.
- Para hallar los parámetros Alpha y beta se utiliza la siguiente fórmula:

$$\alpha = \frac{4(Tm - To)}{Tp - To} + 1$$

$$\beta = \frac{4(Tp - Tm)}{Tp - To} + 1$$

- La duración optimista y pesimista To y Tp .

Teniendo la duración de las actividades, el siguiente paso es calcular la duración de cada uno de los cinco caminos críticos posibles, en cada ronda de la simulación. Eligiendo en cada ronda el camino de más duración como camino crítico.

A continuación, en la Tabla 2 se muestra un extracto de la simulación con 10 iteraciones, pero en la práctica, y para sacar las conclusiones finales, se han corrido 1.000 iteraciones.

	Camino A-B-E-	Camino A-B-F	Camino A-D-E	Camino A-D-F	Camino A-D-C	Duración del proyecto
Ronda 1	8,578	8,798	6,537	6,758	8,320	8,798
Ronda 2	8,884	10,658	8,675	10,449	12,744	12,744
Ronda 3	7,718	9,581	5,253	7,115	8,678	9,581
Ronda 4	9,201	10,028	8,796	9,623	11,673	11,673
Ronda 5	8,619	10,071	6,094	7,546	10,479	10,479
Ronda 6	8,387	8,904	6,788	7,306	10,749	10,749
Ronda 7	9,787	10,317	6,530	7,061	10,897	10,897
Ronda 8	8,584	9,124	7,299	7,840	12,414	12,414
Ronda 9	9,299	10,281	6,838	7,820	10,179	10,281
Ronda 10	9,124	10,435	7,409	8,719	11,188	11,188

Tabla 2: Simulación de 10 iteraciones. Fuente: Elaboración propia.

Falta por identificar el camino crítico de todo el proyecto, y como la duración de las actividades es aleatoria y está por tanto sujeta a incertidumbre, dicha duración puede variar. Por ello, es posible que en distintas rondas de la

simulación el camino crítico resultante sea diferente. Por esta razón se define el índice de criticidad de una actividad como la probabilidad de que una actividad sea crítica. Además, tal y como se aprecia en la Tabla 2, en las rondas 1, 3, y 7, se ha obtenido como camino crítico el A-B-F, siendo éste distinto al A-D-C, que era el camino crítico invariable que se obtenía con PERT.

El Índice de Criticidad de una actividad se calcula como el porcentaje de veces que dicha actividad pertenece al camino crítico de mayor duración para cada ronda de la simulación, dividido entre el número total de rondas desarrolladas. Esto queda reflejado en el Excel asignando un 1 a la actividad si pertenece al camino más largo, y un 0 en el caso contrario. Para calcularlo, se suman todos los 1's y se divide por el número total de simulaciones (Acebes Senovilla, Poza García, & López Paredes, 2022).

INDICE DE CRITICIDAD DE LAS ACTIVIDADES						
	A	B	C	D	E	F
	100%	27%	73%	73%	1%	26%
Ronda 1	1	0	1	1	0	0
Ronda 2	1	0	1	1	0	0
Ronda 3	1	0	1	1	0	0
Ronda 4	1	1	0	0	0	1
Ronda 5	1	0	1	1	0	0
Ronda 6	1	1	0	0	0	1
Ronda 7	1	1	0	0	0	1
Ronda 8	1	1	0	0	0	1
Ronda 9	1	0	1	1	0	0
Ronda 10	1	0	1	1	0	0

Tabla 3: Índice de criticidad. Fuente: Elaboración propia.

Lo primero que se aprecia en la Tabla 3, es que la actividad A tiene un índice de criticidad de 1, lo cual era de esperar ya que la actividad A forma parte de los 5 caminos críticos posibles. Después, las dos actividades que mayor índice de criticidad tienen son las actividades C y D, ambas con un 73% de probabilidad en este caso. Esto quiere decir, que el 73% de las simulaciones el camino crítico ha estado formado por estas actividades. Identificar estas actividades que poseen el índice de criticidad más alto, permite controlarlas ya que son susceptibles de convertirse en cuellos de botella (Acebes Senovilla, Poza García, & López Paredes, 2022).



Además, en esta Tabla 3 también se observa el índice de criticidad de las actividades B, y F, siendo éste de un 27% y 26% respectivamente. Es decir, casi tres veces de cada diez, el camino crítico pasa por las actividades “B” y “F”, que cuadra con los resultados obtenidos en la simulación mostrada en la Tabla 2, donde a lo largo de diez iteraciones, el camino crítico era el formado por A-B-F, en tres ocasiones.

Una vez realizadas todas las rondas de la simulación, y hallado para cada una su camino crítico, se puede hallar la duración media y la varianza de todo el proyecto.

Duración media del proyecto	11,294
Varianza de la duración del proyecto	1,830

Tabla 4: Duración media y Varianza del proyecto. Fuente: Elaboración propia.

A pesar de haber modelado la duración de las actividades con las mismas funciones de distribución, los resultados obtenidos con la simulación de Montecarlo, muestran una duración superior a la obtenida con PERT, demostrando que la duración de PERT era demasiado optimista.

En este caso, la conclusión a la que se llega utilizando la herramienta PERT, donde se veía que el camino crítico propuesto era el formado por las actividades A-D-C, coincide con las actividades de mayor índice de criticidad utilizando la metodología de simulación de Montecarlo. Aun así, es importante destacar ese aproximadamente 27% de las simulaciones en las cuales, debido a la incertidumbre ocasionada por las duraciones aleatorias de las actividades, el camino crítico era otro.



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**



IV. MÉTODO CCPM

CCPM, que en inglés significa *Critical Chain Project Management* (Método de la Cadena Crítica), es un método de planificación y gestión de proyectos que fue introducido por Eliyahu Goldratt en el año 1997 (Luz, 2017), año en el cual también publica otro de sus libros, “Critical Chain”, donde introduce la Teoría de las Restricciones (TOC), utilizada en la metodología CCPM.

El objetivo de esta metodología es el de vencer los impedimentos que le generaban las técnicas tradicionales de programación de proyectos. Según el PMBOK: “El método de la Cadena Crítica es una técnica de análisis de la red de programación que modifica el cronograma del proyecto para tener en cuenta los recursos limitados” (PMI, 2017).

La particularidad de este método es que no solo se centra en las actividades y su duración y relación de dependencia, sino que introduce un mecanismo para gestionar la incertidumbre, basándose en la naturaleza de los recursos necesarios para ejecutar las actividades del proyecto, lo que le diferencia de otros métodos para la programación de proyectos como el CPM, el PERT o el ROY (ver sección III.1) (Torre, 2015).

La incertidumbre es la causante de la mayoría de los retrasos de los proyectos, que inducen a sobrecostos y retrasos. Esta metodología gestiona la incertidumbre asociada a la duración de las actividades eliminando la protección implícita agregada a cada actividad individualmente, y añadiéndola como un *buffer* (amortiguador) en puntos específicos del proyecto. Es decir, se centra en la finalización del proyecto global antes que en la finalización de cada actividad por separado. Dichos amortiguadores son actividades ficticias de duración determinada, encargadas de absorber la incertidumbre, es decir, cualquier retraso en la ejecución de las actividades (Verma, 2022).

Cabe destacar la diferencia de estos amortiguadores con las ya mencionadas holguras de las metodologías anteriormente explicadas. La mayor diferencia reside en que las holguras ocurren de forma espontánea en el proyecto, y no sirven para hacer frente a las fluctuaciones de la duración de ejecución de las actividades, sin embargo, los *buffers* sí que son capaces de absorber esa incertidumbre.

En función de la ubicación de los amortiguadores en el plan de proyecto, CCPM distingue entre tres tipos: los *buffers* de proyecto, para absorber cualquier retraso en actividades de la cadena crítica, los *buffers* de alimentación, para proteger a la cadena crítica de retrasos en cadenas secundarias; y los *buffers*



de recurso, para advertir a los recursos que van a trabajar en la cadena crítica a continuación.

1. Bases:

1.1 Teoría de las restricciones

TOC (“Theory of Constraints”) es aquel conjunto de metodologías que ayudan a identificar las limitaciones que impiden alcanzar los objetivos del proyecto.

Desarrollada por Eliyahu Goldratt en su libro “Critical Chain”, publicado en 1997, define las limitaciones del sistema como cuellos de botella, causantes de que un alto porcentaje de los proyectos no acaben a tiempo y muchos de los mismos sean cancelados. El origen de estas limitaciones tiene lugar en una mala gestión del tiempo, incluyendo el ansia de “multitarea”, perdiendo eficiencia al trabajar en más de una cosa a la vez e ir cambiando, el mal establecimiento de prioridades, o la dependencia entre tareas.

Se trata de metodología de programación de proyectos para aumentar la productividad. Estableciendo que la mejor forma de hacerlo es la sincronización de la cadena de producción con el factor limitante. Goldratt propone 5 pasos para conocer las limitaciones del sistema, a través de la metodología DBR (Drum, *Buffer*, Rope) explicada a continuación.

1.1.1 Método DBR (*Drum Buffer Rope*)

Este método, en español conocido como Tambor – Amortiguador – Cuerda, se centra en identificar las limitaciones o cuellos de botella del sistema.

Se trata de un método de planificación, que, aunque incapaz de adelantarse a los imprevistos que puedan surgir, busca minimizar los retrasos y paradas durante el proceso.

a. Drum (Tambor)

El tambor es el ritmo marcado por la limitación o cuello de botella. Esta metodología es análoga de una marcha de boy-scouts, en la cual el Drum



(tambor) viene representado por el boy-scout más lento, que ralentiza la marcha y dicta la velocidad de los demás.

Para que no llegue a ser problemático el cuello de botella, el ritmo tiene que ser aquel que coincida con el máximo aprovechamiento de la restricción, y permanecer constante, y el resto de los procesos deben adaptarse a él.

Para que la marcha de boy-scouts prosiga normalmente, todos deben andar al ritmo del boy-scout más lento, si anduvieran más rápido se saturaría y se caería, causando un mayor retraso en la marcha, y si caminaran más despacio no lo estarían aprovechando al máximo.

b. *Buffer* (Amortiguador)

Se utilizan para evitar que las fluctuaciones en las protecciones de cada una de las actividades del proyecto puedan sufrir variaciones y afectar al ritmo marcado por el tambor.

El objetivo de estos *buffers* es el de asegurar un flujo continuo.

En la analogía, el *buffer* viene representado por la distancia entre los boy-scouts, de manera que si alguno de los boy-scouts en cabeza sufre una caída, ese parón no afecte al boy-scout más lento (tambor), sino que sea absorbido por los amortiguadores.

c. *Rope* (Cuerda)

La cuerda que sujetan cada uno de los boy-scouts cuando caminan y que asegura que todos van al mismo ritmo (el marcado por el tambor).

En términos del proceso de producción, la “Cuerda” representa el programa de liberación de materiales, es decir, la velocidad con la que los materiales son liberados y debe ajustarse al ritmo del tambor, para no ocasionar sobrecarga en el cuello de botella. Se encarga de que la limitación este alimentada con la cantidad justa de trabajo.

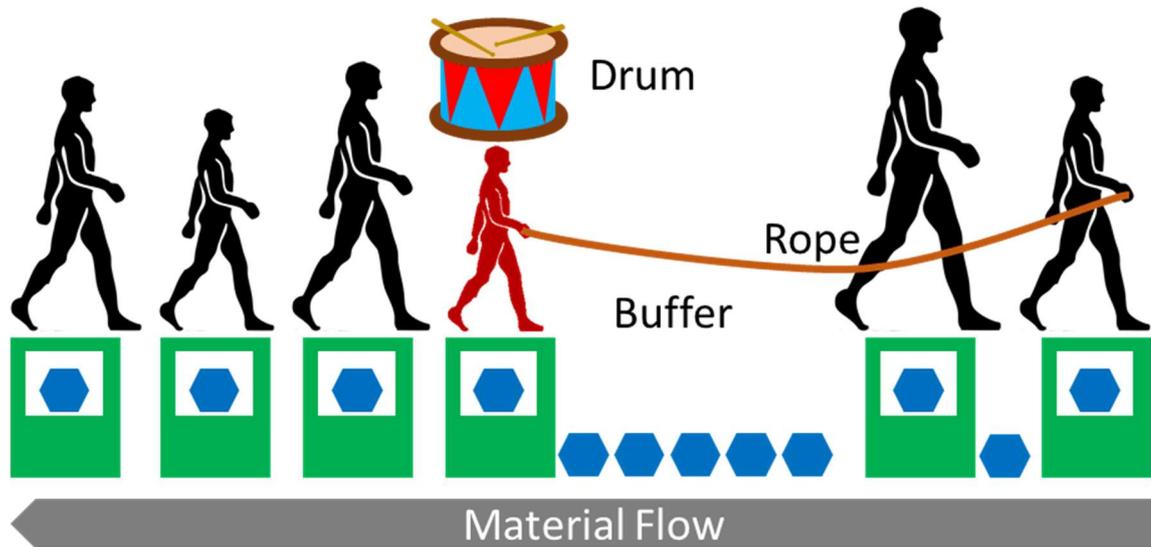


Ilustración 18: Método DBR. Fuente: (Roser, 2014)

En la Ilustración 18 se muestra gráficamente la metodología DBR. El flujo continuo que se consigue con la sincronización al ritmo del cuello de botella, minimiza la cantidad de interrupciones durante el proceso productivo.

La TOC, para aplicarla en la cadena de producción de una empresa, se hace a través de la mencionada metodología DBR, y en cinco pasos permite identificar las restricciones del sistema.

1. Identificar el cuello de botella

- Hay diversos parámetros que pueden ocasionar este problema, como la carga de trabajo, la frecuencia de uso de una máquina, o la tendencia a romperse y ocasionar retraso.

2. Decidir cómo explotarlo

- El cuello de botella hemos visto que marca el ritmo de producción del sistema, y para poder aumentarla, hay que buscar la forma que permita explotar la limitación para que deje de serlo.
- Se deberá priorizar la utilización de este recurso para no detener la producción, pero todavía extraer el máximo beneficio posible.

3. Subordinar a la decisión de como explotarlo

- Convertir la restricción en el tambor (*drum*) del sistema, y adaptar el resto de los procesos al ritmo marcado para que no se sature y detenga la producción.



- Para lograrlo, se utiliza el amortiguador (*buffer*) de alimentación, que hace llegar material a los puntos críticos con antelación, evitando que sucedan interrupciones y asegurándose de que el *drum* siempre tenga material.
- Es importante mirar por el rendimiento global del sistema y no por el rendimiento local de los recursos, puesto que, de ser así, aumentarían los gastos sin ocasionar un beneficio global.

4. Elevar la restricción

- Para poder incrementar la producción hay que conseguir aumentar la capacidad de la limitación.

5. Empezar de nuevo

- Esta metodología está basada en la mejora continua, por tanto, cuando se ha conseguido aumentar la capacidad del que era el cuello de botella y se ha aumentado consecuentemente la producción, se debe reiniciar el proceso y analizar de nuevo el sistema, porque pueden haber surgido nuevas restricciones al hacer los cambios apropiados para elevar el cuello de botella.

2. Origen de los retrasos

La razón por la cual los proyectos se alargan, como dice Goldratt, hasta ocupar todo el tiempo que se les ha sido asignado, es por la aparición de retrasos, que pueden deberse a múltiples factores, tanto externos como internos, pero en cualquiera de ellos están fuera del control del director del proyecto. Las acciones que originan los retrasos habitualmente son Síndrome del Estudiante, la Ley de Parkinson, y la Ley de Murphy, que explicaremos a continuación.

2.1 Síndrome del estudiante

Introducido por Goldratt en su libro *Critical Chain*, donde realiza una analogía con los estudiantes y el fenómeno que experimentan al hacer frente a la realización de una tarea. Los estudiantes normalmente empiezan a dedicarse a fondo a la realización de la tarea cuando se acerca la fecha límite, habiendo desaprovechado el tiempo hasta entonces.

Lo mismo ocurre con los miembros del equipo de proyecto, que posponen la realización de las actividades, empezando a dedicar más tiempo según se acerca la fecha límite, lo que aumenta el riesgo de la aparición de un imprevisto que cause un impacto negativo en la fecha de finalización del proyecto.

