

**DISEÑO UN ALGORITMO DE PROGRAMACIÓN DE
OPERACIONES PARA SISTEMAS DE FABRICACIÓN Y ENSAMBLE**

**DISEÑO UN ALGORITMO DE PROGRAMACIÓN DE
OPERACIONES PARA SISTEMAS DE FABRICIÓN Y ENSAMBLE**

LUIS EDUARDO RAMIREZ POLO

**Trabajo de grado presentado como requisito parcial para optar al título de
Magister en Ingeniería Industrial**

**Director: Yezid Donoso M.
Ingeniero de Sistemas. PhD.**

**UNIVERSIDAD DEL NORTE
PROGRAMA DE INGENIERIA INDUSTRIAL
BARRANQUILLA
2008**

Aprobado por el profesorado de la División de Ingeniería en cumplimiento de los requisitos exigidos para otorgar el título de Maestría en Ingeniería Industrial.

Ing. Yezid Donoso M . Ph.D.

Director del Proyecto

Ing. Carlos D. Paternina Ph.D.

Director del programa de ingeniería industrial

Codirector del proyecto

Ing. Angel L. Gonzalez Ph.D.

Coordinador de la maestría

Lic. Vicente Valls Verdejo Ph.d

Corrector del Proyecto

Ing. Agustín Barrios Sarmiento Ph.d

Corrector del Proyecto

AGRADECIMIENTOS

Este trabajo no lo puede haber realizado sin la guía, comprensión y ayuda de muchas personas que además de apoyarme a realizarlo me han hecho la persona que soy ahora, con mucho cariño les dedico este trabajo.

Principalmente le agradezco a Dios, por haberme dado la posibilidad de Vivir y además, la vida que he vivido, a él, que de todos los lugares del mundo me puso a vivir aquí, que me dio la oportunidad de tener una familia, que además es excelente, tener unos amigos, que son inigualables y tener unos compañeros, que son estupendos, a él, que me abrió las puertas para llegar hasta donde he llegado, infinitas gracias. A mi madre Alba Cristina Polo, mi padre Jorge Eliécer Ramírez y mi hermano Jorge Alberto Ramírez, que siempre me han brindado su amor, cariño y apoyo incondicional, ellos que me enseñaron que la humildad, la perseverancia y el esfuerzo son el camino para lograr objetivos. A mi abuela Carmen Rodríguez, mis familiares y mis amigos, porque siempre me han brindado su apoyo cuando lo necesito y me han hecho feliz, si los citó a cada uno no me alcanzan las páginas.

Quisiera darles especiales gracias a los ingenieros; Yezid Donozo por confiar en mí para este trabajo, infinitas gracias a Carlos Paternina de igual forma, por apoyarme en este proyecto y brindarme su confianza a lo largo de mi vida profesional. Francesco Narducci quien fue uno de los principales causantes de que esta sea mi tesis y me apoyo de gran manera en este trabajo, muchas gracias a Iván Saavedra y su equipo quien ayudo en la codificación del modelo. A los licenciados Vicente Valls y Agustín Barrios quienes desde su puesto de jueces me ayudaron a darle el toque final a este proyecto. A todos los profesores del departamento de ingeniería industrial por haberme ayudado a ser la persona que soy como ingeniero y ahora como Magíster. Gracias al excelente personal de UNINORTE quienes hacen de esta una gran universidad y un buen ambiente para las personas que trabajan y estudian en ella. A estas y a las personas que me faltan por nombrar que quiero mucho, les doy las gracias por hacerme tan feliz.

TABLA DE CONTENIDO

CAPITULO 1	1
PLANTEAMIENTO DE LA INVESTIGACIÓN	1
1.1 FORMULACIÓN DEL PROBLEMA	1
1.1.1 ANTECEDENTES	1
1.1.2 ARBOL CAUSA – EFECTO	3
1.1.3 PLANTEAMIENTO DEL PROBLEMA	4
1.1.4 ÁRBOL DE MEDIOS Y FINES	6
1.1.5 ALTERNATIVA DE SOLUCIÓN	8
1.2 JUSTIFICACIÓN DEL PROYECTO	10
1.3 OBJETIVOS	11
1.3.1 OBJETIVO GENERAL	11
1.3.2 OBJETIVOS ESPECIFICOS	11
1.4 ALCANCES Y LIMITACIONES	12
1.4.1 ALCANCES	12
1.4.2 LIMITACIONES	13
1.5 MARCO DE REFERENCIA	14
1.5.1 MARCO TEORICO	14
1.5.2 MARCO CONCEPTUAL	25
1.6 DISEÑO METODOLOGICO	31
1.6.1 TIPO DE ESTUDIO	31
1.6.2 FUENTES Y TÉCNICAS	31
1.6.3 TRATAMIENTO DE LA INFORMACIÓN.	31
CAPITULO 2	33
ANÁLISIS Y DISEÑO DEL MODELO	33
2.1 MODELADO DEL SISTEMA DE INFORMACIÓN	33
2.2 INGENIERIA DEL PRODUCTO	34
2.3 INGENIERIA DE REQUISITOS	36
2.3.1 Configuración del sistema productivo	36
2.3.2 Configuración de los productos en el sistema	37

2.3.3	Descripción de las órdenes de pedido	39
2.3.4	Descripción de las órdenes de producción	39
2.4	INGENIERIA DEL PROCESO	40
2.4.1	PROCEDIMIENTO UTILIZADO	41
2.4.2	Etapa II. Construcción de planes.	43
2.4.3	Etapa III. Compresión de planes con el algoritmo del margen de tolerancia.	44
2.4.4	Consideración de programación reactiva, por cambio en el estado del sistema.	49
2.5	MODELADO DEL SISTEMA	51
2.6	MODELADO FUNCIONAL Y FLUJO DE INFORMACIÓN	52
<i>CAPITULO 3</i>		53
ANALISIS DE RESULTADOS		53
3.1 RESULTADOS OBTENIDOS		56
3.1.1	Comparación con la realidad	56
3.1.2	Comparación Con Otros Modelos	58
<i>CONCLUSIONES</i>		61
<i>RECOMENDACIONES</i>		64
<i>FUTURAS INVESTIGACIONES</i>		65
<i>BIBLIOGRAFIA</i>		66

