



**EXPRESIVIDAD DE BPMN 2.0 DESDE LA PERSPECTIVA DEL
PROBLEMA DE SECUENCIAMIENTO**

**MEMORIA PARA OPTAR AL TÍTULO DE
MAGÍSTER EN CONTROL DE GESTIÓN**

DANIELA GONZÁLEZ NAVARRO

**PROFESOR GUÍA:
SIGIFREDO LAENGLER SCARLAZETTA**

**SANTIAGO DE CHILE
DICIEMBRE 2018**

Agradecimientos

Me gustaría agradecer primero que todo a mi profesor guía Sigifredo Laengle, por haberme permitido el desarrollo de esta tesis, por compartir sus conocimientos, cada detalle y momento dedicado para aclarar cualquier tipo de duda que me surgiera, agradecerle por la claridad y exactitud con la que me enseñó. Estuvo siempre dispuesto a resolver mis dudas y a aconsejarme cada vez que fuera necesario, siendo así un apoyo constante y fundamental para lograr el trabajo realizado.

Agradecer también a mi padre Sebastián, por ser el principal promotor de mis sueños, quien desde que tengo recuerdo me ha incentivado aprender, crecer y superarme. Gracias por cada día confiar y creer en mí y en mis expectativas y por siempre desear y anhelar lo mejor para mi vida, gracias por cada consejo y por cada una de sus palabras que me guiaron durante mi vida. Por lo demás, aún ante las dificultades que nos ha impuesto la vida, nunca ha dejado de apoyarme.

A mi hermano Sebastián quien me apoya en cada decisión y proyecto, quien ha estado a mi lado para cada triunfo y derrota que he tenido. Te agradezco por tantas ayudas y tantos aportes no solo para el desarrollo de mi tesis, sino también para mi vida; eres mi inspiración y mi motivación.

Finalmente, mi agradecimiento a María Jose, Stephanie, Kristina y Javier, los cuales han estado a mi lado y estuvieron presentes durante toda o la mayor parte de la realización y el desarrollo de esta tesis y magíster, gracias a aquellos que con respeto realizaron aportes a ésta, gracias a todos. No solo no hubiera sido nada sin ustedes, quienes han estado desde el comienzo y hasta el día de hoy. Que hasta cada momento que disfrutamos es un experiencia totalmente enriquecedora. Gracias por ayudarme a crecer tanto en lo académico como en lo personal.

Resumen Ejecutivo

Business Process Model and Notation (BPMN) se ha convertido en un estándar para describir flujos de tareas. BPMN propone un conjunto de símbolos que representan el comportamiento de los procesos de negocios desde el punto de vista del usuario de un sistema de información y aporta descripción de elementos para la automatización computacional. Por otra parte, el comportamiento de procesos ha sido estudiado por decenas de años en Investigación Operativa bajo el nombre de Problema de *Scheduling*. A pesar de la estrecha relación, no se encuentra en la literatura ninguna referencia a ambos problemas en conjunto, ni ningún aporte desde el problema de *Scheduling* a la representación del comportamiento en BPMN. ¿Es posible enriquecer la capacidad descriptiva de BPMN con elementos del comportamiento estudiados en el problema de *Scheduling*? ¿Es posible que existan elementos del comportamiento de procesos de negocio relevantes que no estén representados en BPMN? ¿Cuáles son esos elementos?.

Para responder a estas preguntas, investigamos en detalle el formalismo BPMN y el problema de *Scheduling*, identificamos diferencias y proponemos un conjunto de símbolos que añaden más descripción al comportamiento de los procesos de negocios desde el punto de vista del usuario. Adicionalmente, se hizo una revisión de dos trabajos previos que proponen mejoras a BPMN, pero que van en una dirección diferente a lo buscado en este trabajo.

En este trabajo presentamos detalladamente el último estándar BPMN 2.0 y revisamos un voluminoso resumen del problema de *Scheduling* (Pinedo, 2016). Durante esta revisión, encontramos varios elementos prácticos relevantes del problema de *Scheduling* que no están representados en BPMN. Entre estos elementos destacan la velocidad de procesamiento, tiempo de procesamiento, la capacidad de procesamiento, la regla de prioridad y la función objetivo.

Estudiando las diferencias, encontramos que hay elementos que describen el problema de *Scheduling* que deberían incorporarse al estándar BPMN, porque contribuyen a una mejor y más completa caracterización de los procesos. De hecho, con los atributos de velocidad, el tiempo de preparación y de procesamiento, se puede describir gráficamente la presencia de cuellos de botella y restricción de recursos. Adicionalmente, la regla de prioridad permite describir colas de espera impuestas por condiciones propias del negocio. Finalmente, la función objetivo permite comunicar de manera gráfica a los participantes del negocio cuáles son los objetivos que la organización requiere de los procesos. Hay algunos elementos propios de *Scheduling* que proponemos agregar sólo opcionalmente. Entre estas opciones están las restricciones que consideran la fecha de llegada de la tarea al sistema, los tiempos de espera entre máquinas, y familia de tareas. Con nuestra propuesta se robustece la capacidad descriptiva del estándar BPMN y lo hace apropiado a un mejor análisis consistente con la teoría de *Scheduling*.

Tabla de Contenido

1. Introducción	1
1.1. Metodología	4
2. Marco Teórico	6
2.1. Business Process Management (BPM)	8
2.1.1. Notaciones para modelado de procesos de negocio	10
2.1.2. Business Process Model and Notation (BPMN)	11
2.2. Scheduling	14
2.3. Conexión entre las dos corrientes en la literatura	15
3. BPMN y Scheduling	16
3.1. Business Process Model and Notation 2.0	17
3.1.1. Clase de Modelos	17
3.1.2. Instancia de Proceso y Token	17
3.1.3. Objetos de flujo	18
3.1.4. Objetos de conexión	42
3.1.5. Carriles	42
3.1.6. Artefactos	43
3.1.7. Ejemplo BPMN 2.0	46
3.1.8. Categorías de procesos	46
3.2. Trabajos Anteriores que Pretenden Mejorar BPMN	49
3.3. Scheduling	50
3.3.1. Campo α : ambiente de máquinas	52
3.3.2. Campo β : restricciones	59
3.3.3. Campo γ : función objetivo	61
4. Desarrollo de Propuesta	63
4.1. Comparación de los elementos propios de <i>Scheduling</i> con los de BPMN 2.0.	63
5. Resultados	71
5.1. Análisis Tabla Comparativa	71
5.1.1. Elementos de <i>Scheduling</i> que son representables en BPMN 2.0	73
5.1.2. Elementos de <i>Scheduling</i> que no se pueden representar en BPMN 2.0	73
5.2. Propuesta de Mejora BPMN 2.0	74
5.2.1. Tiempo de Procesamiento	74
5.2.2. Tiempo de Preparación	75

5.2.3.	Reglas de Prioridad	75
5.2.4.	Capacidad de Procesamiento	78
5.2.5.	Función Objetivo	79
5.3.	Modelo Práctico Numérico	80
5.3.1.	Problema I de tipo Flow Shop: Secuenciación en taller de flujo de dos máquinas	80
5.3.2.	Diagrama Problema I	83
5.3.3.	Diagrama Problema I con Propuesta de Mejora	85
5.3.4.	Problema II de tipo Flow Shop: Secuenciación en taller de flujo de tres máquinas	85
5.3.5.	Diagrama Problema II	88
5.3.6.	Diagrama Problema II con Propuesta de Mejora	89
5.4.	Pasos a seguir para que la propuesta se incorpore al estándar oficial	90
6.	Conclusión	91
	Bibliografía	94
	Referencias	94
	Anexos	I
	Anexo A. Resumen de BPMN 2.0.	I
	Anexo B. Resumen de Scheduling	V
	Anexo C. Tablas Comparativas entre Scheduling y BPMN 2.0 con Ilustraciones	IX

Índice de Tablas

4.1. Comparativa Conceptos básicos de Scheduling ¿Qué elementos de <i>Scheduling</i> se encuentran en BPMN?	64
4.2. Comparativa ambiente Alpha de Scheduling con BPMN 2.0	65
4.3. Continuación comparativa ambiente Alpha de Scheduling con BPMN 2.0	66
4.4. Comparativa ambiente Beta de <i>Scheduling</i> con BPMN 2.0 [1/4]	67
4.5. Continuación comparativa ambiente Beta de <i>Scheduling</i> con BPMN 2.0 [2/4]	68
4.6. Continuación comparativa ambiente Beta de <i>Scheduling</i> con BPMN 2.0 [3/4]	69
4.7. Continuación comparativa ambiente Beta de <i>Scheduling</i> con BPMN 2.0 [4/4]	70
4.8. Comparativa ambiente Gama de <i>Scheduling</i> con BPMN 2.0	70
5.1. Problema de dos máquinas	81
5.2. Problema de dos máquinas, iteración 1	82
5.3. Problema de dos máquinas, iteración 2	82
5.4. Problema de dos máquinas, iteración 3	83
5.5. Problema de dos máquinas, iteración 4	83
5.6. Problema inicial de secuenciamiento de taller de flujo de tres máquinas	86
5.7. Problema transformado a dos máquinas	87
A.1. Objetos de Flujo en BPMN 2.0.	I
A.2. Actividades en BPMN 2.0.	II
A.3. Eventos en BPMN 2.0.	II
A.4. Compuertas en BPMN 2.0.	III
A.5. Objetos de conexión en BPMN 2.0.	III
A.6. Carrieles en BPMN 2.0.	IV
A.7. Artefactos en BPMN 2.0.	IV
B.1. Conceptos de <i>Scheduling</i>	V
B.2. Campo α : Ambiente de máquinas en <i>Scheduling</i>	VI
B.3. Campo β : restricciones en el problema de Scheduling [1/2]	VII
B.4. Campo β : restricciones en el problema de Scheduling [2/2]	VIII
B.5. Campo γ : función objetivo del problema de Scheduling	VIII

Índice de Ilustraciones

1.1. Metodología a seguir	4
2.1. Evolución modelos de gestión	7
2.2. Línea de Tiempo BPM.	9
2.3. Ciclo de Vida Business Process Management.	10
2.4. Ejemplo Modelado de un proceso de Negocio.	12
2.5. Evolución de BPMN.	13
2.6. Evolución Scheduling.	15
3.1. Categorías de elementos gráficos en BPMN.	18
3.2. Representación gráfica de elementos en BPMN.	19
3.3. Tarea en BPMN.	19
3.4. Transacción en BPMN.	20
3.5. Subproceso de eventos en BPMN.	20
3.6. Actividad de llamada en BPMN.	20
3.7. Tarea indefinida en BPMN.	21
3.8. Envío de mensaje en BPMN.	21
3.9. Recepción de mensaje en BPMN.	21
3.10. Tarea de usuario en BPMN.	22
3.11. Tarea manual en BPMN.	22
3.12. Regla de negocio en BPMN.	22
3.13. Tarea de servicio en BPMN.	23
3.14. Ejecución de script en BPMN.	23
3.15. Subproceso en BPMN.	23
3.16. Ciclo en BPMN.	24
3.17. Instancias múltiples en paralelo en BPMN.	24
3.18. Instancias múltiples en secuencia en BPMN.	24
3.19. Subproceso Ad Hoc en BPMN.	25
3.20. Tarea de Compensación en BPMN.	25
3.21. Compuerta exclusiva en BPMN.	25
3.22. Compuerta basada en eventos en BPMN.	26
3.23. Compuerta paralela en BPMN.	26
3.24. Compuerta inclusiva en BPMN.	26
3.25. Compuerta compleja en BPMN.	26
3.26. Eventos de inicio, intermedio y finalización en BPMN.	27
3.27. Tipos de evento en BPMN.	28

3.28. Evento de Inicio Simple	29
3.29. Evento Intermedio de Lanzamiento Simple	29
3.30. Evento Final Simple	29
3.31. Evento de Inicio de Mensaje	29
3.32. Evento Intermedio de Captura de Mensaje	30
3.33. Evento Intermedio Adjunto Interruptor de Mensaje	30
3.34. Evento Intermedio Adjunto No Interruptor de Mensaje	30
3.35. Evento intermedio de Lanzamiento de Mensaje	30
3.36. Evento Fin de Mensaje	31
3.37. Evento de Inicio Temporal	31
3.38. Evento Intermedio de Captura Temporal	31
3.39. Evento Intermedio Adjunto Interruptor Temporal	32
3.40. Evento Intermedio Adjunto No Interruptor Temporal	32
3.41. Evento de Inicio Escalable	32
3.42. Evento Intermedio Adjunto Interruptor Escalable	33
3.43. Evento Intermedio Adjunto No Interruptor Escalable	33
3.44. Evento Intermedio de Lanzamiento Escalable	33
3.45. Evento de Fin Escalable	33
3.46. Evento de Inicio Condicional	34
3.47. Evento Intermedio de Captura Condicional	34
3.48. Evento Intermedio Adjunto Interruptor Condicional	34
3.49. Evento Intermedio Adjunto No Interruptor Condicional	35
3.50. Evento Intermedio de Captura de Enlace	35
3.51. Evento Intermedio de Lanzamiento de Enlace	35
3.52. Evento de Inicio de Error	36
3.53. Evento Intermedio Adjunto Interruptor de Error	36
3.54. Evento Final de Error	36
3.55. Evento Intermedio Adjunto Interruptor de Cancelación	37
3.56. Evento Final de Cancelación	37
3.57. Evento Inicio Compensación	37
3.58. Evento Intermedio Adjunto de Compensación	38
3.59. Evento Intermedio de Lanzamiento de Compensación	38
3.60. Evento Final de Compensación	38
3.61. Evento Inicio de Señal	39
3.62. Evento Intermedio de Captura de Señal	39
3.63. Evento Intermedio Adjunto Interruptor de Señal	39
3.64. Evento Intermedio Adjunto No Interruptor de Señal	39
3.65. Evento Intermedio de Lanzamiento de Señal	40
3.66. Evento de Fin de Señal	40
3.67. Evento de Inicio Múltiple	40
3.68. Evento Intermedio Múltiple	40
3.69. Evento de Fin Múltiple	41
3.70. Evento de Inicio Paralelo Múltiple	41
3.71. Evento Intermedio Paralelo Múltiple	41
3.72. Evento de Fin	42
3.73. Flujo de secuencia en BPMN.	42
3.74. Flujo de mensaje en BPMN.	42

3.75. Asociación en BPMN.	42
3.76. Piscina y carriles en BPMN.	43
3.77. Grupos en BPMN.	43
3.78. Anotación en BPMN.	44
3.79. Objeto de dato en BPMN.	44
3.80. Recopilación de datos en BPMN.	44
3.81. Almacén de datos en BPMN.	45
3.82. Dato de entrada en BPMN.	45
3.83. Dato de salida en BPMN.	45
3.84. Ejemplo de proceso en BPMN.	46
3.85. Ejemplo de diagrama de orquestación en BPMN.	47
3.86. Ejemplo de coreografía en BPMN.	48
3.87. Ejemplo de diagrama de colaboración en BPMN.	49
3.88. Máquinas en Scheduling.	52
3.89. Jobs en Scheduling.	52
3.90. Ambiente de una máquina en Scheduling.	52
3.91. Ambiente de máquinas idénticas en paralelo en Scheduling.	53
3.92. Ambiente de máquinas paralelas con diferentes velocidades en Scheduling.	54
3.93. Ambiente de máquinas sin relacionar en paralelo en Scheduling.	55
3.94. Flow shop en Scheduling.	55
3.95. Flow shop flexible en Scheduling.	56
3.96. Job shop en Scheduling.	57
3.97. Job shop flexible en Scheduling.	58
5.1. Propuesta gráfica tiempo de procesamiento	74
5.2. Propuesta gráfica tiempo de preparación dependiente de secuencia	75
5.3. Propuesta gráfica regla de prioridad FIFO	76
5.4. Propuesta gráfica regla de prioridad LIFO	76
5.5. Propuesta gráfica regla de prioridad Priority	77
5.6. Propuesta gráfica capacidad de procesamiento	78
5.7. Propuesta gráfica función objetivo	79
5.8. Diagrama BPMN problema I	84
5.9. Diagrama BPMN problema I que incluye propuesta de Mejora	85
5.10. Diagrama BPMN problema II	88
5.11. Diagrama BPMN problema II que incluye propuesta de Mejora	89
C.1. Comparativa ¿Qué elementos de Scheduling están en BPMN 2.0 ?	IX
C.2. Comparación entre ambiente alfa de Scheduling y BPMN 2.0	X
C.3. Continuación comparativa entre ambiente alfa de Scheduling y BPMN 2.0	XI
C.4. Comparación entre ambiente beta de Scheduling y BPMN 2.0	XII
C.5. Continuación comparativa entre ambiente beta de Scheduling y BPMN 2.0	XIII

Capítulo 1

Introducción

El modelo de gestión tradicional de las empresas se estructura agrupando en departamentos actividades relacionadas entre sí. Su representación suele ser el organigrama, el cual establece la estructura organizativa, designa las funciones de cada trabajador y establece las relaciones jerárquicas (cadena de mando). Sin embargo, el organigrama no muestra el funcionamiento de la empresa, las responsabilidades, los aspectos estratégicos, los flujos de información ni la comunicación interna. Esta manera de ver la organización, ha conducido casi 300 años la manera en que las empresas llevan a cabo sus objetivos. Sin embargo, debido a la actual dinámica empresarial y a los retos que enfrenta, y que enfrentará la gestión, el modelo se ha agotado.

Este modelo comenzó con los grandes comerciantes artesanos (en los años 1700), publicado por Adam Smith (*The Wealth of Nations*, 1776), se formalizó con las teorías administrativas de Frederick Taylor (*Shop Management*, 1903) y de Henry Fayol (*General and Industrial Management*, 1916), fue puesto en práctica por Henry Ford. Fue exitoso en la era industrial, cuando los monopolios y oligopolios dominaban los mercados, las comunicaciones eran lentas, las tecnologías no eran de fácil acceso y las personas (consumidores y clientes) no eran tan conocedoras de los productos y servicios que los mercados les ofrecían. Claramente, este no es el panorama de estos días.

Actualmente, existe una alta competencia en una gran parte de los mercados, debido a la apertura de éstos, por lo que hay cada vez más oferentes. Los clientes tienen mayor acceso a mejor información, por lo que exigen cada día más. Por ende, el entorno en el que se mueven las empresas exige flexibilidad, rapidez e innovación, además de tecnologías industriales y de información que requieren ser simples y coherentes para integrarse de mejor manera a las organizaciones.

El modelo clásico funcional no tiene las características necesarias para enfrentar estos desafíos, porque se basa en jerarquías de trabajo que hacen a las organizaciones muy especializadas, pero lentas y dispersas (Sepúlveda Jaramillo, Hermes, s.f.). Entonces aparece *Business Process Management (BPM)*, un modelo de gestión organizacional por excelencia originado como consecuencia de la integración entre la administración de la calidad total (*Total Quality Management, TQM*), la reingeniería de procesos, y los desarrollos de las tecnologías de la información. Este modelo busca mejorar la eficiencia a través de la gestión de los procesos de negocios, que se deben modelar, automatizar, integrar, monitorear y optimizar de forma continua. Es en la actualidad el modelo de gestión mejor posicionado en el mundo, y el de mayor proyección para el futuro, debido a que su objetivo es asegurar que todos los procesos de una

organización se desarrollan de forma coordinada, mejorando la efectividad y la satisfacción de todas las partes interesadas (clientes, accionistas, personal, proveedores, sociedad en general).

Al implementar una solución BPM, comúnmente se utiliza un software o un conjunto de herramientas tecnológicas y servicios desarrollados para llevar a cabo el diseño, modelamiento, monitoreo (control y seguimiento de las instancias de las tareas), el cual permite la integración de aplicaciones permitiendo la automatización de cada uno de los procesos. Diversas organizaciones de todo el mundo han optado por utilizar distintas soluciones BPM disponibles en el mercado, debido a que para implementarlas deben utilizar muy poco código (*low code*) o nulo código (*zero code*), con el objetivo de que la integración sea más simple y pueda responder más rápidamente a los cambios en la organización.

Una etapa clave en este modelo de gestión es el modelamiento, ya que se requiere que sea comprensible y escalable. Para ello, se utiliza una notación que estandariza el entendimiento de los procesos, llamada *Business Process Model and Notation 2.0* (BPMN 2.0). Ésta es una notación gráfica que plasma la lógica de las tareas, mensajes, actores y toda aquella información necesaria para que un proceso sea entendido, analizado, simulado y finalmente, implementado. Si bien los procesos son modelados, algunos de los objetivos que persigue la empresa que son sumamente importantes no son capturados por esta notación. Por ejemplo, no hay especificación de velocidades, ni tampoco se consideran los plazos en la que las tareas debieran ser completadas, ni el volumen que debieran tener ciertas operaciones, así como tampoco incorporan las capacidades.

Por otro lado, *Scheduling* es el método que organiza, controla y optimiza el trabajo y las cargas de trabajo en un proceso de producción o de fabricación, una manera de enfrentar de manera eficiente la toma de decisiones, que juega un rol crucial en la industria manufacturera y la de servicios. Ayuda a determinar la asignación óptima de recursos en tareas o *jobs* (procesos), en determinados períodos de tiempo, para cumplir uno o más objetivos (por ejemplo, maximizar el promedio de *jobs* cumplidos en su plazo), cada *job* puede tener un cierto nivel de prioridad, un tiempo de inicio más temprano posible y una fecha de vencimiento. En el ambiente competitivo actual, un *Scheduling* efectivo se ha convertido en una necesidad de supervivencia en el mercado. Esto porque los niveles de servicio cada vez son mayores, por lo que cumplir con los plazos con los clientes resulta muy importante para poder mantener las relaciones comerciales. También, porque tienen que ser capaces de programar sus actividades de tal manera que utilicen sus recursos de manera eficiente.

Scheduling empezó a utilizarse de manera seria en la industria manufacturera al comienzo del siglo XX con el trabajo de Henry Gantt y otros pioneros. En sus inicios fueron considerando distintas formulaciones, utilizando en un principio formulaciones de programación dinámica y programación entera, pasando por la teoría de la complejidad, hasta formulaciones estocásticas. Al mismo tiempo y a medida que las computadoras personales empezaron a ocupar las instalaciones de las fábricas, se desarrollaron sistemas de programación para la generación de programas utilizables en la práctica. Este diseño y desarrollo del sistema fue y está siendo realizado por científicos informáticos, investigadores de operaciones e ingenieros industriales.

Existe una conexión muy estrecha entre el problema de la implementación del flujo de trabajo (BPMN) y el problema de asignación óptima de recursos a tareas en un horizonte de tiempo (*Scheduling*), en la práctica es un solo problema con distintas perspectivas. Históricamente, la literatura no dispone de estudios que las integren. Aunque de distinta manera, ambos métodos se centran en los procesos. BPMN es un estándar utilizado en empresas de todo el mundo para documentar sus procesos, con el objetivo de detectar con facilidad las deficiencias, así como también para automatizar. Sin embargo, no cuenta con

los elementos gráficos para poder representar las capacidades de los procesos, los plazos, etcétera. En esta misma línea, no cuenta con los atributos que permitan realizar, de manera genérica, un control de gestión en los procesos, ya que no incorpora elementos que describan los procesos desde una perspectiva operacional con métricas e indicadores. Por otra parte, *Scheduling* cuenta con distintas implementaciones para poder resolver problemas de procesos en diferentes industrias, pero cuenta con una representación gráfica muy pobre.

El propósito de este estudio es plantear una mejora a BPMN 2.0, utilizando las herramientas propias de *Scheduling*, con el objetivo de que esto pueda acelerar, utilizando la perspectiva del *zero code*, la implementación de mejoras operativas en los procesos de la empresa, considerando elementos que pueden influir en la toma de decisiones. Actualmente, BPMN es la notación más utilizada para describir los procesos de una organización, desde una perspectiva de negocios. Por esta razón, no cuenta con las características que permitan otorgarle un enfoque más operativo, aunque en las organizaciones donde se utiliza sea común que sí existan indicadores de gestión asociados a los procesos (*Key Performance Indicators*, KPI's). Por otra parte, *Scheduling* es una excelente herramienta para describir procesos desde un punto de vista operativo, pero muy pobre desde una perspectiva gráfica y de negocios. Aprovechando esta brecha es que se pensó en estudiar cómo poder conectar estas dos corrientes. Para ello se estudiaron las literaturas de BPMN 2.0 y *Scheduling* por separado, para luego hacer una comparación entre ambas, dar a conocer sus diferencias y así encontrar las características más relevantes que podrían añadirse a BPMN 2.0. De esta forma, se busca entregar una propuesta de mejora al estándar actual, haciéndolo más robusto. Cabe mencionar que en el caso de *Scheduling*, el estudio realizado corresponde al caso determinístico, ya que lo que se busca es comparar la manera en la que ambas herramientas representan un flujo de proceso y no cómo incorporar aleatoriedad en los procesos. En el estudio se encontraron varias diferencias entre BPMN 2.0 y *Scheduling*, resultando una propuesta de mejora la cual es bastante útil, porque captura elementos de *Scheduling* que antes no estaban en BPMN 2.0.

En el estudio se encontraron varios elementos de *Scheduling* que si se pueden representar y otros que no se pueden representar en BPMN 2.0. Existen elementos en los que se parecen, dentro de los cuales se destacan que ambos permiten representar y describir el flujo de un proceso de negocio con máquinas, *jobs*, operaciones, ambientes de máquinas tales como: una máquina, máquinas idénticas en paralelo, flow shop, flow shop flexible, job shop, job shop flexible y open shop. Adicionalmente, el estándar actual de BPMN podría representar restricciones del problema de *Scheduling* tales como: interrupciones, de precedencia, procesamiento por lotes de trabajo, ambos muestran las averías posibles que pudiesen representar las máquinas, elegibilidad de máquinas, bloqueo y recirculación.

Dentro de las diferencias (elementos que no están en BPMN 2.0 pero sí en *Scheduling*) destacan: máquinas en paralelo con diferentes velocidades, máquinas sin relacionar en paralelo, fecha de llegada, tiempo de preparación dependiente de la secuencia, familia de tareas, permutación, no espera y función objetivo. Dentro de éstos elementos solo se propone agregar a BPMN 2.0 los que no lo hagan salir de su calidad de estándar e impliquen en un aumento en su capacidad descriptiva. Entre los elementos consideramos deben incorporarse en versiones posteriores está: tiempo de preparación y procesamiento, capacidad de procesamiento, reglas de prioridad y la función objetivo. Al agregarlos, el estándar es mucho más completo y permite realizar análisis de procesos desde un punto de vista operativo, determinando con mayor facilidad cuáles son las claves para una mejora.

La presente tesis está compuesta de seis capítulos. El capítulo 2 indaga bibliografías existentes de BPMN 2.0 y *Scheduling* de manera separada. Luego, con este estudio, se procede en el capítulo 3 a

describir las características propias de ambas corrientes para luego en el capítulo 4 analizar la información recopilada en el capítulo anterior para en el capítulo 5 encontrar las similitudes y diferencias existentes entre ambos enfoques y presentar una propuesta gráfica de mejora a BPMN 2.0 y, finalmente, en el capítulo 6 las conclusiones.

1.1. Metodología

Para lograr los objetivos planteados anteriormente, se requiere realizar una recopilación de la información bibliográfica existente de BPMN y de *Scheduling*. De esta forma obtenemos la información existente, contando con un fuerte manejo de antecedentes como información del estándar 2.0, póster e información del ambiente de máquinas, restricciones y función objetivo de *Scheduling*; pero, además, es necesario estudiar ambas corrientes para poder obtener sus diferencias y poder proponer una propuesta de mejora para BPMN 2.0. Así, se plantea la metodología resumida en la siguiente figura y que se describe a continuación.

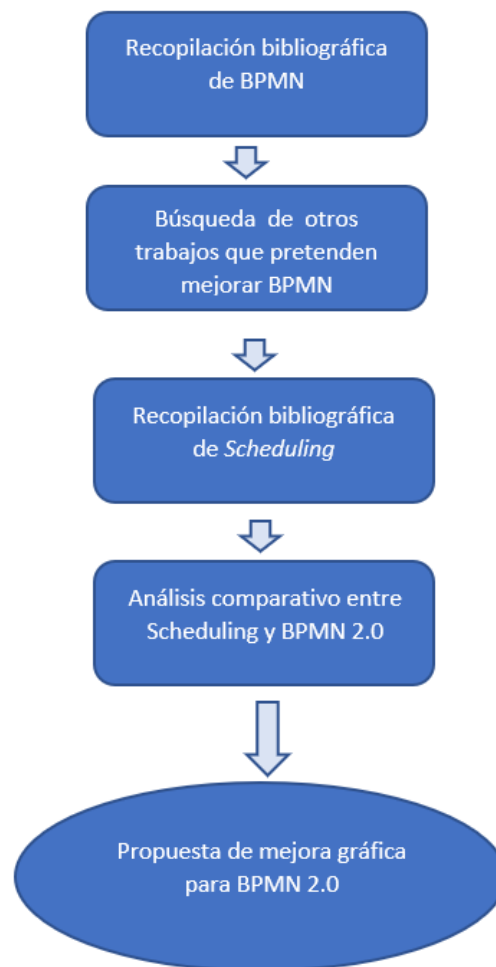


Figura 1.1: Metodología a seguir
Fuente: Elaboración propia

Recopilación bibliográfica de BPMN 2.0: Se requiere recopilar información sobre el estándar BPMN, para así poder tener una visión completa de las características y alcances del estándar actual. Dentro de los contenidos abordados está su definición, historia, evolución desde la versión 1.0 hasta la 2.0, los objetivos del estándar, los distintos clases de modelo que BPMN es capaz de representar, las cuatro categorías de elementos gráficos que tiene BPMN 2.0 las cuales fueron obtenidas desde el póster publicado por la BPM Offensive Berlin en el año 2005. Un ejemplo de modelado de un proceso de negocio de apertura de una cuenta corriente para mostrar la utilización de este estándar.

Búsqueda de otros trabajos que pretenden mejorar BPMN: Se buscaron varios trabajos que pretenden dar mejoras a BPMN con el fin de saber que se había hecho hasta el momento y no proponer lo mismo. Dentro de los trabajos encontramos varios que agregan características operativas, como: dimensión temporal, rendimiento, objetivos, etcétera. Pero, van en una línea distinta al trabajo presentado en esta tesis.

Recopilación bibliográfica de Scheduling: Por otro lado, se busca información sobre *Scheduling*, dado a que se desea encontrar una comparativa entre este último y BPMN 2.0. Para ello se detalla la información que permite modelar un problema de *Scheduling*. Esta información será obtenida de la quinta edición del libro de Michael Pinedo. Se describirá en detalle la tripleta alfa, beta y gamma, la cual describe un problema de *Scheduling*. El campo alfa describe el ambiente de máquinas a utilizar, en este campo se explica en detalle cada ambiente de máquinas y se realizó una ilustración para poder dar mayor entendimiento a estos conceptos y para ayudar a entender mejor el modelado de *Scheduling*. El campo beta entrega los detalles de las características de procesamiento y las restricciones, es por ello que se describirán de manera sencilla cada uno de los tipos de restricciones con ejemplos para dar mayor claridad del contenido expuesto. Finalmente, el campo gamma describe el objetivo a ser minimizado, en el cual se explicarán los distintos tipos de funciones objetivos. Esta información será utilizada tanto para la elaboración de tablas resumen y comparativas.