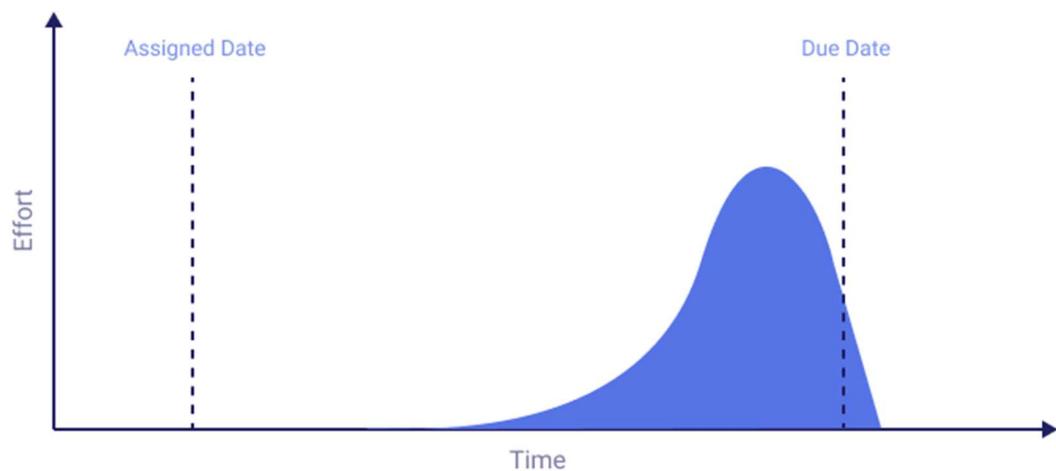


Ilustración 19: Síndrome del estudiante. Fuente: (Balyuk A. , s.f.)

Goldratt dice que “1/3” del trabajo se realiza durante los primeros “2/3” del tiempo, y los “2/3” del trabajo restante, se realizan en el último “1/3” del tiempo antes de cumplir con la fecha límite. Y esas cantidades se mantienen independientemente del tiempo que se posea para la realización de la tarea, de hecho, cuanto más tiempo se tenga, más prioridad pierde la realización de esta, y más ganan otras tareas.

2.2 Ley de Parkinson

Introducido por Cyril Northcote Parkinson, en un ensayo para "The Economist" en 1957, afirma que ningún trabajo se entrega antes de la fecha de finalización planificada, porque el trabajo se expandirá hasta que llene el tiempo disponible sin importar la intensidad del trabajo que se debe llevar a cabo.

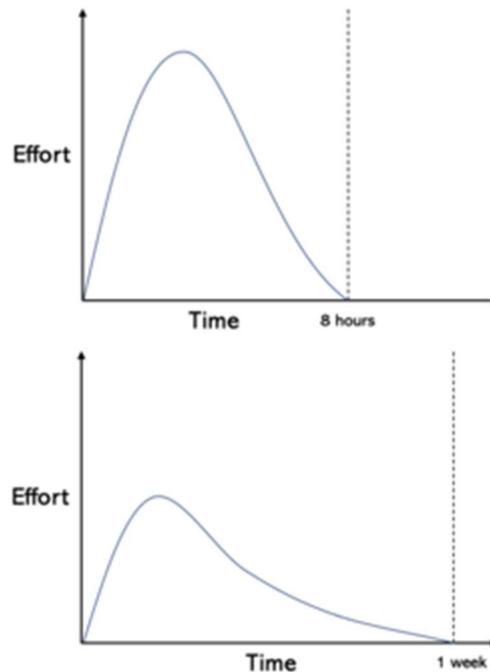


Ilustración 20: Ley de Parkinson. Fuente: (Strom, 2022)

En los gráficos de la Ilustración 20, se ve reflejado que la tarea se puede llevar a cabo en 8 horas, con un esfuerzo elevado, pero si la persona dispone de 1 semana, realizará un esfuerzo mucho menor y acabará tardando los siete días. Aun pudiendo hacerlo en menos tiempo, tendemos a organizarnos en función del tiempo disponible y en muy pocas ocasiones se reporta la finalización de una tarea antes del plazo establecido.

2.3 Ley de Murphy

Enuncia que, si algo puede salir mal, lo hará.

Los imprevistos son inevitables y surgen en la realización de las actividades, pudiendo afectar a la programación, por lo tanto, es importante comenzar a trabajar con suficiente antelación para disponer del tiempo necesario de gestionar los imprevistos que puedan surgir.

Estos retrasos suponen un coste adicional, y la necesidad de aumentar los recursos, por lo que las consecuencias para el proyecto son graves. Incluso si



hay relaciones de precedencia entre actividades, la aparición de retrasos puede tener impacto en la realización del resto de estas.

El método de la cadena crítica pretende solucionar estos problemas definiendo un plazo mínimo de tiempo en el que la actividad debería desarrollarse, forzando a los recursos a trabajar de manera más intensa en un tiempo más corto del que necesitaban a priori. Así se eliminan las protecciones implícitas añadidas a cada actividad, y se evitan los retrasos que podrían poner en peligro la fecha de finalización del proyecto comprometida.

3. *Buffers*

Los *buffers* o amortiguadores se utilizan para mantener el progreso del proyecto ante la aparición de problemas e imprevistos, protegiendo la fecha programada de finalización. Dichos problemas pueden ocasionarse por la interacción con otros proyectos, entre el personal, o a factores externos a la organización.

Se trata de amortiguadores temporales que se insertan en lugares específicos del proyecto, actuando como actividades ficticias que se van consumiendo automáticamente cuando alguna de las actividades requiere más tiempo del que se había programado, es decir, absorben esos retrasos sin absorber a la fecha de finalización programada.

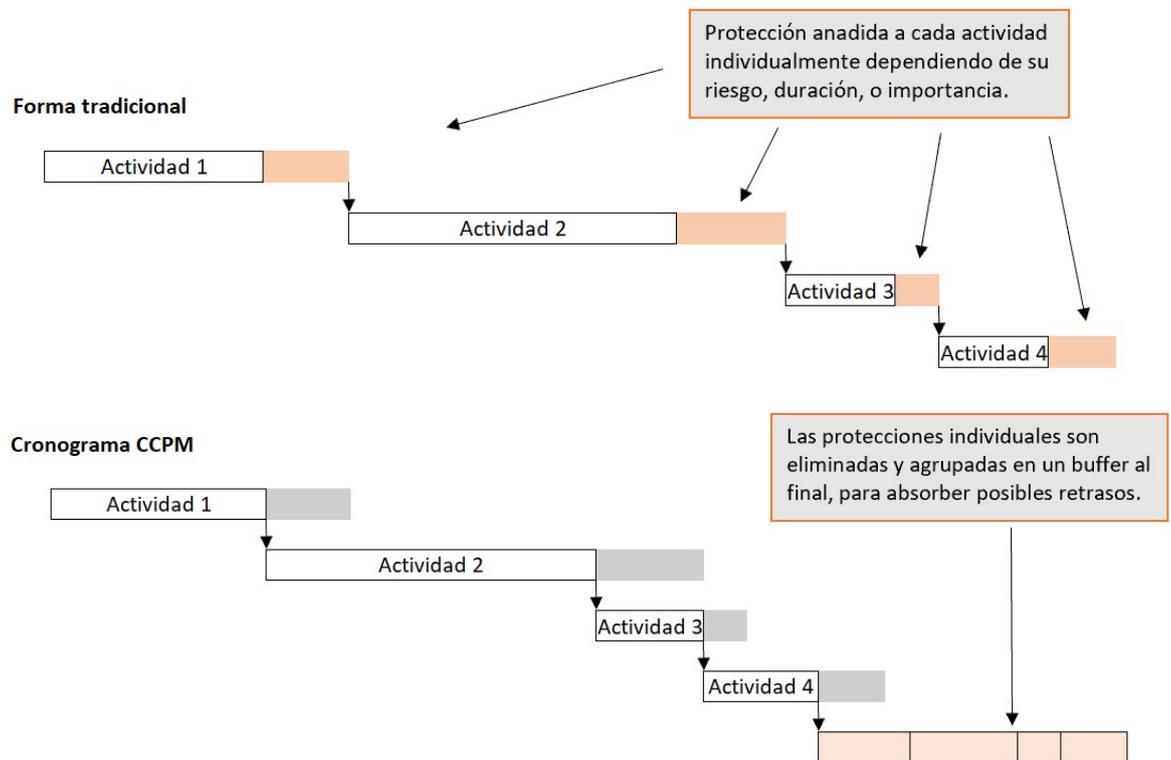


Ilustración 21: Forma tradicional VS. CCPM. Fuente: Elaboración propia.

En la Ilustración 21 se compara gráficamente un mismo proyecto programado primero añadiendo protección individual a cada actividad, de la forma tradicional (parte superior), donde se ve reflejado que, si alguna de las tareas llevara menos tiempo del programado, ese ahorro sería perdido porque la siguiente actividad está programada para comenzar teniendo en cuenta ese consumo de tiempo en la protección. Y en la parte inferior se ve el mismo proyecto programado usando la metodología CCPM, añadiendo solamente un *buffer* al final de la cadena, posibilitando que el proyecto termine antes.

Queda reflejada la importancia del control de los recursos, eliminando la multitarea y dando importancia a la cadena crítica, o añadiendo *buffers* para protegerla. Los proyectos que usan esta metodología reportan un mejor rendimiento, mayor productividad, y menos costos generados por imprevistos.



3.1 Tipos de *buffers*

En función de la localización de los *buffers* en la secuencia de actividades, se distinguen tres tipos, *buffer* de proyecto, alimentación y recursos.

3.1.1 *Buffers de proyecto*

El *buffer* de proyecto se inserta entre la última actividad de la ruta crítica y la fecha de finalización del proyecto, para absorber las perturbaciones que puedan surgir a lo largo de la cadena crítica (Mulvenna, 2019).

El tamaño de este depende de la probabilidad deseada de completar el proyecto a tiempo, (Taylor, 2017), y variará dependiendo de la longitud de las actividades de la cadena crítica, y por tanto del riesgo acumulado.

Ejemplo

- Téngase un proyecto compuesto de cuatro actividades. La duración de cada una de ellas se ha estimado como se hace normalmente, añadiendo una protección implícita a la duración de esta.
- El problema es que con los márgenes implícitos las actividades se demoran hasta ocupar todo ese margen implícito, por la mencionada Ley de Parkinson, o Síndrome del Estudiante.

En la Ilustración 22 se muestra una representación gráfica del ejemplo que se ha mencionado anteriormente, donde se aprecia la diferencia entre el uso del *buffer* de proyecto (correspondiente con la metodología CCPM), y el uso de las metodologías tradicionales.

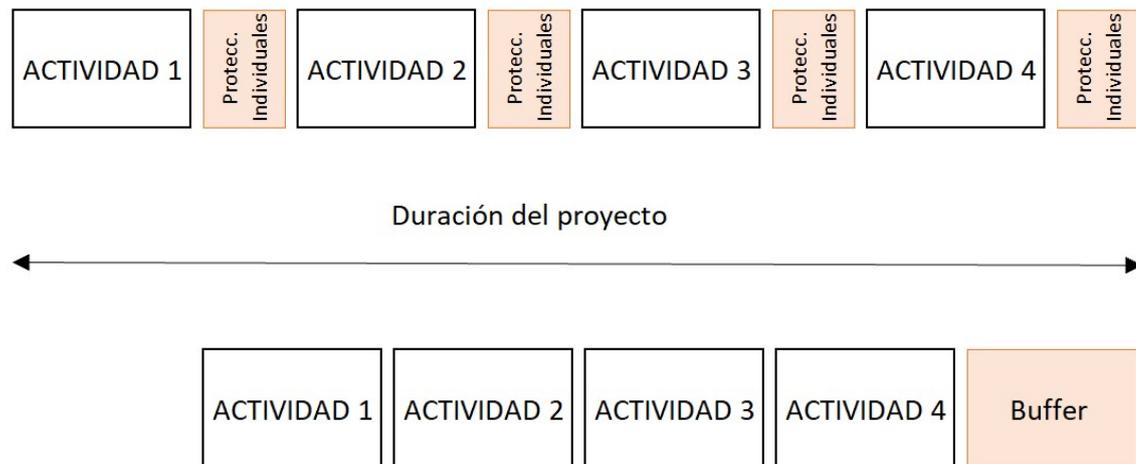


Ilustración 22: Buffer de proyecto comparado con la forma tradicional.

Fuente: Elaboración propia.

La utilización de un *buffer*, eliminando los tiempos de protección a cada actividad de manera individual, e incorporando la suma al proyecto completo (con un tamaño más pequeño que la suma de las protecciones individuales) ayuda a que se cumpla la fecha de finalización del proyecto, siendo ésta, además, una fecha más temprana que la que se conseguía añadiendo las protecciones individuales a cada actividad. La estimación de la duración de las actividades sin el tiempo de protección tiene mayor probabilidad de cumplirse, ya que, de no eliminar esas protecciones individuales, la duración de la de la actividad se va a alargar hasta llenar ese margen individual, de modo que la duración del proyecto acabará siendo, con gran probabilidad, la de la parte superior de la figura. Sin embargo, al quitar esas protecciones y agruparlas en un *buffer* de proyecto, las desviaciones serán absorbidas por el *buffer*, de modo que en la parte inferior de la figura el proyecto acabará más temprano. Además, al dar un tiempo más ajustado para la realización de las actividades, los mencionados orígenes de los retrasos como el síndrome del Estudiante, o la Ley de Parkinson, no afectarán al *buffer*, y solo las causas inevitables tendrán impacto en el *buffer* y su consumo, por tanto, las probabilidades de que el *buffer* no se consuma, y el proyecto acabe antes, son mucho más elevadas.

En conclusión, el *buffer* de proyecto permite absorber la incertidumbre en la duración de las actividades críticas del proyecto. Como el tamaño de este *buffer* es inferior a la suma de las protecciones de cada actividad individual, aumentamos la probabilidad de cumplir con la fecha de finalización del proyecto.

Calcular el tamaño del amortiguador es una de las complicaciones de esta metodología, ya que Goldratt no dejó indicaciones al respecto, pero este tema será abordado en la sección IV.3.2.

3.1.2 Buffers de alimentación

En la mayoría de los proyectos, existen cadenas de actividades paralelas a la crítica que se fusionan en algún momento. Es importante considerar el impacto que podrían tener estas cadenas anexas en el proyecto, porque si sufrieran algún retraso, también impedirían el progreso de las actividades críticas retrasando por tanto la finalización. Por tanto, los *buffers* de alimentación se insertan donde las cadenas de actividades no críticas se fusionan con la cadena crítica (Taylor, 2017), absorbiendo los potenciales retrasos de estas actividades con el objetivo de evitar que éstos tengan un impacto en la duración del proyecto.

En el ejemplo siguiente, se distingue la cadena crítica formada por las actividades 1-2-3-4-5, y las dos cadenas no críticas A-B-C, y I-II. Queda reflejado gráficamente en la Ilustración 23 el hecho de que, si hubiera un retraso en alguna de las actividades de las cadenas no críticas, afectarían directamente a la ejecución de la actividad 5, que es parte del camino crítico.

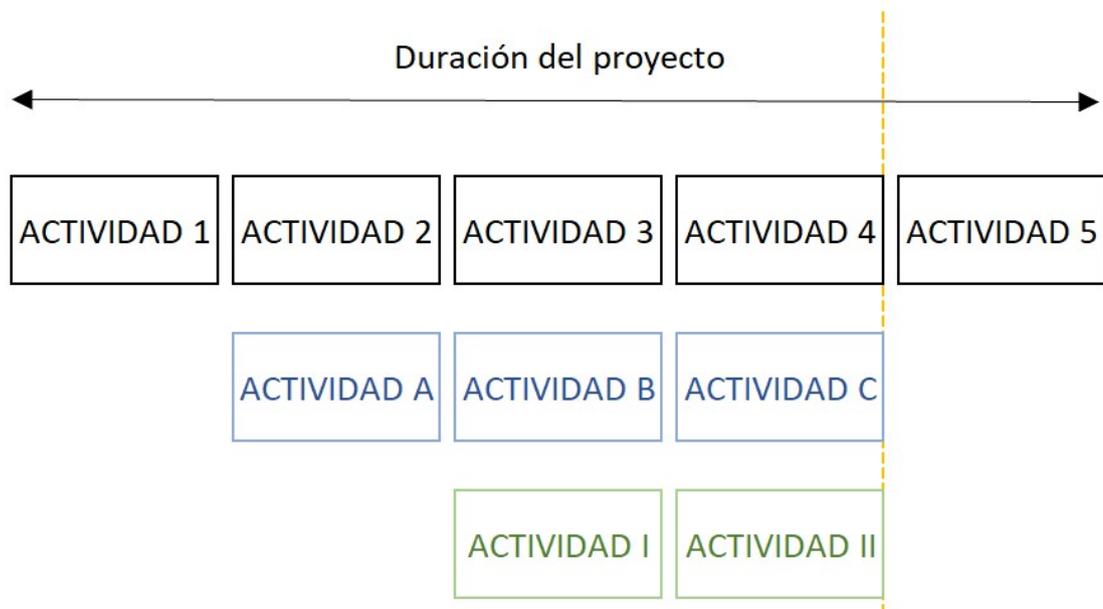


Ilustración 23: Cadenas paralelas a la crítica. Fuente: Elaboración propia.

Con el objetivo de proteger al proyecto de retrasos haremos uso de los *buffers* de alimentación (Ilustración 24) para que absorban la incertidumbre de las actividades de caminos que ‘desembocan’ en el camino crítico, y su tamaño se calculará previamente en función de varios factores como la longitud de la cadena auxiliar, y su riesgo, para asegurar, o aumentar la probabilidad de que estas cadenas lleguen a tiempo a la fusión con la cadena crítica. En la sección IV.3.2 se detallan varias maneras de dimensionarlos.