INDICE DE TABLAS Y GRAFICOS

GRAFICA 1. ÁRBOL CAUSA – EFECTO.....	3
GRAFICA 2. ÁRBOL DE MEDIOS Y FINES.....	7
GRAFICA 3. JOB SHOP FLEXIBLE.....	27
GRAFICA 4. INGENIERÍA DEL PRODUCTO.....	35
GRAFICA 5. PROCEDIMIENTO DE LA SEMILLA.....	41
GRAFICA 6. COMPORTAMIENTO DEL MARGEN DE TOLERANCIA.....	46
GRAFICA 7. MODELADO DEL SISTEMA.....	51
GRAFICA 8. FLUJO DE INFORMACIÓN.....	52
GRÁFICA 9. DE PROGRAMACIÓN PARA REPARACIÓN DE BUQUE.....	56
TABLA 1. RESULTADOS OBTENIDOS COMPARANDO LEKIN VS HS.....	59

INTRODUCCIÓN

En el mundo globalizado en el que se encuentran actualmente las empresas, se hace indispensable el mejoramiento de todos sus sistemas y las diferentes formas de llevar a cabo sus procesos, en especial los que le agregan valor a los productos ó servicios que ofrecen con el fin de satisfacer al cliente. Observando las situaciones que se presentaron en el pasado, se puede ver como las prioridades de las compañías van cambiando, desde el tema de producir más, pasando por el de producir mejor, llegando hasta la prioridad de satisfacer al cliente. Todas estas son diversas estrategias que hacen que las empresas busquen las mejores formas de desarrollar sus procesos.

Disminuir costos, aumentar ganancias y mantener a los clientes, es la lucha del día a día de todos los ingenieros y personas encargadas de tomar decisiones en todo tipo de empresa, es por esta razón que nacen softwares de apoyo a las actividades empresariales, que hacen de las labores tediosas y/o poco confiables algo más eficaces. En la actualidad, con la globalización del mundo, la presencia de software de apoyo a las actividades puede hacer la diferencia en las operaciones de una empresa. Hoy en día existe toda clase de software que van desde aquellos que captan y manejan la información, hasta aquellos que suministran la información acerca de cuál es la mejor forma de realizar las actividades y llevar a cabo procesos; ejemplo de estos últimos son los software que permiten programar las tareas a realizar en una empresa (Scheduling).

En este proyecto se creará un algoritmo heurístico que busque optimizar diferentes criterios en el problema de secuenciación de trabajos que se realizan por proyectos, en sistemas de producción que maneje fabricación y ensamble. El trabajo se encuentra dividido en tres capítulos, en el primero se hace una introducción al problema a tratar, en el segundo se define cómo se plantea la heurística y en el tercero se muestran los resultados obtenidos.

CAPITULO 1

PLANTEAMIENTO DE LA INVESTIGACIÓN

1.1 FORMULACIÓN DEL PROBLEMA

1.1.1 ANTECEDENTES

En un entorno empresarial globalizado es básico minimizar la incertidumbre y disminuir riesgos buscando la mejor forma de realizar todos sus procesos. En la actualidad al momento de buscar mejores soluciones, por ejemplo, el caso de programar las operaciones, muchas empresas creen que la experiencia es el factor más importante para enfrentar este flagelo, al no confiar en formas lógicas que puedan resolver los problemas. Hoy en día, la programación de operaciones reviste una gran importancia en los diversos sistemas productivos, debido a la incidencia de la competitividad y productividad de la empresa.

Los métodos teóricos propuestos para programación de operaciones en los sistemas productivos se basan en supuestos que guardan relación con el entorno de fabricación y los pedidos de los clientes. Sin embargo, en el mundo real es muy raro que se cumplan estos supuestos. Además, resulta necesario modificar los programas de producción implementados frente a la aparición de sucesos inesperados, en lo que respecta a la disponibilidad de los medios productivos o la

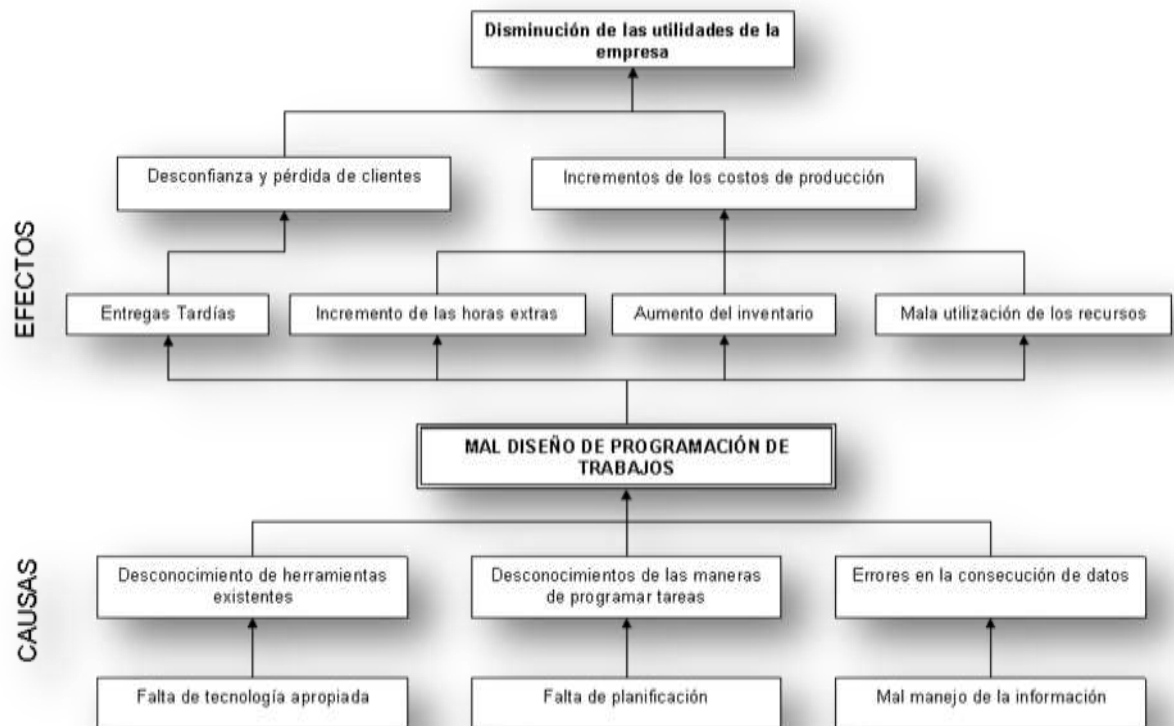
prioridad de las órdenes de trabajo. Esto hace que en la práctica sea muy difícil diseñar las reglas que ayuden a obtener un programa bueno teniendo en cuenta que la distribución del sistema productivo es muy compleja, y que la forma como se mueve el material a través de éste también lo es.