Análisis comparativo entre *Scheduling* y BPMN 2.0: Una vez que se tienen los antecedentes esenciales, se comienza con el desarrollo de las tablas resumen de los antecedentes recopilados anteriormente de BPMN 2.0 y *Scheduling*, respectivamente. Dichas tablas se encuentran en los anexos.

Luego se procede a realizar la comparación entre *Scheduling* y BPMN. Para ello se realiza una tabla con tres columnas: la primera contiene un elemento de *Scheduling*, la segunda corresponde al elemento de BPMN 2.0 al cual, de existir, se asemeja y la tercera columna es una explicación de esta comparación. Al realizar esta comparativa podemos detectar que elementos de *Scheduling* tienen o no una representación en BPMN 2.0. Y se realizan dos listas, una con los elementos que si se pueden representar y la segunda con los elementos que no son representables.

Propuesta de mejora gráfica para BPMN 2.0: Finalmente, con los elementos de *Scheduling* que no tienen representación, se propone una propuesta de mejora para BPMN 2.0 con elementos gráficos que se agregan al estándar para que este sea más completo. Además, en la última parte del desarrollo se resuelven dos problemas de *Scheduling* sencillos y se grafican en BPMN 2.0 con las respectivas propuestas de mejora gráfica, para dar un ejemplo concreto respecto a lo que se propuso.

Capítulo 2

Marco Teórico

Una decisión importante que debe tomar cualquier empresa es el modelo de gestión bajo el cual funcionará. Actualmente, existen dos grandes corrientes: el modelo de gestión tradicional y el modelo de gestión por procesos. El modelo de gestión tradicional (o funcional) surge teóricamente en el año 1776 con el trabajo de Adam Smith, en cuyo libro “La Riqueza de las Naciones” aludió a que el bienestar social se maximiza con la división del trabajo. Según esta teoría, la división del trabajo, a su vez, se profundiza a medida que se amplía la extensión de los mercados y, por ende, la especialización. Luego, el modelo tradicional se formalizó con los primeros pasos del pensamiento administrativo desarrollados por los cuatro principios de Taylor (Taylor, Fayol, y Lender, 2003): organización del trabajo, selección y entrenamiento del trabajador, cooperación y remuneración por rendimiento individual, y, finalmente, responsabilidad y especialización de los directivos en la planeación del trabajo, dividiendo así a las organizaciones en departamentos en los cuales cada uno realiza sus tareas específicas, generando una división del trabajo más acentuada y mayor eficiencia.

Los problemas no abordados por Taylor, el cual puso mucho énfasis en la fábrica, dejando de lado lo que tiene relación con los diferentes niveles administrativos, fueron desarrollados por Fayol (Fayol, 1971), años más adelante. Identificando la diferencia existente entre los niveles gerenciales y de supervisión, le dio más importancia al primero, ya que éste tiene mayor influencia en la organización. Además, estableció las funciones administrativas que tenían que llevar a cabo los directivos, detallando que toda empresa industrial debe tener seis grupos de funciones: técnicas, comerciales, financieras, seguridad, contables y administrativas, lo que se conoce actualmente como la definición funcional de la administración.

Más adelante, estos conceptos fueron implementados por Henry Ford en la producción de su modelo de automóvil Ford T, los cuales se generalizaron con rapidez al resto de las industrias, dándole el éxito en la era industrial al modelo tradicional de gestión, que tiene como directriz principal aumentar la eficiencia de una organización mediante la forma y disposición de sus departamentos y de sus relaciones estructurales. El énfasis en la estructura es la principal característica del modelo tradicional, lo cual genera que la empresa se centre en las necesidades de la empresa y no en las del cliente, caminando hacia la ineficiencia, incrementando la burocracia, lo cual multiplica las tareas a realizar y los tiempos de respuesta hacia el cliente.

Algunas organizaciones se dieron cuenta de los problemas asociados con el modelo tradicional, el cual no se enfoca en el cliente y optaron por reorientar su visión hacia ellos y sus procesos adoptando

el modelo de gestión por procesos. Esto implica un cambio importante en la operación de la empresa, al migrar de una operación funcional a una operación administrada por procesos, de manera de responder rápidamente a las necesidades del mercado actual y no perder competitividad. La filosofía dentro de la cual se desarrolla el modelo de gestión por proceso es llamada *Business Process Management (BPM)*.

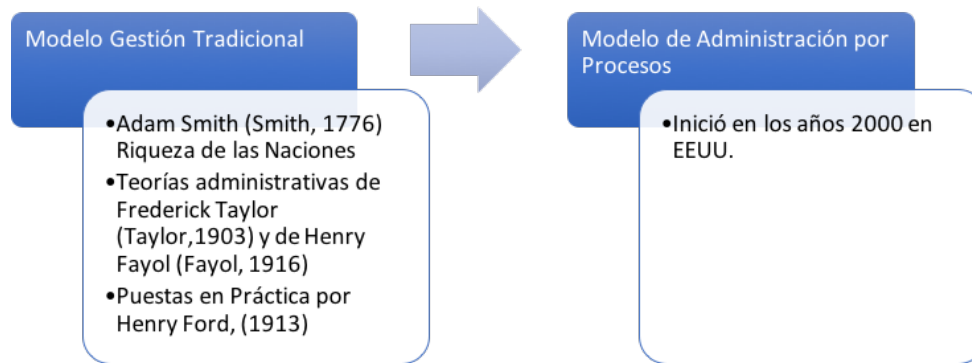


Figura 2.1: Evolución modelos de gestión
Fuente: Elaboración propia

Sin embargo, BPM no es la única filosofía, ya que existe otra metodología para abordar los problemas de gestión de procesos llamada *Scheduling*. Hasta ahora, estos dos enfoques no conversan.

Por una parte, está BPM, cuyo objetivo es mejorar la eficiencia a través de la gestión de los procesos de negocios, que se deben modelar, automatizar, integrar, monitorear y optimizar de forma continua. Este tipo de enfoque no podría tener el impacto que tiene en la actualidad, sin que contara de una notación estandarizada, que permite que todas las personas que trabajan con procesos de negocio hablen un lenguaje común, desde los ejecutivos hasta los trabajadores. Este lenguaje es *Business Process Model and Notation (BPMN)*, notación que se utiliza para representar los modelos de procesos de negocio. A pesar que ha ido incorporando mejoras desde su creación como BPMN 1.0 en 2004, hasta llegar a BPMN 2.0, utilizado desde 2010, hay varios aspectos relevantes en un proceso que no ha incluido este tipo de modelamiento. Entre estos aspectos se destaca que la notación no incorpora las capacidades ni las velocidades de los procesos y servidores (computadores, personas, máquinas, etcétera), no se consideran los tiempos de entrega ni de procesamiento de las tareas. BPM y BPMN se abordan con mayor detalle en la sección 2.1.

Por otro lado, está *Scheduling*, un área de estudio que se enmarca dentro de la Optimización, en la que se busca encontrar el orden en el que la producción resulta ser más eficiente, intentando cumplir con los plazos establecidos. Se modelan los procesos productivos a través de los ambientes de máquinas, que son la manera en la que se estructuran los flujos de cómo se procesarán los *jobs*. Se denomina *job* (o trabajo) a las órdenes de producción y máquinas a los recursos que permiten realizar el procesamiento de dichas órdenes para convertirlas en productos.

Esta área de la Optimización comenzó a ser tomada de manera seria en la industria manufacturera al principio del siglo XX, con el trabajo de Henry Gantt, el creador del famoso Diagrama de Gantt (o Carta Gantt), publicado en 1910. Tuvieron que pasar varios años para que aparecieran las primeras publicaciones de *Scheduling* en la literatura de ingeniería industrial e investigación operativa. De acuerdo a (Pinedo, 2016) uno de los primeros trabajos apareció en la revista científica *Naval Research Logistics Quarterly*

que publica artículos en el campo logístico, especialmente en las áreas de investigación operativa, estadística aplicada y modelado cuantitativo, al principio de los 1950's contenía resultados del trabajo de Smith, Johnson y J.R Jackson. Smith publicó su artículo titulado "*Various Optimizers for Single-Stage Production*", el cual desarrolla un teorema para el caso de un solo escenario de producción minimizando ciertas funciones u optimizándolas de acuerdo a los tiempos de entrega de los trabajos (Smith, 1956). Diez años más tarde aproximadamente, durante los años 60, se llevó a cabo una cantidad importante de trabajo en programación dinámica y programación entera como formulación a problemas de *Scheduling*. Luego del trabajo de Richard Karp en la teoría de la complejidad en 1972, con la publicación de su famosa lista de 21 problemas NP Completos, los cuales tratan sobre combinatoria y teoría de grafos. Dentro de esta lista destacan algunos problemas tales como: problemas de programación lineal entera, problema del conjunto independiente, problema de la cobertura de vértices, problema de la mochila, problema de corte máximo, etcétera. Dichos problemas se resuelven con un algoritmo, el cual consiste en una serie de pasos delimitados (Karp, 1972). Posterior a esto, se dio lugar a que las investigaciones se centraran, principalmente, en la complejidad jerárquica de los problemas de *Scheduling*. Discutiremos detalladamente *Scheduling* en la sección 2.2.

2.1. Business Process Management (BPM)

"Es una filosofía la cual es parte de un campo en la gestión de operaciones que se centra en mejorar el rendimiento corporativo mediante la gestión y la optimización de los procesos empresariales de una empresa." (Panagacos, 2012)

"BPM permite a las organizaciones ser más eficientes, más eficaces y más capaces de cambiar que en un enfoque de gestión jerárquica tradicional funcionalmente enfocado."

En los años 1980, hubo un foco importante en filosofías de calidad. Esto se tomó la atención de las gerencias, con el foco en *Total Quality Management* (TQM), *Six Sigma*, ISO y los conceptos de Kaisen.

Unos años más tarde, en 1990, fue el auge de la integración y la mejora de procesos del negocio. Gracias a esto, aparecieron los estándares del flujo de trabajo, rediseñando los procesos de negocio para conseguir su optimización de acuerdo con los objetivos establecidos en el plan estratégico de la empresa, con el método *Business Process Reengineering* (BPR) y la filosofía para eliminación de desperdicios que afecten a la organización, bajo el nombre de filosofía *Lean*.

A mediados de los 90 se introduce el concepto de automatización con sistemas informáticos destinados a apoyar la planificación en una organización, estos sistemas son conocidos como *Enterprise Resource Planning* (ERP). A finales de los 90 y principios del año 2000 llegan los sistemas con foco en el cliente y experiencia del cliente acompañado de una estrategia orientada a la satisfacción y fidelización de estos, *Customer Relationship Management* (CRM).

Luego de la era de la información, pasamos a la era de los procesos con la sistematización. Para poder lograrlo es necesario un conjunto de herramientas llamadas *Business Process Management Systems* (BPMS) (Jeston y Nelis, 2014). Para apoyar esta estrategia, es necesario contar con un conjunto de herramientas que den el soporte necesario para cumplir con el ciclo de vida de BPM (deben diseñar, modelar, organizar, documentar y optimizar de forma continua) y con ellas se construyen aplicaciones BPM. Normalmente, siguen una notación común denominada *Business Process Model and Notation* (BPMN).

Otras poseen una notación propia y son capaces de generar código basado en lenguajes de programación.

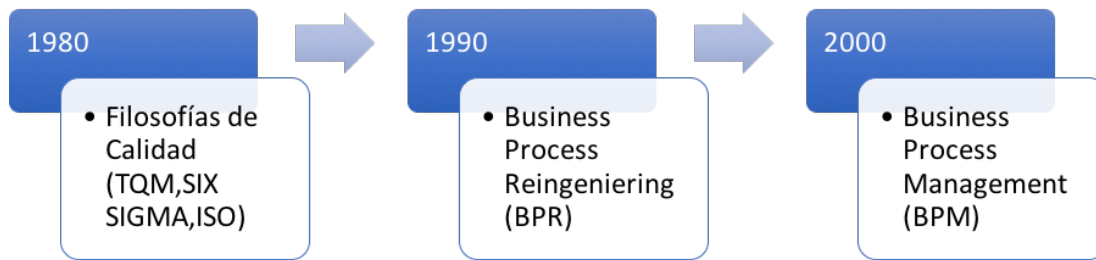


Figura 2.2: Línea de Tiempo BPM.

Fuente: Elaboración propia

Cualquier organización está en la búsqueda de la manera de mejorar las cosas que hacen para el beneficio de las partes interesadas en el negocio, esta idea se encuentra en el corazón de la gestión de procesos de negocio (BPM), una filosofía de gestión con foco en mejora los procesos operacionales de la organización. La filosofía BPM pone un mayor énfasis en los procesos de negocio de la empresa, es por ello que se quiere dar una definición a lo que es un proceso de negocio la cual se detalla a continuación.

Proceso de Negocio o *Business Process*

Un proceso de negocio es un conjunto de tareas relacionadas lógicamente realizadas para alcanzar un resultado definido del negocio (Davenport, Short, y cols., 1990). Este resultado del negocio puede ser generar productos o servicios y atraviesan varias áreas funcionales de la empresa. Para que los resultados de la empresa sean mejorados es que estos procesos son modelados. El modelar consiste en definir un mundo abstracto y teórico tal que las conclusiones que se puedan sacar de él coinciden con las manifestaciones aparentes del mundo real, un modelo es la interpretación explícita de lo que una persona entiende de una situación, o tan solo de las ideas de uno acerca de esa situación (Luza M, 2009).

Es muy importante el modelado de procesos de negocio, debido a que sirve para guiar la forma en que se trabaja en las organizaciones modernas ayudando en la comunicación entre las personas de la empresa que ven cómo organizar el trabajo de la mejor manera para lograr un entendimiento compartido, ayuda a comprender de mejor forma la realidad, pues un modelo es una simplificación de la realidad comprendiendo los elementos y relaciones, disminuye la complejidad pues se pueden entender modelos complejos en su totalidad ayudando a la comprensión, es un respaldo para los programas de mejora,

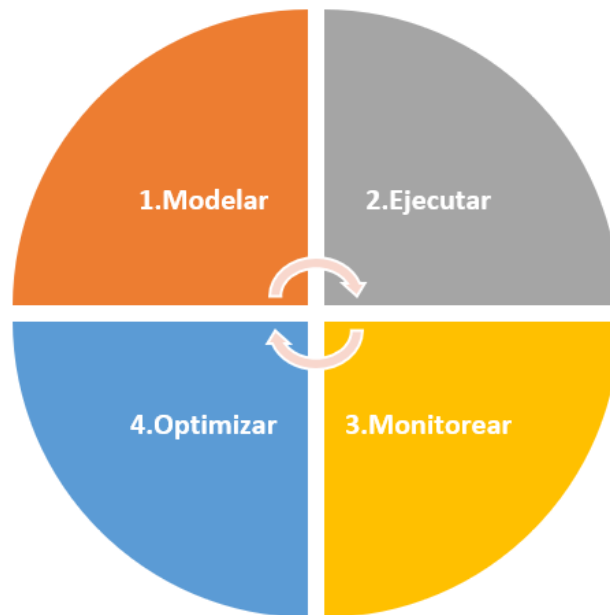


Figura 2.3: Ciclo de Vida Business Process Management.
Fuente: Elaborada por el autor en base a ciclo de mejora continua.

forman parte del material de capacitación debido a que los modelos de proceso son creados observando las operaciones de la empresa en marcha (León León y Asato España, 2009).

El foco de estudio está en la notación común BPMN 2.0, ya que es un estándar para el modelamiento de los procesos de negocios.

Lo que hace BPMN es definir un *Business Process Diagram* (BPD), el cual se basa en una técnica de diagramas de flujo adaptada para crear modelos gráficos de procesos de negocio. Un BPD está constituido por un conjunto de elementos gráficos, estos elementos permiten desarrollar diagramas simples que resultan familiares para la mayoría de los trabajadores en una empresa (White, 2004).

2.1.1. Notaciones para modelado de procesos de negocio

Según (Rodríguez, Fernández-Medina, Piattini, y Trujillo, 2008) el principal objetivo que tiene el modelado de procesos de negocios es poder representar una descripción de la realidad para poder comprenderla de la mejor forma para a futuro poder incorporar mejoras a los determinados procesos de negocio. Para poder cumplir con aquello es primordial poder contar con una notación que permita modelar con la mayor claridad posible.

Hoy en día, es posible identificar en la industria del modelado de procesos de negocio dos importantes notaciones, *Unified Modeling Language* (UML) y *Business Process Modeling and Notation* (BPMN).

BPMN es una notación basada en diagramas de flujo para definir procesos de negocio, desde los más simples hasta los más complejos y sofisticados para dar soporte a la ejecución de procesos cuyo

objetivo principal es proporcionar una notación que sea fácilmente comprensible por todos los usuarios de negocios, desde los analistas de negocio que crean los borradores iniciales de los procesos, hasta los desarrolladores técnicos responsables de la aplicación. Proporciona un lenguaje gráfico común, con el fin de facilitar su comprensión a los usuarios de negocios (White, 2004).

UML, también conocido como el lenguaje unificado de modelado, es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un “plano” del sistema (modelo), incluyendo aspectos conceptuales tales como procesos, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados. El principal objetivo de UML es “proporcionar a los arquitectos de sistemas, ingenieros de software y desarrolladores de software” herramientas para el análisis, diseño e implementación de sistemas basados en software (Martin y Scott, 1999).

UML es un lenguaje que facilita a los desarrolladores la especificación, visualización y documentación de modelos de sistemas de software. Está dirigido en líneas generales a los arquitectos de software e ingenieros de software. Fue desarrollado como un medio para mejorar el proceso de desarrollo de software, desde el diseño de la arquitectura hasta la implementación de la aplicación, para ser utilizado por personas con conocimientos técnicos (analistas de sistemas y programadores). BPMN está dirigido a los analistas de negocio, arquitectos de sistemas e ingenieros de software. Fue desarrollado para mejorar el ciclo de vida del desarrollo de procesos desde el diseño de los mismos. BPMN está emparentado con UML por el hecho que ambos definen una notación gráfica para los procesos de negocio, sin embargo, BPMN y UML usan enfoques diferentes para modelarlos. UML en general ofrece un enfoque orientado a objetos para modelar aplicaciones, mientras que BPMN toma un enfoque centrado en los procesos

Ambas son herramientas que ayudan a la organizaciones a diseñar la arquitectura empresarial modelando los proceso de negocio mediante diagramas, pudiendo también automatizar el funcionamiento de principio a fin y permitiendo su monitorización y control. Ambos son estándares y ambos son aceptados en el ámbito de los negocios, pero UML es dominante en la industria del software. Esto se debe a que es más complejo y no tiene como objetivo ser entendido por todos los miembros de la organización. Es por ello que se opta por trabajar con el estándar BPMN.

2.1.2. Business Process Model and Notation (BPMN)

La literatura que ha estudiado la optimización de los procesos de negocios no es una idea nueva, esta idea tiene sus raíces en la Revolución Industrial, la cual centró la visión que debe tener la optimización de procesos de negocio hacia la sistematización, que es el establecimiento de un sistema u orden que tiene por objetivo permitir obtener los mejores resultados posibles de acuerdo al fin que tenga que alcanzar, aumentando así los ingresos y beneficios de las organizaciones.

Para lograr la sistematización, los procesos deben ser mapeados, seguidos, optimizados y editados. La disciplina para hacerlo no es particularmente nueva; ni siquiera los sistemas para soportar la gestión de procesos de negocios son nuevos. Lo que sí es nuevo es el interés centrado en la utilización de una práctica de gestión para dar apoyo a la agilidad empresarial y una norma mundial para especificar los procesos de negocio ya sean totalmente automatizables: BPMN. El objetivo de BPMN es apoyar la gestión de procesos de negocio (BPM), tanto para usuarios técnicos como para usuarios empresariales, proporcionando una notación intuitiva para los usuarios empresariales, pero capaz de representar una semántica de procesos

MODELADO DEL PROCESO - ACTUAL



Figura 2.4: Ejemplo Modelado de un proceso de Negocio.
Fuente: Julio César, Modelado de Procesos (2012)

compleja.

BPMN es un estándar para el modelado de procesos de negocio, que proporciona una notación gráfica para especificar procesos de negocio en un *Business Process Diagram* (BPD) (Simpson, 2005).

La notación de modelado de proceso de negocio (BPMN) es el apogeo de dos corrientes de trabajo de finales de la década de 1990 e inicios de este siglo. Una de las corrientes se focalizó en la planificación y gestión del flujo de trabajo y tareas asociadas, mientras que la otra se dedicó al modelado y la arquitectura.

Los pilares de esta notación nacen de la fusión de *Business Process Management Initiative* (BPMI), una organización sin fines de lucro que existe para promover la estandarización de los procesos comerciales comunes, como medio de fomentar el comercio electrónico y el desarrollo *Business to Business* (B2B)¹. Esta organización (la BPMI) desarrolló en el 2001 un lenguaje de modelado de procesos de negocio.

Sin embargo, surgió la necesidad de una representación gráfica, debido a que una gran cantidad de empresas que se unieron con el objetivo común de consolidar los principios subyacentes del modelado de proceso de negocio, consideraron que era muy importante llegar a un acuerdo sobre una única notación adaptada por otras herramientas y personas, y que esta fuera simple y acordada para hacer el aprendizaje

¹B2B: Es aquel en el que una empresa vende a otra empresa y no al consumidor final, por ejemplo, la relación que existe entre un fabricante y el distribuidor de un producto y también a la relación entre el distribuidor y el comercio minorista pero no a una relación entre el comerciante y su cliente final.

transferible, proporcionando los mecanismos para pasar del diseño original hasta la ejecución.

Luego, el número cada vez mayor de adopciones de las empresas y el creciente interés por esta notación causó una fusión del antiguo *Object Management Group* (OMG), también una organización sin fines de lucro que promueve el uso de tecnología orientada a objetos mediante guías y especificaciones, con la BPMI que surgió en el 2004, creándose exitosamente una sola organización que se especializó en el modelado de sistemas y que mantuvo el nombre OMG. BPMI se centró en los procesos de negocio, mientras que la OMG se enfocó en el problema de modelado genérico con su arquitectura orientada a modelos, y especialmente en el modelado de sistemas de software.

Con esta fusión, el lenguaje BPMN fue originalmente publicado en mayo 2004 en su versión 1.0 como una notación gráfica para representar el diseño gráfico de los procesos de negocio. En febrero de 2006 la especificación 1.0 fue adoptada como un estándar y en febrero de 2008 la OMG publicó la versión final de BPMN 1.1. Luego, a finales del 2008, se realizaron solo pequeñas correcciones a la versión 1.1, dando origen a la versión 1.2.

En el año 2010 nace una de las versiones más recientes de BPMN, la 2.0, que se diferencia notablemente de la versión 1.0 en muchos aspectos, ya sea añadiendo nuevas características o cambiando las propiedades de los elementos como se definen en las versiones anteriores del estándar, realizando grandes cambios oficializando su versión 2.0 como nuevo estándar y en el año 2013 pasa a ser BPMN 2.0 una norma ISO.



Figura 2.5: Evolución de BPMN.

Fuente: Elaboración propia

Finalmente, la OMG no solamente hizo que el lenguaje BPMN se estableciera como estándar, sino que también logró un sustento técnicamente detallado, con *Model-Driven Architecture* (MDA) ², que es un acercamiento al diseño de software, propuesto y patrocinado por el OMG, integrándolo con lenguajes para expresar el diseño de software, diseño de sistemas de ingeniería e incluso diseño de hardware.

Con BPMN ahora es posible modelar un conjunto diferente de procesos, tales como orquestaciones, coreografías y colaboraciones (Chinosi y Trombetta, 2012), pudiendo comunicar una amplia variedad de información a diferentes públicos, siendo esto uno de sus propósitos, además de “crear un puente estandarizado para la brecha entre el diseño del proceso empresarial y la implementación del proceso” (White, 2004). Parte de esa implementación es el uso de reglas de decisión, que permiten la automatización de algunas de las actividades a realizar dentro de los procesos.

²MDA: Es una arquitectura que proporciona un conjunto de pautas para estructurar especificaciones expresadas como modelos.

El objetivo principal de BPMN es proporcionar una notación estándar fácilmente comprensible por todos los interesados del negocio. Estos incluyen los analistas de negocios que crean y perfeccionan los procesos, los desarrolladores técnicos responsables de implementarlos y los gerentes de negocios que los monitorean y administran. En consecuencia, BPMN sirve como un lenguaje común, superando la brecha de comunicación que ocurre con frecuencia entre el diseño de procesos de negocio y la implementación.

El modelado de procesos empresariales se utiliza para comunicar una amplia variedad de información a diferentes públicos. BPMN está diseñado para cubrir muchos tipos de modelado y permite la creación de segmentos de proceso, así como procesos de negocio de extremo a extremo, en diferentes niveles de fidelidad. Dentro de la variedad de los objetivos de modelado de procesos, hay dos tipos básicos de modelos que se pueden crear con un *Business Process Diagram* (BPD):

- Procesos B2B de colaboración (públicos): un proceso B2B colaborativo describe las interacciones entre dos o más entidades empresariales. Los diagramas para estos tipos de procesos son generalmente desde un punto de vista global.
- Procesos de negocio internos (privados): un proceso de negocio interno generalmente se centrará en el punto de vista de una única organización empresarial. Aunque los procesos internos a menudo muestran interacciones con los participantes externos, definen las actividades que generalmente no son visibles para el público y, por lo tanto, son actividades privadas (White, 2004).

2.2. Scheduling

La importancia de *Scheduling* en la industria manufacturera y la industria de servicio tiene que ver con la necesidad que existe por cumplir con los plazos, organizando de manera eficiente cómo operar las órdenes de trabajo. Por ejemplo, *Scheduling* podría utilizarse en una fábrica para saber qué líneas de producto se necesita fabricar primero y cuales después. Otro ejemplo en el que se utiliza *Scheduling* es para poder determinar la cantidad óptima de vendedores en una tienda se necesitan, además de las horas en las que cada trabajador debe estar en su puesto de venta.

Aunque empezó a formalizarse con Henry Gantt y el Diagrama de Gantt (1910), con Jackson (1955) y Smith (1956) se crearon las primeras reglas de ordenamiento para el problema de *Scheduling* de una sola máquina. Se introdujeron, además, conceptos importantes como la ponderación de *jobs* y se diferenció entre objetivos, como disminuir la mayor tardanza, minimizar el tiempo ponderado de completado, etcétera. Es importante señalar que el problema de una sola máquina es la base para ambientes de máquina y restricciones más complejas.

Con el tiempo, se avanzó en resolver problemas más complejos, en términos de configuraciones y algoritmos. En 1977, Weglarz formuló el problema de máquinas paralelas con preprocesamiento. Esto implica que el procesamiento de una tarea se puede detener antes de terminar y que la cantidad de trabajo procesado no se pierde.

Más adelante, se trabajó en resolver problemas de *Scheduling* en máquinas paralelas sin preprocesamiento. En este problema, en palabras simples, cada tarea se procesa en una sola máquina. Una solución para máquinas uniformes (igual velocidad de procesamiento) fue propuesta por Horowitz y Sahni (1976). Una generalización de este problema se presentó en un paper de Dror (1987).

Para completar, se estudió el *Scheduling* en *job shops*. En este problema, existen conjuntos separados de máquinas especializadas, llamados *estaciones de trabajo*. Cada trabajo necesita ser procesado en al menos una estación, dependiendo de sus características. Este problema fue formulado por Manne (1960), Fisher (1973) y el mismo Fisher presentó un paper en el que mostraba dos formulaciones matemáticas y una estrategia de solución (Blazewicz, Dror, y Weglarz, 1991).

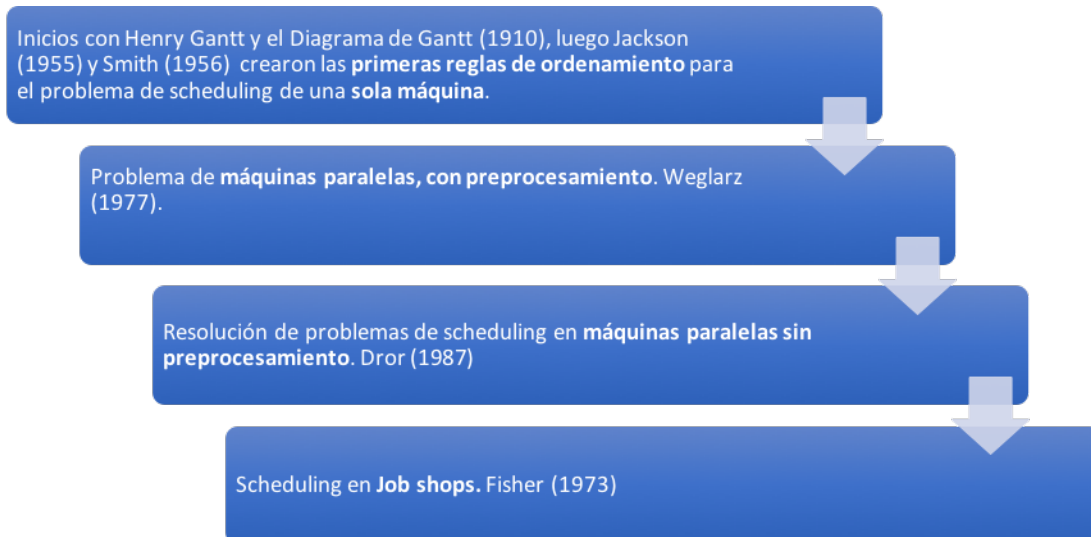


Figura 2.6: Evolución Scheduling.

Fuente: Elaboración propia

Aunque es una herramienta muy potente en términos de modelamiento y resolución de problemas, *Scheduling* cuenta con una representación gráfica bastante pobre, que hace que su interpretación no sea tan robusta como en BPMN 2.0.

2.3. Conexión entre las dos corrientes en la literatura

Ambos tipos de literaturas no interactúan entre sí, creando dos visiones aisladas para abordar el mismo problema. Dicho de otra manera, la literatura dedicada a estudiar esta temática no dispone de estudios que integren ambas corrientes para entender la conexión existente entre el problema de la implementación del flujo de trabajo y el problema de modelamiento, el cual en la práctica es un problema con distintas perspectivas, generando una brecha entre los estudios que la abordan los procesos desde el modelamiento y los que lo hacen desde una perspectiva de resolución. Dado esto, y con el motivo de contar con una base de estos temas, es que son revisados ambos tópicos de literatura en el presente capítulo.

La principal deficiencia de BPMN es no contar con elementos que permitan evaluar el rendimiento de un proceso. Esto implica que no cuenta con mecanismos de control por defecto, por lo que la definición de indicadores de rendimiento no está representada en los diagramas. Por otra parte, *Scheduling* cuenta con una batería de atributos que permiten describir y modelar procesos en términos operacionales (rendimiento, capacidades, etcétera). Dada esta diferencia, se ve la oportunidad de incorporar elementos de *Scheduling* a BPMN 2.0, que impulsen una manera de describir los procesos de forma más completa, dotándolos de herramientas que permitan realizar un control de gestión de manera explícita.