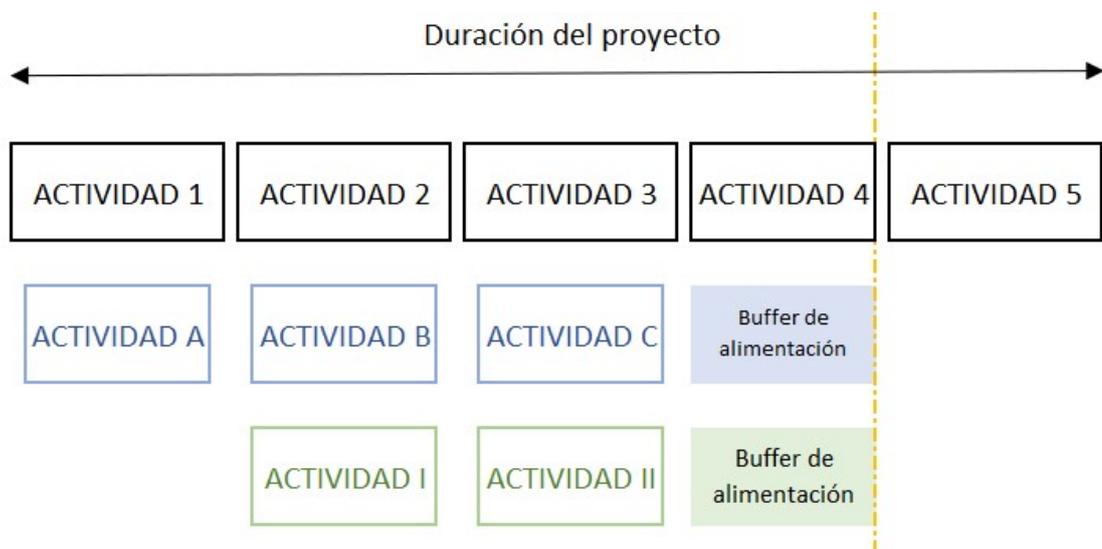


Ilustración 24: Buffers de alimentación. Fuente: Elaboración propia.

3.1.3 Buffers de recurso

Funcionan como señales de advertencia con el objetivo de proteger al proyecto de imprevistos asociados a la disponibilidad de los recursos que forman parte de la ejecución de las actividades críticas, garantizando la disponibilidad de estos recursos cuando vayan a ser necesitados. Son especialmente útiles cuando advierten a los recursos programados para trabajar en la cadena crítica, con antelación, de que son los siguientes en ser necesitados para realizar una actividad de la cadena crítica, aumentando así las probabilidades de que el proyecto avance sin retrasos (Sánchez, 2005).

Es importante notar que son tareas virtuales, por tanto, no modifican la duración del proyecto, ni añaden costes extra.

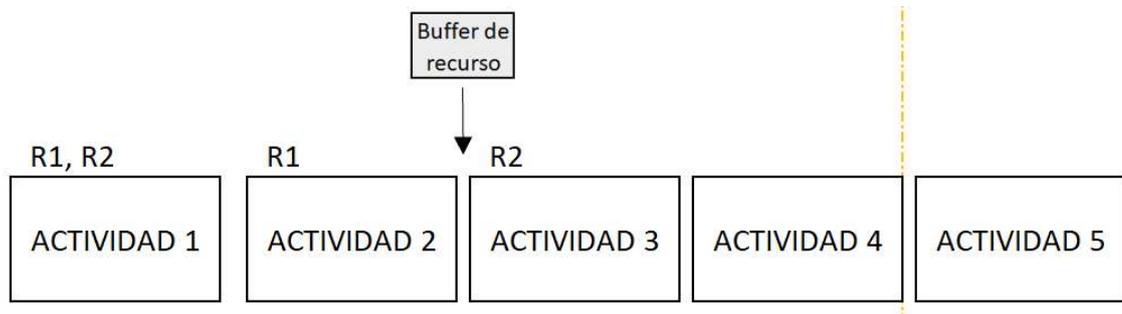


Ilustración 25: Buffers de recurso. Fuente: Elaboración propia.

Continuando con el mismo ejemplo, en la Ilustración 25 se indican los recursos que participan en las primeras actividades de la cadena crítica. Como se puede apreciar, el recurso dos participa en las actividades 1 y 3, pero no participa en la actividad 2. Por esta razón, se introduce un *buffer* de recurso entre las actividades 2 y 3, para advertir al recurso dos, de que la actividad 3, en la que participa, va a comenzar en breve.

3.1.4 Combinación de buffers de proyecto, alimentación, y recurso.

Con los *buffers* de recurso se pretende garantizar el avance del proyecto y que no haya parones que lo impidan debido a la no disponibilidad de los recursos. El *buffer* de proyecto pretende garantizar la fecha de finalización del proyecto global, absorbiendo los posibles retrasos en la ejecución de las tareas críticas del proyecto. Los *buffers* de alimentación protegen el camino crítico de los posibles retrasos en actividades no críticas.

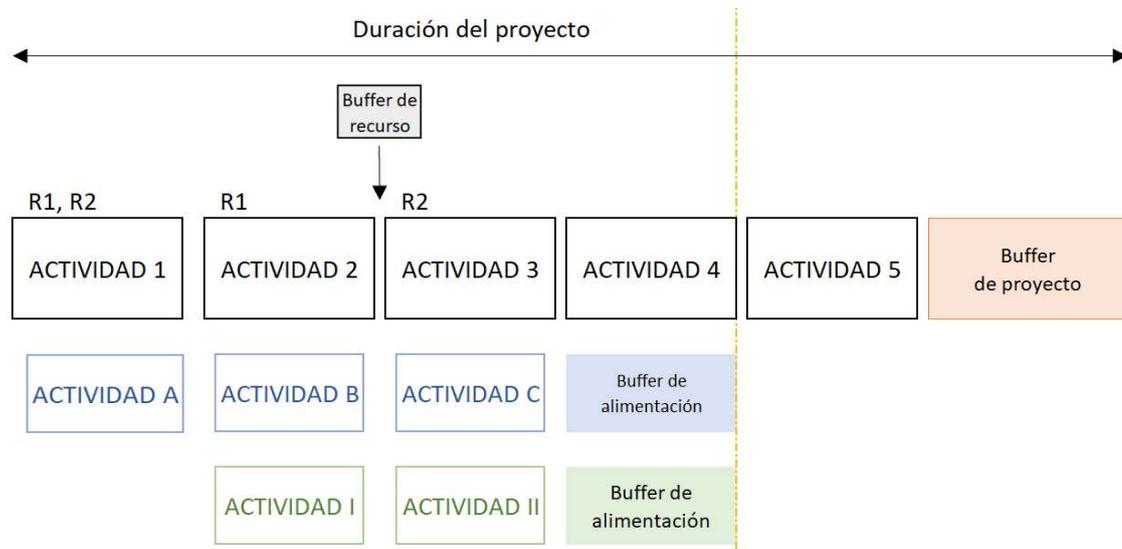


Ilustración 26: Combinación de buffers. Fuente: Elaboración propia.

Las duraciones representadas en la Ilustración 26, se corresponden con las duraciones de las actividades en las que se les ha quitado las protecciones implícitas.

Insertando estos *buffers*, la probabilidad de que el proyecto acabe en la fecha planeada aumenta considerablemente, si los *buffers* de alimentación se consumieran, el impacto en la cadena crítica sería absorbido por el *buffer* de proyecto situado al final de esta, quedando así protegidas tanto la cadena crítica, como las no críticas (Sánchez, 2005).

3.2 Métodos de dimensionado de *buffer* de proyecto

Cuando Goldratt introdujo la metodología CCPM, a pesar de que afirmó la importancia de colocar estos *buffers* en puntos clave específicos del proyecto, no dejó ninguna indicación de cómo se debe determinar su tamaño, que en los últimos años ha sido uno de los mayores desafíos de esta metodología (Taylor, 2017).

El tamaño del *buffer* dependerá de la seguridad que se busque de acabar el proyecto a tiempo, y se determina en función de las características de la ruta crítica, la cantidad de actividades que contiene, su longitud... A lo largo de los años se han propuesto varias técnicas para cumplir con el objetivo de



determinar la duración óptima de los *buffers*. A continuación, explicamos las técnicas más habituales.

3.2.1 La regla del 50% (Cut & Paste Method)

También conocido como el método del "50% de la cadena", se trata de una regla introducida por Goldratt en su libro "Critical Chain" en 1997. Es el método más simple y extendido para dimensionar los *buffers*.

Alega que el amortiguador debe ser tan largo como la mitad de la duración de los tiempos de protección del conjunto de actividades donde se coloca.

Esta regla permite obtener el tamaño de los *buffers* siguiendo los siguientes cuatro pasos:

1. Estimar la duración de la cadena.
2. Separar las protecciones individuales de cada actividad.
3. Mover esos tiempos de protección al final de la cadena.
4. Reducir la duración de los tiempos de protección, al 50%.

Para hacer el cálculo del tamaño del *buffer* (Luz, 2017):

$$Buffer = 0,5 \times \sum_{i=1}^n (dcPi - dsPi)$$

Donde:

- N = número de actividades de la cadena crítica.
- i = actividad.
- dcPi = duración de la actividad i con protección individual.
- dsPi = duración de la actividad i sin protección individual.

La fórmula anterior refleja la diferencia entre las duraciones de las actividades con protección, y las duraciones de las actividades sin protección, de manera que se extrae el sumatorio de todas las protecciones individuales. Una vez hallada esa cantidad, se multiplica por 0,5 para extraer el 50% de la longitud de esa cadena, que se corresponde con la longitud final del *buffer* de proyecto.

En la Ilustración 27, se muestra un ejemplo en el que se utiliza la regla del 50% como método de dimensionamiento del *buffer*. El proyecto está formado por tres actividades, con una duración de 17,5 días cada una, y tal y como se

muestra en la parte superior de la ilustración, la duración real optimista de la actividad son 10 días, y a cada una le han asignado una protección individual de 7,5 días. Aplicando la regla del 50%, se eliminan las protecciones individuales de las actividades, y en su lugar, se coloca un *buffer* al final del proyecto cuyo tamaño es el 50% de la suma de las protecciones individuales eliminadas. En este caso, las protecciones individuales suman: $7,5 + 7,5 + 7,5 = 22,5$ días. Y el 50% de esa cantidad es 11,25 días.

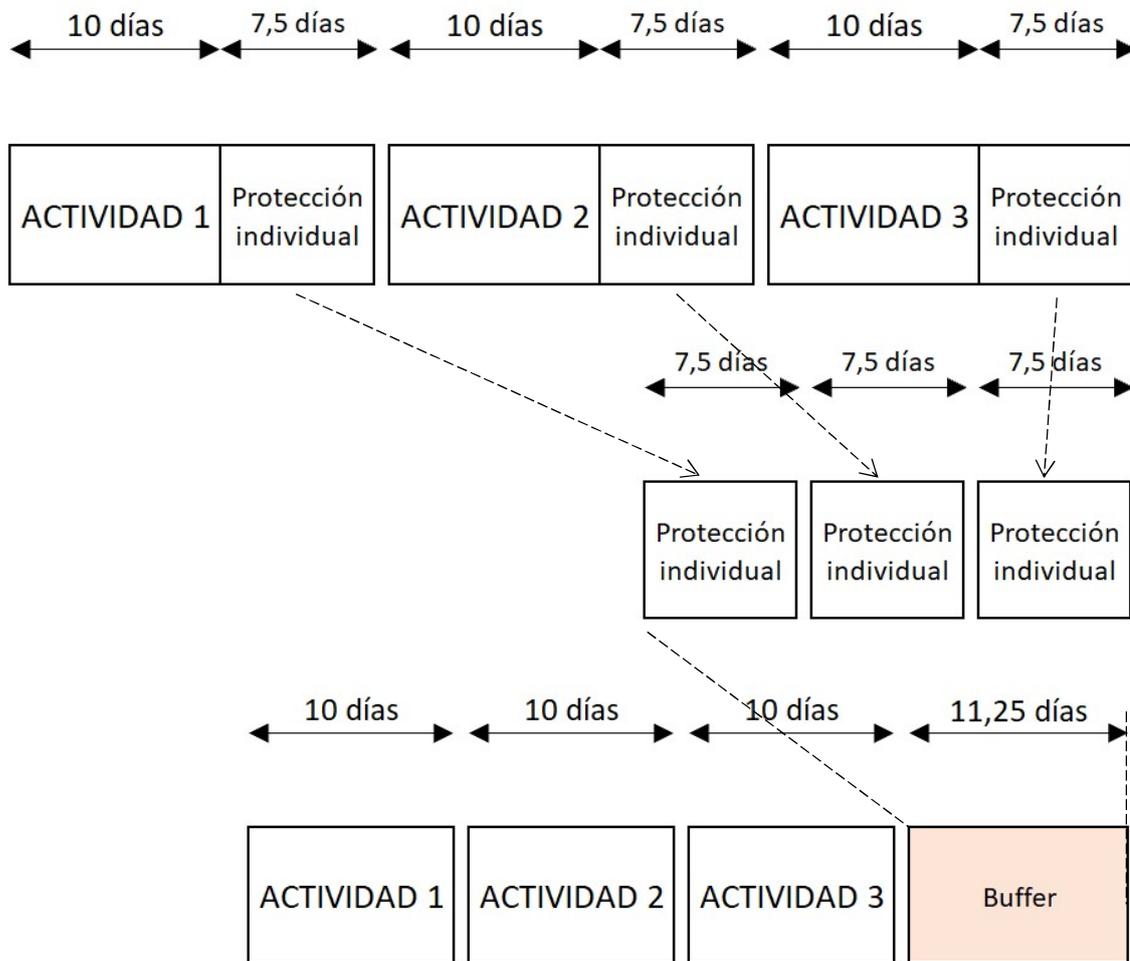


Ilustración 27: Regla del 50%. Fuente: Elaboración propia.

Aplicando un dimensionamiento del *buffer* de proyecto siguiendo este método, se llega a una duración del proyecto de, 41,25 días, resultante de situar un *buffer* de 11,25 días a continuación del camino crítico. Frente a los 52.5 días que se obtenían de la primera manera, y que, con los orígenes de los retrasos



ya estudiados, como son el Síndrome del Estudiante, y la Ley de Parkinson, probablemente acabaría demorándose hasta esa duración.

3.2.2 Método del tercio crítico

Se trata de una variación de la Regla del 50%, con la diferencia de que este método calcula el *buffer* como 1/3 de la longitud total de la cadena, reduciendo, en general, los tiempos de protección en comparación con la regla del 50%.

Los pasos a seguir para el dimensionamiento del *buffer* de proyecto son cuatro:

1. Estimar la duración de la cadena.
2. Eliminar la protección de cada actividad.
3. Calcular la duración de la cadena crítica.
4. Hacer 1/3 de la duración obtenida, e insertarla como *buffer* de proyecto.

$$Buffer = \frac{\sum_{i=1}^n (dsPi)}{3}$$

- N = número de actividades de la cadena crítica.
- i = actividad.
- dsPi = duración de la actividad i sin *buffer*.

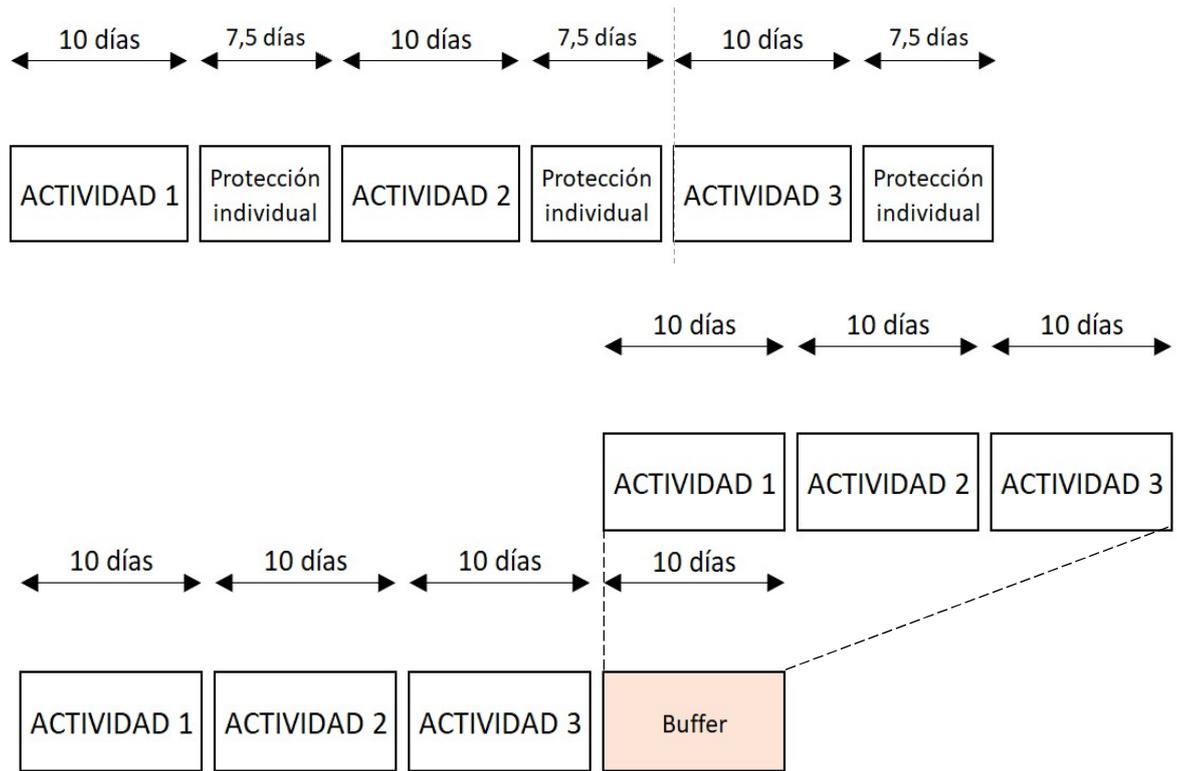


Ilustración 28: Método del tercio crítico. Fuente: Elaboración propia.