Si se buscan las mejores opciones al momento de programar operaciones, por una parte se tiene las reglas heurísticas, las cuales pueden proporcionar buenas soluciones a problemas complejos en tiempo real y por otro lado están los algoritmos meta-heurísticos que también pueden aplicarse con éxito a la programación dinámica de operaciones, gracias a su flexibilidad para adaptarse al tiempo disponible para hallar una solución.

Lo que se desea con éste estudio es diseñar un algoritmo que ayude a obtener una buena solución para la programación de operaciones en sistemas de fabricación y ensamble, cuyo objetivo sea el de minimizar el lapso, que arroje soluciones factibles buenas, en tiempo real corto y que sirva para cualquier entorno empresarial que maneje éste tipo de configuración.

1.1.2 ARBOL CAUSA – EFECTO

A continuación se presenta el árbol causa – efecto que nos da una idea más clara de los diferentes problemas que se presentan en las empresas, resaltando las posibles causas que los producen y efectos que ocasionan. Lo que se observa en éste diagrama es que las raíces del mal diseño de la programación de trabajos son la falta de tecnología junto con el mal manejo de la información y la falta de planificación adecuada. Se puede decir que esta última proviene de la no utilización de reglas heurísticas o algoritmos de optimización que ayuden a esta programación. Todos estos problemas se esbozan en pérdidas para la empresa.



GRAFICA 1. ÁRBOL CAUSA – EFECTO

Fuente: Elaboración Propia.

1.1.3 PLANTEAMIENTO DEL PROBLEMA

En los ambientes empresariales en donde las formas como podemos producir varios productos cumple de una serie grande de restricciones, tales como, un producto pueda tener más de una fase de producción, requiera de diferentes tiempos de procesamiento..., la forma como asignamos tanto: materiales, recursos humanos, fechas, cantidades a producir en cada uno de los recursos materiales ó máquinas con el fin de cumplir con un plan de entrega, se convierte en una multiplicidad de opciones en la cuales seleccionar la mejor es una ardua tarea. El resultado de este proceso se conoce mejor como la elaboración de un plan de secuenciación ó la programación de operaciones. En diversos sistemas productivos, la programación de operaciones reviste una gran importancia, debido a la incidencia de la competitividad y productividad de la empresa.

Programar operaciones es el proceso de asignar recursos limitados a las tareas en un cierto plazo, con el fin de optimizar uno o más objetivos. Los recursos pueden ser máquinas o personas. Los ejemplos de tareas incluyen operaciones en máquinas, mudanzas, transporte, cargar, descargar, etc.

El uso de heurísticas y meta-heurísticas en la programación de operaciones, como su nombre en inglés "Scheduling", ayuda a entregar programas muy cercanos al objetivo óptimo. Estos modelos matemáticos, usados para la programación de

operaciones, dependen de la forma como esté distribuida la empresa, refiriéndose a la estructura física y flujo de producto que posea, pasando desde el más sencillo de una maquina simple en donde entra un solo tipo de producto, a los problemas tipo Job Shop Flexible que poseen una complejidad alta para su solución hasta la programación de proyectos, en los cuales se encuentran actividades de ensamble y pueden trabajar tanto máquinas como operarios en una misma actividad.

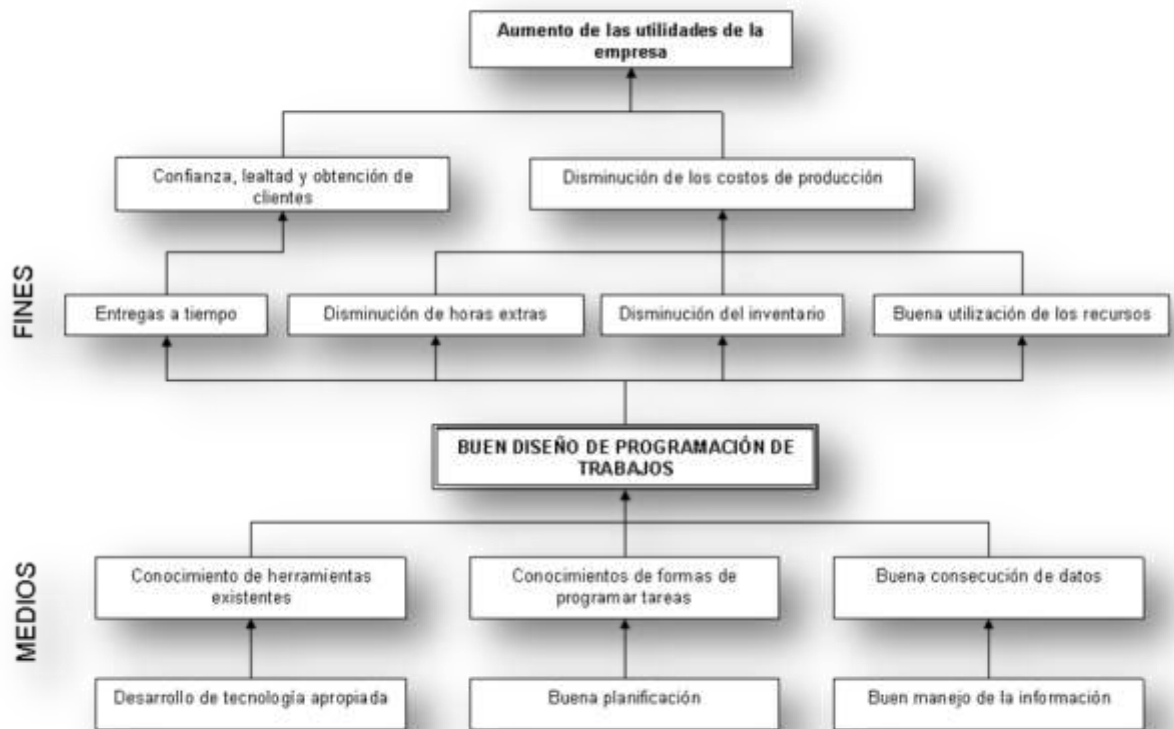
Para el uso de la configuración productiva se pueden encontrar diferentes referencias. Si se estudia el contexto empresarial, podrá encontrarse que existen distintos sistemas de producción en las empresas manufactureras y de servicio.