Capítulo 3

BPMN y Scheduling

El modelado de procesos de negocios puede ayudar a la comunicación y comprensión entre las personas, contribuyendo a lograr un entendimiento compartido, además de guiar el trabajo dentro de la organización. BPMN hace posible lo anterior, proporcionando un lenguaje estándar para poder representar los procesos de negocio, permitiendo detectar con facilidad las deficiencias, así como también para poder implementar mejoras. Sin embargo, esta notación no representa los objetivos de la organización. Si bien los procesos son modelados, algunos de los objetivos que persigue la empresa, que son sumamente importantes, no son capturados por esta notación. Por ejemplo, no hay especificación de velocidades, ni tampoco se consideran los plazos en la que las tareas debieran ser completadas, ni el volumen que debieran tener ciertas operaciones, así como tampoco incorporan las capacidades.

Por otra parte, *Scheduling* cuenta con distintas implementaciones para poder resolver problemas de procesos en diferentes industrias, determina la asignación óptima de recursos en tareas (procesos), en determinados períodos de tiempo, para cumplir uno o más objetivos (por ejemplo, maximizar el promedio de tareas cumplidas en su plazo), cada *job* puede tener un cierto nivel de prioridad, un tiempo de inicio más temprano posible y una fecha de vencimiento, pero cuenta con una representación gráfica muy pobre.

Aunque de distinta manera, ambos métodos se centran en los procesos, en la práctica abordan el mismo problema, pero cada uno desde su perspectiva, pudiéndose complementar y sacar una mejor solución para las organizaciones. Históricamente, la literatura no dispone de estudios que integren esta conexión por ende se analizaran las características de cada método para posteriormente analizar las similitudes y diferencias entre ellos y poder seleccionar las mejores características para agregar estas mismas a una mejora de BPMN 2.0.

En el presente capítulo se presenta la metodología a seguir en esta tesis para luego explicar las características del modelado de procesos de negocios BPMN 2.0, su definición, las tres formas en las que puede utilizarse, para más adelante, presentar las principales características, definiciones y configuraciones presentes en *Scheduling*.

3.1. Business Process Model and Notation 2.0

La versión 2.0 de BPMN es la última especificación del grupo OMG en el modelamiento de procesos de negocios, un estándar más detallado para el modelo de procesos de negocio, mediante el uso de un conjunto más rico de símbolos y notaciones para los diagramas de procesos de negocio.

3.1.1. Clase de Modelos

BPMN es capaz de representar una gran cantidad de niveles de detalle cubriendo distintas clases de modelo, desde los más simples a los más complejos distinguiendo, en el modelado BPMN existen tres niveles de procesos:

- Mapas de Procesos
- Descripción de Procesos
- Modelo de Proceso

Los mapas de procesos son simples diagramas de flujo, con el único detalle que contiene el nombre de las actividades. La descripción de procesos brinda un poco más de información sobre el proceso, como los roles (personas encargadas de llevar a cabo), datos e información. Por último, los modelos de proceso son diagramas de flujo con bastante mayor grado de detalle, con la información suficiente para poder analizar un proceso e importarlo de manera directa a herramientas que ejecuten procesos.

3.1.2. Instancia de Proceso y Token

Para dar mayor claridad al comportamiento de los procesos en los diagramas, es que se muestra a continuación conceptos importantes para no perder la lógica de BPMN. Según lo presentado en (Freund, Rücker, y Hitpass, 2014) se tienen las siguientes definiciones:

Instancia de proceso: Bajo instancia de proceso se entiende un proceso concreto en la realidad, es decir, casos reales, como por ejemplo la reclamación de un cliente crea una instancia del proceso de reclamaciones. Algunos procesos se instancian solo un par de veces en el año y otros más a menudo.

Token (marca): El concepto de marcas se utiliza para visualizar y probar el comportamiento de los procesos diseñados. Las marcas recorren en forma de una animación la lógica por los flujos normales y los de excepción. El token se puede definir como una construcción teórica que podría compararse con un mapa carretero o de una ciudad sobre el cual uno pone un auto de juguete y desplaza este desde un lugar a un destino. Si el auto llega a un cruce el chofer tiene que tomar la decisión que camino seguir dependiendo de la ruta, el tráfico, etc. Es posible que el auto llegue a un cruce donde debe seguir derecho y además doblar porque tiene que dejar pasajeros en dos puntos diferentes. En este caso debería ser clonado, lo que lógicamente no es real. En la realidad los pasajeros van a tomar otro vehículo mientras uno sigue a su destino, o bien el vehículo va primero a un lugar, se vuelve y luego va al otro. En todo caso en nuestro ejemplo el mapa representa el modelo de proceso y una ruta representa un caso de recorrido que

puede tomar una instancia. Si usted aplica el concepto de token va a poder entender cualquier modelo de proceso, incluso los más complejos.

Para poder describir el modelado en BPMN, se utiliza un conjunto de elementos gráficos especializados para poder realizar diagramas simples de procesos de negocio (en inglés *Business Process Diagram*, BPD). Este grupo de elementos gráficos principales ayudan a la sencillez y legibilidad de los modelos de procesos de negocio, es decir, permiten el fácil desarrollo de los diagramas que son entendibles por todos los *stakeholders* de una organización, al mismo tiempo de manejar la complejidad inherente a los procesos de negocio, cumpliendo de esta forma con los objetivos de BPMN. Por ejemplo, las decisiones a tomar son representadas por rombos y las actividades que se desarrollarán por rectángulos. El enfoque adoptado para manejar los objetivos de complejidad y sencillez del modelo, fue organizar los elementos gráficos de la notación en cuatro categorías específicas (Braun y Esswein, 2014).

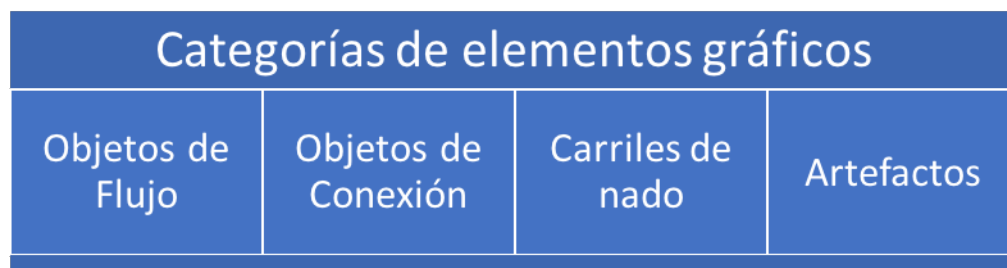


Figura 3.1: Categorías de elementos gráficos en BPMN.

Fuente: Elaboración Propia

3.1.3. Objetos de flujo

Es la primera categoría de los elementos gráficos de BPMN 2.0. Un diagrama de procesos de negocios debe contener tres elementos principales, los cuales se van modificando para poder cubrir un mayor campo de modelado. Estos elementos corresponden a las actividades, compuertas y eventos. Cada uno de estos elementos de flujo tiene una representación sencilla. Los primeros son representados por rectángulos, los segundos por rombos y los últimos por círculos, tal como se muestra a continuación.



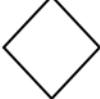
Elemento	Representación Grafica
Actividad	
Eventos	
Decisión/compuerta	

Figura 3.2: Representación gráfica de elementos en BPMN.
Fuente: Elaboración Propia

Actividades

Dentro de la categoría objetos de flujo se encuentran las actividades. Una actividad está representada por un rectángulo de esquina redondeada y describe el tipo de trabajo que será realizado. Son las actividades las que transforman el estado de un objeto de negocio para que el proceso pueda llegar a crear valor para los clientes. Se denominan siempre con un verbo, es decir una acción, seguido de un sustantivo (Ejemplo: comprar alimentos). Existen diferentes tipos de actividades los cuales son: tareas, transacción, subprocesso de evento, actividades de llamada.

1. **Tarea:** es una unidad de trabajo, una actividad que puede ser refinada. Existen varios tipos de tarea, los cuales especifican la naturaleza de la tarea que se desea llevar a cabo.



Figura 3.3: Tarea en BPMN.
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

2. **Transacción:** es una unidad lógica de trabajo que permite agrupar un conjunto de actividades individuales, de manera que tienen éxito o fracaso colectivamente.



Figura 3.4: Transacción en BPMN.

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

3. **Subproceso de Eventos:** se sitúa en el interior de otro subproceso. Este se activa en la ocurrencia del evento de inicio especificado y mientras el proceso que lo contiene permanezca también activo. El subproceso de evento puede interrumpir o no al proceso que lo contiene.



Figura 3.5: Subproceso de eventos en BPMN.

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

4. **Actividad de llamada:** es una referencia a un subproceso definido de forma global que se reutiliza en el proceso actual.



Figura 3.6: Actividad de llamada en BPMN.

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

BPMN nos brinda la posibilidad de clasificar las tareas en diferentes tipos, con una marca en su esquina superior izquierda. Esta marca tiene distintos símbolos que se diferencian dependiendo de la naturaleza de quién lleve a cabo el *job*. Los tipos de tarea son:

- **Indefinida:** no presenta ninguna marca en particular, lo que significa que no está definido quien la ejecuta. Ejemplo: obtener un seguro de auto.



Figura 3.7: Tarea indefinida en BPMN.

Fuente: Elaborado por el autor en base a póster de BPMN 2.0

- **Envío de mensaje:** toda clase de envío se modela con un ícono ennegrecido. Al ser este el de un envío de mensaje, se utiliza una carta. De igual forma esto puede modelarse con un evento. Ejemplo: envío de un correo electrónico por no pago.



Figura 3.8: Envío de mensaje en BPMN.

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Recepción de mensaje:** : la recepción de un mensaje puede modelarse con una tarea con la marca de un mensaje blanco. De manera análoga al caso anterior, un ícono en blanco indica recepción. Esta tarea también puede ser modelada como un evento, dependiendo solo de la manera de modelar el proceso. Ejemplo: recepción de confirmación de pago.



Figura 3.9: Recepción de mensaje en BPMN.

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Tarea de Usuario:** ejecutada por una persona, pero el control de esta tarea es llevada por sistema. Ejemplo: revisión de una factura.



Figura 3.10: Tarea de usuario en BPMN.

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Tarea Manual:** ejecutada por una persona, cuyo control no son llevados por un sistema. Ejemplo: conversación de un ejecutivo con un cliente.

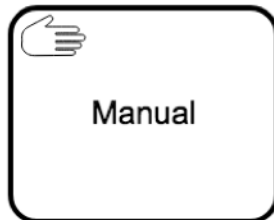


Figura 3.11: Tarea manual en BPMN.

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Regla de Negocio:** las reglas de negocios son tipos específicos de servicios mantenidos por un grupo de trabajo de negocios, en lugar de un grupo de informática. La figura de regla se usa para representar la implementación de una regla de negocios. Ejemplo: Calcular el descuento en un proceso de creación de boleta.



Figura 3.12: Regla de negocio en BPMN.

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Tarea de servicio:** es una actividad automática que es ejecutada por algún software. Ejemplo: Revisar disponibilidad de asiento al comprar pasajes.



Figura 3.13: Tarea de servicio en BPMN.

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Ejecución de Script:** las tareas de la secuencia de comandos son ejecutadas por un motor de procesos de negocios. La secuencia de comandos está escrita en un lenguaje que el motor puede analizar y que en muchos casos es JavaScript.

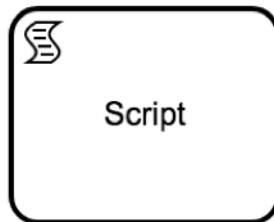


Figura 3.14: Ejecución de script en BPMN.

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

Además de la clasificación de las actividades (tarea, transacción, subprocesso de evento y actividad de llamada) podemos marcarlas con ciertas propiedades (marcas), existiendo los marcadores de actividades, los cuales especifican el comportamiento particular de las actividades durante su ejecución.

- **Subproceso:** esta marca representa un proceso más pequeño que va dentro de un gran proceso.

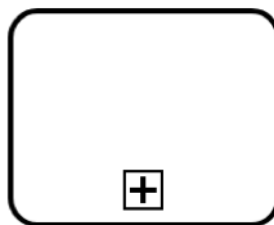


Figura 3.15: Subproceso en BPMN.

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Ciclo:** esta marca también es conocida como loop. Una actividad con esta propiedad se va a repetir cuantas veces sea necesario para que se cumpla la actividad. Es decir, repite la ejecución en forma secuencial.

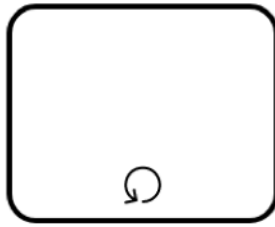


Figura 3.16: Ciclo en BPMN.
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Instancias múltiples en paralelo:** permite la creación de un número deseado de instancias de actividad idénticas que son ejecutadas en paralelo.

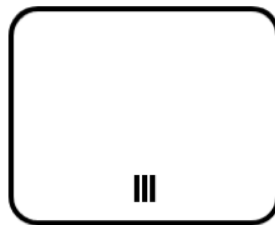


Figura 3.17: Instancias múltiples en paralelo en BPMN.
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Instancias múltiples en secuencia:** permite la creación de un número deseado de instancias de actividad idénticas que se ejecutan en serie.

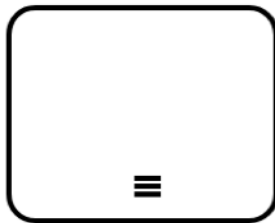


Figura 3.18: Instancias múltiples en secuencia en BPMN.
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Subproceso Ad Hoc:** es un grupo de actividades que no requieren relaciones de secuencia. Se puede definir un conjunto de actividades, pero su secuencia y número de ejecuciones es determinada por sus ejecutantes.

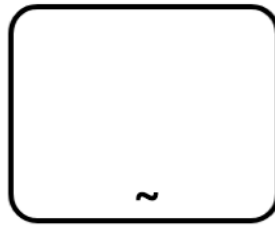


Figura 3.19: Subproceso Ad Hoc en BPMN.

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Compensación:** esta marca se utiliza de forma especializada, que solo sucede cuando otra tarea específica ocurre previamente. Existe un evento que hace exactamente lo mismo.

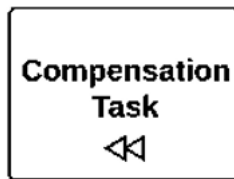


Figura 3.20: Tarea de Compensación en BPMN.

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

Compuertas

Dentro de la categoría objetos de flujo, también encontramos las compuertas. Las compuertas son representadas gráficamente por un rombo. Éstas determinan las decisiones como la bifurcación o fusión de los caminos, dependiendo de las condiciones expresadas anteriormente en el modelo. Existen 5 diferentes tipos de compuertas, los cuales se detallan a continuación:

- **Exclusiva (xor):** selecciona exactamente un flujo de secuencia entre las alternativas existentes, es decir, permite recorrer solo una alternativa. Existen dos símbolos para representar esta compuerta, uno es un rombo con una x en su interior y el segundo es un rombo con su interior en blanco. La convergencia (fusión) se activa cuando una de las alternativas se cumple.

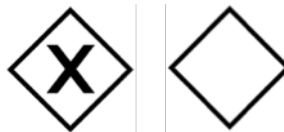


Figura 3.21: Compuerta exclusiva en BPMN.

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Basada en Eventos:** siempre será seguida por eventos o tareas de recepción, y sólo activará un flujo saliente dependiendo del evento que ocurra en primer lugar.



Figura 3.22: Compuerta basada en eventos en BPMN.
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Paralela (and):** en un punto de bifurcación, todos los caminos salientes serán activados simultáneamente, mientras que en un punto de convergencia, la compuerta espera a que todos los flujos incidentes se completen antes de activar el flujo saliente. Sirve para paralelizar lo máximo posible, realizando tareas distintas en forma paralela, disminuyendo el tiempo total del proceso.



Figura 3.23: Compuerta paralela en BPMN.
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Inclusiva (or):** en un punto de bifurcación, al menos un flujo es activado. En un punto de convergencia, la compuerta espera a todos los flujos que fueron activados para activar el saliente.



Figura 3.24: Compuerta inclusiva en BPMN.
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Compleja:** esta compuerta se utiliza cuando un caso de negocio no se puede representar con ninguna otra compuerta, es decir, para modelar el comportamiento de sincronización complejo. No se utiliza mucho esta compuerta.



Figura 3.25: Compuerta compleja en BPMN.
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

Eventos

Finalmente, el último de los elementos dentro de la categoría objetos de flujo, son los eventos. Éstos son representados gráficamente por un círculo y se utilizan para representar alguna acción que ocurre

durante el transcurso de un proceso de negocio. En comparación con una actividad que se realiza (tiempo presente), un evento es algo que ya ocurrió (tiempo pasado). Existen dos clases de eventos: de captura o de disparo. El primero se utiliza cuando un estado, dependiente o no del flujo de negocio, es necesario para llevar a cabo una serie de acciones (actividades, compuertas, etcétera). El segundo se utiliza para “lanzar” un aviso tanto dentro como afuera del flujo de negocio, sobre algún estado en particular.

Dentro de los eventos encontramos eventos de inicio, intermedios y de finalización. Los **eventos de inicio** nos señalan qué tipo de acontecimientos ocurren para que un proceso de negocio comience. Además, son eventos de captura, es decir, algo independiente del proceso ocurrió para que se dé inicio al flujo del proceso de negocio. Los **eventos intermedios** no se utilizan con frecuencia, pero son apropiados si un estado en particular representa un hito, mostrando el estado que el proceso ha logrado mediante un evento intermedio de disparo y que en el algún momento servirá para que el flujo de negocio tome uno u otro camino, dependiendo de dicho estado con un evento intermedio de captura. Finalmente se encuentran los **eventos de finalización (o fin)**, los cuales muestran que se logró al acabar con la trayectoria de un proceso. Ocurren de tal forma que el proceso ya no puede reaccionar a ellos. Los eventos de fin son eventos de disparo y pueden o no señalar un estado hacia afuera del proceso de negocio.



Figura 3.26: Eventos de inicio, intermedio y finalización en BPMN.
Fuente: Elaboración Propia

Los eventos anteriores pueden tener distintos tipos, entre los que están los siguientes:

Eventos

	Inicio			Intermedios				Fin
	Alto Nivel	Evento Interruptor de Subproceso	Evento No Interruptor de Subproceso	Captura	Adjunto Interruptor	Adjunto No Interruptor	Lanzamiento	Alto Nivel
Simple: Eventos sin especificar. Indican puntos de inicio, de fin y situaciones intermedias.								
Mensaje: Recepción y envío de mensajes.								
Temporal: Puntos en el tiempo, lapsos, límites (timeouts). Pueden ser eventos únicos o cíclicos.								
Escalable: Cambio a un nivel más alto de responsabilidad.								
Condicional: Reacción a cambios en las condiciones de negocios o integración de reglas de negocio.								
Enlace: Conectores fuera de página. Dos conectores de enlace equivalen a un flujo de secuencia.								
Error: Captura y lanzamiento de errores conocidos con nombre.								
Cancelación: Reacción a la cancelación de una transacción/Solicitud de cancelación.								
Compensación: Manejo/Solicitud de compensación.								
Señal: Intercambio de señales entre procesos. Una señal puede ser capturada varias veces.								
Multiple: Captura uno de un conjunto de eventos. Lanza todos los eventos definidos.								
Paralela Multiple: Captura todos los eventos de un conjunto de eventos en paralelo.								
Terminación: Terminación inmediata del proceso.								

Figura 3.27: Tipos de evento en BPMN.
Fuente: BPM Offensive Berlin, Tabla de Eventos (2005)

- **Simple:** son eventos no especificados, también llamados eventos “en blanco”. Por ejemplo, significa técnicamente que el desencadenante para iniciar la instancia del proceso no está especificado. Esto significa que el motor no puede anticipar cuándo se debe iniciar la instancia del proceso. Indican puntos de inicio, de fin y situaciones intermedias.

Nota: un subproceso siempre debe tener un evento de inicio simple.

- **Evento de Inicio Simple:** indica el punto donde comienza el proceso además un subproceso siempre debe tener un evento de inicio simple.



Figura 3.28: Evento de Inicio Simple

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Evento Intermedio de Lanzamiento Simple:** a menudo se usa para indicar algún estado logrado en el proceso.



Figura 3.29: Evento Intermedio de Lanzamiento Simple

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Evento Final Simple:** significa que el resultado arrojado cuando se alcanza el evento no está especificado. Como tal, el motor no hará nada además de finalizar la ruta de ejecución actual.



Figura 3.30: Evento Final Simple

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Mensaje:** son eventos para la recepción y envío de mensajes, hacen referencia a un mensaje con nombre. Un mensaje tiene un nombre y una carga útil. A diferencia de una señal, un evento de mensaje siempre se dirige a un único destinatario. Por ejemplo, se refiere a cualquier objeto que transporte información: llamadas, correos, cartas, órdenes de compra, ordenes de despacho, boletas, facturas, etc.
- **Evento de Inicio de Mensaje:** un proceso activo envía un mensaje a un proceso específico para iniciarlo. El Mensaje de inicio siempre recibe mensajes, nunca los envía.



Figura 3.31: Evento de Inicio de Mensaje

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Evento Intermedio de Captura de Mensaje:** cuando un token llega hasta este evento, esperará allí hasta que llegue un mensaje con el nombre propio. Es decir, tiene la opción de “atrapar” un mensaje. Su figura solo se muestra el contorno del sobre.



Figura 3.32: Evento Intermedio de Captura de Mensaje
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Evento Intermedio Adjunto de Mensaje** : los eventos adjuntos son eventos de captura que están adjuntos a una actividad. Esto significa que mientras se está ejecutando la actividad, el evento de límite de mensaje está escuchando el mensaje con nombre. Cuando se detecta esto, pueden ocurrir dos cosas, dependiendo de la configuración del evento de límite:

-Adjunto Interruptor: la actividad se interrumpe y se sigue el flujo de secuencia que sale del evento.



Figura 3.33: Evento Intermedio Adjunto Interruptor de Mensaje
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

-Adjunto No interruptor: un token permanece en la actividad y se crea un token adicional que sigue el flujo de secuencia que sale del evento.



Figura 3.34: Evento Intermedio Adjunto No Interruptor de Mensaje
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Evento Intermedio de Lanzamiento de Mensaje**: este evento envía un mensaje a un servicio externo, es decir tiene la opción de “lanzar un mensaje”, su figura tiene un sobre con relleno sólido.



Figura 3.35: Evento intermedio de Lanzamiento de Mensaje
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Evento de Fin de Mensaje**: cuando la ejecución del proceso llega a un evento de fin de mensaje, la ruta de ejecución actual finaliza y se envía un mensaje a un proceso o caso específico. Este evento siempre envía mensajes, nunca los recibe.



Figura 3.36: Evento Fin de Mensaje

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Temporal:** también llamados eventos temporizador, debido a que se utilizan cuando una condición de tiempo ocurre, son eventos que son activados por un temporizador definido. Este tipo de eventos se representan gráficamente como un reloj dentro del símbolo de un evento (círculo). Se pueden usar como evento de inicio, evento intermedio o evento de límite. Los eventos de límite pueden estar interrumpiendo o no. Algunos ejemplos de eventos temporales pueden ser: puntos en el tiempo, lapsos, límites, pueden ser eventos únicos o cíclicos.
 - **Evento de Inicio Temporal:** se usa para crear una instancia de proceso en un momento dado. Se puede usar tanto para procesos que deben comenzar solo una vez, como para procesos que deben comenzar en intervalos de tiempo específicos.



Figura 3.37: Evento de Inicio Temporal

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Evento Intermedio de Captura de Temporal:** este tipo de eventos actúan como un cronómetro. Cuando llega una ejecución en la actividad que contiene este evento, se inicia un temporizador. Cuando el temporizador se dispara (por ejemplo, después de un intervalo especificado), se sigue el flujo de secuencia que sale del evento intermedio del temporizador.



Figura 3.38: Evento Intermedio de Captura Temporal

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Evento Intermedio Adjunto Temporal :** Un evento de límite de temporizador actúa como un cronómetro y como un reloj despertador. Cuando llega una ejecución a la actividad a la que está conectado el evento de límite, se inicia un temporizador. Cuando el temporizador se dispara (por ejemplo, después de un intervalo especificado), se interrumpe la actividad y se sigue el flujo de la secuencia que sale del evento del límite del temporizador. Existe la diferencia entre un evento de temporizador de interrupción y uno que no interrumpe.
 - Adjunto Interruptor: es el predeterminado mencionado anteriormente.

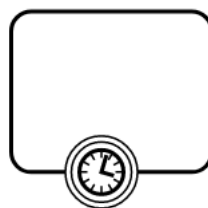


Figura 3.39: Evento Intermedio Adjunto Interruptor Temporal
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

-Adjunto No interruptor: conduce a que la actividad original no se interrumpa, la actividad permanece allí. En cambio, se crea una ejecución adicional y se envía a través de la transición saliente del evento.



Figura 3.40: Evento Intermedio Adjunto No Interruptor Temporal
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Escalamiento:** este evento solo se usa en un subproceso de evento. Son eventos que hacen referencia a una escalada con nombre. En su mayoría se utilizan para comunicarse desde un subproceso a un proceso superior y mejorar la comunicación existente entre subprocesos superiores e inferiores.
 - **Evento de Inicio Escalable** este tipo de evento sólo se puede usar para activar un subproceso de evento; no se puede usar para iniciar una instancia de proceso.



Figura 3.41: Evento de Inicio Escalable
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Evento Intermedio Adjunto Escalable:** un evento de escalamiento intermedio de captura en el límite de una actividad captura escalamientos que se lanzan dentro del alcance de la actividad en la que está definido.

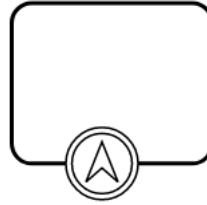


Figura 3.42: Evento Intermedio Adjunto Interruptor Escalable
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.



Figura 3.43: Evento Intermedio Adjunto No Interruptor Escalable
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Evento Intermedio de Lanzamiento Escalable** : cuando la ejecución del proceso llega a este tipo de evento, se lanza un escalado con nombre. Esta escalada se puede capturar mediante un evento de límite de escalamiento o un subprocesso de evento con un evento de inicio de escalada que tiene el mismo o ningún código de escalamiento.



Figura 3.44: Evento Intermedio de Lanzamiento Escalable
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Evento de Fin Escalable**: cuando la ejecución del proceso llega a este tipo de evento, la ruta de ejecución actual finaliza y se lanza una escalada con nombre. Tiene el mismo comportamiento que un evento de lanzamiento intermedio escalable.



Figura 3.45: Evento de Fin Escalable
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Condicional**: define un evento que se desencadena si una condición determinada se evalúa como verdadera. Se puede usar como evento de inicio de un subprocesso de evento, como evento interme-

dio y evento de límite. El evento de inicio y el límite pueden ser interruptores y no interrumpir. Un proceso puede iniciar o continuar bajo ciertas condiciones, una condición puede ser cualquier cosa como una reacción o cambios en las condiciones de negocios o integración de reglas de negocio, sin olvidar que este evento puede ser utilizado si la condición ocurre independiente del proceso.

- **Evento de Inicio Condicional:** este evento sólo se puede usar en combinación con un sub-proceso de evento. De forma similar a los eventos de frontera condicionales, los eventos de inicio condicionales pueden ser interruptores y no interrumpir.



Figura 3.46: Evento de Inicio Condicional

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Evento Intermedio de Captura de Condicional:** este evento es como esperar hasta que la condición sea verdadera. Cuando la ejecución llega a la actividad del evento de captura, la condición se evalúa por primera vez. Si se cumple la condición, el proceso de ejecución continúa a la siguiente actividad. Si la condición no se cumple, la ejecución permanece en esta actividad hasta que se cumpla la condición.



Figura 3.47: Evento Intermedio de Captura Condicional

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Evento Intermedio Adjunto Condicional :** un evento de frontera condicional actúa como un observador que se activa si se cumple una condición específica. Hay una diferencia entre un evento condicional de interrupción y uno sin interrupción.

-Adjunto Interruptor: es el explicado anteriormente.

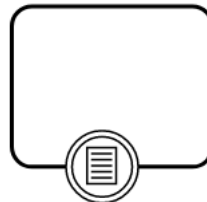


Figura 3.48: Evento Intermedio Adjunto Interruptor Condicional

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

-Adjunto No Interruptor: el evento sin interrupción provoca que la actividad original no se interrumpa, la instancia permanece activa. En su lugar, se crea una ruta de ejecución adicional, tomando la transición saliente del evento. Un evento condicional que no interrumpa puede

activarse más de una vez, siempre que la actividad a la que esté conectada esté activa.



Figura 3.49: Evento Intermedio Adjunto No Interruptor Condicional

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Enlace:** los eventos de enlace son un caso especial: no tiene una semántica de ejecución especial, pero sirve como un “ir” a otro punto en el mismo modelo de proceso (para ser precisos: en el mismo subproceso). Este evento no tiene un significado para el proceso de negocio, tiene como objetivo dividir diagramas muy grandes sin perder la conexión del flujo de secuencia que sigue el proceso, en otras palabras, no perder el hilo. Son usados como conectores cuando un diagrama queda en varias páginas, dos conectores de enlace equivalen a un flujo de secuencia. Este evento solo puede ser usado como evento intermedio.

- **Evento Intermedio de Captura de Enlace:** Se utiliza para representar gráficamente una continuidad de un flujo de secuencia y es el evento link que recibe la redirección.



Figura 3.50: Evento Intermedio de Captura de Enlace

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Evento Intermedio de Lanzamiento de Enlace :** se utiliza para representar gráficamente una continuidad de un flujo de secuencia. El evento que inicia el “go to” debe ser del tipo “lanzamiento”.



Figura 3.51: Evento Intermedio de Lanzamiento de Enlace

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Error:** los eventos de error son eventos que se desencadenan por un error definido. Gráficamente se identifican con el signo de rayo dentro del círculo del evento captura y lanzamiento de errores conocidos con nombre.

- **Evento de Inicio de Error:** solo se puede usar este evento para desencadenar un subproceso de evento; no se puede usar para iniciar una instancia de proceso. El evento de inicio de error siempre está interrumpiendo.



Figura 3.52: Evento de Inicio de Error

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Evento Intermedio Adjunto interruptor de error** : un evento intermedio de error de captura en el límite de una actividad, captura los errores que se lanzan dentro del alcance de la actividad en la que está definido. La definición de un evento de límite de error tiene más sentido en un subproceso incrustado, o una actividad de llamada, ya que un subproceso crea un alcance para todas las actividades dentro del subproceso. Los errores son arrojados por eventos finales de error. Tal error propagará sus ámbitos principales hacia arriba hasta que se encuentre un ámbito en el que se defina un evento límite de error que coincida con la definición del evento de error. Cuando se detecta un evento de error, se destruye la actividad en la que se define el evento de límite, destruyendo también todas las ejecuciones actuales en el mismo (por ejemplo, actividades concurrentes, subprocesos anidados, etc.). La ejecución del proceso continúa siguiendo el flujo de secuencia de salida del evento de límite.