En la Ilustración 28, el cálculo del tamaño del *buffer* se hace siguiendo el mencionado método del tercio crítico, y los cálculos son aún más sencillos en este caso. Partimos de las mismas tres actividades con 10 días de duración que teníamos en la Ilustración 27, y siguiendo este método, se eliminan las protecciones individuales, y se añade un *buffer* de longitud $1/3$ la suma de la duración de las actividades que forman la cadena. Es decir, tenemos tres actividades que duran 10 días, en total 30 días, y $1/3$ de esa cantidad son 10 días.

3.2.3 Método de la raíz cuadrada de la suma de cuadrados (SSQ)

El método más complejo de los tres descritos. También conocido como Fórmula Newbold (Newbold, 1998), no sigue una regla empírica simple para el dimensionamiento de los *buffers*, basándose en la variación de la duración de las actividades (D_i).

Calcula el tamaño del *buffer* en función de la variación de la duración de las actividades, calculando dicha variación como la duración de la actividad con protección ($dcPi$), menos la duración de la actividad sin ella ($dsPi$), es decir, las protecciones individuales (Cui & Li, 2012). Y el tamaño del *buffer* será la raíz cuadrada del sumatorio de las protecciones individuales, elevadas al cuadrado (Ashtiani & Jalali, 2007).

$$Buffer = \sqrt{\sum_{i=1}^n (dcPi - dsPi)^2}$$

- N = número de actividades de la cadena crítica.
- i = actividad.
- $dcPi$ = duración de la actividad i con *buffer*.
- $dsPi$ = duración de la actividad i sin *buffer*.

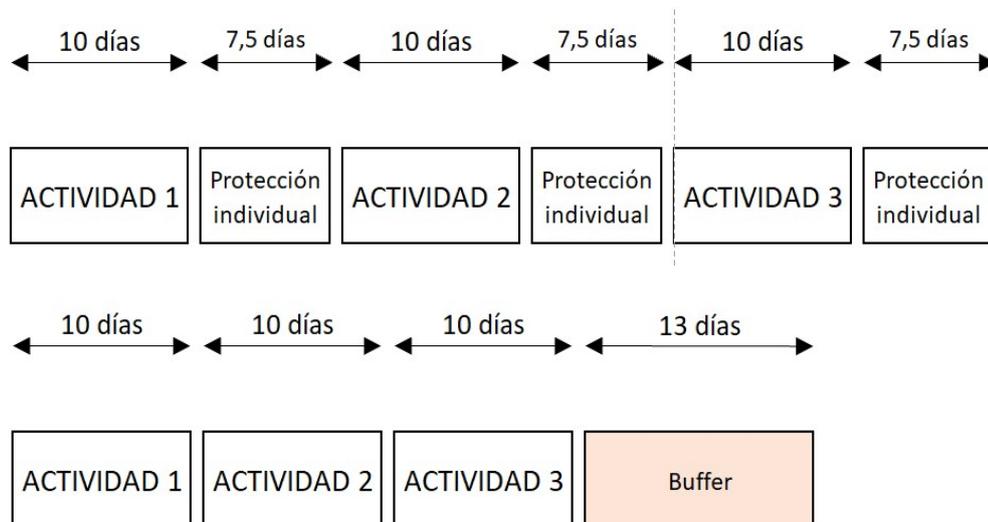


Ilustración 29: Método de la raíz cuadrada de la suma de cuadrados.
Fuente: Elaboración propia.

En la Ilustración 29, se muestra un ejemplo en el que se utiliza el método de la raíz cuadrada de la suma de cuadrados como método de dimensionamiento del *buffer*. El proyecto está formado por las mismas tres actividades, con una duración de 17,5 días cada una, y tal y como se muestra en la parte superior de la ilustración, a cada una le han asignado una protección individual de 7,5

días. Aplicando este método, se halla la variación en la duración de las actividades, como la diferencia entre la duración de la actividad con protección (10 + 7,5 días), y la duración de la actividad sin ella (10 días), para cada actividad. Posteriormente se suman todas las variaciones para cada actividad elevándolas al cuadrado, y se hace la raíz cuadrada. El resultado de esa operación es el tamaño del *buffer* utilizado para sustituir a las protecciones individuales de cada actividad y absorber los imprevistos.

$$Buffer = \sqrt{(17,5 - 10)^2 + (17,5 - 10)^2 + (17,5 - 10)^2} = 13 \text{ días}$$

3.3 Métodos de dimensionado de *buffer* de alimentación

Los *buffers* de alimentación se colocan en el punto donde se fusionan las cadenas de actividades no críticas, con la cadena crítica, para proteger a esta última ya que cualquier retraso en la misma tendría un impacto directo en la fecha de finalización del proyecto. De nuevo, Goldratt no proporciona una metodología para dimensionar estos *buffers* (Taylor, 2017).

Las técnicas más conocidas para el dimensionamiento de los *buffers* de alimentación coinciden con las de dimensionamiento de los *buffers* de proyecto (García, Area, & Rioja, 2017):

1. Método de cortar y pegar.
2. Método del tercio crítico.
3. Método de la raíz cuadrada de la suma de cuadrado.

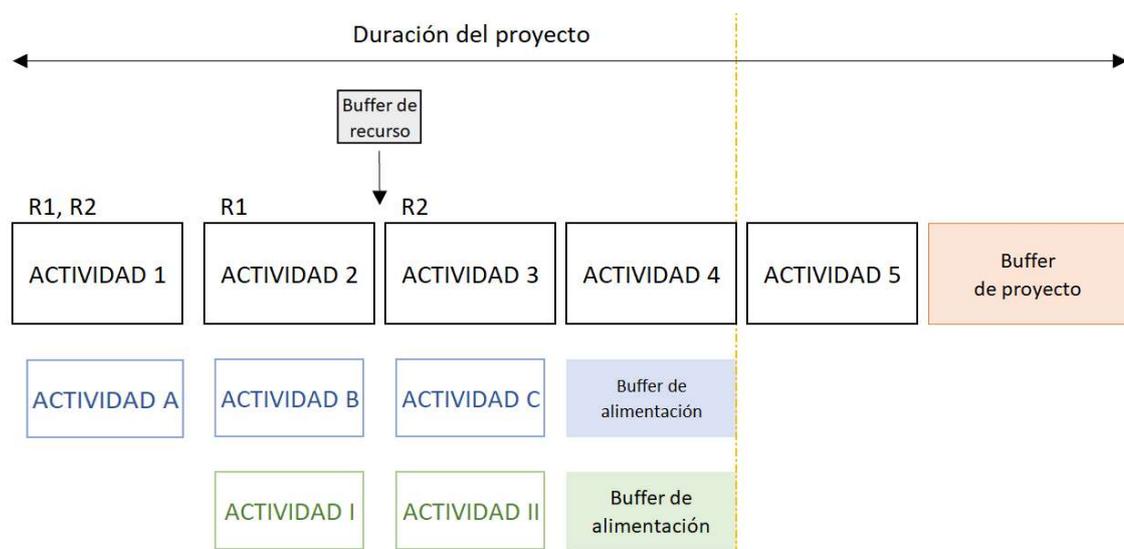


Ilustración 30: Cronograma con buffers incluidos. Fuente: Elaboración propia.



En el dibujo de la Ilustración 30, se observa que existe una cadena crítica (formada por las actividades 1-5). Según se vio en la sección anterior, se inserta un *buffer* de proyecto al final de la cadena crítica para proteger la fecha de finalización del proyecto de los posibles retrasos en la ejecución de las actividades críticas. No obstante, en la figura observamos que, además, existen dos cadenas no críticas (la de color azul, formada por la secuencia de actividades A, B y C), y la cadena no crítica de color verde (formada por la secuencia de actividades I y II). Estas dos cadenas desembocan en el camino crítico del proyecto por lo que un retraso en cualquiera de estas dos cadenas no críticas superior a sus respectivas holguras provocaría un retraso en el comienzo de la actividad 5 y, por tanto, potencialmente también en la fecha de finalización del proyecto. Recordemos que el *buffer* de proyecto, sea cual sea el método de dimensionamiento, se dimensiona teniendo en cuenta solamente las características de las actividades que componen la cadena crítica. Por tanto, para proteger la fecha de finalización del proyecto de los retrasos en las cadenas no críticas debemos insertar un *buffer* de alimentación justo antes de la 'desembocadura' de estos caminos no críticos en el camino crítico.

El dimensionamiento de estos *buffers* de alimentación puede hacerse de cualquiera de las tres maneras estudiadas:

- Regla del 50%: eliminar las protecciones individuales de las actividades, sumarlas, insertar esa cantidad al final de la cadena, y posteriormente reducir esa cantidad a la mitad.
- Método del tercio crítico: estimar la duración de la cadena crítica, sin tener en cuenta la duración añadida debida a las protecciones individuales, y calcular $1/3$ de esa cantidad.
- Método de la raíz cuadrada de la suma de cuadrados: calcular la raíz cuadrada de la suma de cada protección individual elevada al cuadrado.

En la Ilustración 30 se han programado las actividades no críticas para que comiencen lo más tarde posible (programación ALAP, *As Late As Possible*) y se ha insertado un *buffer* de alimentación justo antes de la desembocadura de estas cadenas en la cadena crítica. Una vez insertados estos *buffers* se puede decidir si dejar programadas las cadenas no críticas para que empiecen ALAP junto con el *buffer* de alimentación, o bien programar las cadenas no críticas para que empiecen lo más pronto posible (ASAP, *As Soon As Possible*). Cuando se programan las cadenas no críticas de esta segunda manera, aparecerá cierta holgura en cada una de las cadenas (esta holgura es, de hecho, la característica que hace que estas actividades sean no críticas). Si la holgura que aparece de manera natural al programar ASAP es más grande que el *buffer* calculado, no es necesario añadir un *buffer* de alimentación (pues la protección



espontánea generada por la holgura es superior a la del *buffer* calculado). Sin embargo, si la holgura fuera más pequeña que el tamaño del *buffer* calculado sería una señal de advertencia para que la cadena no crítica comenzara antes de la fecha considerada inicialmente.



Universidad de Valladolid



ESCUELA DE INGENIERÍAS
INDUSTRIALES



V. CASO PRÁCTICO: DIMENSIONAMIENTO DE *BUFFERS* UTILIZANDO SIMULACIÓN DE MONTECARLO FRENTE A LOS MÉTODOS TRADICIONALES

En entornos de incertidumbre, las duraciones de las actividades de un proyecto, se modelan como un número aleatorio que sigue determinada función de distribución, en función de las características de la actividad y de la certeza en cuanto a sus posibles duraciones. Por ejemplo, la función Triangular, BetaPert, Uniforme... En este caso, el dimensionamiento de los *buffers* usando técnicas más tradicionales descritas en el capítulo anterior, a pesar de su sencillez de aplicación, dan lugar a tamaños de *buffer* que no tienen en cuenta el impacto asociado a la incertidumbre en la duración de las actividades del proyecto (Vanhoucke, 2012). Como consecuencia, los tamaños de *buffer* obtenidos no son siempre consecuentes con la variabilidad esperada de las actividades. Como alternativa, podemos dimensionar los *buffers* utilizando simulación de Montecarlo. A pesar de que el procedimiento es más complejo, como se vio en la sección III.2.2, la simulación de Montecarlo sí tiene en cuenta el impacto de toda la incertidumbre en cuanto a la duración de las actividades en la duración del proyecto.

Por este motivo, el objetivo de este caso práctico es realizar el dimensionamiento de los *buffers* de alimentación y de proyecto en dos escenarios distintos. En el primer escenario, utilizando las técnicas de dimensionamiento tradicionales estudiadas, como la Regla del 50%, el Método del tercio crítico, y el Método de la raíz cuadrada de la suma de cuadrados. Frente al segundo escenario, donde se dimensionarán los *buffers* utilizando simulación de Montecarlo. Después, se compararán los resultados obtenidos, y se extraerán las correspondientes conclusiones.

1. Definición del proyecto

A continuación, se presenta el proyecto cuyas actividades van a ser objeto de estudio en este caso práctico.

La realización de dicho proyecto conlleva la ejecución de siete actividades, como se muestra a continuación en la Tabla 5:

Actividad	Precedente
A	-
B	A
C	A
D	A
E	B
F	C, E
G	D, F

Tabla 5: Actividades y sus precedentes. Fuente: Elaboración propia.

En primer lugar, se realiza la actividad A, posteriormente se ejecutan en paralelo las actividades B, C, y D. Una vez ejecutada la actividad B, se ejecuta E, y cuando E ha terminado, y suponiendo que también C ha finalizado, comienza la actividad F. Por último, tras finalizar F y contando con que D finalizó también, se ejecuta la actividad G. Estas relaciones de precedencia se aprecian en el grafo que se muestra a continuación.

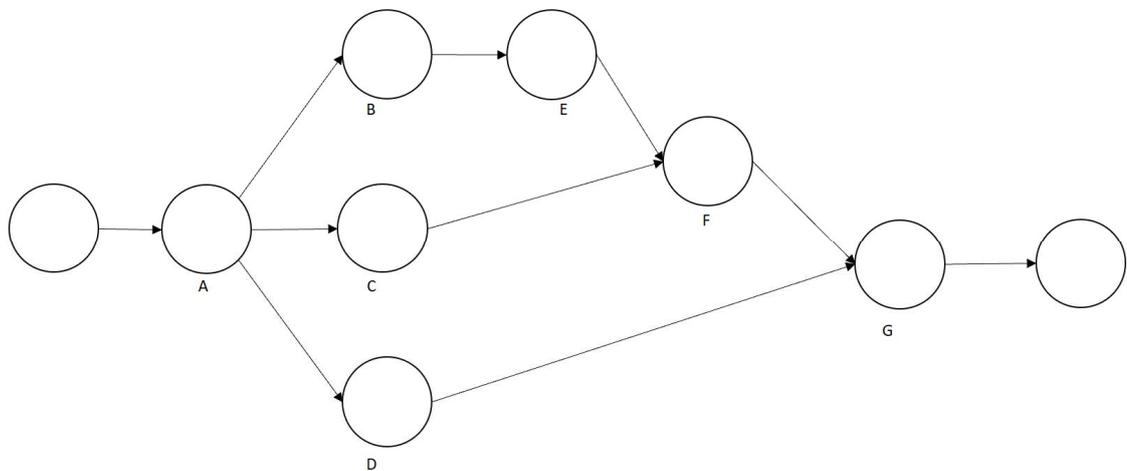


Ilustración 31: Grafo del proyecto. Fuente: Elaboración propia.

Las duraciones de cada actividad están sujetas a incertidumbre y se definen por diversos parámetros, ajustándose en cada caso a una función de distribución diferente, que es detallada a continuación en la Tabla 6.

Actividad	Precedente	Función de distribución de probabilidad	Parámetros	Optimista	Más probable	Pesimista	μ
A	-	Triangular	(min, +prob, máx)	8	10	13	10,33
B	A	BetaPert	(min, +prob, máx)	4	5	6	5
C	A	Uniforme	(min, máx)	10		13	11,50
D	A	BetaPert	(min, +prob, máx)	2	3,5	4	3,33
E	B	Triangular	(min, +prob, máx)	3	4,5	6	4,5
F	C, E	BetaPert	(min, +prob, máx)	4,5	6	7,5	6
G	D, F	Uniforme	(min, máx)	4		6	5

Tabla 6: Duraciones de las actividades. Fuente: Elaboración propia.

Según (Vanhoucke, 2012), en condiciones de incertidumbre, la duración que debe considerarse para cada actividad sin la protección implícita es aquella que tiene el 50% de probabilidades de cumplirse (Es decir, su percentil 50, $D_{50\%}$). Por otro lado, la duración que debe considerarse para la actividad con la protección implícita es aquella que tiene un 90% de probabilidades de cumplirse (es decir, su percentil 90, $D_{90\%}$). Queda reflejado en la Ilustración 32.