Las tareas pueden tener fechas de entrega y algunas tareas pueden tener prioridad sobre otras. Entre los objetivos de programar las operaciones es relevante considerar el objetivo de reducir al mínimo el tiempo de terminación de todos los trabajos (MAKESPAN) maximizando el número de las tareas terminadas en un tiempo dado y reduciendo el inventario en proceso.

1.1.4 ÁRBOL DE MEDIOS Y FINES

A continuación se presenta el árbol de medios y fines, en el cual se plasman los diferentes medios de solución a los problemas planteados y los resultados o metas que se esperan ser obtenidas.

En este diagrama se puede observar cómo la interacción de una buena tecnología en conjunto con una buena planificación o reglas adecuadas junto con un buen manejo de la información, generan buenos planes con los se puede mejorar las utilidades de la empresa, dando mejores beneficios. Lo que se busca con este proyecto es desarrollar buenas herramientas de programación, que en conjunto con un programa adecuado, ayuden a empresas que posean configuración de fabricación y ensamble, a realizar un buen diseño de los programas del trabajo.



GRAFICA 2. ÁRBOL DE MEDIOS Y FINES

Fuente: Elaboración Propia

1.1.5 ALTERNATIVA DE SOLUCIÓN

En el árbol de medios y fines se muestran las posibles soluciones que dan respuesta a los problemas presentados por la mayoría de las empresas. Para llegar a solucionar estos problemas son indispensables:

- ✓ **Buen manejo de la información.** Esta solución se plantea debido a las deficiencias que maneja el sistema productivo al momento de almacenar la información de producción. En la mayoría de las empresas sólo se manejan datos relacionados a grandes cantidades en periodos de tiempo largos (días o semanas), pero no cuentan con la información de tal manera que se pueda conocer de manera precisa a qué hora entra, se procesa y sale una orden para su posterior estudio.

- ✓ **Buena Planificación.** En las empresas no se cuenta con un sistema que planifique la producción, es decir, no disponen de sistemas que a través de lógicas generen la mejor forma de producir, lo que hace que sea más difícil organizar la producción al momento de entrar las órdenes. Para éste problema sería bueno que se generaran modelos que ayudaran a la toma de decisiones al momento de organizar el programa de producción.

- ✓ **Tecnología Apropriada.** Para dar solución a los problemas de programación de la producción, se podría desarrollar un modelo sistemático (software) que ayude en la programación de las operaciones en la empresa.

Una conjunción de las tres soluciones sería la opción más adecuada para solucionar el problema planteado, ya que con la ayuda de una herramienta que facilite la programación de los trabajos, ayudando al manejo de información y que además utilice las lógicas correctas al momento de programar, contribuiría a disminuir las entregas tardías, eliminando inventario y minimizando los costos.

1.2 JUSTIFICACIÓN DEL PROYECTO

Dar respuesta a la necesidad de encontrar un sistema que ayude a obtener una solución rápida para una buena programación de operaciones (Scheduling) en sistemas de producción complejos, como son los sistemas de que manejan fabricación y ensamble, en los cuales en muchas ocasiones se trabaja en órdenes por proyectos y en los cuales las órdenes no siempre tienen las mismas características, es un reto importante. En la realidad, a la mayoría de las empresas que poseen un sistema de programación de órdenes, y concretamente a las empresas donde se piensa implementar el nuevo sistema, SUPER BRIX y COTECMAR, se les plantean una serie de problemas al momento de programar la producción, debido a la multiplicidad de opciones que se les presentan y no encuentran una manera rápida de obtener la mejor ó a lo menos una buena programación que implique optimizar los objetivos planteados.

Lo que se quiere en éste proyecto es modelar un sistema de optimización en la programación de operaciones de los sistemas de manufactura los cuales trabajan por proyectos cuya configuración productiva va regida por una fabricación y ensamble, en la cual las ordenes no siempre tienen las mismas características; que ayude a las empresas a hallar una buena programación de sus trabajos, que si no es la óptima sea muy buena, y obtener ésta en un lapso de tiempo corto; y que ayude a las empresas a tener mayor coordinación en la programación de sus trabajos.

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

Diseñar un algoritmo de programación de operaciones de proyectos para sistemas cuya configuración sea de fabricación y ensamble, buscando arrojar un diagrama web de decisiones al momento de seleccionar un plan, bajo criterios de fechas de entrega, costo y makespan.

1.3.2 OBJETIVOS ESPECIFICOS

- Optimizar la forma de programar las operaciones en un entorno productivo donde sus trabajos se realicen por proyectos, tomando en cuenta las reacciones que se presenten en el día a día.

- Generar unos formatos de historia de producción y detenciones de la planta con el fin de poseer archivo detallado de los sucesos que ocurren en la empresa.

1.4 ALCANCES Y LIMITACIONES

1.4.1 ALCANCES

Con este proyecto se busca desarrollar un modelo que ayude a la programación de las operaciones en sistemas cuya configuración sea de fabricación y ensamble de órdenes de producción, en donde se trabaja por proyectos y pueda manejar reactividad.