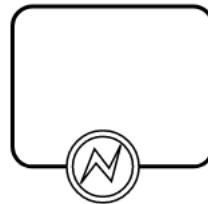


Figura 3.53: Evento Intermedio Adjunto Interruptor de Error

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Evento Final de Error** : cuando la ejecución del proceso llega a un evento final de error, la ruta de ejecución actual finaliza y se produce un error. Este error puede detectarse mediante un evento de límite de error intermedio coincidente. En caso de que no se encuentre un evento de límite de error coincidente, la semántica de ejecución se predetermina a la semántica de evento sin fin.



Figura 3.54: Evento Final de Error

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Cancelación**: este evento se utiliza como una reacción a la cancelación de una transacción/solicitud de cancelación.
 - **Evento Intermedio Adjunto Interruptor de Cancelación**: un evento de frontera de cancelación, se activa cuando se cancela una transacción. Cuando se desencadena el evento de límite de cancelación, primero interrumpe todas las ejecuciones activas en el ámbito actual. A

continuación, inicia la compensación de todos los eventos de límite de compensación activos en el alcance de la transacción. La compensación se realiza de forma síncrona, es decir, el evento límite espera antes de que se complete la compensación antes de abandonar la transacción. Cuando se completa la compensación, el subproceso de transacción se deja usando el (los) flujo (s) de secuencia que se está quedando sin el evento de límite de cancelación.

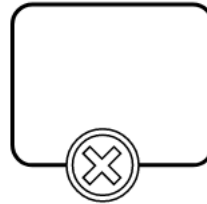


Figura 3.55: Evento Intermedio Adjunto Interruptor de Cancelación
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Evento Final de Cancelación:** este evento solo se puede usar en combinación con un subproceso de transacción. Cuando se llega al evento de cancelación de finalización, se produce un evento de cancelación que debe capturarse mediante un evento de límite de cancelación. El evento límite de cancelación luego cancela la transacción y activa la compensación.



Figura 3.56: Evento Final de Cancelación
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Compensación:** en BPMN compensar significa volver al estado inicial de una actividad. El evento de compensación permite representar de manera gráfica el reverso en forma comprimida, dando la opción a no tener que modelar todo el proceso de reversa, ahorrando modelamiento. Este evento de compensación asume todos los pasos necesarios para volver al estado inicial de un proceso de negocio. manejo/ solicitud de compensación.
 - **Evento Inicio Compensación:** un evento de inicio de compensación solo se puede usar para activar un subproceso de evento; no se puede usar para iniciar una instancia de proceso. Este tipo de subproceso de evento se denomina subproceso de evento de compensación. Este evento se activa cuando existe una compensación, que se inicia mediante un lanzamiento de compensación en el diagrama principal del proceso. La compensación sirve para que las acciones se deshagan.



Figura 3.57: Evento Inicio Compensación
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Evento Intermedio Adjunto de Compensación:** un evento de límite de compensación, se

puede utilizar para capturar el evento de compensación, cuando esto ocurre, una actividad de compensación será ejecutada. Este evento tiene una política de activación diferente a otros eventos de límite. Otros eventos de límite, como el evento de límite de señal, se activan cuando se inicia la actividad a la que están conectados. Cuando se deja la actividad, se desactivan y se cancela la suscripción al evento correspondiente. El evento de límite de compensación es diferente. El límite de compensación se activa cuando la actividad a la que se adjunta se completa con éxito.

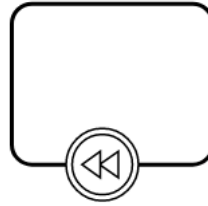


Figura 3.58: Evento Intermedio Adjunto de Compensación
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Evento Intermedio de Lanzamiento de Compensación:** este evento se utiliza activar la compensación.



Figura 3.59: Evento Intermedio de Lanzamiento de Compensación
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Evento Final de Compensación:** este evento finaliza el proceso o “token” e inicia el flujo de tratamiento de compensaciones, que se utilizan para deshacer las acciones previamente realizadas en el proceso. Tiene el mismo comportamiento y limitaciones que un evento de lanzamiento intermedio de compensación.



Figura 3.60: Evento Final de Compensación
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Señal:** son eventos que hacen referencia a una señal nombrada, gráficamente se identifican por un triángulo, estos eventos son similares a los eventos de mensaje. La diferencia radica en que los eventos de mensaje tienen un destinatario predeterminado y una señal es un mensaje con un destino sin definir. Como por ejemplo, avisos en el diario de esta forma se logra un intercambio de señales entre procesos y cualquier persona que percibe la señal puede reaccionar a esta. Una señal puede ser capturada varias veces.

- **Evento Inicio de Señal:** este tipo de eventos se puede usar para iniciar una instancia de

proceso usando una señal nombrada.



Figura 3.61: Evento Inicio de Señal

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Evento Intermedio de Captura de Señal:** cuando un token llega a este evento, esperará allí hasta que llegue una señal con el nombre propio.



Figura 3.62: Evento Intermedio de Captura de Señal

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Evento Intermedio Adjunto de Señal:** cuando llega una ejecución en la actividad a la que está conectado el evento de límite de señal, el evento de límite de señal capta señales con el nombre propio.

Nota: Al contrario de otros eventos, como el evento de límite de error, un evento de límite de señal no solo captura eventos de señal lanzados desde el ámbito al que está conectado. Un evento de señal tiene un alcance global, lo que significa que la señal puede emitirse desde cualquier lugar, incluso desde una instancia de proceso diferente.

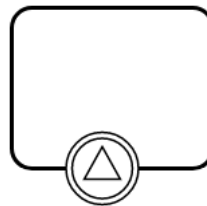


Figura 3.63: Evento Intermedio Adjunto Interruptor de Señal

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

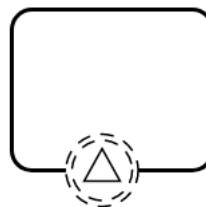


Figura 3.64: Evento Intermedio Adjunto No Interruptor de Señal

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Evento Intermedio de Lanzamiento de Señal:** este evento arroja un evento de señal para una señal definida. La señal se transmite a todos los controladores activos (es decir, todos los eventos de señal de captura).



Figura 3.65: Evento Intermedio de Lanzamiento de Señal
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Evento de Fin de Señal:** este evento, arroja un evento de señal para una señal definida y la ruta de ejecución actual finaliza. Tiene el mismo comportamiento que un evento de lanzamiento intermedio de señal.



Figura 3.66: Evento de Fin de Señal
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Múltiple:** con un símbolo se captura un conjunto de eventos, se puede utilizar de dos formas, una es siendo evento de captura y otra siendo evento disparador. El primero inicia el proceso con el hecho de ocurrir uno de los eventos posibles y el segundo es como un disparador múltiple lanzando todos los eventos contenidos.
 - **Evento de Inicio Múltiple:** representa un inicio de proceso basado en varias reglas posibles, siendo que sólo una es obligatoria para disparar el proceso.



Figura 3.67: Evento de Inicio Múltiple
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Evento Intermedio Múltiple:** este se dispara si sólo uno de los intermediarios contenidos se dispara.



Figura 3.68: Evento Intermedio Múltiple
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Evento de Fin Múltiple:** está compuesto por varios otros finalizadores, que se ejecutan todos al término del proceso



Figura 3.69: Evento de Fin Múltiple

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Paralela Múltiple:** este evento es un complemento del evento múltiple, captura todos los eventos de un conjunto de eventos en paralelo. Trabaja bajo la semántica AND. En otras palabras, tienen que haber ocurrido todos los eventos para que se pueda iniciar, continuar o interrumpir una actividad.
 - **Evento de Inicio Paralelo Múltiple:** representa un inicio de proceso basado en varias reglas posibles, siendo que sólo una es obligatoria para disparar el proceso.



Figura 3.70: Evento de Inicio Paralelo Múltiple

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Evento Intermedio Paralelo Múltiple:** igual a los eventos condicional y timer, pero con la posibilidad de contener varios intermediarios y todos se deben atender para que el paralelo se ejecute.



Figura 3.71: Evento Intermedio Paralelo Múltiple

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Terminación:** un evento de terminación finaliza el alcance completo donde se genera el evento y todos los ámbitos internos. Si lo usa en el nivel de instancia de proceso, todo el proceso finaliza. En un nivel de subprocesso, el alcance actual y todos los procesos internos finalizarán. Es decir, corresponde a la terminación inmediata del proceso, desactiva de inmediato todo el sistema que está antes de él.



Figura 3.72: Evento de Fin

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

3.1.4. Objetos de conexión

Los objetos de flujos (actividades, eventos y compuertas) se ensamblan entre sí a través los objetos de conexión, los cuales son de tres tipos: flujo de secuencia, flujo de mensaje y asociación.

- **Flujo de secuencia:** conecta los objetos de flujo, indicando en qué orden se deben realizar las actividades.



Figura 3.73: Flujo de secuencia en BPMN.

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Flujo de mensaje:** representa mensajes de un participante del proceso a otro.



Figura 3.74: Flujo de mensaje en BPMN.

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Asociación:** muestra relaciones entre artefactos y objetos de flujo.



Figura 3.75: Asociación en BPMN.

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

3.1.5. Carriles

Es la tercera categoría de los elementos gráficos de BPMN 2.0. Esta categoría organiza visualmente las actividades, separándolas para ilustrar quiénes son los responsables de ejecutar las actividades, es decir, las diferentes responsabilidades funcionales que existen en un proceso de negocio. Hay dos elementos en esta categoría: la piscina o contenedor y el carril o compartimento.

- **Piscina o contenedor (*pool*):** representada gráficamente por un rectángulo grande, es un contenedor de una serie de actividades representando a los participantes principales en un proceso, identificando a las distintas organizaciones involucradas. Esta piscina es la analogía a una piscina en la

realidad, pudiendo contener uno o más carriles. Representa los límites de un proceso de principio a fin.

- **Carril o compartimento (*lane*):** se representa gráficamente como una subdivisión dentro de una piscina, con un rectángulo que se extiende a la altura y al ancho de la piscina. Es una analogía a los carriles en las piscinas de natación, en el cual cada nadador tiene su propio carril, separado por las bandas de boyas flotantes. Contiene los objetos de flujo (actividades, eventos y compuertas), objetos de conexión y artefactos. Por lo general, se usan carriles en la práctica para: cargos de algún área (gerente, supervisar y obrero), roles (auditor y ejecutivo), roles generalizados (clientes, proveedores, fiscalizadores), departamentos de una organización (contabilidad, finanzas, ventas, recursos humanos, operaciones).

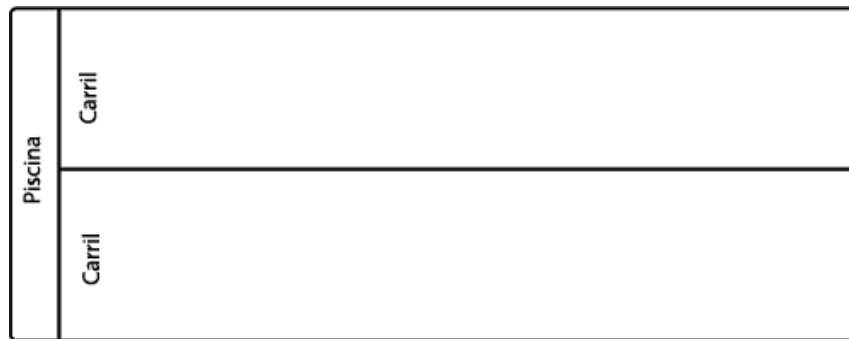


Figura 3.76: Piscina y carriles en BPMN.

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

3.1.6. Artefactos

Es la cuarta categoría de los elementos gráficos de BPMN 2.0. Estos elementos permiten a las personas que están desarrollando el modelo, llevar más información que les contribuye a agregar un contexto al diagrama, lo que se traduce en que el modelo sea lo más cercano a la realidad posible. Existen tres artefactos definidos, los cuales son: grupo, anotación y objetos de datos.

- **Grupo:** representado gráficamente por un rectángulo con esquinas redondeadas, construido con líneas segmentadas, se utiliza para agrupar diferentes actividades no afectando el flujo del diagrama general.



Figura 3.77: Grupos en BPMN.

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Anotación:** se representa por un paréntesis de apertura cuadrado, junto con una línea diagonal segmentada. Es un mecanismo para proporcionar información de texto adicional y de esta forma el

lector entenderá de mejor manera el modelo.

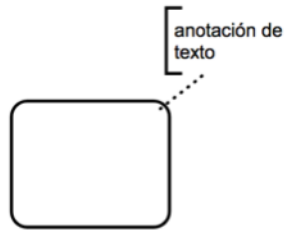


Figura 3.78: Anotación en BPMN.

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- **Objetos de datos:** proveen información sobre cómo documentos, datos y otros objetos son utilizados y actualizados durante el proceso. Existen varios tipos de datos los cuales se detallan a continuación:

- Objeto de dato: representa información que fluye a través del proceso como, por ejemplo, documentos, correos electrónicos o cartas.



Figura 3.79: Objeto de dato en BPMN.

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- Recopilación de datos: representa una colección de información, por ejemplo, una lista de artículos.



Figura 3.80: Recopilación de datos en BPMN.

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- Almacén de datos: es un lugar donde el proceso puede leer o escribir datos, por ejemplo, una base de datos. Esta información persiste más allá del alcance del proceso.



Figura 3.81: Almacén de datos en BPMN.
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- Dato de Entrada: es una entrada externa a todo el proceso y se utiliza en una actividad.

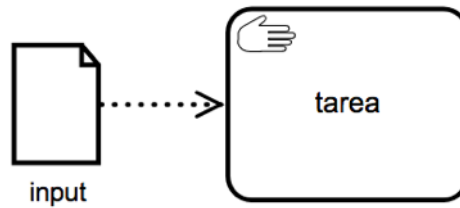


Figura 3.82: Dato de entrada en BPMN.
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

- Dato de Salida: es una variable disponible como resultado del proceso.

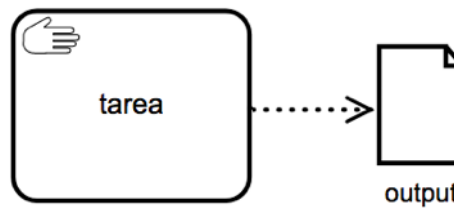


Figura 3.83: Dato de salida en BPMN.
Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

3.1.7. Ejemplo BPMN 2.0

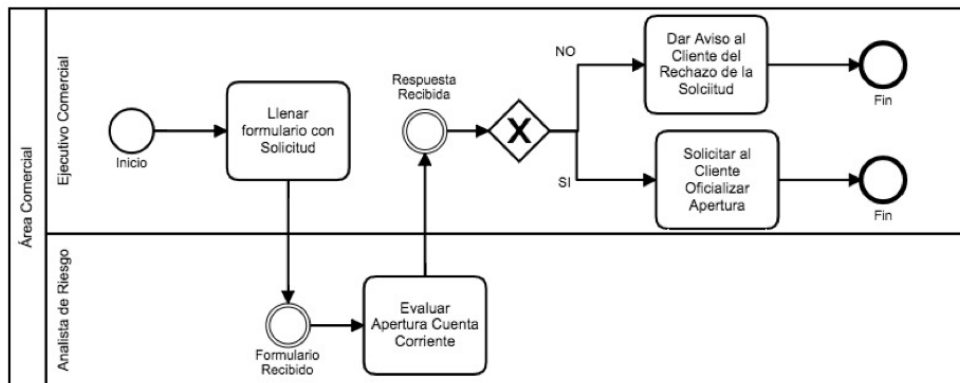


Figura 3.84: Ejemplo de proceso en BPMN.

Fuente: Elaborado por el autor en base a póster de BPMN 2.0.

En la figura anterior, se puede observar un diagrama BPMN, que corresponde al proceso de apertura de una cuenta corriente. Este proceso está a cargo del área comercial de un banco, por lo que la piscina representa a dicha área. El proceso comienza cuando el ejecutivo comercial (carril), el cual llena el formulario de solicitud de cuenta corriente con los datos del cliente (tarea). Luego, se lo envía al analista de riesgo (carril), el cual recibe este formulario (evento) y evalúa la factibilidad de abrir la cuenta corriente para este cliente (tarea), enviando posteriormente su respuesta al ejecutivo comercial. Éste recibe dicha respuesta, que tiene dos posibles resultados: le abren o no la cuenta corriente al cliente, dando lugar a una compuerta exclusiva. Es decir, solo una de las respuestas pasará, nunca ambas. En el caso que la respuesta sea SÍ, el ejecutivo comercial debe solicitarle al cliente oficializar la apertura (tarea), y en el caso que la respuesta sea NO, se debe dar aviso al cliente del rechazo de la solicitud (tarea), luego de que ocurra cualquiera de las dos alternativas se finaliza el proceso con un evento de término (círculo con borde grueso).

3.1.8. Categorías de procesos

Desde la publicación de la versión 2.0 de BPMN, el único diagrama BPD (solo existía éste en las versiones anteriores), se dividió en dos diagramas, dando origen a los de colaboración y orquestación. Además, se agregó el diagrama de coreografía. Estas categorías distintas de procesos que da soporte BPMN 2.0 son detallados a continuación:

Orquestación

El concepto de orquestación se utiliza en BPMN como una analogía a una orquesta de música, en la cual el director guía y controla a cada uno de los músicos. En BPMN, el contenedor o piscina dirige y controla el flujo de actividades del proceso.

Un proceso de orquestación describe como una única entidad de negocio procede a realizar las cosas representando una mirada propia de la organización. Considerando dicha mirada que representan estos

procesos, implican una punto de vista único de coordinación.

Los modelos de orquestación están siempre contenidos en un contenedor o piscina. Esto indica que consisten de elementos de procesos que coexisten dentro de un contexto bien definido. Un diagrama BPMN puede tener más de una orquestación, pero siempre cada proceso de orquestación se encuentra dentro de su respectivo contenedor o piscina. Este punto (que se encuentre siempre dentro de su contenedor) es muy importante, debido a que ayuda a comprender la diferencia entre un proceso orquestación y uno de coreografía, que es el que se describirá en el punto siguiente.

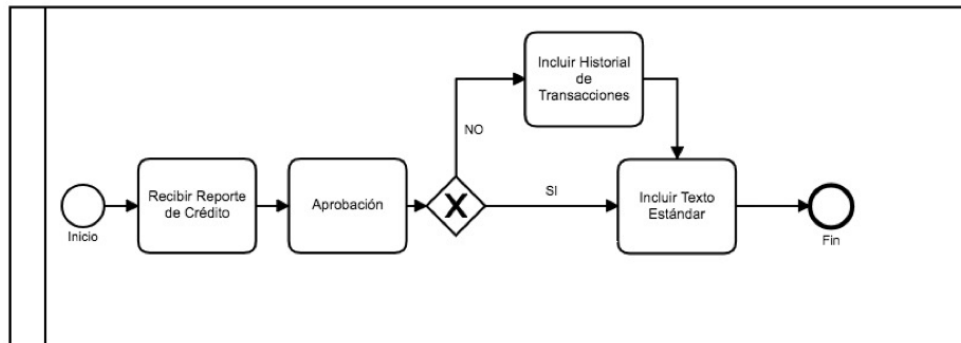


Figura 3.85: Ejemplo de diagrama de orquestación en BPMN.

Fuente: Stephen White and Derek Miers, Guía de Referencia y Modelado BPMN (2009)

Coreografía

El término coreografía se utiliza en BPMN de la misma forma como se hace en la definición de un ballet: es una definición del comportamiento esperado (una clase de contrato procedimientos o protocolo) entre los participantes que interactúan. Una coreografía muestra el conjunto ordenado (protocolo) de interacciones entre los participantes, un proceso de coreografía detalla las interacciones (en BPMN las interacciones con la comunicación, mediante la cual se intercambian mensajes entre dos participantes) entre dos o más participantes. Estos participantes pueden corresponder a roles de negocio generales (por ejemplo, un despachador) o una entidad específica de negocio (por ejemplo, FedEx como empresa de transporte).

Un modelo de coreografía BPMN comparte muchas de las características de un modelo de orquestación, en cuanto a que tienen un diagrama de flujo. Incluye tanto caminos alternativos y paralelos, como subprocesos. De esta manera, los objetos de flujo (actividades, eventos y compuertas) de los modelos de orquestación, también aplican a los modelos de coreografía. Para evitar confusiones, se mencionarán algunas de sus diferencias:

- Un proceso de orquestación está siempre dentro de un contenedor o piscina, dentro de un contexto claramente definido.
- Un proceso de coreografía no existe dentro de un contexto bien definido. No hay mecanismo central que guíe o mantenga trazo de una coreografía. Por lo cual, no hay datos compartidos disponibles para todos los elementos de la coreografía.

- La ubicación de una coreografía dentro de un diagrama BPMN es entre los contenedores, mientras que un proceso de orquestación está dentro de un contenedor.

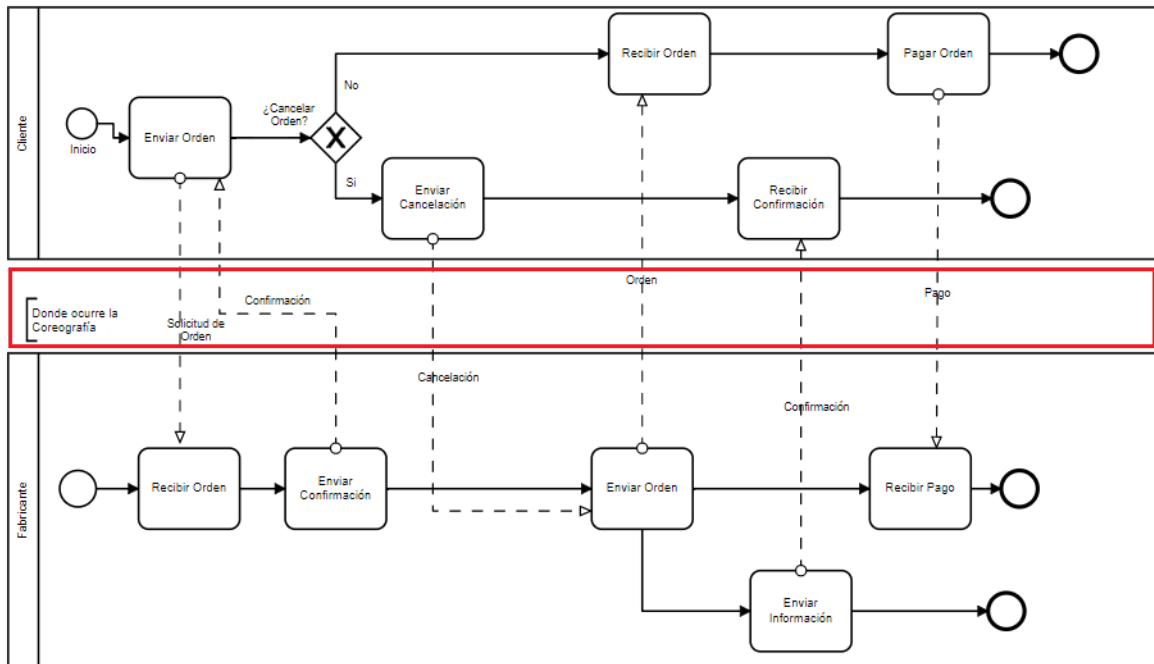


Figura 3.86: Ejemplo de coreografía en BPMN.

Fuente: Stephen White and Derek Miers, Guía de Referencia y Modelado BPMN (2009)

Colaboración

La colaboración tiene un significado específico en BPMN, una colaboración es cualquier diagrama BPMN que contenga dos o más participantes como se muestra en la figura. Los contenedores tienen flujo de mensajes entre ellos. Cualquiera de los contenedores puede llegar a contener un proceso, es decir, una colaboración puede contener también una coreografía y una o más orquestaciones, pero no está requerido.

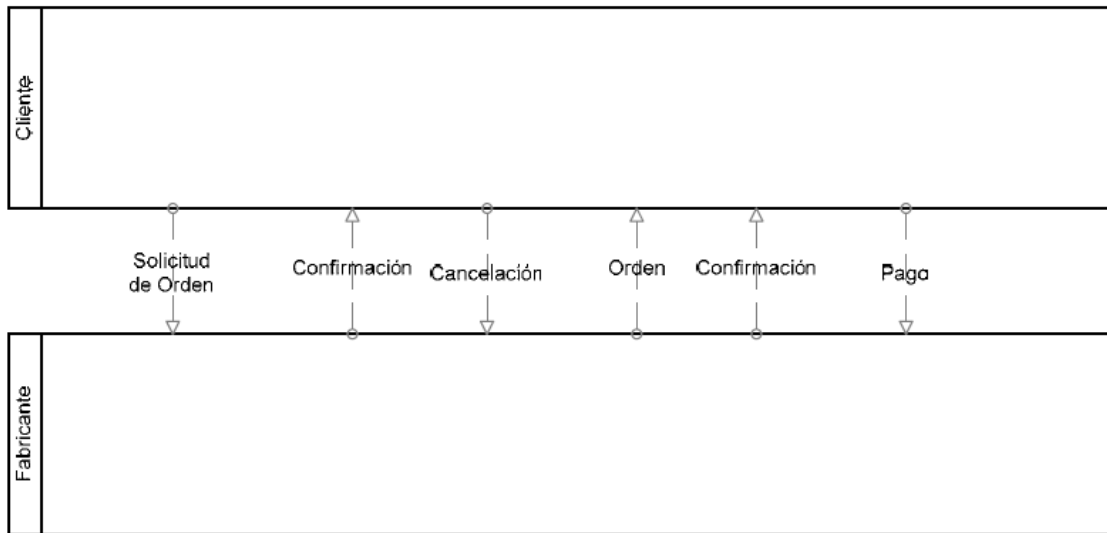


Figura 3.87: Ejemplo de diagrama de colaboración en BPMN.
Fuente: Stephen White, Guía de Referencia y Modelado BPMN (2009)

3.2. Trabajos Anteriores que Pretenden Mejorar BPMN

Existe un gran cantidad de trabajos que proponen extender y mejorar las características actuales de BPMN 2.0. En (Braun y Esswein, 2014) se realiza una revisión y caracterización de ellos. Para el trabajo presentado en esta tesis son relevantes los que tienen relación con el tópico “Medición de rendimiento” (*Performance measurement*).

Uno de los trabajos ahí presentados es el de Bocciarelli y D’Ambrogio (Bocciarelli y D’Ambrogio, 2011). Se trata de una extensión de las características de BPMN, incorporando elementos para medir rendimiento y confiabilidad. Para ello, introduce una notación que puede ser utilizada para especificar propiedades funcionales y no funcionales de un proceso de negocio, llamada PyBPMN. Se basa en un enfoque que explota los principios y estándares introducidos por la arquitectura impulsada por modelos (MDA por sus siglas en inglés), obteniendo así ventajas significativas en términos de fácil personalización y automatización mejorada. El objetivo final es que con las características añadidas, se pase de un enfoque descriptivo a uno productivo, en el cual la descripción conduzca el análisis automatizado y la implementación del proceso de negocio.

En términos de objetivos, es similar al análisis propuesto en el trabajo de esta tesis, no se realiza una descripción detallada de las características agregadas, dejándolas abiertas a la necesidad del proceso. En cambio, esta tesis propone que exista una manera estandarizada de representar los procesos y sus características, añadiendo elementos específicos de *Scheduling* que mejorarán la capacidad descriptiva de BPMN y permitirán tener un análisis más directo en términos operativos.

El segundo trabajo que se revisó fue el presentado por (Friedenstab, Janiesch, Matznerand, y Müller, 2012). Este trabajo utiliza el concepto de Monitoreo de Actividad del Negocio (*Business Activity Monitoring*), que se refiere a “la observación, el análisis y la presentación de información en tiempo real sobre

las actividades de negocio a través de las fronteras de los sistemas y las empresas”. Para ello, introduce algunos elementos de temporalidad como la duración y la frecuencia, así como también la definición de un objetivo. Lo que persigue, finalmente, es poder obtener indicadores de rendimiento claves (*KPI's*) en tiempo real.

Este es uno de los pocos trabajos revisados que utiliza el concepto de objetivo dentro de su propuesta y es por ello que se agregó a esta sección. La diferencia es el enfoque, ya que es utilizado como un rango a cumplir, mientras que el utilizar el concepto de función objetivo u objetivo desde *Scheduling* implica la búsqueda de un óptimo, es decir, del mejor valor posible.

Otros trabajos que se revisaron corresponden a los de (Cheikhrouhou, Kallel, Guermouche, y Jmaiel, 2013) y (Gagne y Trudel, 2009). Ambos proponen una extensión que agrega la dimensión de tiempo en los procesos de BPMN, con el objetivo de poder responder a algunas restricciones propias de procesos. Estas restricciones tienen que ver con condiciones propias de algunos procesos: como por ejemplo es una pizzería que promete llegar en menos de 30 minutos, o de lo contrario las pizzas entregadas son gratis. Para diagramar esto se tendrían que utilizar eventos temporales, de señales, etcétera, por lo que al incorporar la dimensión temporal, el diagrama quedaría más limpio, sobre todo en el caso de que exista más de una restricción temporal. Dentro de los elementos que este trabajo incorpora para proveer la dimensión temporal está la duración mínima y máxima de una actividad, tiempo entre actividades, entre otros.

Para poder incorporar elementos de *Scheduling* a BPMN, es requisito mínimo incorporar la dimensión temporal. Sin embargo, dicha implementación sería más simple a lo propuesto en el trabajo descrito anteriormente, porque el objetivo es muy distinto. El problema de *Scheduling* resuelve la asignación óptima de recursos a las ejecuciones de *jobs*, buscando lograr un objetivo asociado a la temporalidad, sin que existan restricciones asociadas a los plazos. Por tanto, al agregar elementos de tiempo de *Scheduling* a BPMN, se busca que el diagrama provea una descripción operativa del proceso, más que agregar restricciones.

Los trabajos revisados, si bien, incorporan elementos operativos a BPMN, van en una línea distinta al trabajo presentado en esta tesis. El primer trabajo señalado no entrega una directriz clara de lo que se puede considerar como rendimiento o confiabilidad, mientras que el segundo tiene un foco distinto respecto a la definición de objetivos y horizontes temporales al presentado en este trabajo. Los últimos dos están enfocados en poder representar las restricciones de temporalidad que existen en algunos procesos de negocio. Por otra parte, con el trabajo de esta tesis se busca mejorar el estándar actual de BPMN, agregándole características de *Scheduling* que sean comunes a distintos tipos de procesos y que permitan aumentar la capacidad descriptiva, sin dejar de lado la propiedad de estándar que tiene BPMN.

3.3. Scheduling

No existe una única definición de lo que es *scheduling*. Sin embargo en lo que las definiciones existentes convergen (T'kindt y Billaut, 2006), es que se trata de un proceso de toma de decisión en el que se busca asignar, de manera óptima considerando uno o más objetivos, recursos limitados para la realización de tareas (*jobs*) predeterminadas.

Para poder introducir las distintas formulaciones que existen en *scheduling*, se hace necesario hablar primero de la notación que se utilizará. Los conceptos base para esta notación son las operaciones y las máquinas.

Una operación es toda actividad, dentro de un negocio, que requiere de acciones manuales (“máquinas” humanas) o automáticas (máquinas, computadores, servidores, etcétera) para ser completada. Las máquinas corresponden a los recursos con los que se llevan a cabo las operaciones. Adicionalmente, existe el concepto *job*, que consiste en una serie de operaciones.