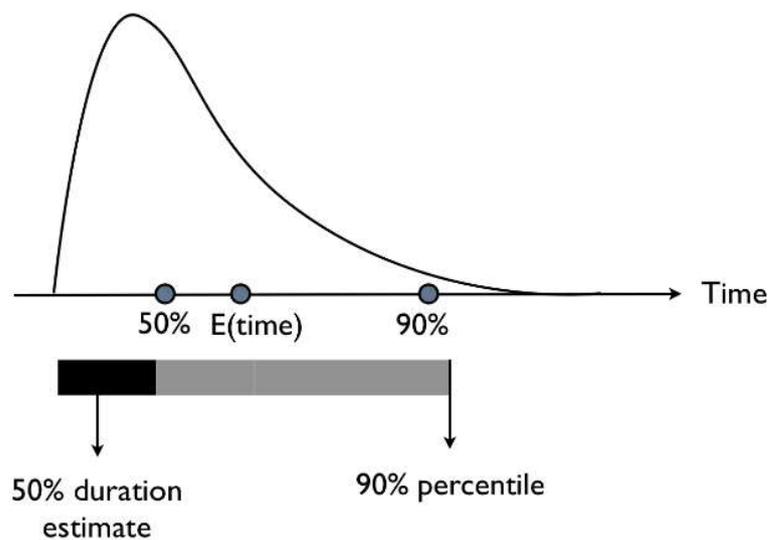


Ilustración 32: Percentiles 50, y 90. Fuente: (Vanhoucke, 2012)

Por tanto, consideraremos que la duración de las protecciones individuales de cada actividad se corresponde con la diferencia entre la duración con un percentil 90 y la duración con un percentil 50, ($D_{90\%} - D_{50\%}$). La Tabla 7

muestra, para cada actividad del proyecto, su duración con la protección implícita ($D_{90\%}$), su duración sin la protección implícita ($D_{50\%}$) y las protecciones individuales ($D_{90\%} - D_{50\%}$).

Actividad	Función de distribución de probabilidad	Duración sin la protección implícita $D_{50\%}$ (mediana)	Duración con la protección implícita $D_{90\%}$	Protección implícita $D_{90\%} - D_{50\%}$
A	Triangular	10,26	11,78	1,52
B	BetaPert	5	5,51	0,51
C	Uniforme	11,50	12,70	1,20
D	BetaPert	3,37	3,78	0,41
E	Triangular	4,5	5,33	0,83
F	BetaPert	5,94	6,87	0,93
G	Uniforme	5	5,8	0,8

Tabla 7: Protecciones individuales. Fuente: Elaboración propia.

En el grafo representado en la Ilustración 33, se muestra el cronograma del proyecto realizado teniendo en cuenta duraciones de las actividades según el percentil 50, es decir, cuando se considera la duración de las actividades una vez eliminados los márgenes de protección individuales.

De esta manera, el proyecto podría finalizar al cabo de 32,7 días. Esa duración es la correspondiente con la duración de las actividades que forman el camino crítico, que está formado por las actividades A-C-F-G, tal y como se aprecia en la Ilustración 33. Además, también se identifican dos caminos no-críticos: el formado por las actividades B-E, y el formado por la actividad D, que se fusionan con el camino crítico.

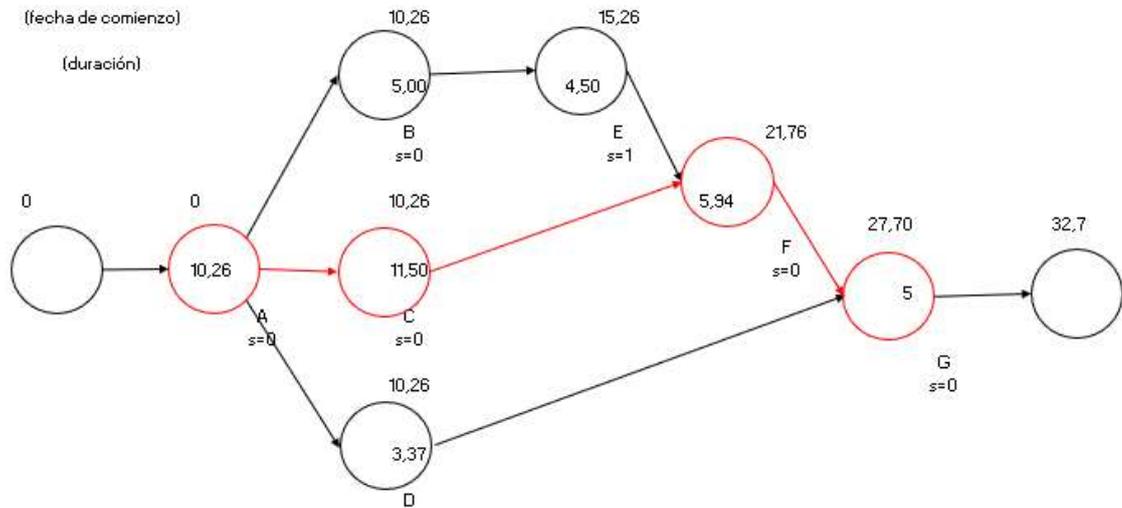


Ilustración 33: Grafo del proyecto programando actividades sin las protecciones individuales (percentil 50). Fuente: Elaboración propia.

La duración del proyecto, calculada como la suma de las duraciones de las actividades del camino crítico:

$$\text{Duración del proyecto} = A + C + F + G = 10,26 + 11,5 + 5,94 + 5 = 32,7 \text{ días}$$

Por otro lado, en el grafo de la Ilustración 34, se muestra el cronograma cuando se realiza estimando las duraciones de las actividades, más un tiempo de protección, para proteger la fecha de comienzo programada de las actividades siguientes, es decir, considerando las duraciones pesimistas de las actividades. En este caso, la duración del proyecto aumenta, ya que la duración de las actividades que forman el camino crítico se alarga. En el proyecto, programando de esta manera, el proyecto se alarga hasta durar 37,15 días.

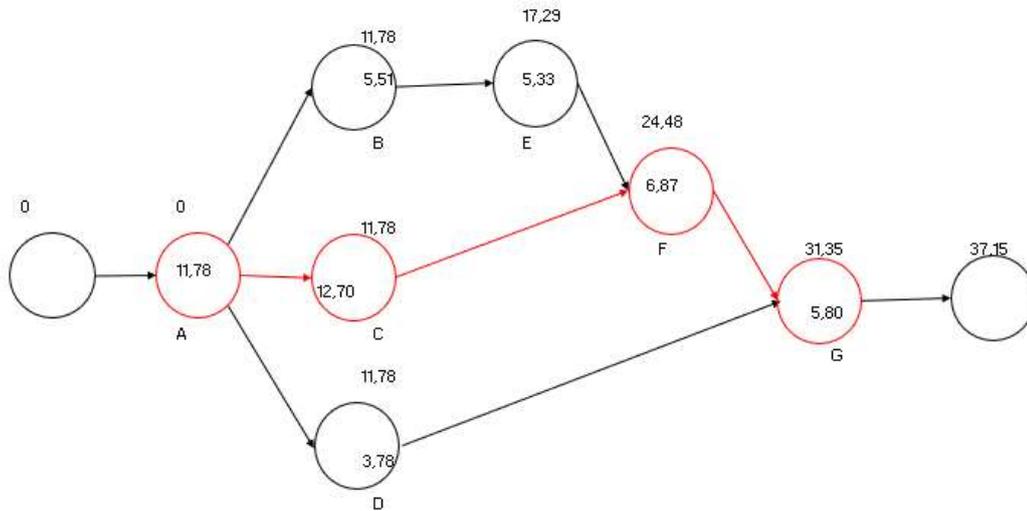


Ilustración 34: Grafo del proyecto programando actividades con las protecciones individuales (percentil 90). Fuente: Elaboración propia.

La duración del proyecto añadiendo protecciones de manera que se tenga 90% de probabilidades de que se cumpla la estimación, es:

$$\begin{aligned} \text{Duración del proyecto} &= A + C + F + G = 11,78 + 12,7 + 6,87 + 5,8 = \\ &= 37,15 \text{ días} \end{aligned}$$

Comparando ambas maneras de programar el cronograma del proyecto, observamos una diferencia de 4,45 días en la fecha de finalización del proyecto. Por un lado, la programación teniendo en cuenta las duraciones según el percentil 50 (es decir, la estimación de lo que debería tardarse en realizar dicha actividad sin tener en cuenta la protección), donde la duración del proyecto sería de 32,7 días; y, por otro lado, la programación del cronograma incluyendo una protección individual para cada actividad, donde el proyecto se alarga hasta 37,15 días.

Esos 4,45 días de diferencia se corresponden con la suma del conjunto de protecciones individuales implícitas que forman parte de las duraciones de las actividades críticas del proyecto.

Siguiendo la metodología de la Cadena Crítica (CCPM, Critical Chain Project Management), el objetivo es eliminar las protecciones individuales de cada actividad para mitigar los efectos de la ley de Parkinson, síndrome del estudiante, etc. en la duración de la ejecución real de las actividades. Según esta metodología, cuando los recursos conocen el tiempo del que disponen



para realizar la actividad, y ese tiempo es superior al necesario para realizarla, la duración se va a alargar hasta cubrir la cantidad de tiempo que le ha sido asignada. Es decir, el recurso es capaz de ejecutar la actividad en el tiempo considerado optimista, pero cuando se programa el cronograma con estimaciones en las que se incluye un tiempo extra, de protección, el recurso tenderá a alargar la realización de la actividad hasta ocupar dicho tiempo. Esto es así, porque frecuentemente se dan los casos de procrastinación de recursos definidos por el Síndrome del Estudiante, o la Ley de Parkinson según la cual el trabajo para realizar una actividad se expande hasta abarcar todo el tiempo que ha sido asignado para ello, o la aparición de imprevistos definidos por la Ley de Murphy.

Por tanto, se trata de considerar para cada actividad del proyecto, su duración sin las protecciones implícitas (es decir, aquel que tiene un 50% de probabilidades de cumplirse, en otras palabras, su duración según el percentil 50), y añadir un *buffer* al final de la cadena crítica del proyecto, que tenga en cuenta esos tiempos de protección individuales que han sido eliminados de las actividades.

Este *buffer* de proyecto, situado al final de la cadena crítica, tendría la misión de absorber las variaciones que se puedan ocasionar en las actividades de la cadena crítica, y lo que se persigue es obtener un *buffer* más corto que la suma de las protecciones asignadas a las actividades individualmente. El mismo procedimiento se haría en las cadenas no-críticas, insertando dos *buffers* de alimentación en la fusión entre los dos caminos no críticos mencionados, y la cadena crítica.

Volviendo a nuestro caso de estudio, se pueden convertir esos 4,45 días destinados a proteger individualmente a cada actividad, para implantar un *buffer* de proyecto, y dar a los recursos un plazo más ajustado para la realización de las actividades, intentando disminuir el efecto de los mencionados Síndrome del estudiante o Ley de Parkinson.

2. Aplicación de la Metodología de la Cadena Crítica.

Como se ha indicado anteriormente, la forma de calcular las protecciones individuales añadidas a cada actividad es hallar la diferencia entre las duraciones de las actividades según el percentil 90 (cuya estimación incluye un tiempo de protección), y las duraciones según el percentil 50 (que se corresponde con las duraciones que tendrían un 50% de probabilidad de



cumplirse) (Vanhoucke, 2012). Esas diferencias se aprecian en la Tabla 7 mostrada anteriormente.

Una vez expuesto el método para obtener la duración de las protecciones individuales, se procede al cálculo de los *buffers* de proyecto y de alimentación.

3. Escenario 1: dimensionamiento de los *buffers* según los métodos tradicionales.

Para el primer escenario, se van a calcular utilizando los tres métodos tradicionales explicados en la sección IV.3.2, y se citan brevemente a continuación.

3.1 Regla del 50%

$$\begin{aligned} \text{Buffer} &= \\ &= 0,5 \times \sum_{i=1}^n (\text{Duración activ. con protección} - \text{Duración activ. sin protección}) \end{aligned}$$

3.2 Método del tercio crítico

$$\text{Buffer} = \frac{\sum_{i=1}^n (\text{Duración actividad sin protección})}{3}$$

3.3 Método de la raíz cuadrada de la suma de cuadrados (SSQ)

$$\begin{aligned} \text{Buffer} &= \\ &= \sqrt{\sum_{i=1}^n (\text{Duración activ. con protección} - \text{Duración activ. sin protección})^2} \end{aligned}$$



3.4 Cálculo de *buffers*.

Estos tres métodos van a ser utilizados para hallar los tres *buffers* que hay que insertar en este proyecto (es decir, el *buffer* de proyecto y los dos *buffers* de alimentación).

A. *Buffer de proyecto*

Se insertará al final del camino crítico formado por A-C-F-G.

1) Regla del 50%

Se toman los datos de la Tabla 7, donde se muestran las actividades del proyecto, la duración de estas incluyendo protección, su duración sin protección, y la protección extraída de cada una de ellas.

Las protecciones de las actividades del camino crítico son:

Actividad	Protección individual = D _{90%} - D _{50%}
Actividad A	1,52
Actividad C	1,20
Actividad F	0,93
Actividad G	0,80

Tabla 8: Regla del 50% al *buffer* de proyecto. Fuente: Elaboración propia.

Aplicando la regla del 50%, el tamaño del *buffer* quedaría dimensionado de la siguiente manera.

$$\begin{aligned}
 \text{Buffer} &= \\
 &= 0,5 \times \sum_{i=1}^n (\text{Duración activ. con protección} - \text{Duración activ. sin protección}) = \\
 &= 0,5 \times (1,52 + 1,2 + 0,93 + 0,8) = 2,225 \text{ días}
 \end{aligned}$$



2) Método del tercio crítico

Las duraciones de las actividades del camino crítico, sin la protección añadida, son:

Actividad	D _{50%}
Actividad A	10,26
Actividad C	11,5
Actividad F	5,94
Actividad G	5

Tabla 9: Método del tercio crítico al buffer de proyecto. Fuente: Elaboración propia.

Aplicando la regla del tercio crítico, el tamaño del *buffer* quedaría dimensionado de la siguiente manera.

$$Buffer = \frac{\sum_{i=1}^n (Duración\ activ.\ sin\ protección)}{3} = \frac{10,26 + 11,5 + 5,94 + 5}{3} = 10,9\ días$$

3) Método de la raíz cuadrada de la suma de cuadrados (SSQ)

Para este método, de nuevo necesitamos las protecciones individuales de cada actividad que forma parte del camino crítico, y se muestran en la Tabla 10.

Actividad	Protección individual = D _{90%} - D _{50%}
Actividad A	1,52
Actividad C	1,20
Actividad F	0,93
Actividad G	0,80

Tabla 10: Método de la raíz cuadrada de la suma de cuadrados (SSQ) al buffer de proyecto. Fuente: Elaboración propia.

Aplicando el método de la raíz cuadrada de la suma de cuadrados, el tamaño del *buffer* quedaría dimensionado de la siguiente manera.

Buffer =

$$= \sqrt{\sum_{i=1}^n (Duración\ activ.\ con\ protección - Duración\ activ.\ sin\ protección)^2} =$$
$$= \sqrt{(1,52)^2 + (1,2)^2 + (0,93)^2 + (0,8)^2} = 2,29\ días$$

B. Buffer de alimentación 1

El primer *buffer* de alimentación se insertará al final del camino no-crítico formado por B-E, antes de que se fusione con la actividad F del camino crítico.

1) Regla del 50%

Se toman los datos de la Tabla 7, donde se muestran las actividades del proyecto, la duración de estas incluyendo protección, su duración sin protección, y la protección extraída de cada una de ellas.

Las protecciones de las actividades del camino no-crítico B-E son:

Actividad	Protección individual = $D_{90\%} - D_{50\%}$
Actividad B	0,51
Actividad E	0,83

Tabla 11: Regla del 50% al buffer de alimentación 1. Fuente: Elaboración propia.

Aplicando la regla del 50%, el tamaño del *buffer* quedaría dimensionado de la siguiente manera.

Buffer =

$$0,5 \times \sum_{i=1}^n (Duración\ actividad\ con\ protección - Duración\ actividad\ sin\ protección)$$
$$= 0,5 \times (0,51 + 0,83) = 0,67\ días$$

2) Método del tercio crítico

Las duraciones de las actividades del camino no-crítico B-E, sin la protección implícita, son:

Actividad	D _{50%}
Actividad B	5
Actividad E	4,5

Tabla 12: Método del tercio crítico al buffer de alimentación 1. Fuente: Elaboración propia.

Aplicando la regla del tercio crítico, el tamaño del *buffer* quedaría dimensionado de la siguiente manera.

$$Buffer = \frac{\sum_{i=1}^n (Duración actividad sin protección)}{3} = \frac{5 + 4,5}{3} = 3,167 \text{ días}$$

3) Método de la raíz cuadrada de la suma de cuadrados (SSQ)

Para este método, de nuevo necesitamos las protecciones individuales de cada actividad que forma parte del camino no-crítico B-E, y se muestran en la Tabla 13.