Este proyecto busca encontrar soluciones buenas en tiempo muy corto, trabajando con técnicas de priorización bajo los conceptos de semillas de inicio de producción.

1.4.2 LIMITACIONES

Son varias las limitaciones que se tendrán al momento realizar éste proyecto. Una es la complejidad de suministrar los escenarios de fabricación, debido a que se realizan las pruebas con restricciones asociadas al manejo de la empresa.

Otra dificultad seria al momento de validarlo, debido a que se necesita toda la información utilizada en la realidad si se quiere compara con ésta. También es muy difícil adquirir y encontrar software que cuente con la potencia suficiente de lo planteado en este trabajo para poder comparar.

1.5 MARCO DE REFERENCIA

1.5.1 MARCO TEORICO

1.5.1.1 ¿Qué es un algoritmo?

Un algoritmo es una “receta” para obtener una solución de un modelo. Los algoritmos exactos proporcionan una solución óptima a un problema, los algoritmos heurísticos dan soluciones que se espera sean las óptimas o cercanas a la óptima, estos últimos mediante un conjunto de reglas “racionales” obtienen una solución “buena”, y se utilizan cuando no es posible obtener una solución óptima¹.

1.5.1.2 Programación de operaciones y algoritmos.

La programación de operaciones se basa en la utilización de técnicas matemáticas o métodos heurísticos para la asignación de trabajos a unos recursos limitados. Una asignación apropiada de los recursos hace que la compañía optimice sus objetivos y alcance sus metas².

¹ SIPPER, D. Y Bulfin R. Planeación y Control de la Producción. 1998 Mc Graw Hill, p. 403.

² PINEDO M. Chao X. Operation of Scheduling with applications in manufacturing and services. Irving- Mc Graw Hill. 1999.

Morton y Pentico (1993) afirman: “Programar es el proceso de organizar, elegir y dar tiempos al uso de los recursos para llevar a cabo las actividades necesarias, para producir las salidas deseadas en los tiempos deseados, satisfaciendo a la vez un gran número de restricciones de tiempo y relaciones de los recursos”³.

La programación de las operaciones ocurre en gran parte de las actividades que le agregan valor a un producto, siempre involucrando la mejor forma de utilizar un recurso en un periodo de tiempo. Los recursos tienen una capacidad limitada. Los procesos a realizar pueden ser llamados “trabajos”, “proyectos” ó “asignaciones” y están formados por partes elementales llamadas “actividades” ú “operaciones” y “demoras”. Cada actividad requiere de cierta cantidad de recursos específicos en un tiempo específico llamado “tiempo de procesamiento”. Los recursos también tienen partes elementales, llamadas “máquinas”, “celdas”, “trasportes”, entre otras⁴.

Al programar las operaciones se presentan problemas muy complejos, en su mayoría debido al gran numero de restricciones que relacionan actividades unas con otras, recursos con actividades y recursos ó actividades con eventos externos al sistema.

³ SIPPEN, Op. cit., p. 399.

⁴ PINEDO, OP. cit, p. 372.

Un modelo de programación de operaciones reactivo se encarga de hacer modificaciones al programa de producción original en respuesta a la aparición de sucesos imprevistos que puedan afectar a la viabilidad del programa de producción en curso. Una vez que el programa de producción generado de forma off-line se lanza a la planta, se convierte en el programa de producción activo. A partir de éste momento, dicho programa de producción se somete a una monitorización por si alguna perturbación pudiera afectar a su validez⁵.

1.5.1.3 Programación de proyectos.

Los problemas de secuenciación de proyectos (PSP) están formados por actividades, recursos, relaciones de precedencia y funciones de evaluación (Slowinski et al., 1994). Las primeras aproximaciones fueron hechas en los 50. Algunos artículos y libros históricos son Kelley 1961, Malcon et al., 1959, Morder et al., 1983 y Pritsker y Happ, 1966.⁶

1.5.1.4 Ventajas de la Programación de Operaciones

Determinar la “mejor” secuencia parece sencillo: sólo habría que enumerar todas las secuencias y elegir la que optimizara alguna medida de desempeño. Para 47 trabajos, el número de secuencias posibles es de $32!$, aproximadamente 2.6×10^{35} . Suponga que una computadora puede examinar mil millones de secuencias por segundo, tomaría 8.4×10^{15} siglos enumerarlas todas. Una computadora que

⁵ PINEDO. Op. cit, p. 384.

⁶ BALLESTIN F 2002

fuera un millón de veces más rápida tardaría 8.4×10^9 siglos para examinarlas. Con solo 16 trabajos existen más de 20 millones de posibles soluciones, que a una tasa de mil millones de secuencias por segundo, podrían enumerarse más o menos en ocho meses. Muy pocos problemas son tan sencillos. Los recursos adicionales (mano de obra, materia prima, etc.) y las dependencias de los trabajos (como la preparación) complican más el problema. La utilización de algoritmos heurísticos para la programación de las operaciones hace que éste proceso se haga más rápido, debido a que sólo busca una solución (la óptima ó una que se le acerque en caso de que ésta sea difícil de encontrar), ignorando las que son menos efectivas⁷.

1.5.1.5 Evolución de la Programación

Desde el comienzo de nuestra época ha existido sin duda la programación de las operaciones. Gantt (1911) fue, tal vez, la primera persona en impulsar el enfoque cuantitativo de la programación. Como área de atención, la programación en realidad tuvo su inicio a mediados de los 50. Johnson (1954), Smith (1956), Jackson (1956) y McNaughton (1959) desarrollaron los modelos y las soluciones clásicas. Estas publicaciones dieron fundamento al trabajo que siguió. Los primeros resultados fueron consolidados por Muth y Thompson (1963). Se modelaron muchos problemas complicados de programación como problemas de

⁷ SIPPER, Op. cit., p. 399

programación entera mixta, pero su naturaleza combinatoria los hizo imposibles de resolver en un tiempo razonable. Esto hizo que los esfuerzos se tornaran a los algoritmos heurísticos. Conway et al (1967) escribieron el primer libro de texto sobre la programación de la producción. Una generación de investigadores y profesionales aprendió programación en el riguroso pero comprensible libro de Baker (1974)⁸.