Un ejemplo sencillo es un pedido en una pizzería: un cliente realiza un pedido (*job*) a un garzón (máquina 1), quien digita el pedido en una computadora (máquina 2), el pedido llega al cocinero para que lo prepare (máquina 3) en el horno (máquina 4) y al barman (máquina 5), el garzón entrega el pedido al cliente, el cliente pide la cuenta, el garzón solicita el detalle al cajero (máquina 6), el cajero recibe el dinero y entrega el garzón el vuelto al cliente, y así concluye el pedido. Cada acción realizada por una máquina correspondería a una operación, mientras que el conjunto de dichas operaciones constituiría el *job*. Asignar los pedidos de manera que se atrase la menor cantidad, correspondería a un problema de *Scheduling*.

Todos los problemas de *scheduling* consideran un determinado número de *jobs*, n , y un número determinado de máquinas, m . Para un determinado *job* se utilizará el índice j y para una máquina determinada se utilizará el índice i . El par (i, j) se refiere a la operación del *job* j en la máquina i . Dentro del concepto de *job*, términos relevantes son el tiempo de procesamiento, fecha de llegada, fecha de entrega y peso. Las definiciones de dichos términos relacionados al *job* j son:

1. Tiempo de procesamiento ($p_{ij} \geq 0$): representa el tiempo de procesamiento del *job* j en la máquina i . Si el índice i se omite, el tiempo de procesamiento no depende de la máquina o el *job* j solo se procesará en una máquina.
2. Fecha de llegada ($r_j \geq 0$): es el periodo en el que el *job* j llega al sistema, por lo que es el periodo más temprano en el que el trabajo puede empezar su procesamiento.
3. Fecha de entrega ($d_j \geq 0$): representa el periodo en el que está comprometida la entrega o finalización (pasa por todas las operaciones necesarias) de el *job* j . La finalización de una tarea después de la fecha de entrega es aceptada, pero se incurre en una multa. Cuando se debe cumplir con una fecha de entrega, se la denomina fecha límite y se denota por \bar{d}_j
4. Peso ($w_j \geq 0, \sum_{j=1}^n w_j = 1$): el peso de un *job* se refiere a un factor de prioridad, denotando la importancia de el *job* j respecto a las otras *jobs* del sistema. Por ejemplo, puede representar el costo de mantener el *job* en el sistema.

Al mismo tiempo, un problema de *Scheduling* es descrito por una tripleta $\alpha|\beta|\gamma$.

El campo α describe el ambiente de las máquinas y contiene solo una entrada. El campo β entrega detalles de las características de procesamiento y restricciones, y puede contener ninguna entrada, una sola entrada o múltiples entradas. El campo γ describe el objetivo a ser minimizado y suele tener una sola entrada. Esta tripleta es para facilitar la comprensión del ambiente, las características de procesamiento y el objetivo del problema de *Scheduling*, siendo una especie de notación común para describirlos.

Con el objetivo de mostrar los diferentes campos descritos por la tripleta $\alpha|\beta|\gamma$, se utilizará el siguiente apoyo gráfico. Cabe mencionar que estos son ejemplos para facilitar el entendimiento de los diferentes ambientes de máquinas y no son la manera oficial de representar un diagrama de *Scheduling*.

- **Máquinas:** las distintas máquinas serán graficadas por engranajes, de la siguiente manera.



Figura 3.88: Máquinas en Scheduling.
Fuente: Elaboración Propia

Cada subconjunto de engranajes representa una máquina, por lo que aquí habrían 5 máquinas distintas.

- **Jobs:** los distintos jobs serán graficados por papeles, de la siguiente manera.



Figura 3.89: Jobs en Scheduling.
Fuente: Elaboración Propia

3.3.1. Campo α : ambiente de máquinas

Como se señaló previamente, en el campo α se explicita el ambiente de máquinas a utilizar. Algunos de los ambientes de máquinas que son comúnmente utilizados serán descritos a continuación.

- **Una máquina (1):** el caso de una máquina es el más simple de todos los ambientes de máquinas posibles y es un caso especial con respecto a otros ambientes. En este ambiente hay solo una máquina que realiza el procesamiento de los diferentes *jobs*.

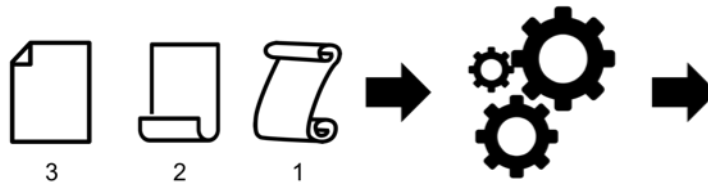


Figura 3.90: Ambiente de una máquina en Scheduling.
Fuente: Elaboración Propia

Los números indican el *job* y el orden de procesamientos.

- **Máquinas idénticas en paralelo (P_m):** hay m máquinas idénticas en paralelo. Un *job* j requiere una sola operación y puede ser procesada en cualquiera de las máquinas o en cualquiera que pertenezca a algún subconjunto determinado. Si este último es el caso, digamos que el *job* j debe ser procesado en un subconjunto M_j , entonces la entrada M_j aparece en el campo β .

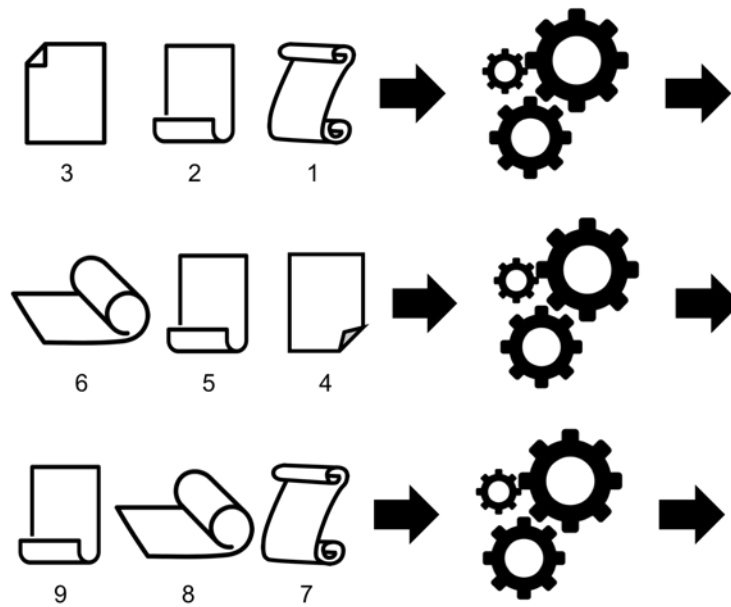


Figura 3.91: Ambiente de máquinas idénticas en paralelo en Scheduling.
Fuente: Elaboración Propia

En este caso, se tienen 9 *jobs* en un ambiente de 3 máquinas idénticas en paralelo, por lo que sería descrito por un $P3$ en el campo α . Los números indican el *job* a procesar y su resultado.

- **Máquinas paralelas con diferentes velocidades (Q_m):** hay m máquinas en paralelo con diferentes velocidades. La velocidad de una máquina i está denotada por v_i . El tiempo p_{ij} que un *job* j pasa en una máquina i es igual a p_j/v_i (asumiendo que el *job* j recibe todo su procesamiento en la máquina i). Este ambiente es referido como de máquinas uniformes. Si todas las máquinas tienen las mismas velocidades, es igual al ambiente anterior (P_m).

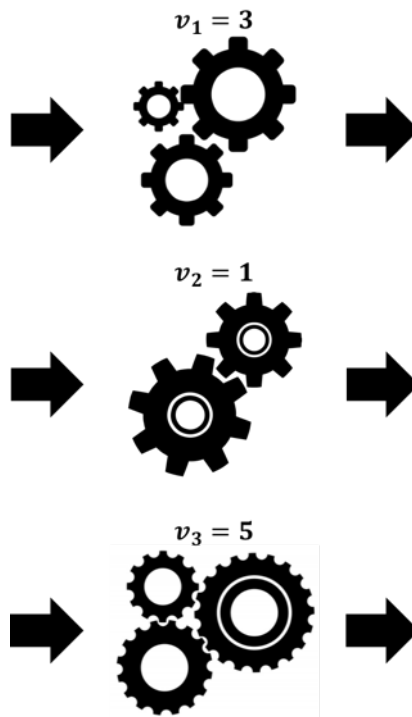


Figura 3.92: Ambiente de máquinas paralelas con diferentes velocidades en Scheduling.
Fuente: Elaboración Propia

En este ambiente, hay tres máquinas, todas con una velocidad distinta, por lo que el campo α quedaría descrito con un $Q3$.

- **Máquinas sin relacionar en paralelo (Rm):** este ambiente es una generalización más del ambiente anterior. Hay m máquinas diferentes trabajando en paralelo. La gran diferencia es que la máquina i puede procesar un *job* j a una velocidad v_{ij} . Es decir, la velocidad de procesamiento depende tanto de la máquina como de el *job*. El tiempo p_{ij} que un trabajo j pasa en una máquina i es igual a p_j/v_{ij} (asumiendo que el *job* j recibe todo su procesamiento en la máquina i). Si la velocidad de las máquinas es independiente de los *jobs*, es decir, $v_{ij} = v_i$ para todo i y j , entonces el ambiente es igual al anterior. Un ejemplo está descrito por la siguiente matriz, en la que la velocidad de procesamiento dependen tanto de la máquina como del *job*.

	$v_{11} = 3$	$v_{12} = 1$	$v_{13} = 4$
	$v_{21} = 2$	$v_{22} = 3$	$v_{23} = 1$
	$v_{31} = 7$	$v_{32} = 1$	$v_{33} = 5$

Figura 3.93: Ambiente de máquinas sin relacionar en paralelo en Scheduling.
Fuente: Elaboración Propia

Si las máquinas que están en esta tabla, estuvieran en paralelo y estas fueran sus velocidades, dependiendo de los *jobs*, estaríamos en el caso tenemos de un ambiente $R3$ en el campo α .

- **Flow shop (Fm):** hay m máquinas en serie. Cada *job* tiene que ser procesada en cada una de las m máquinas. Todas los *jobs* siguen la misma ruta, es decir, pasan primero por la máquina 1, luego por la máquina 2, y así sucesivamente. Después de ser procesada en una máquina, el *job* pasa a ser parte de la cola de la máquina siguiente. Usualmente, se asume que las colas operan bajo la disciplina *FIFO* (*First In First Out*). Esto quiere decir que la primera que llega es la primera que sale, además de que una tarea no se puede saltar a otra en la fila. Si la disciplina *FIFO* es utilizada, el *flow shop* se denomina como de permutación y el campo β incluye la entrada *prmu*.

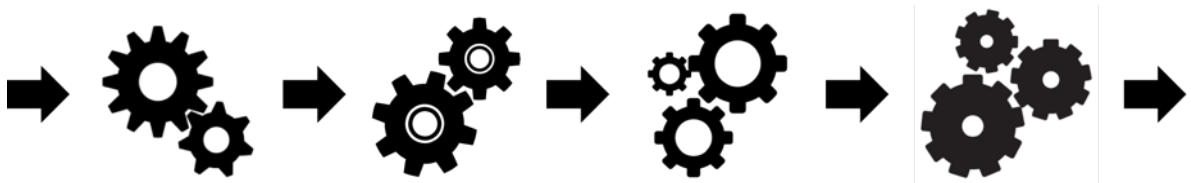


Figura 3.94: Flow shop en Scheduling.
Fuente: Elaboración Propia

En el *flow shop* de la Figura de arriba, hay 4 máquinas en serie. Todas los *jobs* que entren a este ambiente tendrán que pasar por las cuatro máquinas. En este caso, el campo α estaría descrito por $F4$.

- **Flow shop flexible (FFc):** un *flow shop* flexible es una generalización del *flow shop* y de los ambientes de máquinas en paralelo. En vez de m máquinas en serie, hay c etapas en series, con un

número de máquinas idénticas en paralelo en cada etapa. Cada *job* tiene que ser procesado en la etapa 1, luego en la etapa 2, y así. Una etapa funciona como un banco de máquinas en paralelo; en cada etapa, el *job j* requiere ser procesada solo por una máquina y cualquier máquina lo puede hacer. Las colas entre las distintas etapas pueden o no estar operando bajo la disciplina *FIFO*.

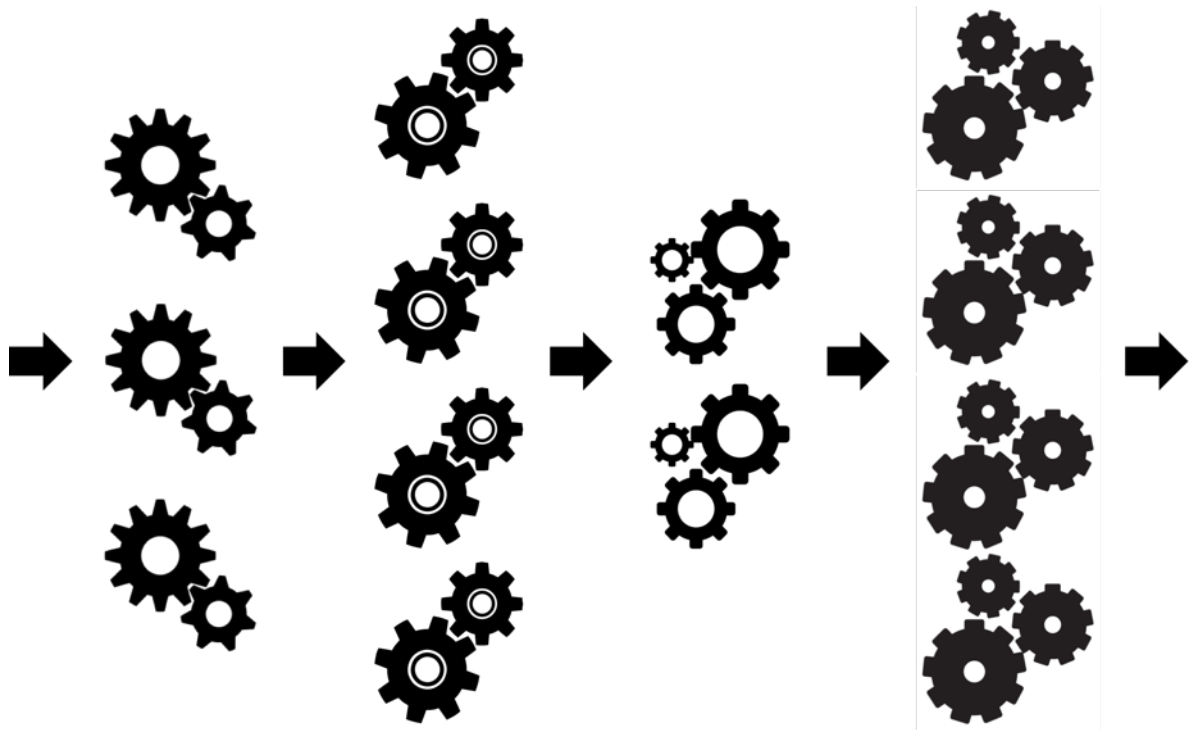


Figura 3.95: Flow shop flexible en Scheduling.
Fuente: Elaboración Propia

En este ejemplo existen 4 etapas y en cada etapa solo hay máquinas idénticas. Por lo tanto, este ambiente tendría *FF4* como descripción en el campo α .

- **Job shop (Jm):** en un *job shop* con m máquinas, cada trabajo tiene su propia ruta predeterminada a seguir. Se hace una distinción entre los *job shops* en los cuales cada *job* visita cada máquina al menos una vez y aquellos en los que los *jobs* pueden visitar una máquina más de una vez. En el último caso, el campo β contiene una entrada *rcrc* por recirculación.

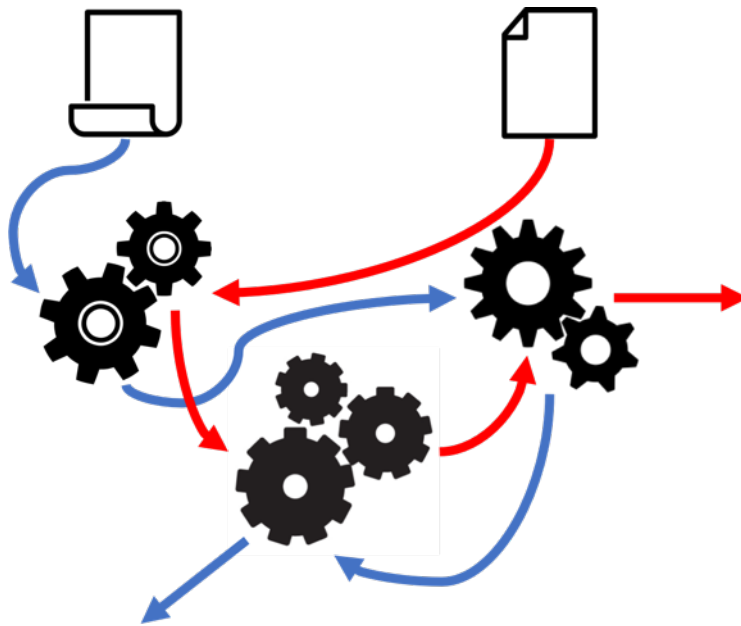


Figura 3.96: Job shop en Scheduling.
Fuente: Elaboración Propia

En esta figura se puede apreciar lo que sería un *job shop*. Cada tipo de *job* sigue su propia ruta, utilizando cada una de las máquinas. Puede darse el caso en el que un *job* no requiera procesamiento en alguna de las máquinas. Lo importante es resaltar que el mismo tipo de *job* seguiría el mismo camino, ya que las operaciones para procesar cada tipo de *job* son conocidas y están fijas. En este caso, se tiene un campo α descrito por $J3$.

- **Job shop flexible (FJc):** un *job shop* flexible es una generalización del *job shop* y los ambientes de máquinas en paralelo. En vez de m máquinas en serie, hay c centros de trabajo, con un número de máquinas idénticas en paralelo en cada centro de trabajo. Cada *job* tiene su propia ruta que seguir a través del shop; el *job* j requiere ser procesado en cada centro de trabajo en una sola máquina y puede ser en cualquier máquina. Si algún *job* en su ruta debe pasar por un centro de trabajo más de una vez, entonces el campo β contiene una entrada *rcrc* por recirculación.

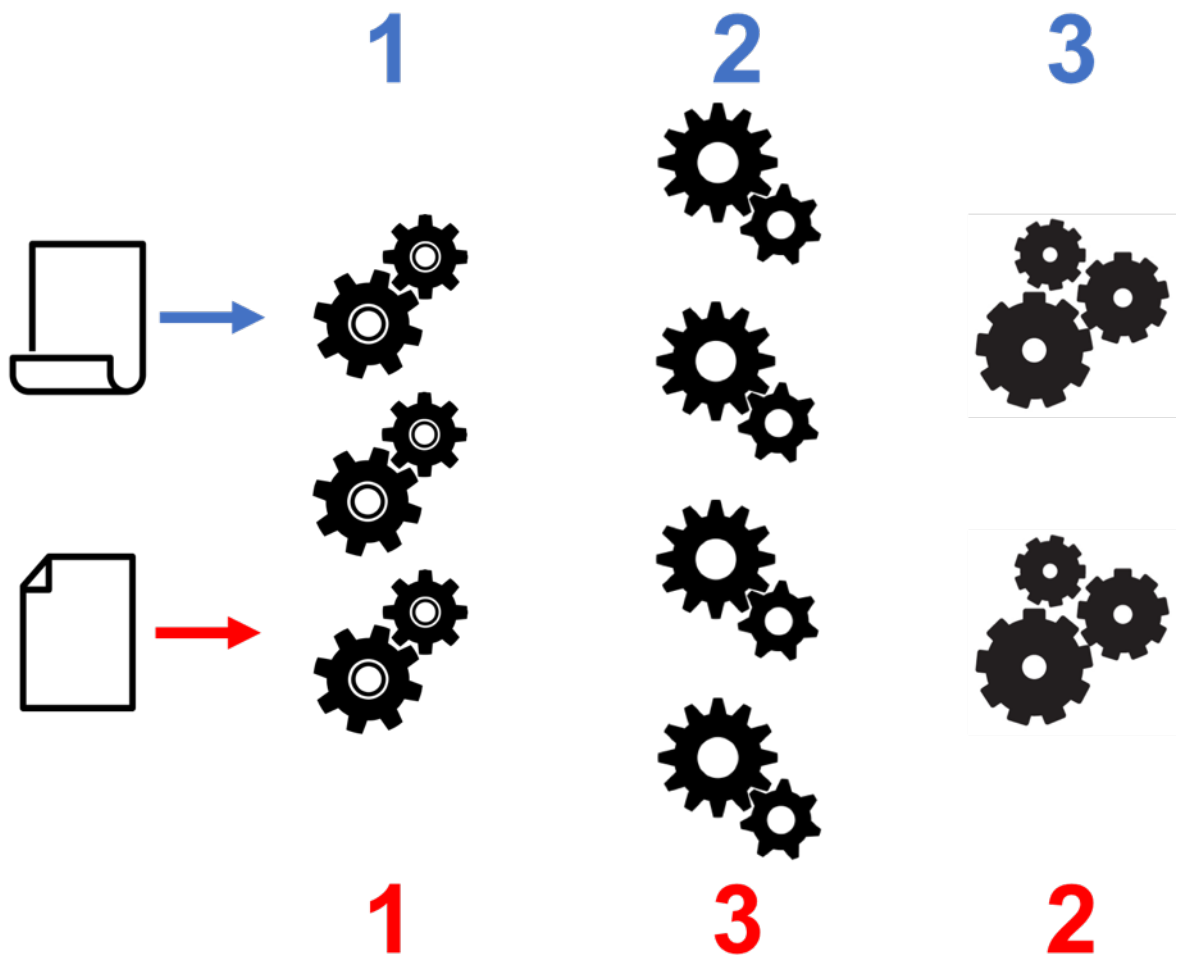


Figura 3.97: Job shop flexible en Scheduling.
Fuente: Elaboración Propia

Utilizando el mismo ejemplo anterior, pero ahora con centros de trabajo. Los números representan el orden que seguirían, por tipo de *job*. En este caso, correspondería que el campo α esté descrito por *FJ3*, al existir 3 centros de trabajo.

- **Open shop** (Om): hay m máquinas. Cada *job* debe ser procesado en cada una de las máquinas. Sin embargo, algunos de los tiempos de procesamiento pueden ser cero. No hay restricciones con respecto al ruteo de cada trabajo a través del ambiente de las máquinas. El *scheduler* puede determinar la ruta para cada uno de los *jobs* y las distintos *jobs* pueden tener distintas rutas. Este es un caso especial de un *job shop*, por lo que no será ilustrado por una imagen.

3.3.2. Campo β : restricciones

En el campo β se presentan las características de procesamiento y restricciones específicas, por lo que pueden incluir múltiples entradas. Algunas posibles entradas son:

- **Fecha de llegada (r_j):** si este símbolo aparece en el campo β , entonces el *job* j no puede comenzar su procesamiento antes de su fecha de llegada r_j . Si este símbolo no aparece en el campo, el procesamiento de dicho *job* puede comenzar en cualquier periodo. En contraste a las fechas de llegada, las fechas de entrega no son especificadas en este campo. El tipo de función objetivo entrega indicaciones suficientes para determinar si hay o no fechas de entrega.
- **Interrupciones ($prmp$):** implica que no es necesario mantener ininterrumpidamente un *job* en una máquina desde que se inicia, hasta completar su procesamiento. El *scheduler* está autorizado a interrumpir el procesamiento de un *job* (*preempt*) en cualquier periodo y poner un *job* distinto en esa máquina. Lo que se ha procesado de un *job* en una máquina no se pierde al ser interrumpida. Cuando un trabajo interrumpido se procesa de nuevo en la máquina (o en otra, en el caso de máquinas en paralelo), solo necesita estar el tiempo que le queda de procesamiento. Cuando se permiten *interrupciones*, se incluye $prmp$ en el campo β ; cuando no se incluye, no se aceptan *interrupciones*.
- **Restricciones de precedencia ($prec$):** las restricciones de precedencia pueden aparecer tanto en ambientes de una máquina o de varias máquinas en paralelo, necesitan que uno o más *jobs* se completen antes de que se permita que otro *job* se comience a procesar. Hay muchas formas especiales de restricciones de precedencia: si cada *job* tiene a lo más un predecesor y a lo más un sucesor, las restricciones son denominadas como cadenas. Si cada *job* tiene a lo más un sucesor, las restricciones se denominan como *intree*. Si cada *job* tiene a lo más un predecesor, las restricciones se denominan como *outtree*. Si no aparece $prec$ en el campo β , no hay restricciones de precedencia. Este tipo de restricción tiene sentido cuando un *job* es necesario para poder realizar otro. Un ejemplo sería el armado de una pizza: antes de agregar el queso, es estrictamente necesario tener lista la masa, por lo que el *job* “hacer la masa” es precededora de el *job* “agregar el queso”.
- **Tiempos de preparación dependiente de la secuencia (s_{jk}):** representa el tiempo de configuración en el que se incurre entre los procesamientos de el *job* j y el *job* k ; s_{0k} denota el tiempo de configuración para el *job* k , cuando corresponde a la primera tarea en la secuencia y s_{j0} el tiempo de limpieza después de el *job* j si ésta es la última de la secuencia (tanto s_{0k} como s_{j0} pueden ser 0). Si el tiempo de configuración depende además de la máquina se agrega el índice de la máquina i , por ejemplo, s_{ijk} . Si s_{jk} no aparece en el campo β , se asume que todos los tiempos de configuración son 0 o que no dependen de las secuencia. En el último caso se incluyen en los tiempos de procesamiento. Un ejemplo son los restaurantes. Al procesar distintos productos, los cocineros deben evitar la contaminación cruzada en la preparación, limpiando y cambiando recurrentemente los utensilios que utilizan.
- **Familias de tareas ($fmls$):** en el caso cuando hay familias de tareas, los n *jobs* pertenecen a F familias de *jobs* diferentes. Los *jobs* que pertenecen a la misma familia pueden tener tiempos de procesamiento diferentes, pero pueden ser procesadas en una máquina después de otra tarea sin necesitar ninguna configuración entre medio. Sin embargo, si una máquina cambia de una familia a otra, digamos que de la familia g a la familia h , se requiere una configuración de cambio. Si el tiempo de configuración depende de ambas familias y de la secuencia, se denota entonces como

s_{gh} . Si este tiempo solo depende de la familia que está por procesarse (en el ejemplo la familia h), entonces se denota como s_h . Si no depende de ninguna de las dos familias, se denota como s . Un ejemplo de familias de tareas es cuando en una línea de producción, es necesario hacer cambios a la línea para distintos productos, pero no para productos del mismo tipo. Esto ocurre en las embotelladoras, cuando cambian entre sabores de refrescos y formatos.

- **Procesamiento por lotes** ($batch(b)$): una máquina puede estar habilitada para procesar un número de $jobs$, digamos que un número b , de manera simultánea; esto es que pueda procesar un lote de hasta b $jobs$ al mismo tiempo. Los tiempos de procesamiento de los $jobs$ puede que no sean los mismos, por lo que el procesamiento de un lote termina cuando el último job es completado, lo que implica que el tiempo de procesamiento de un lote completo corresponde al de el job con el mayor tiempo de procesamiento. Si $b = 1$, entonces el problema se reduce a un ambiente de *Scheduling* común. Otro caso especial de interés es cuando $b = \infty$, es decir, no hay límite en el número de $jobs$ que una máquina puede procesar en cualquier momento.
- **Averías** ($brkdn$): las averías de las máquinas implican que éstas no estén continuamente disponibles. Los periodos en los que una máquina no esté disponible se asumen fijas para el alcance de esta memoria (por ejemplo, debido a cambios o mantenciones), ya que se revisa solo lo que tiene relación con *Scheduling* determinístico. Si hay un número de máquinas idénticas en paralelo, el número de máquinas disponibles en punto de tiempo es una función del periodo, es decir, $m(t)$. Las averías de las máquinas son también referidas como restricciones de disponibilidad de máquinas en periodos.
- **Restricciones de elegibilidad de máquinas** (M_j): este símbolo puede aparecer en el campo β cuando el ambiente de máquinas corresponde a m máquinas en paralelo (Pm). Cuando M_j está presente, representa al subconjunto de máquinas que pueden procesar el job j . Si en el campo β , M_j no está presente, el job j puede ser procesada en cualquiera de las m máquinas.
- **Permutación** ($prmu$): esta restricción puede aparecer en el ambiente *Flow Shop* y se refiere a que las colas en las que los $jobs$ esperan por las máquinas, operan acorde a la disciplina *FIFO*. Esto implica que el orden (o permutación) en la que los $jobs$ pasan por la primera máquina, es el orden que mantienen hasta que salen.
- **Bloqueo** ($block$): el bloqueo es un fenómeno que puede ocurrir en los *flow shops*. Si un *flow shop* tiene una capacidad limitada de almacenamiento de espera (búfer) entre dos máquinas sucesivas, entonces, puede ocurrir que cuando dicha capacidad esté ocupada al máximo, no se permite que la primera máquina libere una tarea ya procesada. El bloqueo implica que un job completado tiene que mantenerse dentro de la primera máquina, evitando (bloqueando) que la esta máquina procese el siguiente job .
- **No espera** (nwt): el requerimiento de no espera es otro fenómeno que puede ocurrir en *flow shops*. Los $jobs$ no tienen permitido esperar entre dos máquinas consecutivas. Esto implica que el inicio del procesamiento de un job en la primera máquina tiene que ser retrasado hasta asegurar que no tendrá que esperar cualquier máquina. Un ejemplo de dicha operación es un tren de laminación de acero en la cual una losa de acero no tiene permitido una espera, dado a que se enfriaría durante ella.
- **Recirculación** ($rcrc$): la recirculación puede ocurrir en un *job shop* o un *job shop* flexible, cuando

un *job* puede visitar una máquina o una estación de trabajo más de una vez.

Cualquier otra entrada que aparezca en el campo β se explica sola. Por ejemplo, $p_j = p$ implica que todos los tiempos de procesamiento son iguales y $d_j = d$ implica que todas las fechas de entrega son la misma. Como se ha dicho anteriormente, las fechas de entrega, en contraste con las fechas de llegada, usualmente no se explicitan en este campo; el tipo de función objetivo da la suficiente información sobre si los *jobs* tienen o no fechas de entrega.