Actividad	Protección individual = D _{90%} - D _{50%}
Actividad B	0,51
Actividad E	0,83

Tabla 13: Método SSQ al buffer de alimentación 1. Fuente: Elaboración propia.

Aplicando el método de la raíz cuadrada de la suma de cuadrados, el tamaño del *buffer* quedaría dimensionado de la siguiente manera.

$$Buffer =$$

$$\sqrt{\sum_{i=1}^n (Duración actividad con protección - Duración actividad sin protección)^2}$$

$$= \sqrt{(0,51)^2 + (0,83)^2} = 0,97 \text{ días}$$

C. Buffer de alimentación 2

El segundo *buffer* de alimentación, que se insertará al final del camino no-crítico formado por D.

1) Regla del 50%

Se toman los datos de la Tabla 7, donde se muestran las actividades del proyecto, la duración de estas incluyendo protección, su duración sin protección, y la protección extraída de cada una de ellas.

La protección de la actividad D, que forma el segundo camino no-crítico que se fusiona con el camino crítico antes de la realización de la actividad G, es:

Actividad	Protección individual = D _{90%} - D _{50%}
Actividad D	0,41

Tabla 14: Regla del 50% al buffer de alimentación 2. Fuente: Elaboración propia.

Aplicando la regla del 50%, el tamaño del *buffer* quedaría dimensionado de la siguiente manera.

$$Buffer =$$

$$= 0,5 \times \sum_{i=1}^n (Duración acti. con protección - Duración activ. sin protección) =$$

$$= 0,5 \times (0,41) = 0,205 \text{ días}$$

2) Método del tercio crítico

La duración de la actividad D, que forma el segundo camino no-crítico, sin la protección añadida, es:

Actividad	D _{50%}
Actividad D	3,37

Tabla 15: Método del tercio crítico al buffer de alimentación 2. Fuente: Elaboración propia.

Aplicando la regla del tercio crítico, el tamaño del *buffer* quedaría dimensionado de la siguiente manera.

$$\text{Buffer} = \frac{\sum_{i=1}^n (\text{Duración actividad sin protección})}{3} = \frac{3,37}{3} = 1,12 \text{ días}$$

3) Método de la raíz cuadrada de la suma de cuadrados (SSQ)

Para este método, de nuevo necesitamos la protección individual de la actividad que forma parte el segundo camino no-crítico, D, y se muestra en la Tabla 16.

Actividad	Protección individual = D _{90%} - D _{50%}
Actividad D	0,41

Tabla 16: Método SSQ al buffer de alimentación 2. Fuente: Elaboración propia.

Aplicando el método de la raíz cuadrada de la suma de cuadrados, el tamaño del *buffer* quedaría dimensionado de la siguiente manera.

$$\begin{aligned} \text{Buffer} &= \\ &= \sqrt{\sum_{i=1}^n (\text{Duración activ. con protección} - \text{Duración activ. sin protección})^2} \\ &= \\ &= \sqrt{(0,41)^2} = 0,41 \text{ días} \end{aligned}$$

Los resultados de aplicar los tres métodos tradicionales para el dimensionamiento de los *buffers* se resumen en la Tabla 17.

Método	Buffer de proyecto	Buffer de alimentación 1	Buffer de alimentación 2
50%	2,225	0,67	0,205
Tercio crítico	10,9	3,167	1,12
SSQ	2,29	0,97	0,41

Tabla 17: Resumen de buffers en escenario 1. Fuente: Elaboración propia.

Los *buffers* obtenidos se sumarían a las duraciones del proyecto según el grafo que tiene en cuenta las duraciones medias (el de la Ilustración 33, el que tiene una duración de proyecto de 32,7 días).

4. Escenario 2: dimensionamiento de *buffers* con simulación de Montecarlo.

El dimensionamiento de los *buffers* es un factor determinante en la Gestión de Proyectos por Cadena Crítica. Si se dimensionan *buffers* demasiado grandes, se incurre en costos innecesarios, sin embargo, si la duración del *buffer* es insuficiente, también se incurren en sobrecostos para intentar terminar el trabajo a tiempo. Por tanto, y para tratar de disminuir ese riesgo al dimensionar los *buffers*, se recurre a la simulación de Montecarlo, que ya fue propuesta por Hoel y Taylor (1999) (Alexandra B., 2008). La simulación de Montecarlo, tal como se indicó en el capítulo III.2.2, ha sido utilizada durante mucho tiempo para analizar las probabilidades de los proyectos de terminar en la fecha



programada. En este caso, vamos a utilizar la simulación de Montecarlo para determinar los tamaños de los *buffers* del proyecto.

El software utilizado para aplicar la simulación de Montecarlo al proyecto es *Crystal Ball* (Oracle. Crystall Ball, s.f.). El programa permite generar números aleatorios que siguen cierta función de distribución de probabilidad, definida por el usuario, que se empleará para modelar la duración de las actividades del proyecto. Estas son las variables de entrada del programa. De esta manera, *Crystal Ball* permite realizar simulaciones de Montecarlo para obtener, como variable de salida, la función de distribución de la duración del proyecto.

Partiendo del mismo grafo (AON) de la Ilustración 31, se definen las variables de entrada a la simulación, es decir las duraciones para actividad según la función de distribución correspondiente especificando los parámetros descritos en la Tabla 6. A continuación, en la Ilustración 35, se muestra la introducción de dichos parámetros al programa, mediante el uso de *Define Assumption*.

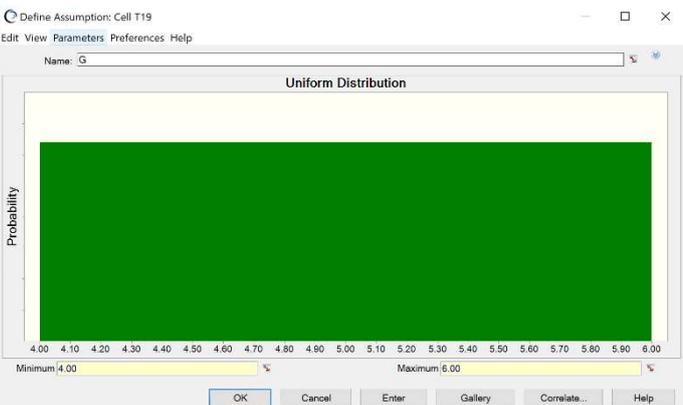
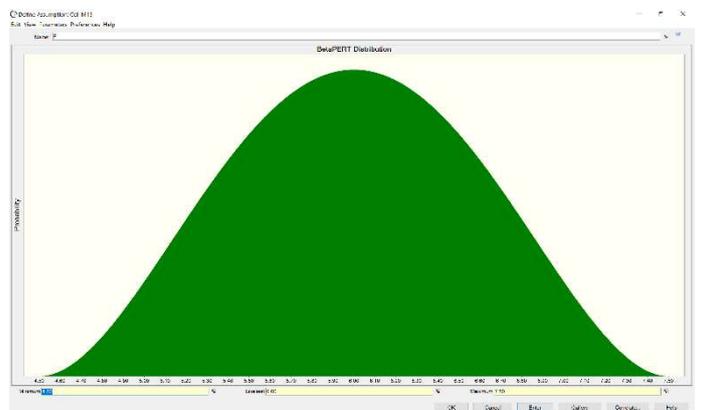
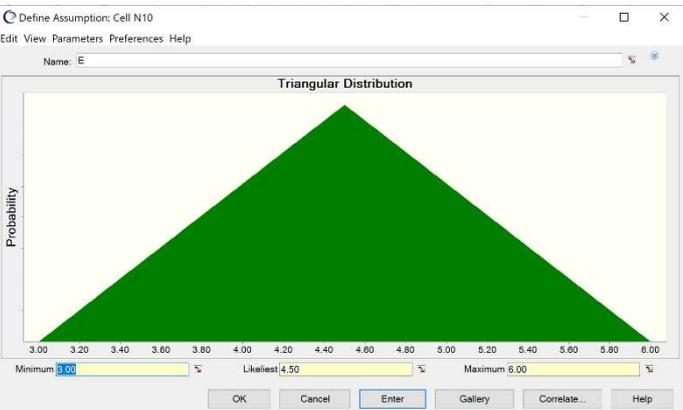
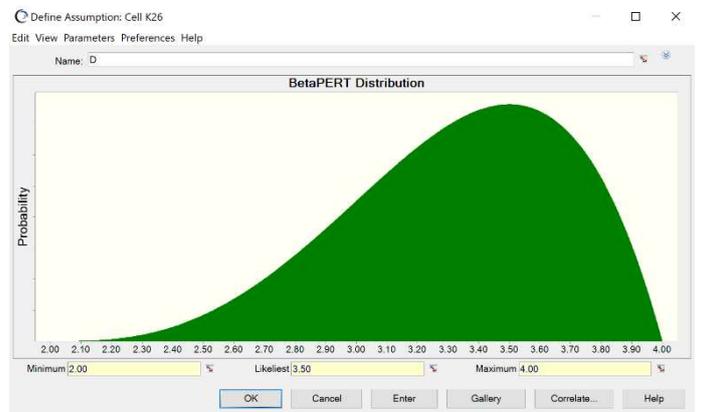
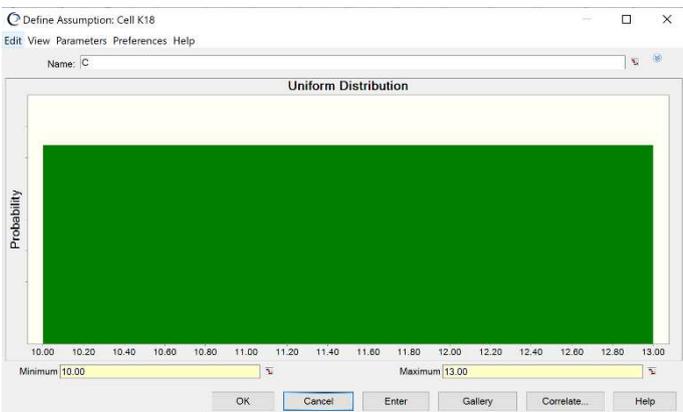
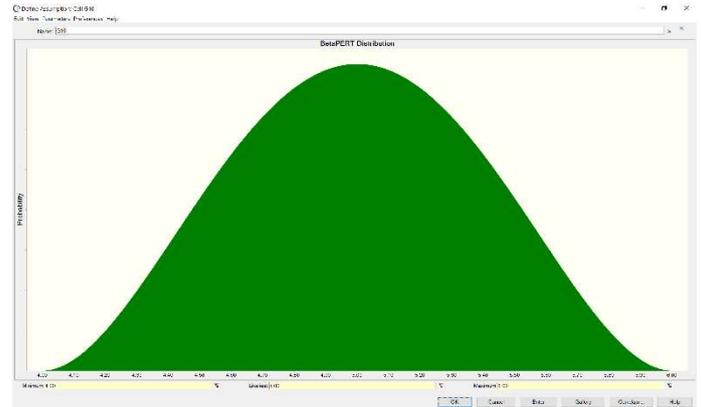
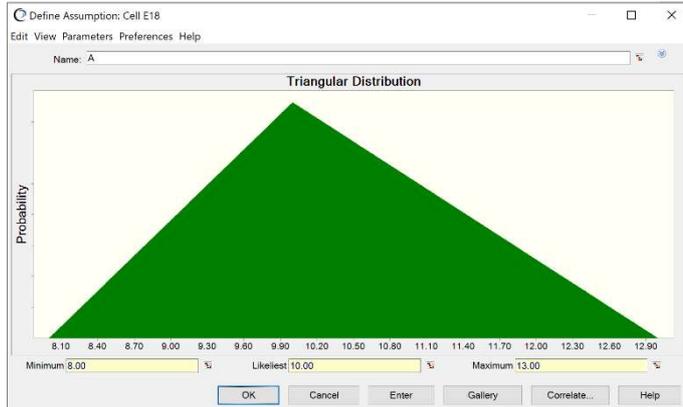


Ilustración 35: Funciones de distribución en Crystal Ball. Fuente: Crystal Ball.

Una vez definidas las variables de entrada, se define las variables de salida, mediante *Define Forecast*. Las variables de salida que se quieren obtener son, en primer lugar, la duración del proyecto, es decir, la duración del camino crítico, para poder usarlo en el cálculo del *buffer* de proyecto que se insertará al final de este.

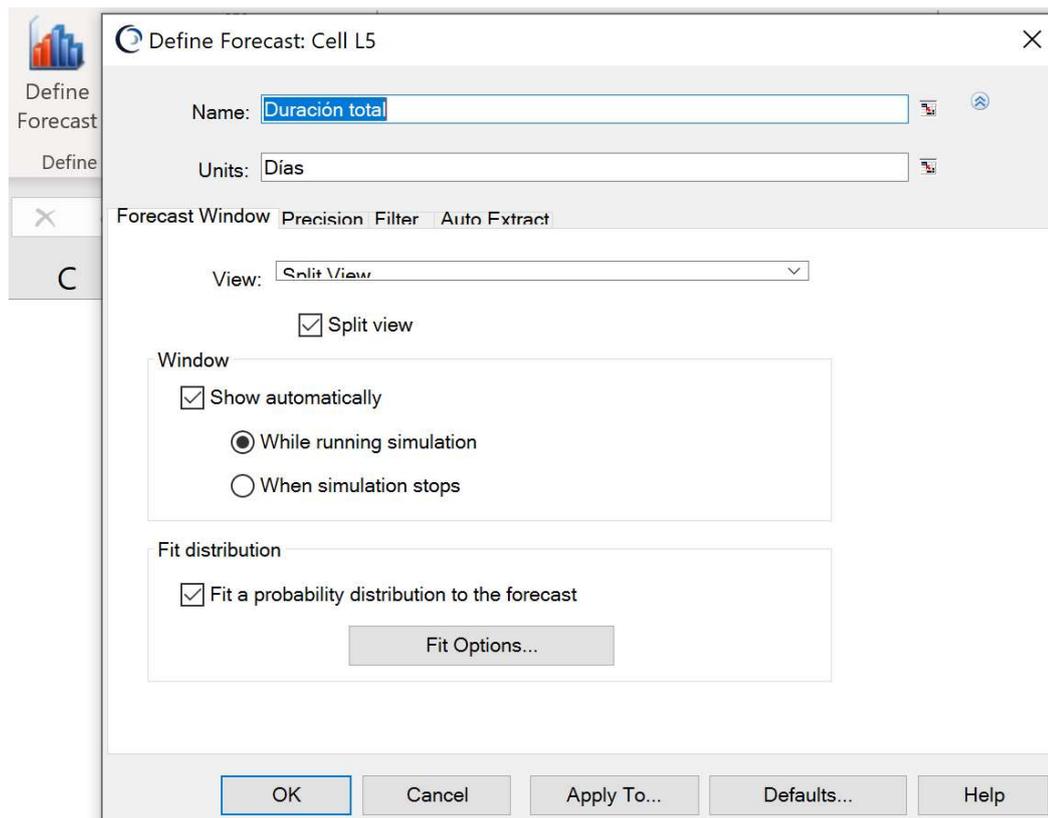


Ilustración 36: Definición de la duración del camino crítico en Crystal Ball. Fuente: Crystal Ball.

En segundo lugar, la duración de la cadena no-crítica BE, para calcular el tamaño del primer *buffer* de alimentación de este proyecto, que se insertará al final de la cadena BE, antes de la fusión con el camino crítico. Por último, la tercera variable de salida buscada es la duración del camino D, es decir, la duración de la actividad D, utilizada para calcular el segundo *buffer* de alimentación, que se insertará en la fusión de D con el camino crítico.

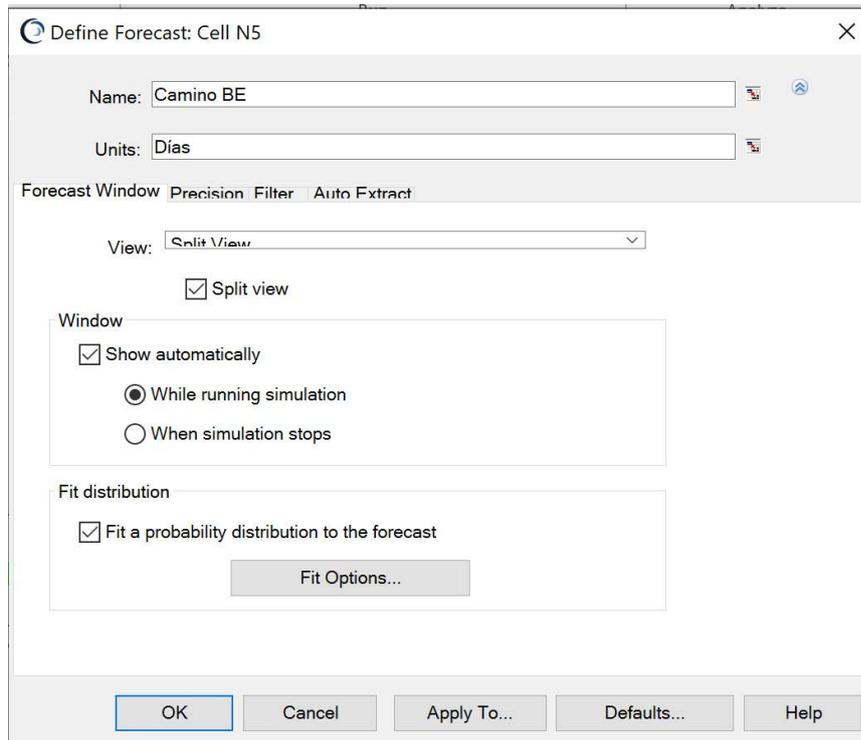


Ilustración 37: Definición de la duración de la cadena no crítica B-E. Fuente: Crystal Ball.