Los años 70 produjeron más resultados teóricos. El trabajo pionero sobre análisis heurístico para el peor caso (Graham, 1969) y la complejidad de varios modelos de programación de la producción (Karp, 1972) condujeron a resultados importantes. Rinnooy Kan (1976) popularizó estos enfoques. La investigación actual sobre la programación está dedicada a resultados más prácticos. Panwalker et al (1973) y McKay et al (1988) investigaron la programación de la producción en la industria e identificaron aspectos relevantes. La reorganización de los sistemas de manufactura, en particular el movimiento para reducir inventarios y la necesidad de reducir tiempos de entrega, ha hecho que la programación de operaciones sea mucho más importante.

Las computadoras actuales más rápidas, al alcance de todos, e interfaces graficas amigables, han hecho que sea más sencillo implantar una programación de la

⁸ SIPPER, Op. cit., p, 469.

producción. APICS: *The Performance Advantage*, una revista orientada a los profesionales en la práctica, ha cambiado su evaluación del software, de los paquetes MRP II a los paquetes de programación de capacidad finita. Se han publicado varios artículos sobre el creciente interés de la programación de la producción, que entre más avances metodológicos más difícil se hace la distinción entre la planeación y la programación⁹.

En lo que respecta a las reglas heurísticas, uno de los problemas más importantes es que no hay una regla que sea globalmente mejor que las demás. Estas reglas emplean datos como el tiempo de procesamiento, las fechas de entrega o los instantes de llegada. Su eficacia depende, en gran medida, de las condiciones operacionales (por ejemplo, la presencia de cuellos de botella, la proximidad de las fechas de entrega, o la carga de trabajo de la planta) o de las medidas de rendimiento elegidas para medir la calidad del programa de producción resultante. Esto significa que una determinada regla puede resultar elegida gracias a que genera buenos resultados en algunos indicadores, aunque pueda arrojar peores resultados en otros. Por ello, muchas veces resulta difícil decidir qué regla es más adecuada a cada caso. Además de esto, existen combinaciones de reglas que se obtienen a partir de reglas más sencillas. Estas reglas suelen depender de la

⁹ SIPPER, Op. cit., p, 469.

situación en la planta. Esto evita que los trabajos cuyos tiempos de procesamiento sean largos se mantengan en la cola durante mucho tiempo¹⁰.

Baker (1974) clasifica las reglas en dos grupos: locales y globales. Una regla es local si la asignación de prioridad sólo se basa en información acerca de los trabajos que se encuentran a la espera en la cola de la máquina afectada (por ejemplo, LPT). Una regla es global si, además, utiliza información de otras máquinas (por ejemplo, AWINQ). Shafei y Brunn (2000) investigan el rendimiento de varias reglas de secuenciación a partir de un método de horizonte de planificación continuo en un entorno de fabricación dinámico y sus resultados muestran que una regla de reciente creación, llamada SPT-C/R, es la más adecuada para minimizar el coste global y que existe una relación entre la calidad del resultado y la longitud del horizonte de replanificación empleado. Los resultados de éste estudio de simulación muestran que esta regla da los mejores resultados a lo largo de distintos horizontes de replanificación y bajo condiciones operativas diferentes. Jayamohan y Rajendran (2000) proporcionan un conjunto de nuevas reglas heurísticas (por ejemplo, EDD, PT+PW, PT+PW+EDD, etc.) para conseguir minimizar distintas medidas de rendimiento tales como el tiempo de flujo máximo y el retraso en plantas de producción muy dinámicas¹¹.

¹⁰ MORTON, T. Pentico W. Heuristic scheduling systems: with applications to productions systems and project management. Jonh Wiley and sons. 1993.

¹¹ MORTON, Op. Cit., p. 327.

Tradicionalmente, la investigación en plantas de tipo Job-shop sólo ha tenido en cuenta restricciones relativas a la disponibilidad de máquina y de mano de obra. Con la llegada de los sistemas de fabricación flexible y la fabricación justo a tiempo, se ha empezado a tener en cuenta la importancia de los recursos auxiliares en las actividades de control de planta y de programación de operaciones. A éste respecto, Gargeya y Deane (1999) desarrollan un método de programación basada en contingencias en un entorno de fabricación dinámico restringido por recursos auxiliares. Se emplearon distintos tipos de indicadores, incluyendo la raíz cuadrada del retraso, el tiempo medio de estancia en el sistema y el número de cambios de recursos auxiliares. A medida que aumenta la utilización de la planta, el estudio revela que el algoritmo proporciona muy buenos resultados en los tres tipos de indicadores mencionados¹².

Los algoritmos genéticos (AG) son algoritmos de optimización aproximados que utilizan una terminología basada en los procesos de evolución genéticos de las especies. En realidad, los AG emplean los mecanismos de selección natural y la genética para encontrar soluciones óptimas a problemas de optimización combinatorios. Los AG surgieron a partir de los estudios llevados a cabo por John Holland en la Universidad de Michigan. Goldberg (1989) proporciona una interesante recopilación del trabajo práctico llevado a cabo en esta área.

¹² MORTON, Op. Cit., p. 338.

Los AG pueden entenderse como una generalización de los algoritmos estocásticos de búsqueda local. El espacio de búsqueda de las soluciones candidatas se asemeja a un conjunto de organismos agrupados en poblaciones que evolucionan en el tiempo. La función objetivo del problema de optimización se implementa como una función de adaptación de organismos. La función de adaptación representa el grado de adaptación al entorno del organismo. Los AG se aplican a campos como la planificación de operaciones en planta, problemas de corte de materiales, problemas de testing de componentes irregulares, el entrenamiento de redes neuronales y muchos otros problemas de optimización. Los AG presentan muchas ventajas de cara a su empleo en problemas de programación de operaciones en planta. La calidad de la solución y el tiempo de proceso dependen directamente de los parámetros de los AG, concretamente del número de organismos de la población y del máximo número de generaciones a probar. Esta característica les confiere gran flexibilidad. Rossi y Dini (2000) afirman que es posible encontrar los mejores valores de los parámetros de configuración de un AG a través de otro procedimiento de optimización como, por ejemplo, el enfriamiento simulado.