3.3.3. Campo γ : función objetivo

Finalmente, el campo γ corresponde a la función objetivo, es decir, a lo que se busca lograr realizando el *scheduling*. El objetivo a ser minimizado es siempre una función de los tiempos de finalización de los *jobs*, la cual, obviamente, depende del agendamiento (*Schedule*). El tiempo de finalización de operación de el *job* j en la máquina i se denota C_{ij} . El tiempo en el que el *job* j sale del sistema (es decir, el tiempo de finalización en la última máquina en la que requiere procesamiento) es denotado por C_j . El objetivo también puede ser una función de los tiempos de entrega. El atraso (lateness) de un *job* j se define como

$$L_j = C_j - d_j \quad (3.1)$$

que es positivo cuando el *job* j es completada tarde y negativo cuando se completa antes de tiempo. La tardanza (*tardiness*) de un *job* j se define como

$$T_j = \max(C_j - d_j, 0) = \max(L_j, 0) \quad (3.2)$$

La diferencia entre tardanza y atraso yace en el hecho de que la tardanza nunca es negativa. La unidad de castigo (*penalty unit*) del *job* j es definida como

$$U_j = \begin{cases} 1 & \text{si } C_j > d_j \\ 0 & \text{si no} \end{cases} \quad (3.3)$$

Algunos ejemplos de posibles funciones objetivos a ser minimizadas son:

- **Makespan** (C_{max}): el *makespan*, definido como $\max(C_1, \dots, C_n)$ es equivalente al tiempo de finalización del último trabajo que abandona el sistema. Un mínimo *makespan* implica, usualmente, una buena utilización de las máquinas.
- **Máximo atraso** (L_{max}): el máximo atraso es definido como $\max(L_1, \dots, L_n)$. Mide la peor infracción a las fechas de entrega.
- **Tiempo de finalización total ponderado** ($\sum w_j C_j$): la suma ponderada de los tiempos de finalización de las n *jobs* da una indicación de la retención total o los costos de inventario incurridos por el agendamiento. La suma de los tiempos de finalización se encuentra en la literatura a menudo denominada tiempo de flujo. El tiempo de finalización total ponderado se denomina tiempo de flujo ponderado.

- **Tiempo de finalización total ponderado descontado** ($\sum w_j(1 - e^{-rC_j})$): esto es una función de costos más general que la anterior, donde los costos son descontados a una tasa r , $0 < r < 1$, por unidad de tiempo. Esto es, si una tarea j no es completada en un periodo t un costo adicional $w_j re^{-rt} dt$ es incurrido en el lapso $[t, t + dt]$. Si el *job* es finalizada en el periodo t , el costo total incurrido en el lapso $[0, t]$ es $w_j (1 - e^{-rt})$. El valor de r es cercano a 0, como 0,1.
- **Tardanza total ponderada** ($\sum w_j T_j$): esta también es una función de costos más general que el tiempo de finalización total ponderado.
- **Número ponderado de tareas tardías** ($\sum w_j U_j$): el número ponderado de tareas tardías no solo es una medida de interés académico, si no que también se toma como objetivo en la práctica, ya que es una medida que registrarse con mucha facilidad.

Todas las funciones objetivo mencionadas recientemente son llamadas medidas de rendimiento *regular*. Una medida de rendimiento regular es una función que es no decreciente en C_1, \dots, C_n . Recientemente, los investigadores han comenzado a estudiar funciones objetivo que no son regulares, pero que no se mencionarán en esta tesis.

Capítulo 4

Desarrollo de Propuesta

En el presente capítulo comprende el análisis para poder cumplir con el propósito de este estudio, el cual es plantear una mejora a BPMN 2.0, utilizando características propias de *Scheduling*, buscando robustecer la herramienta actual. Esto mejorará el entendimiento de los procesos y su contexto, considerando elementos cruciales que pueden afectar en la toma de decisiones.

Para ello se realiza una tabla comparativa la que nos permite reconocer que elementos de *Scheduling* se encuentran presentes en BPMN 2.0.

Con el fin de analizar las similitudes entre BPMN 2.0 y *Scheduling* para determinar qué aspectos en común tienen ambas corrientes, para luego determinar las diferencias existentes entre BPMN 2.0 y *Scheduling* y, de esta forma, con las diferencias encontradas establecer un criterio a utilizar para posteriormente hacer una selección de las características que se agregarán como una mejora a BPMN.

4.1. Comparación de los elementos propios de *Scheduling* con los de BPMN 2.0.

En las siguientes tablas se realiza la comparación de los distintos elementos existentes en *Scheduling* con los presentes en el estándar BPMN 2.0. Con esto se busca identificar aquellos que no están presentes en BPMN 2.0 y que podrían servir para aumentar la capacidad explicativa de los procesos, además de dotarlos con características que lo acercaría a resolver el problema de la asignación eficiente de recursos a la ejecución de tareas.

Para ello, se detalla en la primera columna cada uno de los elementos de *Scheduling* descritos en los capítulos anteriores, señalando en la segunda columna cuál de los elementos de BPMN (si existe) corresponde a su símil. Luego, en una tercera columna, una explicación de esta comparación.

Tabla 4.1: Comparativa Conceptos básicos de Scheduling ¿Qué elementos de *Scheduling* se encuentran en BPMN?

Scheduling	BPMN 2.0	Explicación
Máquina	Compartimento	Una máquina de <i>Scheduling</i> está representada por un compartimento en BPMN, ya que corresponde a la unidad de trabajo, parte de una organización, que lleva a cabo las actividades que componen el flujo de proceso. En particular, un ambiente de una máquina correspondería a un compartimento en el que solo hay una persona/máquina que ejecuta las actividades.
<i>Job</i>	Instancia	Un <i>job</i> consiste en una serie de acciones que es necesario se lleven a cabo. Por ende, está representada por una instancia en el flujo de proceso, en el que las acciones corresponden a las tareas realizadas dentro de los compartimentos, los cuales representan a las máquinas.
Operación	Actividad	Las diferentes operaciones necesarias para completar un <i>job</i> en <i>Scheduling</i> , están representadas por las actividades en BPMN, ya que representan una tarea puntual a llevarse a cabo dentro de los compartimentos.

Fuente: Elaboración propia.

Tabla 4.2: Comparativa ambiente Alpha de Scheduling con BPMN 2.0

Scheduling	BPMN 2.0	Explicación
Una Máquina	Compartimento	El ambiente de una máquina consiste, en una sola máquina procesando las diferentes tareas. Por ende, se puede representar por un carril, en donde se lleven a cabo todas las actividades. Sin embargo, cabe mencionar que el concepto de capacidad no existe en BPMN 2.0, por lo que un compartimento puede representar a una máquina o a varias máquinas en paralelo.
Máquinas idénticas en paralelo	Compartimento	El concepto de capacidad no existe en BPMN 2.0, por lo que, para representar máquinas idénticas en paralelo, también se utiliza un compartimento.
Máquinas paralelas con diferentes velocidades	No existe, porque no hay concepto de velocidad en BPMN	En BPMN 2.0, no existe el concepto de velocidad, en ninguna parte del estándar encontramos que una tarea es realizada a determinados unidades por segundo, por ejemplo. Es por esto que no es posible diagramar en BPMN 2.0 máquinas paralelas con diferentes velocidades.
Máquinas sin relacionar en paralelo	No existe, porque no hay concepto de velocidad en BPMN	Al no existir en BPMN 2.0 el concepto de velocidad no es posible diagramar máquinas sin relacionar en paralelo, debido a que en este caso existe una cantidad de máquinas diferentes trabajando en paralelo, pero la velocidad de procesamiento depende tanto de la máquina como del <i>job</i> .

Fuente: Elaboración propia.

Tabla 4.3: Continuación comparativa ambiente Alpha de Scheduling con BPMN 2.0

Scheduling	BPMN 2.0	Explicación
Flow shop flexible	Esquema de varios compartimentos, en los que solo hay actividades	Un <i>flow shop flexible</i> es un tipo de <i>flow shop</i> en el que existe una serie de etapas, que corresponden a máquinas idénticas en paralelo. Todos los <i>jobs</i> pasan por todas las etapas, siguiendo una misma ruta. Por esta razón, puede estar representado al igual que el <i>flow shop</i> , por un esquema básico en BPMN, con varios compartimentos. Dado que no existe el concepto capacidad en BPMN, no se puede explicitar lo de las etapas, ya que no se puede representar la cantidad de máquinas por etapa.
Job shop	Varios compartimentos, con un flujo de proceso que contenga compuertas exclusivas en algunas partes del proceso	Un <i>job shop</i> es un tipo de proceso que se caracteriza en que sus tareas no necesariamente pasan a través de todos sus procesos. Es decir, cada tarea pasa sólo por las actividades necesarias para llevarse a cabo. Por esta misma razón, un flujo de procesos con compuertas exclusivas en BPMN permite diagramar la situación, debido a la compuerta exclusiva evalúa el estado del proceso de negocio y, según esa condición, separa el flujo en una o más rutas que se excluyen mutuamente.
Job shop flexible	Varios compartimentos, con un flujo de proceso que contenga compuertas exclusivas en algunas partes del proceso	Un <i>job shop flexible</i> corresponde a un ambiente en el que cada <i>job</i> tiene una ruta predeterminada a seguir por estaciones de trabajo. Éstas corresponden a máquinas idénticas en paralelo. No se requiere que los jobs pasen por todas las estaciones. Dado que en BPMN 2.0 no se puede representar el concepto de capacidad (cantidad de máquinas), se diagramaría igual que un <i>job shop</i> .
Open shop	Proceso Ad Hoc	Un <i>open shop</i> es un tipo de ambiente en el cual cada <i>job</i> sigue su propio camino a través de las máquinas y cada <i>job</i> debe pasar al menos una vez por todas las máquinas. Este tipo de proceso complejo se puede representar a través de un proceso Ad Hoc, el cual es un modelado avanzado de BPMN. Esto se modela con el nuevo estándar CMMN (Case Management Model and Notation), también administrado por la OMG. Consiste en una serie de actividades que no tienen un orden definido. Pero este nuevo estándar no es parte de este trabajo.

Fuente: Elaboración propia.

Tabla 4.4: Comparativa ambiente Beta de *Scheduling* con BPMN 2.0 [1/4]

Scheduling	BPMN 2.0	Explicación
Fecha de llegada	No existe	En BPMN 2.0 no existe actualmente un símbolo que represente la fecha en la que entra el trabajo al sistema.
Interrupciones	Existe Parcialmente	Una restricción de interrupciones implica que no es necesario mantener ininterrumpidamente un <i>job</i> en una máquina desde que se inicia hasta completar su procesamiento. Es decir, está autorizado el interrumpir el procesamiento de un <i>job</i> en cualquier periodo y poner un <i>job</i> distinto en esa máquina. Lo que se ha procesado de un <i>job</i> no se pierde al ser interrumpido. En BPMN existe el evento intermedio adjunto interruptor. Este tipo de evento detiene la actividad que se está llevando a cabo, dirigiendo el flujo por otro camino. Sin embargo, no es modelable una interrupción en cada actividad, además de que no se pueda retornar a la actividad interrumpida previamente, por lo que esta restricción no está completamente representada en BPMN 2.0.
Restricciones de precedencia	Puede ser representado por un diagrama común con un solo carril o con un evento de inicio condicional en un diagrama de un solo carril	Las restricciones de precedencia pueden aparecer tanto en ambientes de una maquina o de varias máquinas en paralelo. En presencia de esta restricción, existen algunos <i>jobs</i> que requieren que previamente se hayan ejecutado algún subconjunto específico de <i>jobs</i> . En BPMN 2.0 esto se puede representar mediante un evento de inicio condicional, en el que la condición corresponda a que se haya completado el subconjunto específico para dicho <i>job</i> . O también con un diagrama tradicional de BPMN, ya que cada actividad requiere que las demás anteriores se hayan completado para poder llevarse a cabo.

Fuente: Elaboración propia.

Tabla 4.5: Continuación comparativa ambiente Beta de *Scheduling* con BPMN 2.0 [2/4]

Scheduling	BPMN 2.0	Explicación
Tiempo de preparación dependiente de la secuencia	No existe	Esta restricción representa el tiempo de preparación en el que se incurre entre los procesamientos de dos <i>jobs</i> consecutivos. En otras palabras, es el tiempo para preparar las máquinas para el cambio de operaciones. Esto no es posible graficar en BPMN, debido a que este estándar no considera tiempos de ningún tipo.
Familia de tareas	No existe	En BPMN no existen los dos conceptos necesarios detrás de la restricción de familia de tareas: no existe el concepto de agrupación de <i>jobs</i> de un mismo tipo (familia), así como tampoco el concepto de tiempo de preparación, ya que no se consideran tiempos en BPMN.
Procesamiento por lotes	Un compartimento que contenga en su interior actividades con marcador de instancias múltiples en paralelo.	Una máquina puede estar habilitada para procesar un número de <i>jobs</i> , digamos que un número <i>b</i> , de manera simultánea, lo que significa que pueda procesar un lote de hasta <i>b jobs</i> al mismo tiempo, por lo que podría representarse con un compartimento en el cual todas las actividades tengan marcador de instancias múltiples en paralelo, donde la cantidad de instancias múltiples sería <i>b</i> .
Averías	Evento error	La restricción avería se refiere a la falla en una máquina, es decir, a que no estén continuamente disponibles. Es por ello que la podemos representar en BPMN con un evento que normalmente interrumpirá el proceso o requerirá corrección, este es el evento error.

Fuente: Elaboración propia.

Tabla 4.6: Continuación comparativa ambiente Beta de *Scheduling* con BPMN 2.0 [3/4]

Scheduling	BPMN 2.0	Explicación
Restricciones de elegibilidad de máquinas	Flujo de procesos con compuertas exclusivas	Esta restricción de elegibilidad de máquinas, representa al subconjunto de máquinas que pueden procesar determinado <i>job</i> . Si esta restricción no está presente, el <i>job</i> puede ser procesada en cualquiera de las máquinas. Es por ello que es posible representar en BPMN como un flujo de procesos con compuertas exclusivas, debido a que este tipo de compuerta selecciona exactamente un flujo de secuencia de entre las alternativas existentes siendo la compuerta la que guía a que un subconjunto determinado de máquinas procese el <i>job</i> .
Permutación	No existe o no se declara la regla de ordenamiento en colas de procesamiento en BPMN	La restricción de permutación puede aparecer en un <i>flow shop</i> y se refiere a que las colas en las que los <i>jobs</i> esperan por las máquinas, operan acorde a la disciplina FIFO. Esto no es posible de diagramar en BPMN, debido a que no existen reglas de ordenamiento en las colas de los procesos existentes.
Bloqueo	Evento señal	La restricción de bloqueo ocurre en los <i>flow shops</i> . Si un <i>flow shop</i> tiene una capacidad limitada de almacenamiento de espera entre dos máquinas sucesivas, entonces ocurre un bloqueo cuando la capacidad de la primera máquina este ocupada al máximo implique el que no se permita que la primera máquina libere una tarea ya procesada, manteniéndose dentro de la primera máquina, bloqueando que la máquina procese la siguiente tarea. Esto es posible diagramar en BPMN con un evento de señal el cual es como una señal de bengala o una sirena de incendios la cual indicaría el bloqueo para que la máquina no procese la siguiente tarea. Estas señales no están dirigidas a un destinatario específico, en lugar de ello todos los que estén interesados pueden escuchar y detectar la señal y actuar en consecuencia.

Fuente: Elaboración propia.

Tabla 4.7: Continuación comparativa ambiente Beta de *Scheduling* con BPMN 2.0 [4/4]

Scheduling	BPMN 2.0	Explicación
No espera	No existe	La restricción de no espera ocurre en los <i>flow shops</i> , en los cuales las tareas no tienen permitido esperar entre dos máquinas consecutivas. Esto es no posible graficar en BPMN, debido a que no existe el concepto de tiempo de procesamiento ni el concepto de tiempo de espera en cola.
Recirculación	Flujo de proceso, en el cual se puede volver a utilizar uno o más compartimientos (iteración).	La restricción de recirculación puede ocurrir en los <i>job shops</i> cuando un <i>job</i> puede visitar una máquina o una estación de trabajo más de una vez. Esto lo podemos graficar en BPMN como un flujo de proceso, en el cual se vuelve a utilizar uno o más carriles (iteración), para llevar a cabo las mismas actividades más de una vez.

Fuente: Elaboración propia.

Tabla 4.8: Comparativa ambiente Gama de *Scheduling* con BPMN 2.0

Scheduling	BPMN 2.0	Explicación
Makespan	No existe	La función objetivo de <i>Scheduling</i> no es capturada por BPMN 2.0
Máximo atraso	No existe	La función objetivo de <i>Scheduling</i> no es capturada por BPMN 2.0
Tiempo de finalización total ponderado	No existe	El concepto gamma no fue expuesto en esta tabla debido a que la función objetivo de <i>Scheduling</i> no es capturada por BPMN 2.0
Tiempo de finalización total ponderado descontado	No existe	La función objetivo de <i>Scheduling</i> no es capturada por BPMN 2.0
Tardanza total ponderada	No existe	La función objetivo de <i>Scheduling</i> no es capturada por BPMN 2.0
Número ponderado de tareas tardías	No existe	La función objetivo de <i>Scheduling</i> no es capturada por BPMN 2.0

Fuente: Elaboración propia.

Capítulo 5

Resultados

5.1. Análisis Tabla Comparativa

Con respecto a las últimas tablas correspondientes a la comparativa de que elementos de *Scheduling* están presentes en BPMN 2.0 tenemos lo siguiente:

Del ambiente Alpha:

- Una máquina de *Scheduling* es posible representarla por un compartimento en BPMN 2.0
- Maquinas idénticas en paralelo es posible representarla por un compartimento en BPMN 2.0
- Máquinas paralelas con diferentes velocidades no es posible representar en BPMN.
- Maquinas sin relacionar en paralelo no es posible representar en BPMN.
- Flow shop es posible representarlo con un esquema de varios compartimentos que solo tengan actividades.
- Flow shop flexible de igual forma es posible representarlo de la misma forma que el Flow shop anterior debido a que no existe concepto de capacidad en BPMN.
- Job shop es posible representarlo con varios compartimentos que contengan un flujo con compuertas exclusivas.
- Job shop Flexible de igual forma es posible representarlo de la misma forma que el Flow shop anterior debido a que no existe concepto de capacidad en BPMN.
- Open shop este es un proceso más complejo, pero es posible representarlo con un nuevo estándar llamado CMMN, el cual no es parte de este trabajo.

Beta: Restricciones

- Fecha de llegada no es posible representar en BPMN.
- Interrupciones es posible graficarlo de manera parcial en BPMN con el evento intermedio adjunto interruptor.
- Restricciones de precedencia es posible representar con BPMN 2.0 con un solo compartimento.
- Tiempo de preparación dependiente de la secuencia no es posible representar en BPMN.
- Familia de tareas no es posible representar en BPMN.
- Procesamiento por lotes es posible representar en BPMN con un compartimento que tenga un flujo con actividades que tengan el marcador de instancias múltiples en paralelo.
- Averías es posible representar en BPMN con el evento error.
- Restricciones de elegibilidad de máquinas es posible representar con BPMN a través de un flujo que contenga compuertas exclusivas.
- Permutación no es posible representar en BPMN.
- Bloqueo es posible representar en BPMN con el evento de señal.
- No espera no es posible representar en BPMN.
- Recirculación es posible representar en BPMN con un flujo de procesos en el cual se vuelva a utilizar uno o más compartimentos.

Gamma: Función objetivo

- Makespan no es posible representarlo con el estándar BPMN 2.0.
- Máximo atraso no es posible representarlo con el estándar BPMN 2.0.
- Tiempo de finalización total ponderado no es posible representarlo con el estándar BPMN 2.0.
- Tiempo de finalización total ponderado descontado no es posible representarlo con el estándar BPMN 2.0.
- Tardanza total ponderada no es posible representarlo con el estándar BPMN 2.0.
- Número ponderado de tareas tardías no es posible representarlo con el estándar BPMN 2.0.

De esta forma podemos realizar dos listas, la primera que muestre los elementos de *Scheduling* que sí se pueden representar en BPMN 2.0 y la segunda que muestre los elementos de *Scheduling* que no es posible representar con BPMN 2.0.

5.1.1. Elementos de *Scheduling* que son representables en BPMN 2.0

- ✓ Una máquina
- ✓ Máquinas idénticas en paralelo
- ✓ Flow shop
- ✓ Flow shop flexible
- ✓ Job shop
- ✓ Job shop flexible
- ✓ Open shop
- ✓ Interrupciones
- ✓ Restricciones de precedencia
- ✓ Procesamiento por lotes
- ✓ Averías
- ✓ Restricciones de elegibilidad de máquinas
- ✓ Bloqueo
- ✓ Recirculación

BPMN 2.0 y *Scheduling* permiten representar y describir el flujo de un proceso. El primero representa a un proceso de manera gráfica (con sus cuatro categorías de elementos gráficos: objetos de flujo, objetos de conexión, carriles de rudo y artefactos). El segundo describe la manera en la que se llevará a cabo el flujo del proceso (ambiente de máquinas) y el orden preestablecido que pueden tener algunas tareas (restricciones de precedencia), sus características particulares y toma en cuenta alguna estructura para decidir cuál es el orden óptimo de procesamiento (función objetivo).

5.1.2. Elementos de *Scheduling* que no se pueden representar en BPMN 2.0

- ✗ Máquinas paralelas con diferentes velocidades
- ✗ Máquinas sin relacionar en paralelo
- ✗ Fecha de llegada
- ✗ Tiempo de preparación dependiente de la secuencia
- ✗ Familia de tareas

X Permutación

X No espera

De esta última lista, tomando en cuenta los elementos de *Scheduling* que no se pueden representar actualmente, es que se realiza una propuesta de mejora gráfica. Es importante destacar que no se agregaron todos los mencionados en la lista anterior como elementos gráficos a BPMN 2.0, lo cual es analizado en detalle a continuación.

5.2. Propuesta de Mejora BPMN 2.0

5.2.1. Tiempo de Procesamiento

Para que se pueda obtener la solución a un problema de *Scheduling*, es necesario contar con los tiempos de ejecución de las operaciones (tiempo de procesamiento).

Como el uso de procesos también es transversal a lo realizado en BPMN, se considera necesario agregar el tiempo de procesamiento de cada una de las actividades que se realizan en un flujo de proceso.

Para ello, se recomienda agregar en cada tarea de BPMN su tiempo de procesamiento promedio dentro de un paréntesis cuadrado. En la siguiente figura, se incluyen en color rojo en la parte inferior de cada tarea. De esta forma podemos determinar el tiempo de procesamiento del proceso completo, considerando distintos caminos que recorran las instancias.

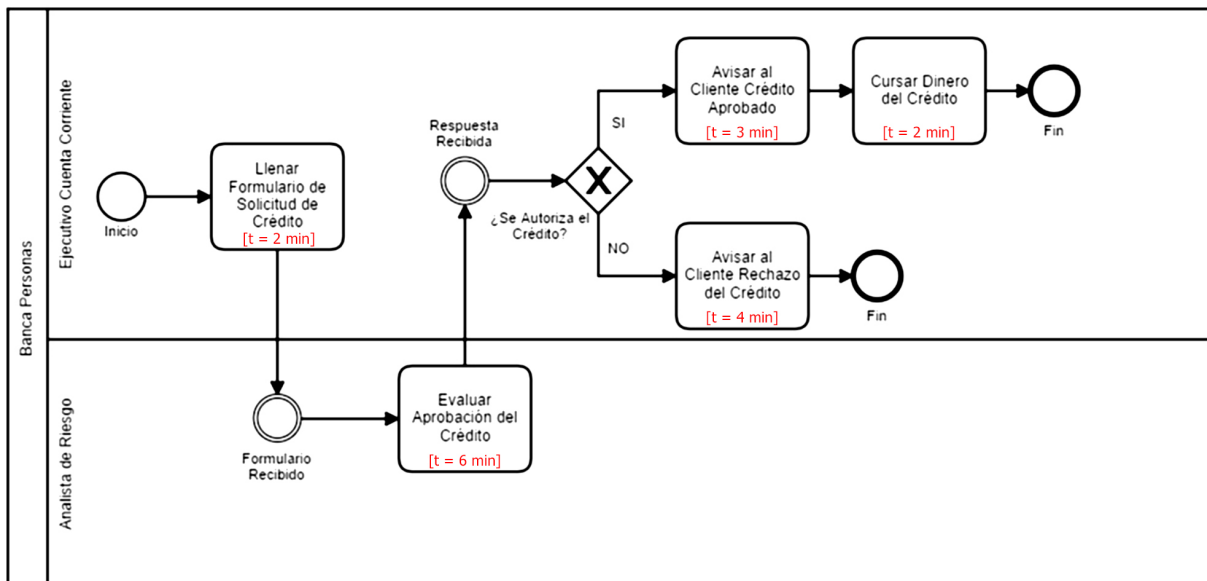


Figura 5.1: Propuesta gráfica tiempo de procesamiento

Fuente: Elaboración propia

La fecha de llegada de un *job* al sistema no es posible diagramar, debido a que una instancia no es visible en BPMN.

5.2.2. Tiempo de Preparación

Para el tiempo de preparación dependiente de la secuencia se propone agregar una actividad denominada “actividad de preparación”, la cual consiste en el tiempo que tardará en prepararse el sistema para procesar la tarea siguiente.

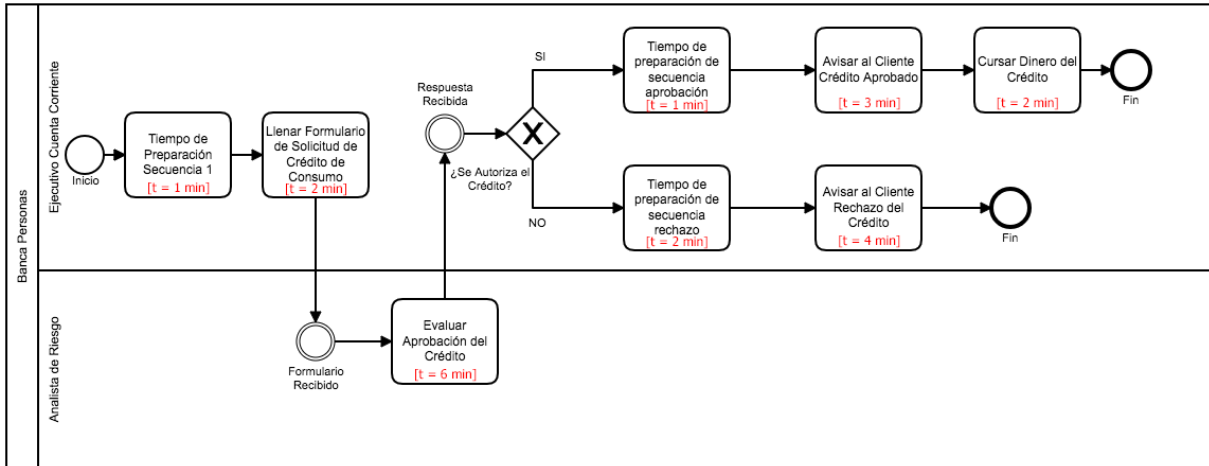


Figura 5.2: Propuesta gráfica tiempo de preparación dependiente de secuencia
Fuente: Elaboración propia

Para la familia de tareas no es posible diagramarlo: las tareas son instancias, en cambio todas las otras restricciones se refieren a las operaciones.

Para la permutación se plantean 3 tipos de reglas de ordenamiento de los *jobs*, es decir que estas guiarán la espera de las operaciones en las respectivas colas. Dichas reglas de ordenamiento se explican a continuación:

5.2.3. Reglas de Prioridad

Para esta mejora se propone que para el contenedor en su lado inferior derecho ubicar una flecha que indica el sentido del flujo y la regla de ordenamiento que tendrán las colas en todo el proceso.

- FIFO: primero en entrar primero en salir.

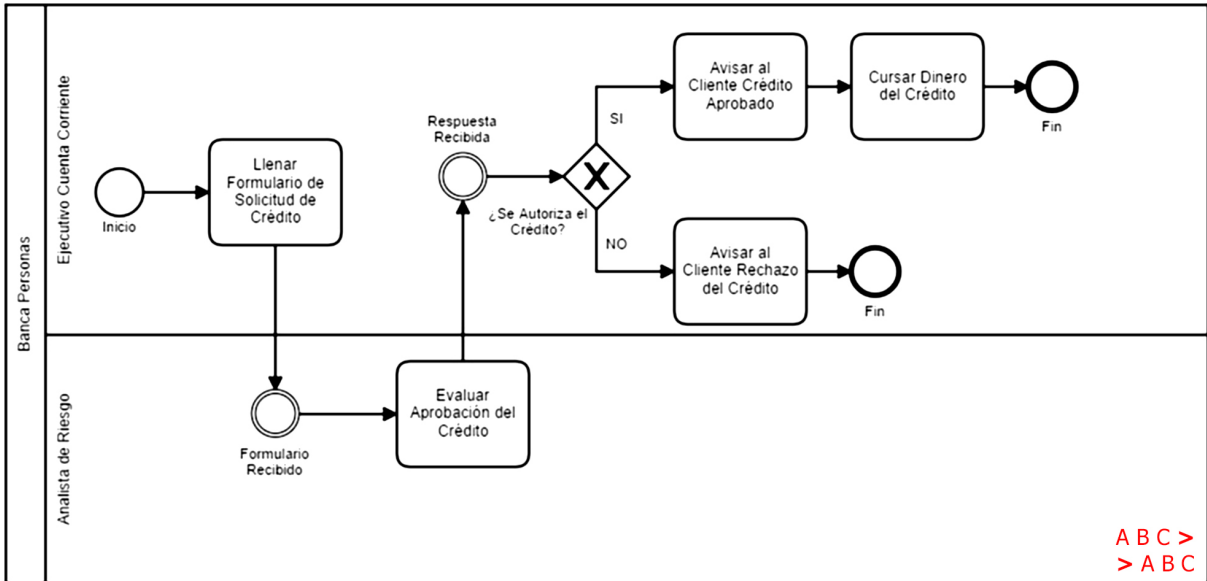


Figura 5.3: Propuesta gráfica regla de prioridad FIFO
Fuente: Elaboración propia

- LIFO: último en entrar, primero en salir.

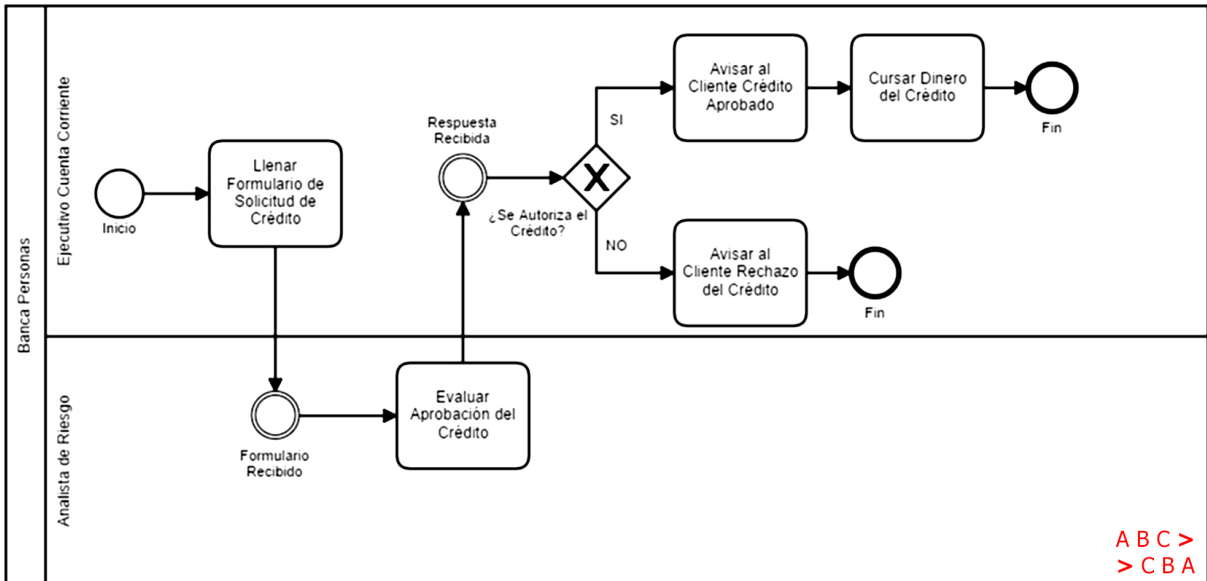


Figura 5.4: Propuesta gráfica regla de prioridad LIFO
Fuente: Elaboración propia

- Priority: los *jobs* se ordenan en las colas dependiendo de su prioridad (peso relativo).