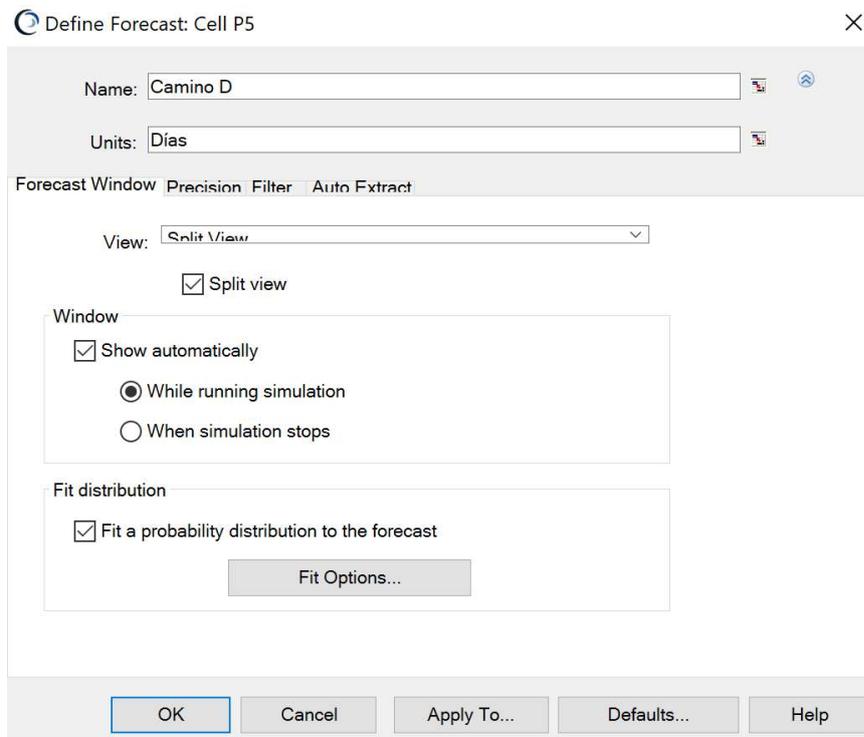


Ilustración 38: Definición de la duración de la cadena no crítica D. Fuente: Crystal Ball.

A. *Buffer* de proyecto.

Utilizando los datos de la Tabla 6 (la primera de todas), se corren 10.000 iteraciones, y los resultados obtenidos se muestran a continuación en la Ilustración 39, donde se aprecia la función de distribución de la duración del proyecto.

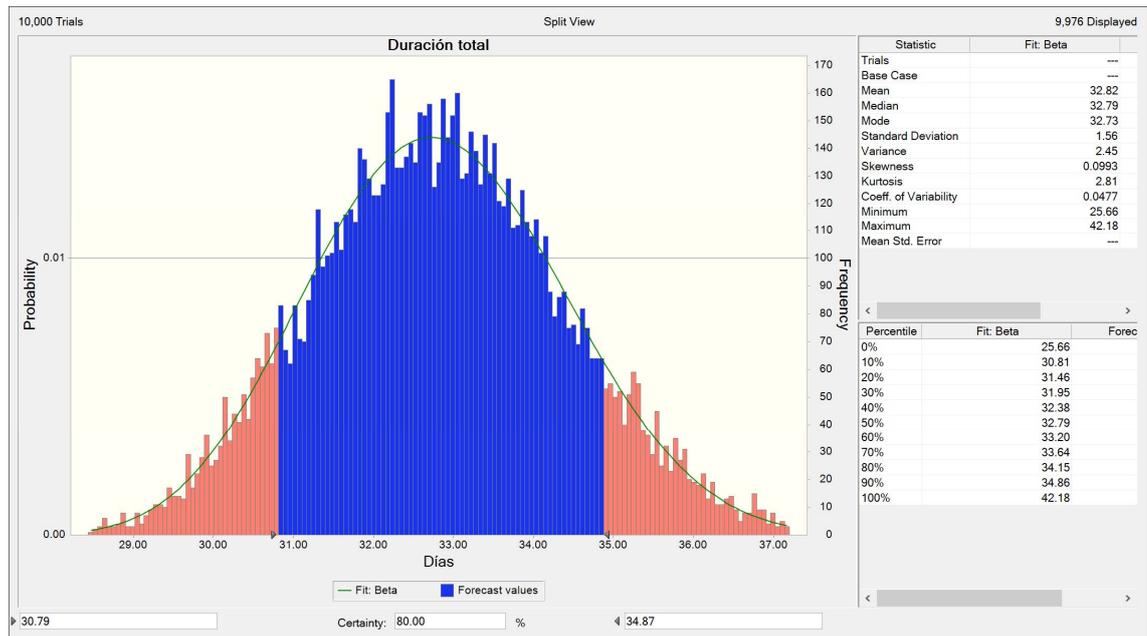


Ilustración 39: Resultados de la simulación del camino crítico. Fuente: Elaboración propia mediante el software Crystal Ball.

Las tablas de la parte derecha de la Ilustración 39, se muestran a continuación en las Tabla 18 y Tabla 19, con más detalle. La primera tabla corresponde con los estadísticos, resultando de especial interés la media, la desviación, y los valores mínimo y máximo.



Statistic	Fit: Beta	Forecast values
Trials	---	10,000
Base Case	---	0.00
Mean	32.82	32.82
Median	32.79	32.79
Mode	32.73	---
Standard Deviation	1.56	1.56
Variance	2.45	2.45
Skewness	0.0993	0.0993
Kurtosis	2.81	2.81
Coeff. of Variability	0.0477	0.0477
Minimum	25.66	27.47
Maximum	42.18	38.29
Mean Std. Error	---	0.02

Tabla 18: Estadísticos de la simulación del camino crítico. Fuente: Elaboración propia mediante el software Crystal Ball.

La duración media del proyecto, como se aprecia en la Tabla 18, es de 32,82 días. Con una variación de 1,56, una duración máxima de 42,18 días, y una duración mínima de 25,66 días.

También se muestran en la Tabla 19, los percentiles que utilizaremos para calcular el *buffer* de proyecto.



Percentile	Fit: Beta	Forecast values
0%	25.66	27.47
10%	30.81	30.79
20%	31.46	31.47
30%	31.95	31.96
40%	32.38	32.38
50%	32.79	32.79
60%	33.20	33.19
70%	33.64	33.62
80%	34.15	34.13
90%	34.86	34.87
100%	42.18	38.29

Tabla 19: Percentiles de la simulación del camino crítico. Fuente: Elaboración propia mediante el software Crystal Ball.

Según (Joaquín, 2019), la duración del *buffer* de proyecto puede definirse como la diferencia entre el percentil 90%, y el percentil 50%. Es decir:

$$\text{Buffer de proyecto} = D_{90\%} - D_{50\%} = 34,86 - 32,79 = 2,07 \text{ días}$$

B. *Buffer* de alimentación 1.

A continuación, se muestran los resultados de correr 10.000 iteraciones con simulación de Montecarlo, para el camino no crítico BE, y se utilizarán para calcular el tamaño del primer *buffer* de alimentación del proyecto, con el objetivo de proteger al camino crítico de retrasos en la cadena no crítica BE.

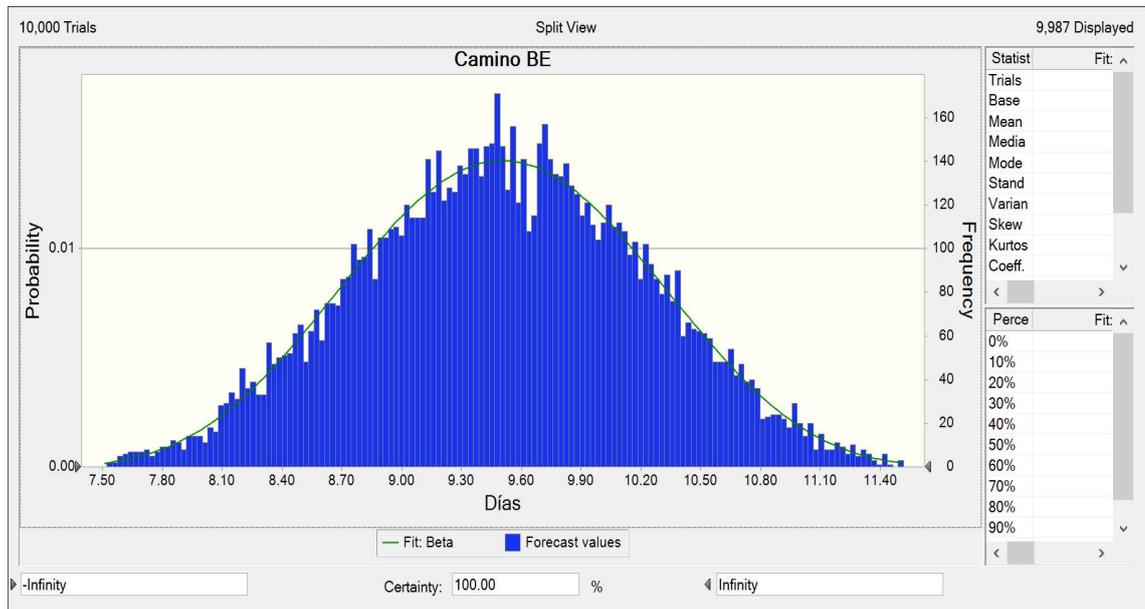


Ilustración 40: Resultados de la simulación de la cadena no-crítica B-E. Fuente: Elaboración propia mediante el software Crystal Ball.

En la Tabla 20, debajo del texto, se aprecian de nuevo los estadísticos del camino no-crítico BE. La duración media de este camino es de 9,51 días, con una desviación de 0,72. Además, también se observa que como máximo este camino ha durado 12,29 días, y como mínimo 6,74 días.

Statistic	Fit: Beta	Forecast values
Trials	---	10,000
Base Case	---	0.00
Mean	9.51	9.51
Median	9.51	9.50
Mode	9.50	---
Standard Deviation	0.72	0.72
Variance	0.52	0.52
Skewness	0.0035	0.0035
Kurtosis	2.65	2.65
Coeff. of Variability	0.0755	0.0755
Minimum	6.74	7.48
Maximum	12.29	11.72
Mean Std. Error	---	0.01

Tabla 20: Estadísticos de la simulación del camino no crítico B-E. Fuente: Elaboración propia mediante el software Crystal Ball.

Para el cálculo del *buffer* de alimentación, que se insertará al final de la cadena BE, antes de la fusión con el camino crítico, se utilizan los datos de los percentiles mostrados en la Tabla 21.

Perce	Fit: Beta	Forecast values
0%	6.74	7.48
10%	8.57	8.56
20%	8.88	8.88
30%	9.11	9.12
40%	9.31	9.32
50%	9.51	9.50
60%	9.70	9.70
70%	9.90	9.90
80%	10.14	10.13
90%	10.45	10.44
100%	12.29	11.72

Tabla 21: Percentiles de la simulación del camino no crítico B-E. Fuente: Elaboración propia mediante el software Crystal Ball.

El percentil 90% del camino BE es 10,45, y el perfil 50% es 9,51. Por tanto, la longitud del *buffer* de alimentación 1 es de:

$$\text{Buffer de alimentación 1} = D_{90\%} - D_{50\%} = 10,45 - 9,51 = 0,94 \text{ días}$$

C. Buffer de alimentación 2

Para el cálculo del *buffer* de alimentación 2, se procede de la misma manera. Los resultados de correr las 10.000 iteraciones se muestran a continuación en la Ilustración 41, donde se aprecia la función de distribución que tiene el camino D.

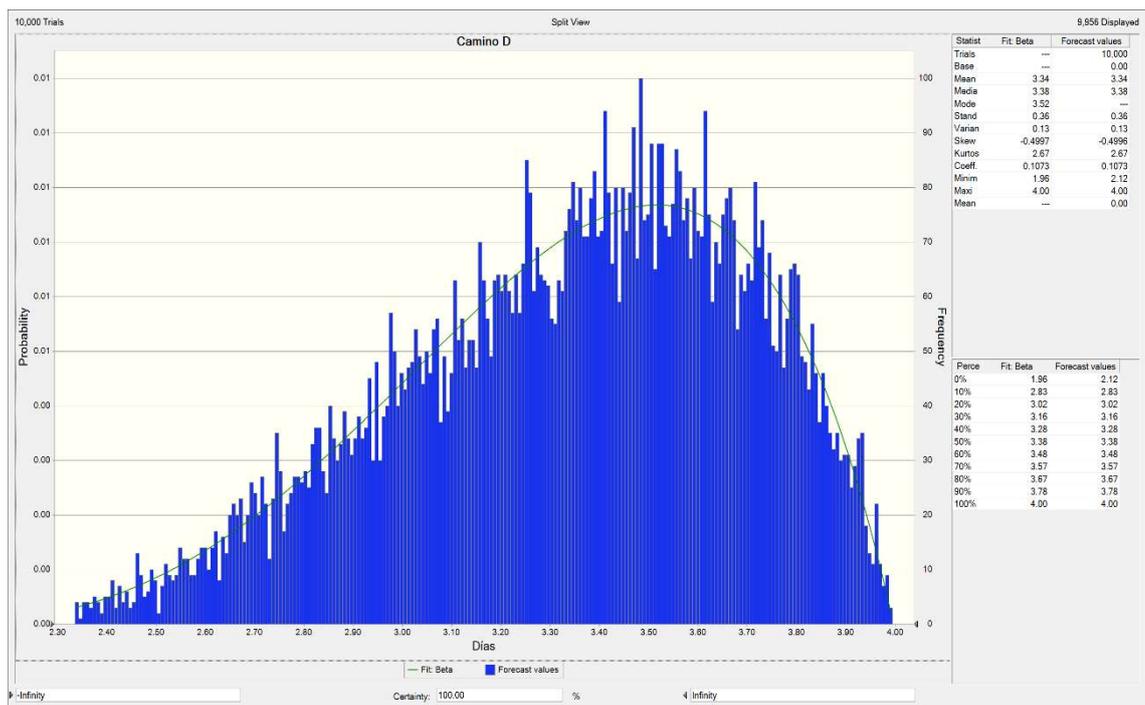


Ilustración 41: Resultados de la simulación del camino no crítico D. Fuente: Elaboración propia mediante el software Crystal Ball.

De igual manera, en las tablas de la parte superior derecha se aprecian tanto los estadísticos, como los percentiles. Con los estadísticos de la Tabla 22, se observa que el camino no crítico D, es decir, la actividad D, tiene una duración



media de 3,34 días, con una desviación de 0,36, una duración máxima de 4 días, y una duración mínima de 1,96 días.

Statistic	Fit: Beta	Forecast values
Trials	---	10,000
Base Case	---	0.00
Mean	3.34	3.34
Median	3.38	3.38
Mode	3.52	---
Standard	0.36	0.36
Variance	0.13	0.13
Skewness	-0.4997	-0.4996
Kurtosis	2.67	2.67
Coeff. of V	0.1073	0.1073
Minimum	1.96	2.12
Maximum	4.00	4.00
Mean Std.	---	0.00

Tabla 22: Estadísticos de la simulación del camino no crítico D. Fuente: Elaboración propia mediante el software Crystal Ball.

Perce	Fit: Beta	Forecast values
0%	1.96	2.12
10%	2.83	2.83
20%	3.02	3.02
30%	3.16	3.16
40%	3.28	3.28
50%	3.38	3.38
60%	3.48	3.48
70%	3.57	3.57
80%	3.67	3.67
90%	3.78	3.78
100%	4.00	4.00

Tabla 23: Percentiles de la simulación del camino no crítico D. Fuente: Elaboración propia mediante el software Crystal Ball.

Con los percentiles de la Tabla 23 se procede a calcular el tamaño del *buffer* de alimentación 2. El percentil 90% del camino C es 3,78, y el perfil 50% es 3,38. Por tanto, la longitud del *buffer* de alimentación 2 es de:

$$\text{Buffer de alimentación 2} = D90\% - D50\% = 3,78 - 3,38 = 0,4$$

En la Tabla 24, quedan reflejados los tamaños de los *buffers* obtenidos para este proyecto, utilizando la simulación de Montecarlo para dimensionarlos.