Con respecto al software, se dispone de paquetes de programación de producción en abundancia. Estos paquetes van desde software para PC de dominio público hasta aplicaciones hechas a la medida que cuestan millones de dólares. Con respecto a estos software, han ido evolucionando y mejorando cada vez más

hasta el punto que muchas empresas lo aplican de manera eficaz y confiando mucho en los resultados que arroje éste. Podemos nombrar algunos ejemplos como los siguientes:

La multinacional FEDEX (Federal Express) realiza la asignación de sus trabajos maximizando la utilización de sus vuelos y minimizando las rutas realizadas y sus entregas tardías por medio de un sistema de scheduling¹³.

En la industria maderera de Chile se aplican modelos de scheduling para la utilización de los carros transportadores, lo cual hace que se aproveche este recurso al máximo, por medio del programa ASICAM, el cual programa los trabajos y a su vez hace la simulación¹⁴.

En la operación Tormenta del Desierto en el plazo de tres meses, manejaban un sistema de instrumentos de apoyo de decisión para manejar la información sobre la carga y los pasajeros que se movían y los recursos disponibles, así como las

¹³ CAMPBELL, Kevin. Bret, Robert. FedEx Generates Bid Lines Using Simulated Annealing. Revista INTERFACES Volume 27, Number 2, Mar/Apr 1997 pp:1-16 disponible en internet: <<http://www.interfaces.smeal.psu.edu/pdf/v27n2a1.pdf>>

¹⁴ WEINTRAUB, Andres. EPSTEIN, Rafael. A Truck Scheduling System Improves Efficiency in the Forest Industries. Revista INTERFACES Volume 26, Number 4, Jul/Aug 1996 pp:1-12 disponible en internet: <<http://www.interfaces.smeal.psu.edu/pdf/v26n4a1.pdf>>

herramientas para programar misiones, para analizar el horario y distribuir éste al comando y al sistema de control mundial¹⁵.

A nivel deportivo, para la organización de los juegos olímpicos se utilizaron sistemas de programación DSS (Decision Support System) que ayudaban a programar el mejor horario para la disposición de cada uno de los juegos¹⁶.

En el ámbito local pocas empresas manejan estas herramientas en su planta, entre éstas se encuentran:

- Retycol utiliza el sistema Finanzoft: módulo de programación (INTEGRA) (No optimizado).
- Procaps utiliza el sistema Visual Manufacturing (no optimizado).
- RYMCO utiliza un módulo de programación de tareas (no optimizado).

Y cada vez más las empresas van llevadas de la mano para mejorar la asignación de sus trabajos haciendo de la programación algo muy sencillo y eficaz y preocupándose más por otros efectos ajenos a la programación que afecten en su entorno.

¹⁵ HILLIARD, Michael. LIU, Cheng. BUSCH, Ingrid. HARRISON, Glen. Scheduling the Operation Desert Storm Airlift: An Advanced Automated Scheduling Support System. Revista INTERFACES Volume 22, Number 1, Jan/Feb 1992 pp:131-146. disponible en internet. <<http://www.interfaces.smeal.psu.edu/pdf/v22n1a7.pdf>>

¹⁶ ANDREU, Rafael. COROMINAS, Albert. A DSS for Scheduling the Olympic Games. Revista INTERFACES Volume 19, Number 5, Sep/Oct 1989 pp:1-12. disponible en Internet: <<http://www.interfaces.smeal.psu.edu/pdf/v19n5a1.pdf>>

1.5.2 MARCO CONCEPTUAL

1.5.2.1 Diagrama de Gantt

Es un diagrama o gráfica de barras que se usa cuando es necesario representar la programación de las tareas en un periodo determinado de tiempo. Muestra la ocurrencia de actividades en paralelo o en serie en un determinado período de tiempo.

1.5.2.2 Reglas Heurísticas

Método lógico que proporciona soluciones buenas a un problema planteado y en ocasiones esta solución puede lograr ser la óptima.

Las utilizadas en este trabajo son:

- SPT- LPT: (Shortest ó Largest Processing Time): se ordenan los trabajos ya sea de menor o mayor tiempo de procesamiento para ser programados.
- EDD: (Early Due Date) se ordenan los trabajos de acuerdo a la fecha de entrega más cercana.
- W: (Weigth) se ordenan los trabajos teniendo en cuenta la importancia ó peso.
- MANP-MINP: (Maximums ó minimum number of process). Se ordenan los trabajos de acuerdo a la cantidad de trabajos ó procesos a realizar.

Además de estas heurísticas también se pueden trabajar combinaciones de éstas, por ejemplo, el WSPT ó WEDD entre otras.

1.5.2.3 Lاپso de Producción

Es la duración que tiene la suma de los tiempos de producción y tiempos ociosos de un programa de producción en un tiempo determinado. Un objetivo importante es minimizar este tiempo con el fin de utilizar la capacidad de las máquinas al máximo y minimizar los tiempos ociosos.

1.5.2.4 Modelos de Máquinas En Paralelo

Con frecuencia los problemas de programación consideran varias máquinas que pueden realizar la misma operación, en las cuales un trabajo puede realizarse en cualquier maquina aunque sus velocidades de proceso sean diferentes.

1.5.2.5 Modelos de un Job Shop flexible

En este tipo de configuración un trabajo se mueve a través de diferentes estaciones de trabajo, cada una de ellas contiene máquinas en paralelo, es decir, el trabajo puede pasar por cualquiera de las maquinas en esta estación, sin embargo la ruta que siga cada trabajo puede variar. En el grafico se muestra un ejemplo que posee 13 estaciones y 4 posibles secuencias de los trabajos.