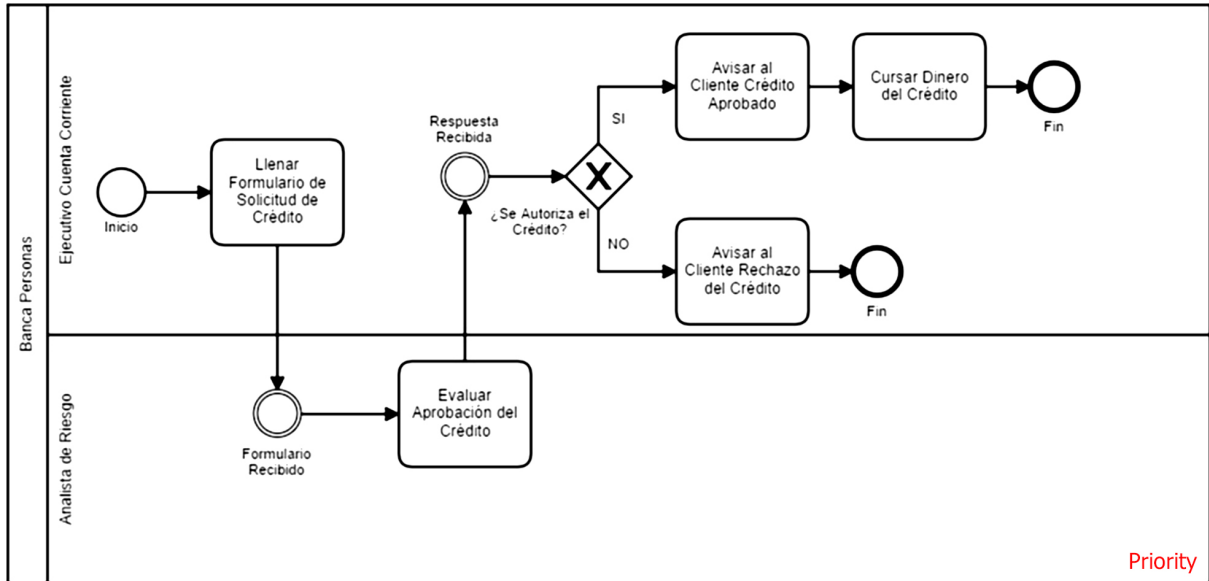


Figura 5.5: Propuesta gráfica regla de prioridad Priority

Fuente: Elaboración propia

La restricción de no espera no es posible diagramarla, ya que no se puede representar el tiempo que la tarea espera para ser procesada.

Finalmente, para complementar el estándar BPMN 2.0 se agregan dos mejoras, las cuales son la capacidad de procesamiento y la función objetivo.

5.2.4. Capacidad de Procesamiento

Esta mejora consiste en agregar en el extremo izquierdo superior de cada compartimento de BPMN la capacidad de cada contenedor, para poder saber la cantidad existente de cada máquina en cada compartimento. De esta forma es posible determinar de mejor manera el tiempo de procesamiento del proceso de un conjunto de tareas, ya que es distinto que un trabajo lo hagan 2 máquinas a que una, por ejemplo:

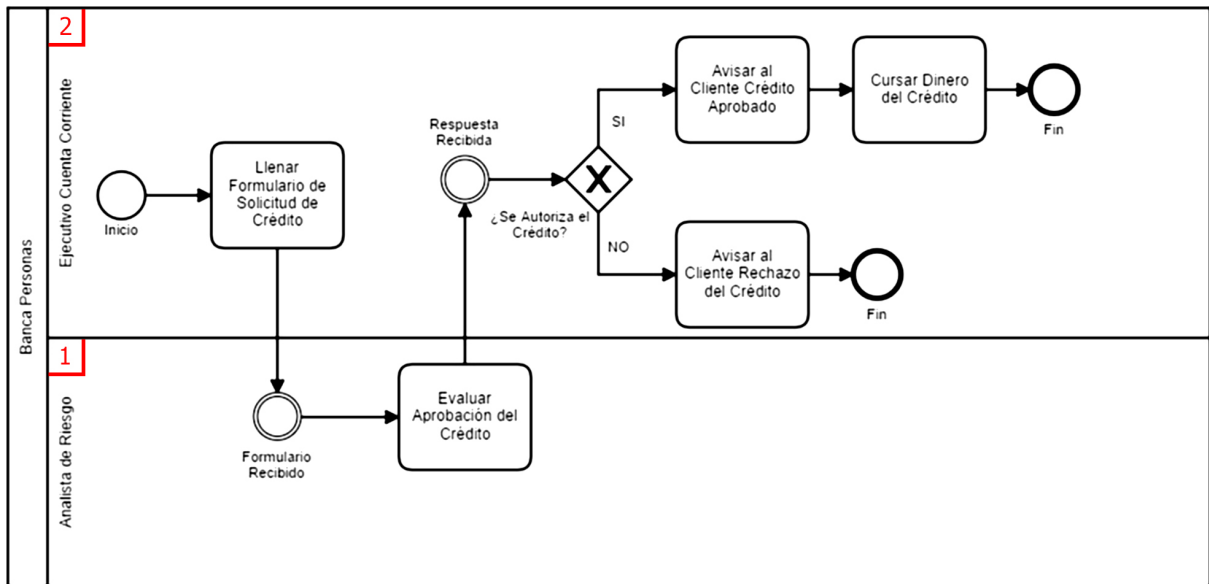


Figura 5.6: Propuesta gráfica capacidad de procesamiento

Fuente: Elaboración propia

5.2.5. Función Objetivo

Para esta mejora se propone agregar un rectángulo en la parte central superior del diagrama en la cual se indique con una imagen de punto al blanco la función objetivo y al lado el respectivo tipo de función objetivo.

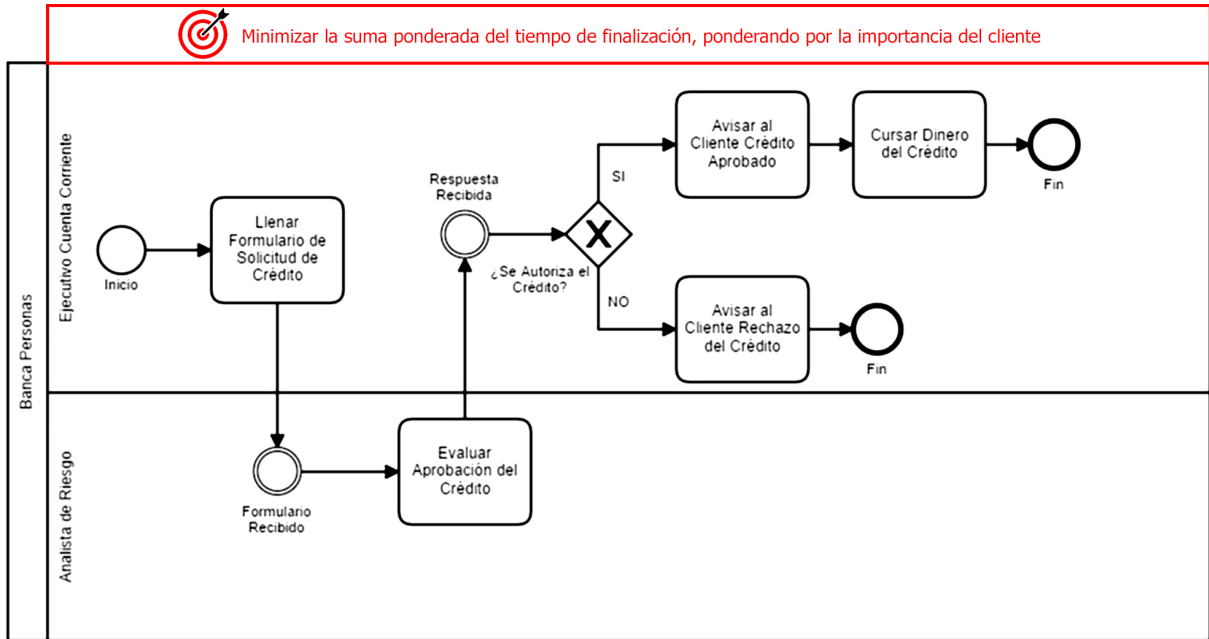


Figura 5.7: Propuesta gráfica función objetivo
Fuente: Elaboración propia

5.3. Modelo Práctico Numérico

La presente sección tiene como fin poder mostrar, por medio de dos ejemplos sencillos, la aplicabilidad de las propuestas gráficas presentadas anteriormente, además de la inclusión de la implementación del algoritmo para dar con la solución de estos problemas.

El primer problema es de secuenciación en un taller de flujo de dos máquinas (problema tipo *flow shop*) y el segundo es un problema de secuenciación de flujo también pero con tres máquinas (problema tipo *flow shop*). Ambos problemas son de secuenciamiento, es decir, se quiere determinar una secuencia óptima para realizar trabajos en un taller que tiene una determinada cantidad de máquinas, donde todos los trabajos siguen el mismo orden, es decir primero se ejecutan las labores en la primera máquina y luego en la segunda máquina. De esta forma se busca determinar el tiempo mínimo para completar los trabajos en el taller.

Ambos problemas se resuelven de forma sencilla en comparación con el resto de algoritmos que resuelven problemas de Scheduling, debido a que el ambiente de máquinas corresponde a un flow shop de dos máquinas o tres máquinas, dos de las configuraciones más simples.

Estos problemas se utilizaron a modo de ejemplo porque se resuelven con un algoritmo sencillo, el algoritmo de Johnson, el cual sigue los criterios generales de escoger el tiempo de menor de procesamiento entre todos los trabajos. Si este tiempo se encuentra en la máquina uno, el trabajo se programa al inicio. De lo contrario, si se encuentra en la máquina 2 se programa al final. Al programar el trabajo, se elimina de la lista y se repite el proceso para los trabajos sin programar hasta que estén todos asignados al orden final.

5.3.1. Problema I de tipo Flow Shop: Secuenciación en taller de flujo de dos máquinas

Enunciado: se desea fotocopiar unos apuntes y disponemos de dos máquinas, cada una de ellas la usamos para fotocopiar una cara de los apuntes. De esta manera fotocopiamos una cara del documento en la primera máquina y posteriormente se fotocopiará la otra cara en la segunda máquina. Se ha realizado un pedido de varios lotes de fotocopias que debe estar finalizado antes de 30 horas. A modo de ejemplo, para explicar la tabla, el trabajo uno toma 1 minuto de tiempo de procesamiento en la máquina fotocopidora 1 y 4 minutos en la máquina fotocopidora 2, y así sucesivamente con el resto de los trabajos.

	p_{i1}	p_{i2}
J_1	1	4
J_2	2	1
J_3	3	5
J_4	7	2
J_5	4	4
J_6	3	6
J_7	1	2
J_8	7	4

Tabla 5.1: Problema de dos máquinas

Objetivo del problema: minimizar la fecha de finalización del último trabajo.

Para dar mayor claridad al problema planteado repasaremos el concepto de flujo de trabajo o *flow shop*.

Un *flow shop* es una secuencia de máquinas en el taller, de tal manera que los trabajos pasan por las mismas máquinas en el mismo orden (todos los trabajos reciben las operaciones en el mismo orden) y todos los trabajos pasan por todas las máquinas .

Algoritmo de Johnson: este algoritmo permite resolver de manera sencilla el problema de secuenciación en dos máquinas de manera óptima, cuando se busca minimizar el tiempo de finalización del último trabajo. En palabras simples según (Rivera y De la Calle, 2011), este algoritmo consiste básicamente en ordenar los trabajos en una secuencia de dos maquinas, utilizando como criterio el tiempo de proceso de cada uno de los trabajos. Se elige en primer lugar aquel trabajo que tiene menos tiempo de procesamiento, bien sea en la máquina primera o en la máquina segunda. Si ese tiempo mínimo pertenece a la primera máquina lo ponemos al inicio de la secuencia y si pertenece a la segunda lo ponemos al final. Lo eliminamos de la lista de trabajos candidatos a elegir y repetimos con este proceso el trabajo de mínimo tiempo de procesamiento de la lista restante.

Paso 1: elegir aquel trabajo que tenga menor tiempo del procesamiento (tanto de la máquina A como la máquina B).

Paso 2: si ese tiempo pertenece a la primera máquina, el trabajo en cuestión se secuencia antes que los demás. Si por el contrario pertenece a la máquina B se secuencia detrás de todas las que resten por secuenciar.

Paso 3: se suprime el trabajo seleccionado (secuenciado) y se repite el proceso otra vez desde el paso 1 hasta que todos los trabajos estén secuenciados.

Aplicando el algoritmo de Johnson en nuestro problema

El mínimo tiempo de procesamiento es 1 para el J_1 y el J_7 en la máquina uno y para J_2 que pertenece en la máquina dos. Eso implica que podemos secuenciar el *job 7* y el *job 1* en primer lugar y el *job 2* en último lugar, además como 7 y 1 tiene el mismo de proceso en la maquina uno podemos secuenciarlos el 7 primero y luego el 1 o viceversa ambas secuencias son válidas. Luego, es necesario eliminar los *jobs* ya secuenciados de la lista.

	p_{i1}	p_{i2}
J_1	1	4
J_2	2	1
J_3	3	5
J_4	7	2
J_5	4	4
J_6	3	6
J_7	1	2
J_8	7	4

Tabla 5.2: Problema de dos máquinas, iteración 1

A continuación, el menor tiempo es 2 para J_4 en la máquina 2. El cual lo secuenciaremos en penúltimo lugar.

	p_{i1}	p_{i2}
J_1	1	4
J_2	2	1
J_3	3	5
J_4	7	2
J_5	4	4
J_6	3	6
J_7	1	2
J_8	7	4

Tabla 5.3: Problema de dos máquinas, iteración 2

El siguiente mínimo tiempo de procesamiento es 3 para J_3 y J_6 , ambos en la máquina 1, por lo que irían después del J_7 .

	p_{i1}	p_{i2}
J_1	1	4
J_2	2	1
J_3	3	5
J_4	7	2
J_5	4	4
J_6	3	6
J_7	1	2
J_8	7	4

Tabla 5.4: Problema de dos máquinas, iteración 3

De la lista restante, el mínimo tiempo de procesamiento es 4 para J_5 , tanto en la máquina 1 como en la máquina 2, y J_8 perteneciente a la máquina 2.

	p_{i1}	p_{i2}
J_1	1	4
J_2	2	1
J_3	3	5
J_4	7	2
J_5	4	4
J_6	3	6
J_7	1	2
J_8	7	4

Tabla 5.5: Problema de dos máquinas, iteración 4

Resultado: una de las secuencias óptimas es la 7-1-3-6-8-5-4-2, con un tiempo de finalización de 29 periodos.

5.3.2. Diagrama Problema I

Para una mejor comprensión del trabajo realizado en esta tesis es que se quiso diagramar este problema resuelto anteriormente en BPMN 2.0 y en BPMN 2.0 más las mejoras expuestas en este capítulo.

El diagrama del primer problema de flow shop con dos máquinas sería un contenedor con dos compartimentos, en los que cada compartimento representa una fotocopiadora. Dentro del contenedor deben estar las tareas a realizar que son fotocopiar el documento por una cara y luego por la otra. Tal como se muestra en la imagen a continuación:

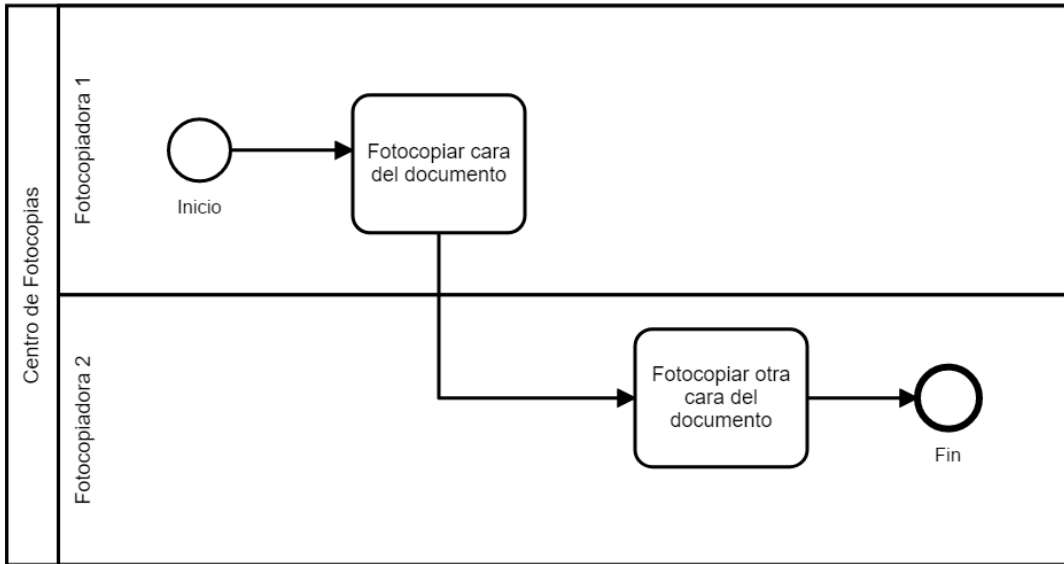


Figura 5.8: Diagrama BPMN problema I
Fuente: Elaboración propia

A continuación se presenta el diagrama del problema I incluyendo las propuestas de mejora explicadas en este capítulo.

5.3.3. Diagrama Problema I con Propuesta de Mejora

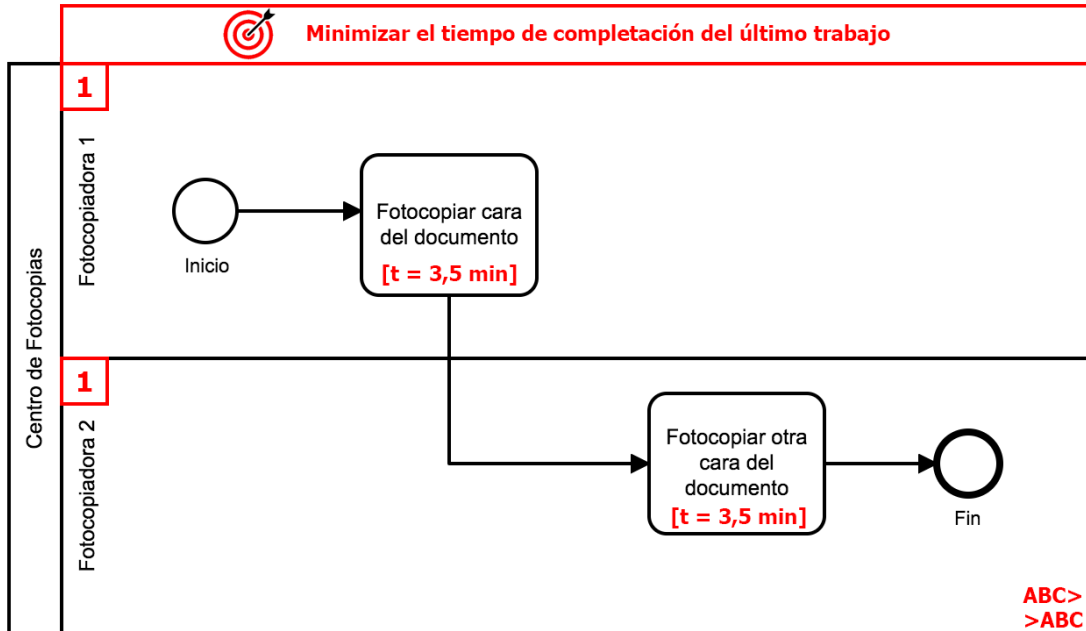


Figura 5.9: Diagrama BPMN problema 1 que incluye propuesta de Mejora
Fuente: Elaboración propia

5.3.4. Problema II de tipo Flow Shop: Secuenciación en taller de flujo de tres máquinas

Enunciado: se presenta un taller de fotocopias en el que se utilizan dos máquinas fotocopadoras y una encuadernadora. En la primera máquina fotocopadora se obtiene una cara de la fotocopia, en la segunda maquina fotocopadora la otra cara. Luego hay una tercera máquina, una encuadernadora. Esto permite procesar una serie de libros. Recordar que en un flow shop todos los trabajos pasan de manera secuencial a través de las máquinas.

Hay 8 trabajos a procesar (*jobs*). En la siguiente tabla se presentan los tiempos de procesamiento de éstos.

	p_{i1}	p_{i2}	p_{i3}
J_1	7	5	14
J_2	9	6	13
J_3	9	5	8
J_4	4	3	6
J_5	9	6	14
J_6	8	3	10
J_7	15	4	6
J_8	6	6	7

Tabla 5.6: Problema inicial de secuenciamiento de taller de flujo de tres máquinas

Existe una solución al problema de taller de flujo de tres máquinas usando algoritmo de Johnson si se cumple una de las tres condiciones siguientes:

- El menor de los tiempos de procesamiento en la máquina 1 (M1) es mayor o igual que el mayor de los tiempos de proceso en la máquina 2 (M2).
- El menor de los tiempos de procesamiento en la máquina 3 (M3) es mayor o igual que el mayor de los tiempos de procesamiento en M2.
- Cuando considerando por separado el problema M1-M2 y el problema M2-M3, la secuencia obtenida es la misma.

Primero, se debe verificar si se cumplen las condiciones del algoritmo de Johnson expuestas anteriormente, para ver si aplica resolverlo con este algoritmo.

Para ello es necesario ver los tiempos de procesamiento en la primera operación y seleccionar el menor. En este caso es 4, correspondiente al *job* 4 (J_4).

Segundo, se comprueba cuál es el tiempo de procesamiento más alto en la segunda operación, que se realiza en la máquina 2. Se tiene que dicho tiempo es 6, correspondiente al J_2 o al J_5 .

Finalmente, se revisa el menor tiempo de procesamiento en la tercera y última máquina que sería 6, correspondiente a J_4 .

Con estos datos, se puede ver que la primera condición no se cumple, porque el mínimo tiempo de procesamiento es 4 en la máquina 1 y no es mayor o igual que el tiempo de procesamiento del mayor de los tiempos de proceso en la maquina 2, que es 6.

Sin embargo, la segunda condición vemos que sí se cumple debido a que el menor de los tiempos de proceso en la maquina 3 es 6 y éste es igual al mayor de los tiempos de proceso en la máquina 2.

Al cumplirse una de las tres condiciones, se puede aplicar el algoritmo de Johnson, pero antes se debe convertir el problema a uno de dos máquinas.

Para ello se crean dos máquinas ficticias, sumando los tiempos de procesamiento de la primera y segunda máquina reales. Dicha suma de máquinas se denomina máquina A. Luego, se suman los tiempos de procesamiento de la segunda y tercera máquina del problema original y se crea una nueva máquina B.

Por ejemplo, para la máquina 1, los tiempos de procesamiento de la máquina A sería de 7 más 5 y para la maquina B sería de 7 más 14. Así se realizan los cálculos sucesivamente con cada trabajo, reflejando los resultados en la siguiente tabla.

	P_{iA}	P_{iB}
J_1	12	19
J_2	15	19
J_3	14	13
J_4	7	9
J_5	15	20
J_6	11	13
J_7	19	10
J_8	12	13

Tabla 5.7: Problema transformado a dos máquinas

La resolución del nuevo problema mediante el algoritmo de Johnson nos da una solución óptima para ambos problemas (problema de 2 máquinas y para el de 3).

Continuando con la utilización del algoritmo de Johnson y la nueva configuración de dos máquinas, se selecciona el menor tiempo de procesamiento de la lista, perteneciente al J_4 en la máquina A, por ende se pone en primer lugar.

Luego, se selecciona el siguiente *job* con el menor tiempo de procesamiento, correspondiente al J_7 en la máquina B. Por lo tanto, se coloca en último lugar. A continuación, se elige el tiempo de procesamiento del trabajo 6 en la máquina A que es 11 y se coloca en la segunda posición.

Después, se tienen como candidatos a J_8 y a J_1 , ya que ambos tienen tiempo de proceso 12 en la primera máquina.

Esto da origen a dos posibles secuencias totalmente válidas: 4–6–8–1 y 4–6–1–8. Repetimos este proceso para todos los trabajos hasta que los hayamos secuenciado todos, obteniendo una de las posibles secuencias óptimas, que sería 4–6–8–1–2–5–3–7. Esta secuencia es óptima tanto para el problema de dos máquinas como para de tres máquinas.

El algoritmo de Johnson permite resolver de manera sencilla el problema de secuenciación en tres máquinas de manera óptima cuando buscamos minimizar la fecha de finalización del último trabajo en algunos casos especiales de tres máquinas, teniendo en cuenta que se debe cumplir una de las tres condiciones expresadas anteriormente. Los problemas de flow shop con más de tres máquinas se pueden resolver, pero con algoritmos muchos más complejos.

El diagrama del segundo problema consistiría en un contenedor con tres compartimentos en los que cada compartimento representa a una máquina, el primero representaría la fotocopidora 1, el segundo la fotocopidora 2 y el tercero la encuadernadora. En el modelo deben existir tres actividades (operaciones), la primera es fotocopiar por una cara, la segunda fotocopiar la otra cara y la última tarea sería encuadernar. Tal como se muestra en la imagen a continuación.

5.3.5. Diagrama Problema II

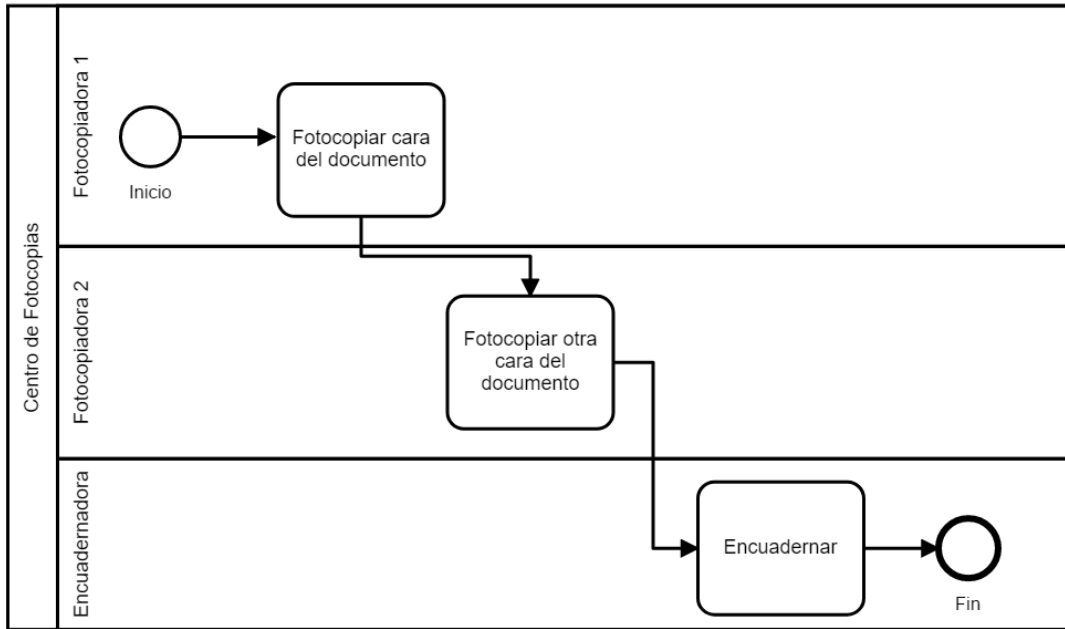


Figura 5.10: Diagrama BPMN problema II

Fuente: Elaboración propia

De esta misma manera, se puede diagramar del problema 2 incluyendo las propuestas de mejora explicadas en este capítulo.

5.3.6. Diagrama Problema II con Propuesta de Mejora

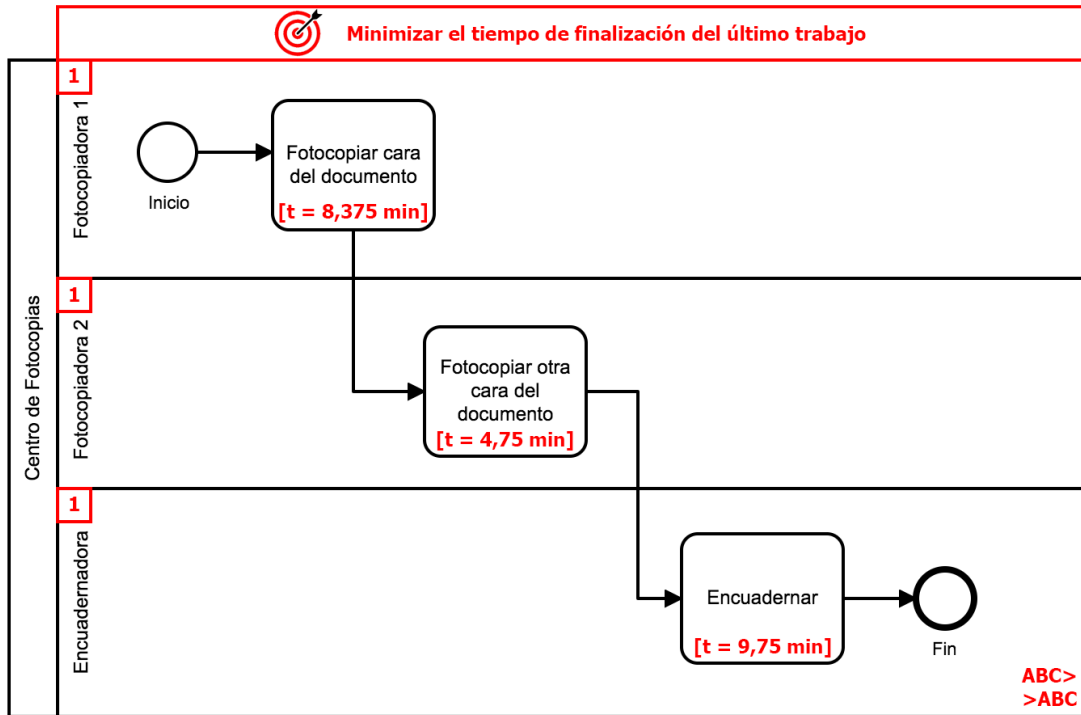


Figura 5.11: Diagrama BPMN problema II que incluye propuesta de Mejora
Fuente: Elaboración propia

A modo de resumen, dentro de las propuestas de mejoras que se incluyeron en los diagramas, se encuentran la función objetivo, la capacidad de cada máquina, la regla de prioridad existente en las colas de los trabajos y tiempo de procesamiento. El principal producto que se obtiene con esta propuesta de mejora a BPMN 2.0 es poder incorporar elementos que permiten representar más características de los procesos. Esto es un gran avance que amplía el alcance del estándar actual, convirtiéndolo en una herramienta más poderosa para la toma de decisiones.