Método	Buffer de proyecto	Buffer de alimentación 1	Buffer de alimentación 2
Simulación de Montecarlo	2,1	0,94	0,4

Tabla 24: Resumen de buffers en escenario 2. Fuente: Elaboración propia.



5. Análisis de los resultados.

El objetivo de utilizar simulación de Montecarlo es que los *buffers* se ajusten mejor al comportamiento esperado en la duración de las actividades del proyecto.

En el escenario 2, en el que se usa la simulación de Montecarlo, la duración media del proyecto es 32,82 días, superior a la que se obtiene en el escenario 1, usando los métodos tradicionales, que es 32,7 días. La razón es que en el escenario 1, la duración que se obtiene para el proyecto es simplemente la suma de un único estadístico de las actividades críticas; simplemente la duración que tiene un 50% de probabilidades de cumplirse. Sin embargo, cuando utilizamos simulación de Montecarlo en el escenario 2, la duración esperada para el proyecto tiene en cuenta toda la variabilidad asociada a las funciones de distribución que realmente tienen las actividades del proyecto (y, por tanto, al comportamiento esperado en las duraciones de las actividades).

Duración del proyecto	Días
Métodos tradicionales	32,7
Simulación de Montecarlo	32,82

Tabla 25: Comparación entre escenarios 1 y 2. Fuente: Elaboración propia.

En cuanto al tamaño de los *buffers*, tanto del *buffer* de proyecto como de los *buffers* de alimentación, el director del proyecto observará que los métodos más tradicionales ofrecen unos valores dispares, como se muestra en el resumen de la Tabla 26, ya que cada uno de estos métodos se fijan en características distintas de la actividad (tiempos de protección implícita, duración de las actividades sin protección). Esto hace que, por ejemplo, la duración del *buffer* de proyecto con el método del tercio crítico dé 10,9 días, muy superior al resultante de aplicar los métodos 50% y SSQ (2,225 y 2,29 días respectivamente).

Método	Buffer de proyecto	Buffer de alimentación 1	Buffer de alimentación 2
50%	2,225	0,67	0,205
Tercio crítico	10,9	3,167	1,12
SSQ	2,29	0,97	0,41
Simulación de Montecarlo	2,07	0,94	0,4

Tabla 26: Resumen del tamaño de los buffers en ambos escenarios. Fuente: Elaboración propia.

La duración del *buffer* de proyecto obtenida con simulación de Montecarlo, utilizando el percentil 90, da como resultado 2,07 días. Esta duración es el margen de protección que debe añadirse a la duración esperada para el proyecto (32,7 días). El dimensionamiento de este *buffer* ha tenido en cuenta no sólo las características de las actividades críticas (márgenes de protección individuales o duraciones de las actividades sin estas protecciones) como hacen los métodos tradicionales, sino toda la incertidumbre asociada a la duración aleatoria de las actividades del proyecto. Esto aumenta las probabilidades de que la dimensión del *buffer* de proyecto obtenido por este método se ajuste a las necesidades reales del proyecto teniendo en cuenta la incertidumbre real introducida por cada actividad.

Esta conclusión se hace extensiva al dimensionamiento de los *buffers* de alimentación, donde se aprecia que, mediante la utilización de la simulación de Montecarlo, se obtienen duraciones que se ajustan mejor a la incertidumbre real introducida por las actividades que forman las cadenas no-críticas.

Sin embargo, el empleo de simulación de Montecarlo para el dimensionamiento de los *buffers* también presenta algunos inconvenientes. En primer lugar, añade complejidad al análisis, ya que es necesario usar herramientas software para hacer la simulación. Además, debido a la aleatoriedad de las duraciones de las actividades del proyecto, y a que en cada simulación se generan escenarios diferentes, es posible que existan varios caminos críticos, cada uno de ellos con cierta probabilidad de ser crítico (según el índice de criticidad de sus actividades), por lo que cada cadena crítica daría lugar a diferentes cadenas de alimentación, lo que dificultaría la selección de un tamaño adecuado para los *buffers* de alimentación y de proyecto. No obstante, el hecho de que el dimensionamiento de *buffers* mediante simulación de Montecarlo genere tamaños de *buffer* más realistas al tener en cuenta la aleatoriedad real en la duración de las actividades hará recomendable su empleo en proyectos complejos con alta incertidumbre en la



duración de las actividades, en los que el cumplimiento con la fecha de finalización sea un factor decisivo.



VI. CONCLUSIONES

En este Trabajo Fin de Grado (TFG) se han comparado los métodos de dimensionamiento de *buffers* tradicionales, con las técnicas más recientes, basadas en simulación de Montecarlo, y se ha ilustrado el proceso de dimensionamiento de los *buffers* con ambos tipos de técnicas. Como resultado, hemos concluido que mediante simulación de Montecarlo podemos obtener unos tamaños de *buffers* más próximos a lo esperable según la incertidumbre real asociada a la naturaleza aleatoria de cada actividad del proyecto. Estos *buffers* se insertan en la cadena del proyecto, en sitios estratégicos en función de la criticidad o no criticidad de las actividades que forman la cadena donde se insertan, siguiendo el método de Gestión de Proyectos por Cadena Crítica (CCPM), que, ante la presencia de los mencionados retrasos (Ley de Parkinson, Síndrome del Estudiante etc.), garantiza un plazo de ejecución que se ajusta más a la línea base del proyecto.

Los métodos tradicionales de dimensionamiento de *buffers* presentan ciertas limitaciones. Cuando se trata de dimensionar estos *buffers* en entornos de incertidumbre, los métodos tradicionales, entre los que se han destacado la regla del 50%, el método del tercio crítico, o el método de la suma de los cuadrados (SSQ), no tienen en cuenta toda la incertidumbre de las actividades, ya que dimensionan los *buffers* reduciendo la aleatoriedad en la duración de la actividad a un parámetro fijo. Por esta razón, en este TFG se ha utilizado la simulación de Montecarlo para el dimensionamiento de los *buffers*, ya que, a diferencia de los métodos tradicionales, recoge toda la incertidumbre asociada a la naturaleza aleatoria de cada actividad, generando duraciones aleatorias consecuentes con las funciones de distribución que modelan, para cada actividad, el riesgo o posibilidades de retraso que éstas tienen. Así, el posible retraso va a estar particularizado para cada actividad (por eso cada actividad tiene una función de distribución distinta, en función de cómo de probables sean los retrasos). De esta manera se consiguen unos tamaños de *buffers* más próximos a lo esperable durante la ejecución del proyecto.

Este trabajo se podría extender analizando más proyectos, formados por redes más complejas, con el objetivo de llegar a una conclusión al respecto de qué método es óptimo en cada caso, teniendo en cuenta diferentes índices de criticidad de las actividades, así como variados niveles de incertidumbre y de longitud de las mismas. Aun así, se ha demostrado que la simulación de Montecarlo aplicada al dimensionamiento de los *buffers* propuestos por CCPM, proporciona unos resultados que se aproximan más a la realidad al tener en cuenta toda la incertidumbre de las actividades.



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**



VII. REFERENCIAS

- [1] Acebes Senovilla, F., Poza García, D., & López Paredes, A. (2022). *Aplicaciones informáticas para la gestión de proyectos*. Valladolid: Marcial Pons.
- [2] Tenera, A. B. (2008). Critical chain *buffer* sizing: a comparative study. Paper presented at PMI® Research Conference: Defining the Future of Project Management, Warsaw, Poland. Newtown Square, PA: Project Management Institute
- [3] Ashtiani, B., & Jalali, G. R. (2007). *New approach for buffer sizing in Critical Chain scheduling*.
- [4] Balyuk, A. (Abril de 2020). *Student Syndrome in Project Management: Real Constraint or Just Human Factor?* Obtenido de <https://www.epicflow.com/blog/student-syndrome-in-project-management-real-constraint-or-just-human-factor/>. [Último acceso: Mayo 2022]
- [5] Biblioteca de SAP Business. (Mayo de 2021). Obtenido de https://help.sap.com/docs/SAP_BUSINESS_BYDESIGN/2754875d2d2a403f95e58a41a9c7d6de/2cddf182722d10149915fe886da5e1d4.html?version=2105&locale=es-ES. [Último acceso: Mayo 2022]
- [6] Cáceres, J. C. (Diciembre de 2021). *Comunicación Soltel*. Obtenido de <https://www.soltel.es/pm%C2%B2-la-metodologia-de-direccion-de-proyectos-de-la-union-europea/>. [Último acceso: Abril 2022]
- [7] Cui, N., & Li, B. (2012). *Buffer sizing approach with dependence assumption between activities in critical chain scheduling*.
- [8] Edraw. (2022). Obtenido de <https://www.edrawsoft.com/es/article/precedence-diagramming-method-the-ultimate-guide.html>. [Último acceso: Abril 2022]
- [9] Excellence, I. T. (Julio de 2017). *Norma ISO 21500: Guía para la gestión de proyectos*.
- [10] García, E. V., Area, C. A., & Rioja, M. B. (Julio de 2017). *Critical chain: Sizing the project buffers*. Obtenido de <http://dspace.aepro.com/xmlui/bitstream/handle/123456789/338/AT01-011.pdf?sequence=1&isAllowed=y>
- [11] Gillis, A. S. (s.f.). *Gestión ágil de proyectos o APM*. Obtenido de <https://www.computerweekly.com/es/definicion/Gestion-agil-de-proyectos-o-APM>. [Último acceso: Mayo 2022]
- [12] Gómez, T. V. (2021). *Dirección del proyecto software: Creación de página web y plataforma e-commerce combinando metodologías de Dirección de Proyectos*. Universidad de Valladolid.



- [13] Haughey, D. (Octubre de 2019). *Escuela de Project Management*. Obtenido de <http://www.proyectosfindecarrera.com/ICB-IPMA%20COMPETENCES%20BASELINE.htm>. [Último acceso: Abril 2022]
- [14] IBM, C. L. (August de 2020). *What is Monte Carlo Simulation?* Obtenido de <https://www.ibm.com/cloud/learn/monte-carlo-simulation>. [Último acceso: Junio 2022]
- [15] iSolution. (s.f.). Obtenido de <https://isolution.pro/es/t/management-concepts/project-management-triangle/triangulo-de-gestion-de-proyectos#:~:text=Las%20tres%20limitaciones%20en%20un%20tri%C3%A1ngulo%20de%20gesti%C3%B3n,pueden%20tardar%20m%C3%A1s%20o%20menos%20tiempo%20en%20completars>. [Último acceso: Abril 2022]
- [16] Joaquín, F. d. (2019). *Aplicación de la simulación Monte Carlo al dimensionado de buffers en la gestión de proyectos por el Método de la Cadena Crítica*. Universidad de Castilla la Mancha.
- [17] Josafat Gascón Busio, Ó. (s.f.). *Todo PMP & Agile*. Obtenido de <https://todopmp.com/estimar-la-duracion-las-actividades/#close>. [Último acceso: Junio 2022]
- [18] Kenton, W. (2021 February). *Critical PAtH Analysis*. Obtenido de <https://www.investopedia.com/terms/c/critical-path-analysis.asp#:~:text=What%20is%20Critical%20Path%20Analysis,each%20activity%20on%20any%20others>. [Último acceso: Mayo 2022]
- [19] Kopp, C. M. (Marzo de 2022). Obtenido de <https://www.investopedia.com/terms/p/pert-chart.asp#:~:text=A%20PERT%20chart%20is%20a,of%20a%20project%20for%20analysis>. [Último acceso: Mayo 2022]
- [20] Luz, J. V. (Septiembre de 2017). *SciFlo. Prácticas PMBOK y Cadena Crítica: antagonismos y oportunidades de complementación*. Obtenido de <https://www.scielo.br/j/gp/a/pBNgmJgwhQvgVykW4ZhR6Qr/?lang=en&format=html>
- [21] Marcos, C. S. (2021). *Gestión de Proyectos mediante cadena crítica (CCPM Critical Chain Project Management)*. Universidad de Valladolid.
- [22] Martínez Pérez, B. (2013). *Simulación del blindaje de una fuente de fotones gamma mediante el método de Monte Carlo*. Obtenido de <http://uvadoc.uva.es/handle/10324/2777>



- [23] Neira, W. (2019). *Seguimiento a la gerencia para proyectos de vías terciarias en Colombia valorando alcance, tiempo y costo, basados en el PMBOK (Tesis de postgrado)*. Bogotá.
- [24] Newbold, R. (1998). *Project Management in the fast lane: applying the theory of constraints*. Londres.
- [25] Oracle. *Crystall Ball*. (s.f.). Obtenido de <https://www.oracle.com/es/applications/crystalball/>. [Último acceso: Junio 2022]
- [26] Orobio, A., & Gómez, H. D. (2015). *Efectos de la incertidumbre en la programación de proyectos de construcción de carreteras*. Medellín, Colombia.
- [27] Palisade. (s.f.). *Monte Carlo Simulation*. Obtenido de https://www.palisade.com/risk/monte_carlo_simulation.asp. [Último acceso: Junio 2022]
- [28] PMI. (2017). *Fundamentos para la Dirección de Proyectos (Guía del PMBOK) -Sexta Edición*.
- [29] Posadas, M. (2018). *Ingeniería en Organización Industrial. Tema 6. Planificación, Programación y Control de Proyectos*. Universidad de Valladolid.
- [30] Poza, D. J. (2019). *Dirección de Proyectos. Simulación de Montecarlo*. Universidad de Valladolid.
- [31] Poza, D. J. (2020). *Dirección de proyectos. Tema 11. Gestión de los Interesados del proyecto*.
- [32] Poza, D. J. (2020). *Dirección de Proyectos. Tema 5. Gestión del cronograma del proyecto*. Universidad de Valladolid.
- [33] Project Manager Soy. (Junio de 2021). Obtenido de <https://www.projectmanager.soy/metodologia-pm2/?cn-reloaded=1>. [Último acceso: Abril 2022]
- [34] Roseke, B. (2017). *Project Engineer*. Obtenido de <https://www.projectengineer.net/the-pmboks-five-project-phases/>. [Último acceso: Abril 2022]
- [35] Roser, C. (Noviembre de 2014). *A Critical Look at Goldratt's Drum-Buffer-Rope Method*. Obtenido de <https://www.allaboutlean.com/drum-buffer-rope/>. [Último acceso: Mayo 2022]
- [36] Sánchez, J. L. (2005). *Gestión de Proyectos: Los buffers de proyecto*.
- [37] Sanz, R. (2017). *Módulo 1. Competencias IPMA*. Obtenido de <https://uv-mdap.com/programa-desarrollado/bloque-7-certificacion-ipma/competencias-ipma/>. [Último acceso: Mayo 2022]



- [38] Sanz, R. (2017). *Módulo 3. Procesos de Dirección de Proyectos*. Obtenido de <https://uv-mdap.com/programa-desarrollado/bloque-ii-certificacion-pmp-pmi/procesos-de-la-direccion-de-proyectos-pmp-pmi/#:~:text=Los%20Procesos%20de%20Direcci%C3%B3n%20de%20Proyectos%20son%20un,Proyectos%20se%20logra%20mediante%20la%20realizaci%C3%B3n%20d.> [Último acceso: Mayo 2022]
- [39] Staff HDC. (2021). *Metodología Scrum, una herramienta útil para agilizar tus proyectos*.
- [40] Strom, E. M. (Enero de 2022). *Parkinson's Law in Project Management*. Obtenido de http://wiki.doing-projects.org/index.php/Parkinson%27s_Law_in_Project_Management. [Último acceso: Abril 2022]
- [41] Taylor, S. G. (s.f.). *Quantifying buffers for project schedules*. College of Business, University of Wyoming.
- [42] Team Asana. (2021). Obtenido de <https://asana.com/es/resources/project-management-triangle>. [Último acceso: Mayo 2022]
- [43] Torre, R. (2015). *Critical Chain Project Management to cope with uncertainty*. Obtenido de http://apppm.man.dtu.dk/index.php/Critical_Chain_Project_Management_to_cope_with_uncertainty. [Último acceso: Mayo 2022]
- [44] Vanhoucke, M. (Diciembre de 2012). *PM Knowledge Center. Aggressive activity time estimates: Protecting against activity delays*. Obtenido de http://www.pmknowledgecenter.be/dynamic_scheduling/risk/aggressive-activity-time-estimates-protecting-against-activity-delays. [Último acceso: Junio 2022]
- [45] Vanhoucke, M. (Enero de 2012). *PM Knowledge Center. Sizing Buffers: The cut and paste method*. Obtenido de http://www.pmknowledgecenter.com/dynamic_scheduling/risk/sizing-ccbm-buffers-cut-and-paste-method. [Último acceso: Junio 2022]
- [46] Verma, E. (Junio de 2022). *What is Critical Chain Project Management*. Obtenido de <https://www.simplilearn.com/what-is-critical-chain-project-management-rar68-article>. [Último acceso: Abril 2022]
- [47] Wrike. (2021). Obtenido de <https://www.wrike.com/es/project-management-guide/faq/que-es-pmi-en-gestion-de-proyectos/>, [Último acceso: Mayo 2022]