Esto permite que toda la organización sepa para donde ir, pudiendo alinear, con mayor facilidad, los objetivos del proceso con los objetivos de la organización. Por ejemplo, al tener mayor claridad de las capacidades de las máquinas que desarrollan los distintos trabajos es más fácil saber cuánto personal contratar, cuántas máquinas comprar, con cuánto tiempo de holgura se cuenta para poder entregar un trabajo, etcétera. Además, teniendo a vista los tiempos de procesamiento y preparación, es posible detectar donde existen fallas y poner énfasis en esos errores, teniendo claridad cuáles son las tareas que se resuelven de manera eficiente y cuáles no.

5.4. Pasos a seguir para que la propuesta se incorpore al estándar oficial

- Realizar un documento con la propuesta y sus antecedentes

- Enviar este documento a la Object Management Group (OMG) el cual es un consorcio internacional de estándares de tecnología sin fines de lucro y membresía abierta, quien fue uno de los creadores de BPMN y quien actualmente lo representa.

- La OMG analizará y evaluará la propuesta y enviará su respuesta.

- De ser positiva la respuesta la propuesta será incluida en la próxima versión de BPMN.

Capítulo 6

Conclusión

Con el objetivo de encontrar las principales diferencias que existen entre BPMN y *Scheduling*, y las mejoras que a partir de ellas podrían realizarse a BPMN, se realizaron distintas actividades. Primero, se hizo una revisión sistemática de BPMN, entendiendo que es un estándar para el modelamiento de procesos de negocio, en el marco de una revolución en la forma de ver la organización y la manera en cómo mejorar. Se revisó cada elemento que lo compone, así como también algunas de las configuraciones más importantes. Adicionalmente se revisaron dos trabajos que proponen mejoras a BPMN, en el que se agregan elementos de rendimientos y confiabilidad, así como también una dimensión de temporalidad. Ambos trabajos se enfocan en implementar dichas mejoras en las *suites* de BPMN (BPMS), por lo que se alejan de lo que se busca realizar en este trabajo. Luego, se hizo una revisión de los principales elementos que describen un problema de *Scheduling*, empezando con los ambientes de máquinas que corresponden a la configuración de los recursos que llevan a cabo el procesamiento de las tareas (o *jobs*). También se revisaron algunas de las restricciones más comunes, asociadas a tiempos, capacidades, etcétera. Finalmente, se describieron algunas de las funciones objetivo que se persiguen en este tipo de problemas.

Con todos estos antecedentes, se realizó una comparación de ambas corrientes de representación y análisis de procesos, buscando encontrar los elementos comunes. En dicha comparación se llegó a una lista de elementos de *Scheduling* que podrían representarse con algunos de BPMN, así como también los elementos que no. Dentro de las características de *Scheduling* que se pueden representar con BPMN están algunos de los ambientes de máquina como: una máquina, máquinas idénticas en paralelo, flow shop, flow shop flexible, job shop, job shop flexible y open shop. También se encontró que era posible representar algunas de las restricciones como interrupciones, precedencia, procesamiento por lotes, averías, de elegibilidad de máquinas, bloqueo y recirculación.

De los elementos de *Scheduling* para los cuales no se encontró un símil en BPMN, respecto a los ambientes de máquina están las máquinas paralelas con diferentes velocidades, máquinas sin relacionar en paralelo. Esto se debe, principalmente, a que en BPMN no existe el concepto velocidad o tiempo de procesamiento, así como tampoco el de capacidad. Respecto a las restricciones del problema de *Scheduling*, no se puede representar la fecha de llegada, el tiempo de preparación dependiente de la secuencia, las familias de tareas, la permutación y la no espera. Nuevamente, queda en evidencia a partir de estas diferencias, que en BPMN el tiempo no se puede representar de manera simple. Finalmente, se encontró

que BPMN no considera los objetivos de la organización dentro de su estándar, por lo que no es posible representar una función objetivo.

Gracias a este análisis previo se pudo identificar los elementos de *Scheduling* que permiten aumentar la capacidad descriptiva de BPMN, además de agregarle elementos operativos que, entre otras cosas, logran caracterizar mejor el proceso y facilitan la evaluación de éste. Entre las características que se propone agregar a BPMN está la capacidad de procesamiento de los carriles, así como también el tiempo que tomaría realizar cada una de las actividades. En esta misma línea, también se propuso incorporar el tiempo de preparación dependiente de las secuencias. Estos elementos entregan una visión operativa del proceso, pudiendo calcular, por ejemplo, el tiempo total de proceso. También se decidió incorporar los objetivos de los procesos, así como también las reglas de ordenamiento en las colas de las actividades.

El tiempo de procesamiento nos permite saber lo que tarda en ejecutarse cada tarea del proceso, por lo que nos permite ver de manera gráfica cuáles son las tareas que se demoran más, cuales menos, además del tiempo de procesamiento total del proceso de negocio. Esto nos puede ayudar a entender mejor los procesos, pudiendo detectar cuáles son aquellas actividades que requieren de mayor cantidad de recursos.

El tiempo de preparación permitiría poder visualizar cuánto tiempo tardará en prepararse una máquina para iniciar su funcionamiento, también llamado tiempo de configuración.

Para la propuesta de las reglas de prioridad, se presentan tres símbolos. Cada uno representa una regla de ordenamiento de procesamiento en las colas. La primera es la regla de prioridad FIFO, que indica que el primero en entrar al sistema el primero en salir. La segunda regla de prioridad es LIFO, que indica que el último en entrar será el primero en salir. La última corresponde a la regla de prioridad, que ordena los trabajos de acuerdo con su importancia relativa.

La capacidad de procesamiento nos permite ver cuántas máquinas están disponibles para ejecutar las tareas, pudiendo calcular con mayor exactitud el tiempo de procesamiento de un proceso, debido a que es muy distinto si un proceso de negocio lo realiza una sola maquina a que tres, por ejemplo.

La última mejora que se propone es incluir la función objetivo que tenga la empresa dentro para el proceso. De esta forma toda la organización sabe para dónde ir y alinea el objetivo del proceso al objetivo de la organización. Así, se fomenta la coordinación entre los trabajadores, evitando perder el horizonte de la empresa y se evita el uso inadecuado de los procesos y recursos, logrando un alineamiento entre las distintas entidades de la organización. Toda organización es un conjunto de etapas y procesos en la que una depende de la otra, por ende, cualquier inconsistencia puede afectar el producto final.

Los objetivos claros proveen una clara idea de las metas para los dueños y los empleados. Los objetivos que se pueden medir permiten que a las organizaciones puedan revisar sus metas, pudiendo realizar los ajustes necesarios que las lleven a cumplirlas.

Las mejoras que se incluyeron sirven, principalmente, para poder llevar un control de la gestión a toda escala. Es decir, entrega una herramienta que permite a la organización tomar decisiones de manera rápida y certera con elementos que contribuya discernir ante distintas opciones y distintos problemas. Otra característica común a estas mejoras propuestas es que, al incorporarlas, la propiedad de estándar de BPMN no se perdería, ya que son elementos comunes a todos los procesos operativos. En todos los procesos existe un objetivo, una manera de priorizar entre las tareas a realizar, capacidades y tiempos de procesamiento, así como también un tiempo de preparación (o configuración) para poder realizarlos.

Así mismo, se decidió no incorporar los demás elementos de *Scheduling* que no existen en BPMN, ya que son muy específicos a ciertos tipo de problemas, por lo que no servirían para representar todo tipo de procesos y harían más compleja la representación.

Adicionalmente, considerando las mejoras señaladas anteriormente, se entrega una representación gráfica de cada una de ellas. A partir de ésta, se muestran dos ejemplos de problemas de *Scheduling* representados por este BPMN mejorado.

Finalmente, esta tesis puede ser de los primeros intentos que reduzca la brecha existente entre estas dos líneas de conocimiento. Para futuras investigaciones se propone implementar un software que integre el modelamiento de procesos con características de *Scheduling* logrando tener una herramienta mucho más robusta y con mayor poder llegando hasta la automatización. Este software permitiría entregar una solución rápida y sencilla, con información clave de los procesos ante las distintas necesidades que puede tener una empresa al momento de analizar sus procesos de negocio y de implementar mejoras.

Referencias

- Blazewicz, J., Dror, M., y Weglarz, J. (1991). Mathematical programming formulations for machine scheduling: A survey. *European Journal of Operational Research*, 51(3), 283 - 300. Descargado de <http://www.sciencedirect.com/science/article/pii/037722179190304E> doi: [https://doi.org/10.1016/0377-2217\(91\)90304-E](https://doi.org/10.1016/0377-2217(91)90304-E)
- Bocciarelli, P., y D'Ambrogio, A. (2011). A BPMN extension for modeling non functional properties of business processes. En *Proceedings of the 2011 symposium on theory of modeling & simulation: Devs integrative m&s symposium* (pp. 160–168).
- Braun, R., y Esswein, W. (2014). Classification of domain-specific bpmn extensions. En U. Frank, P. Loucopoulos, Ó. Pastor, y I. Petrounias (Eds.), *The practice of enterprise modeling* (pp. 42–57). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Cheikhrouhou, S., Kallel, S., Guermouche, N., y Jmaiel, M. (2013). Toward a time-centric modeling of business processes in BPMN 2.0. En *Proceedings of international conference on information integration and web-based applications & services* (p. 154).
- Chinosi, M., y Trombetta, A. (2012). Bpmn: An introduction to the standard. *Computer Standards & Interfaces*, 34(1), 124–134.
- Davenport, T. H., Short, J. E., y cols. (1990). The new industrial engineering: information technology and business process redesign.
- Fayol, H. (1971). *Administración industrial y general: previsión, organización, mando, coordinación, control*. Editorial Universitaria. Descargado de <https://books.google.cl/books?id=Rzo-AAAAYAAJ>
- Freund, J., Rücker, B., y Hitpass, B. (2014). *BPMN 2.0: manual de referencia y guía práctica*. BPM Center.
- Friedenstab, J.-P., Janiesch, C., Matznerand, M., y Müller, O. (2012, 01). Extending bpmn for business activity monitoring. *Proceedings of the Annual Hawaii International Conference on System Sciences*, 4158-4167. doi: 10.1109/HICSS.2012.276
- Gagne, D., y Trudel, A. (2009). Time-bpmn. En *Proceedings of the 2009 ieee conference on commerce and enterprise computing* (pp. 361–367). Washington, DC, USA: IEEE Computer Society. Descargado de <https://doi.org/10.1109/CEC.2009.71> doi: 10.1109/CEC.2009.71
- Jeston, J., y Nelis, J. (2014). *Business Process Management*. Taylor & Francis. Descargado de <https://books.google.cl/books?id=8Q6pAgAAQBAJ>
- Karp, R. M. (1972). Reducibility among combinatorial problems. En R. E. Miller, J. W. Thatcher, y J. D. Bohlinger (Eds.), *Complexity of computer computations: Proceedings of a symposium on the complexity of computer computations, held march 20–22, 1972, at the ibm thomas j. watson research center, yorktown heights, new york, and sponsored by the office of naval research, mathematics program, ibm world trade corporation, and the ibm research mathematical sciences department* (pp. 85–103). Boston, MA: Springer US. Descargado de https://doi.org/10.1007/978-1-4684-2001-2_9 doi: 10.1007/978-1-4684-2001-2_9
- León León, O. M., y Asato España, J. A. (2009). La importancia del modelado de procesos de negocio como herramienta para la mejora e innovación. *Revista Raites*, 4(7), 61-72. Descargado de <http://itcelaya.edu.mx/ojs/index.php/raites/article/view/7/5>
- Luza M, C. (2009). *Técnicas de modelamiento sesión 1: El modelado en el desarrollo de software*. Descargado 2018-01-15, de <https://es.slideshare.net/juliopari/tm01-el-modelado-en-el-desarrollo-de-software>
- Martin, F., y Scott, K. (1999). Uml gota a gota. *Ed. Pearson-AddisonWeasley, Massachussets, Año.*

- Montero, Ildefonso, García-Bañuelos, Luciano, y Dumas, Marlon. (2010). *Póster BPMN 2.0 – Business Process Model and Notation*. Descargado 2018-01-15, de http://www.bpmb.de/images/BPMN2_0_Poster_ES.pdf
- Panagacos, T. (2012). *The ultimate guide to Business Process Management: Everything you need to know and how to apply it to your organization*. Createspace Independent Pub. Descargado de <https://books.google.cl/books?id=AyCQMQEACAAJ>
- Pinedo, M. (2016). *Scheduling: Theory, algorithms, and systems*. Springer International Publishing. Descargado de <https://books.google.cl/books?id=M4-RCwAAQBAJ>
- Rivera, J., y De la Calle, J. (2011). *Algoritmos constructivos, de búsqueda aleatoria, de búsqueda local, genéticos y genéticos híbridos para la solución de problemas de lot streaming en ambiente flow shop con makespan como función objetivo* (Tesis Doctoral no publicada).
- Rodríguez, A., Fernández-Medina, E., Piattini, M., y Trujillo, J. (2008). Secure business processes defined through a uml 2.0 extension. *IEEE Latin America Transactions*, 6(4), 339–346.
- Sepúlveda Jaramillo, Hermes. (s.f.). *BPM se está posicionando en el mundo como el modelo de gestión organizacional por excelencia*. Descargado 2018-01-15, de <http://www.club-bpm.com/Noticias/art00112.htm>
- Simpson, R. C. (2005). An XML representation for crew procedures.
- Smith, W. E. (1956). Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, 3(1-2), 59–66.
- Taylor, F., Fayol, H., y Lender, H. (2003). *Principios de la administración científica*. Edigrama. Descargado de <https://books.google.cl/books?id=YR41twAACAAJ>
- T'kindt, V., y Billaut, J.-C. (2006). *Multicriteria scheduling: theory, models and algorithms*. Springer Science & Business Media.
- White, S. A. (2004). Introduction to BPMN. 2004. *IBM Corporation*.

Anexo A

Resumen de BPMN 2.0.

En las siguientes tablas se hace un resumen de lo presentado en el capítulo anterior respecto al estándar BPMN 2.0.

Tabla A.1: Objetos de Flujo en BPMN 2.0.

Objetos de Flujo	Primera categoría de los elementos gráficos de BPMN 2.0. Corresponde a los tres elementos principales de un diagrama.
Actividades	Representan el trabajo realizado dentro de una organización y consumen recursos.
Eventos	Representan algo que ocurre o que puede ocurrir dentro del transcurso de un proceso de negocio.
Compuertas	Elementos que controlan los puntos de convergencia y divergencia.

Fuente: Elaboración propia, basado en (Montero, Ildefonso y cols., 2010).

Tabla A.2: Actividades en BPMN 2.0.

Actividades	Representan el trabajo realizado dentro de una organización y consumen recursos.
Tarea	Es una unidad de trabajo, el trabajo a realizar.
Transacción	Es un conjunto de actividades relacionadas lógicamente, adhiriéndose a un protocolo transaccional particular.
Subproceso de eventos	Se sitúa en el interior de otro (sub)proceso. Este se activa en la ocurrencia del evento de inicio especificado y mientras el proceso que lo contiene permanezca también activo. El subproceso de evento puede interrumpir o no al proceso que lo contiene.
Actividad de llamada	Es una referencia a un subproceso o tarea definido de forma global, que se reutiliza en el proceso actual.

Fuente: Elaboración propia, basado en (Montero, Ildefonso y cols., 2010).

Tabla A.3: Eventos en BPMN 2.0.

Eventos	Representan algo que ocurre o que puede ocurrir dentro del transcurso de un proceso de negocio.
Simple	Eventos sin especificar. Indican puntos de inicio, de fin y situaciones intermedias.
Mensaje	Recepción y envío de mensajes.
Temporal	Puntos en el tiempo, lapsos, límites (timeouts). Pueden ser eventos únicos o cíclicos.
Escalable	Cambio a un nivel más alto de responsabilidad.
Condicional	Reacción a cambios en las condiciones de negocios o integración de reglas de negocio.
Enlace	Conectores fuera de página. Dos conectores de enlace equivalen a un flujo de secuencia.
Error	Captura y lanzamiento de errores conocidos con nombre.
Cancelación	Reacción a la cancelación de una transacción/solicitud de cancelación.
Compensación	Manejo/solicitud de compensación.
Señal	Intercambio de señales entre procesos. Una señal puede ser capturada varias veces
Múltiple	Captura uno de un conjunto de eventos. Lanza todos los eventos definidos.
Paralela múltiple	Captura todos los eventos de un conjunto de eventos en paralelo.
Terminación	Terminación inmediata del proceso.

Fuente: Elaboración propia, basado en (Montero, Ildefonso y cols., 2010).

Tabla A.4: Compuertas en BPMN 2.0.

Compuertas	Elementos que controlan los puntos de convergencia y divergencia.
Exclusiva	En un punto de bifurcación, selecciona exactamente un flujo de secuencia de entre las alternativas existentes. En un punto de convergencia, la compuerta espera a que un flujo incidente complete para activar el flujo saliente.
Basada en eventos	Esta compuerta siempre será seguida por eventos o tareas de recepción, y sólo activará un flujo saliente dependiendo del evento que ocurra en primer lugar.
Paralela	En un punto de bifurcación, todos los caminos salientes serán activados simultáneamente. En un punto de convergencia, la compuerta espera a que todos los flujos incidentes completen antes de activar el flujo saliente.
Inclusiva	En un punto de bifurcación, al menos un flujo es activado. En un punto de convergencia, espera a todos los flujos que fueron activados para activar el saliente.
Compleja	Comportamiento complejo de convergencia o bifurcación no capturado por el resto de compuertas.
Exclusiva Basada en Eventos (generadora de instancias)	En la ocurrencia de uno de los eventos subsecuentes se crea una nueva instancia del proceso.
Paralela Basada en Eventos (generadora de instancias)	En la ocurrencia de todos los eventos subsecuentes se crea una nueva instancia del proceso.

Fuente: Elaboración propia, basado en (Montero, Ildefonso y cols., 2010).

Tabla A.5: Objetos de conexión en BPMN 2.0.

Objetos de conexión	Segunda categoría de los elementos gráficos de BPMN 2.0, ensamblan a los objetos de flujo.
Flujo de secuencia	Indica en qué orden deben llevarse a cabo las actividades y conecta los objetos de flujo.
Flujo de mensaje	Indica mensajes entre dos participantes de un proceso.
Asociación	Indica relaciones entre los objetos de flujo (tareas, eventos y compuertas) y los artefactos.

Fuente: Elaboración propia, basado en (Montero, Ildefonso y cols., 2010).

Tabla A.6: Carreles en BPMN 2.0.

Carriles	Tercera categoría de los elementos gráficos de BPMN 2.0 y organiza visualmente las actividades.
Compartimento (carril)	Forma de representar los participantes en los modelos de proceso de negocio, por lo general son departamentos o roles de algún área.
Contenedor (<i>pool</i> o piscina)	Los compartimentos siempre se encuentran dentro de un contenedor, el cual representa al mismo tiempo los límites de un proceso de principio a fin, pero además representa una instancia visible que dirige y controla el flujo de actividades en el proceso, tomando el pool un rol de “dirigente”.

Fuente: Elaboración propia, basado en (Montero, Ildefonso y cols., 2010).

Tabla A.7: Artefactos en BPMN 2.0.

Artefactos	Cuarta categoría de los elementos gráficos de BPMN 2.0, permiten tener más información del modelo.
Grupo	Organizan las tareas que tienen importancia en el proceso global agrupando las diferentes actividades.
Anotación	Proporciona información de texto adicional al modelo para que el lector entienda mejor este.
Objetos de datos	<p>Representan los datos ubicados en el proceso (datos resultantes, almacenados o recopilados). Existen varios tipos de objetos de datos los cuales se detallan a continuación:</p> <ul style="list-style-type: none"> • Dato de Entrada o Input: Es una entrada externa a todo el proceso. Puede ser leído por una actividad. • Dato de Salida u Output: Es una variable disponible como resultado del proceso. • Dato de tipo objeto: Representa información que fluye a través del proceso tales como documentos, correos electrónicos o cartas • Colección de Objetos de Datos: Representa una colección de información, p.e una lista de artículos. • Almacén: Es un lugar donde el proceso puede leer o escribir datos. La información en un almacén persiste más allá de la vida de la instancia del proceso. • Mensaje: Es utilizado para representar el contenido de una comunicación entre dos participantes.

Fuente: Elaboración propia, basado en (Montero, Ildefonso y cols., 2010).

Anexo B

Resumen de Scheduling

En las siguientes tablas se presenta un resumen de lo presentado en el capítulo anterior, respecto a *Scheduling*. Para facilitar el entendimiento de lo mostrado, se separan por los distintos campos (α , β y γ).

Tabla B.1: Conceptos de *Scheduling*

Conceptos	Descripción
Máquina	Es el recurso que lleva a cabo las acciones específicas y necesarias para completar un <i>job</i> . Dichas acciones, en conjunto, se denominan procesamiento.
Operación	Es la entidad a programar (asignar). Esto quiere decir que la operación es la unidad lógica de trabajo que es llevado a cabo por una máquina, por lo que el problema de <i>Scheduling</i> intenta resolver la asignación óptima de las operaciones a las máquinas.
<i>Job</i> (trabajo)	Serie de operaciones. Un <i>job</i> puede tener una sola operación o varias.

Fuente: Elaboración propia, adaptado de (T'kindt y Billaut, 2006) y (Pinedo, 2016).

Tabla B.2: Campo α : Ambiente de máquinas en *Scheduling*

Campo α	Representación del ambiente de máquinas
Una Máquina	Es el ambiente más simple. Consiste en una máquina que procesa todos los <i>jobs</i> del sistema.
Máquinas idénticas en paralelo	Un número de máquinas con las mismas características, procesan los distintos <i>jobs</i> en paralelo. Un <i>job</i> solo pasa por una máquina, dado que solo tiene una operación.
Máquinas en paralelo con diferentes velocidades	Un número de máquinas en paralelo, tienen diferentes velocidades de procesamiento.
Máquinas sin relacionar en paralelo	Un número de máquinas en paralelo, procesan los distintos <i>jobs</i> . Las máquinas tienen una velocidad de procesamiento diferente para cada <i>job</i> .
Flow shop	Existe un número determinado de máquinas conectadas en serie, por las que deben pasar todos los <i>jobs</i> .
Flow shop flexible	Existen etapas, que son máquinas idénticas en paralelo, conectadas en serie, por las que cada uno de los <i>jobs</i> debe pasar.
Job shop	Consiste en un número determinado de máquinas, por las que cada <i>job</i> tiene una ruta predeterminada de máquinas a seguir. Además, puede darse el caso de que un <i>job</i> no pase por todas las máquinas.
Job shop flexible	A diferencia de un <i>job shop</i> , existen centros de trabajo, que corresponden a máquinas idénticas en paralelo. Los <i>jobs</i> tienen una ruta predeterminada a seguir por dichos centros de trabajo, sin la necesidad de pasar por cada uno de ellos.
Open shop	Consiste en una cantidad determinada de máquinas, por las que cada <i>job</i> tiene que pasar por lo menos una vez. Se diferencia con el <i>job shop</i> con que no hay un camino predeterminado de los <i>jobs</i> a través de las máquinas.

Fuente: Elaboración propia, basado en (Pinedo, 2016).

Tabla B.3: Campo β : restricciones en el problema de Scheduling [1/2]

Campo β	Representa las restricciones del problema de Scheduling
Fecha de llegada	Indica la fecha de llegada de un <i>job</i> al sistema. Esto implica que el procesamiento de un <i>job</i> debe realizarse en un instante igual o posterior a la llegada.
Interrupciones	Consiste en que, una vez comenzado el procesamiento de un <i>job</i> en una máquina, puede ser detenido, sin que se pierda el procesamiento ya realizado. Es decir, se permite el procesamiento parcial de un <i>job</i> , detenerlo y continuarlo en un futuro, sin pérdidas.
Restricciones de precedencia	Existen <i>jobs</i> que requieren que se hayan procesado uno o más <i>jobs</i> específicos previamente, para poder llevar a cabo su propio procesamiento.
Tiempo de preparación dependiente de la secuencia	Es el tiempo que se requiere para configurar la máquina entre un <i>job</i> y otro. Depende del par de <i>jobs</i> que están en secuencia, pudiendo también depender de la máquina en la que son procesados. Si no está presente, se asume que el tiempo de preparación es cero entre todos los pares de <i>jobs</i> .
Familia de tareas	Existe un determinado número de familias de <i>jobs</i> y todos los <i>jobs</i> pertenecen a alguna familia. En este caso, el tiempo de preparación depende de las familias de tareas a las que pertenezcan dos <i>jobs</i> consecutivos. Si son de la misma familia, el tiempo de configuración es cero.
Procesamiento por lotes	Consiste en que las máquinas están habilitadas a procesar más de un <i>job</i> al mismo tiempo y lo realizan en paralelo. Tienen una capacidad o tamaño de lote, que puede ser ilimitado. El tiempo de procesamiento del lote está determinado por el mayor tiempo de procesamiento entre los <i>jobs</i> .
Averías	Consiste en que algunas máquinas no están continuamente disponibles, por lo que en algunos periodos no se pueden utilizar.
Restricciones de elegibilidad de máquinas	Esta restricción se puede dar en un ambiente de máquinas en paralelo. Consiste en que hay <i>jobs</i> que solo pueden ser procesadas por un subconjunto de las máquinas que están en paralelo.
Permutación	Implica que en un ambiente flow shop, los <i>jobs</i> se procesan según la disciplina FIFO (primero en entrar primero en salir). Por lo tanto, los <i>jobs</i> mantienen el orden (o permutación) que tienen en la primera máquina hasta que salen del sistema.

Fuente: Elaboración propia, basado en (Pinedo, 2016).

Tabla B.4: Campo β : restricciones en el problema de Scheduling [2/2]

Campo β	Representa las restricciones del problema de Scheduling
Bloqueo	Puede ocurrir en los <i>flow shop</i> . Consiste en que algunas máquinas tienen un largo limitado para la cola de espera por procesar, por lo que se realiza un bloqueo en la máquina anterior, de modo que, aunque se haya procesado el <i>job</i> que tenga dentro, no la puede liberar hasta que se haga un espacio en la cola de espera de la máquina siguiente.
No espera	Esta es otra restricción que puede estar presente en los <i>flow shop</i> . Implica que un <i>job</i> no puede esperar entre una máquina y otra para ser procesada, por lo que solo se procesa solo cuando se pueda realizar de manera continua, sin interrupciones entre una máquina y otra.
Recirculación	Implica que un <i>job</i> puede pasar por una máquina o una estación de trabajo, más de una vez. Esto se da en <i>job shops</i> o <i>job shops</i> flexibles.

Fuente: Elaboración propia, basado en (Pinedo, 2016).

Tabla B.5: Campo γ : función objetivo del problema de Scheduling

Campo γ	Representa la función objetivo del problema de Scheduling
Makespan	Consiste en minimizar el tiempo de finalización del último trabajo que abandona el sistema.
Máximo atraso	Implica minimizar el máximo atraso, es decir, la peor infracción a las fechas de entrega.
Tiempo de finalización total ponderado	Consiste en minimizar la suma ponderada del tiempo de finalización, ponderando por la importancia del trabajo.
Tiempo de finalización total ponderado descontado	Tiene un sentido similar a la función objetivo anterior, pero se utiliza una función de decaimiento exponencial, dependiente del tiempo de finalización y de una tasa de descuento dada.
Tardanza total ponderada	Consiste en minimizar la tardanza total ponderada, por lo que se busca disminuir la tardanza de los trabajos más importantes primero.
Número ponderado de tareas tardías	Implica minimizar la cantidad de trabajos tardíos, ponderando por la importancia, por lo que se busca que los trabajos más importantes no tengan atrasos.

Fuente: Elaboración propia, basado en (Pinedo, 2016).

Anexo C

Tablas Comparativas entre Scheduling y BPMN 2.0 con Ilustraciones

Elemento de Scheduling		BPMN 2.0	Ilustración
Máquina	•→	Compartimento	
Job	•→	Instancia	No es visible
Operación	•→	Actividad	

Figura C.1: Comparativa ¿Qué elementos de Scheduling están en BPMN 2.0 ?

Fuente: Elaboración Propia

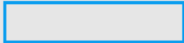
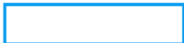
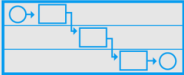
Elemento de Scheduling	BPMN 2.0	Ilustración
Una máquina	• → Compartimento	
Maquinas identicas en paralelo	• → Compartimento	
Maquinas paralelas con diferentes velocidades	• → No existe	No es posible ilustrar
Maquinas sin relacionar en paralelo	• → No existe	No es posible ilustrar
Flow Show	• → Esquemas de varios compartimentos que sólo tengan actividades	



Figura C.2: Comparación entre ambiente alfa de Scheduling y BPMN 2.0
Fuente: Elaboración Propia

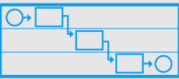
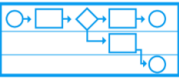

Elemento de Scheduling	BPMN 2.0	Ilustración
Flow shop flexible	Esquemas de varios compartimentos que sólo tengan actividades	
Job shop	Varios compartimentos que contengan un flujo de compuertas exclusivas	
Job shop flexible	Varios compartimentos que contengan un flujo de compuertas exclusivas	
Open shop	Proceso ad hoc	No es abordado en esta tesis

Figura C.3: Continuación comparativa entre ambiente alfa de Scheduling y BPMN 2.0
Fuente: Elaboración Propia



Elemento de Scheduling		BPMN 2.0	Ilustración
Fecha de llegada	• →	No existe	No es posible ilustrar
Interrupciones	• →	Es posible representar de manera parcial, con el evento intermedio adjunto interruptor	
Restricciones de precedencia	• →	Diagrama común con un compartimento o con un evento de inicio condicional en un diagrama	
Tiempo de preparación dependiente de la secuencia	• →	No existe	No es posible ilustrar
Familia de tareas	• →	No existe	No es posible ilustrar



Figura C.4: Comparación entre ambiente beta de Scheduling y BPMN 2.0
Fuente: Elaboración Propia






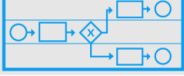





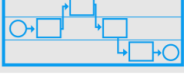
Elemento de Scheduling	BPMN 2.0	Ilustración
Procesamiento por lotes	 Un compartimento que tenga un flujo con actividades que tengan el marcador de instancias múltiples en paralelo	
Averías	 Evento error	
Restricción de elegibilidad de máquinas	 Un flujo de procesos con compuertas exclusivas.	
Permutación	 No Existe	No es posible ilustrar
Bloqueo	 Evento señal	
No espera	 No existe	No es posible ilustrar
Recirculación	 Un flujo de procesos en el cual se vuelva a utilizar uno o más compartimentos	

Figura C.5: Continuación comparativa entre ambiente beta de Scheduling y BPMN 2.0
Fuente: Elaboración Propia