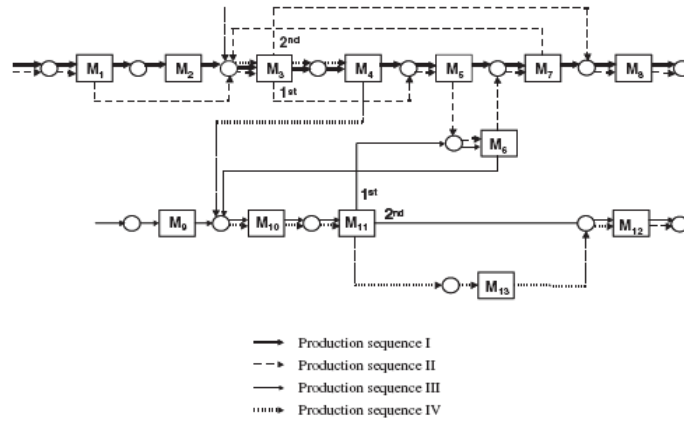


Ilustración 1 Job Shop Flexible

1.5.2.6 Notación matemática:

Un problema de Scheduling se describe como una tripleta del tipo $\alpha | \beta | \gamma$. α describe la configuración de recursos, β proporciona detalles sobre las características de procesamiento del taller y γ describe el objetivo a optimizar.

1.5.2.7 Notación matemática para el campo α :

1: 1 máquina

P_m : Máquinas idénticas en paralelo

R_{em} : Máquinas en paralelo con velocidades diferentes

F_m : m Máquinas en serie

FC_c : flowshop flexible (híbrido) con c estaciones en serie

J_m : Jobshop con m máquinas

FJ_c : jobshop flexible (híbrido) con c estaciones de m máquinas idénticas

1.5.2.8 Notación matemática para el campo β :

r_i : Tiempo de liberación de la orden del trabajo i

S_{jk} : Tiempos de preparación dependientes de la secuencia

$prpm$: \exists la posibilidad de culminar la operación sobre un trabajo (orden) en diferentes máquinas

$prec$: Restricciones de precedencia

$brkdwn$: Fallas en los recursos

M_j : Elegibilidad de recursos (para problemas P_m)

$prmu$: programa de Permutación (secuencia se mantiene)

$block$: bloqueos (flowshop) debido a buffers insuficientes

nwt : no-wait

$recrc$: Recirculación de trabajos

Cualquier otra información en el campo b es auto-explicatoria, por ejemplo

$p_i = p$ implica que todos los tiempos de procesamiento son iguales.

1.5.2.9 Notación matemática para el campo γ :

C_i : tiempo de terminación del trabajo i

$F_i = c_i - r_i$: tiempo de flujo del trabajo i

$L_i = c_i - d_i$: retraso del trabajo i

$T_i = \max. \{0, l_i\}$: tardanza del trabajo i

$E_i = \max. \{0, -l_i\}$: adelanto del trabajo i

$C_{\max.}$: Makespan (lapso), tiempo máximo de terminación de todos los trabajos

$L_{\max.}$: Retraso máximo de todos los trabajos

$T_{\max.}$: Tardanza máxima de todos los

Trabajos

1.5.2.10 Programación de operaciones por proyectos

El conjunto de trabajos requeridos por una orden de demanda constituye un proyecto cuya estructura es generalmente distinta para cada orden.

1.5.2.11 Recursos renovables

Son todos aquellos recursos que, dependiendo de su disponibilidad, se pueden usar a lo largo de todo el proyecto. Su disponibilidad puede estar restringida a por un calendario. El ejemplo más claro son empleados que trabajan fijos en la empresa ó que contrata la empresa por cierto tiempo del proyecto.

1.5.2.12 Recursos no renovables

Son todos aquellos recursos que al utilizarlo se consumen y disminuye su disponibilidad.

1.5.2.13 Eventos disruptivos

Una vez se está ejecutando una programación, vamos a considerar dos tipos de eventos disruptivos: avería de máquina y llegada de nueva orden de demanda. Al ocurrir un evento disruptivo, el plan se modifica; las modificaciones pueden consistir en operaciones que han sido reasignadas a otra máquina o a otro operario, que han sido retrasadas o adelantadas o que han quedado pendientes de programación.

1.5.2.14 Modelo de Reglas Heurísticas (RH)

El modelo RH gestiona cada evento según un enfoque basado en el caso; es decir, sigue un procedimiento específico que tiene en cuenta los objetivos de fabricación. Por tanto, el modelo RH utilizará una lista de trabajos no programados que intentará asignar en alguna máquina que esté libre. Si no

existe suficiente capacidad a lo largo del horizonte de planificación para realizar un determinado trabajo en ninguna de las máquinas disponibles, dicha trabajo quedará pendiente. Esto implica que todos los trabajos que siguen al trabajo pendiente también quedarán pendientes. Se han utilizado dos reglas básicas en éste modelo (Smith, 1995):

- **Left Shifter (LSH):** Esta regla mueve hacia la izquierda (adelanta) los trabajos con respecto a la fecha planificada siempre que la disponibilidad de los recursos y las restricciones de precedencia lo permitan.

- **Right Shifter (RSH):** Esta regla mueve hacia la derecha (retrasa) los trabajos, con respecto a la fecha original planificada.

1.5.2.15 Tiempo De Flujo

Se puede decir que el tiempo de flujo es un tiempo real que toma el material en atravesar el sistema de producción; su duración depende de la programación y el planificador quiere generalmente reducirla cuanto sea posible.

1.6 DISEÑO METODOLOGICO

1.6.1 TIPO DE ESTUDIO

Debido a la naturaleza de investigación y puesta en práctica de un modelo teórico, se puede afirmar que este proyecto posee un tipo de estudio exploratorio. Se puede considerar de este tipo, debido a que se tiene un amplio conocimiento del lugar de aplicación y resultados que se quieren obtener con este estudio.

1.6.2 FUENTES Y TÉCNICAS

Para la recolección de los datos necesarios para el desarrollo del proyecto, se usarán dos tipos de fuentes técnicas. Se tendrá una fuente de información *primaria*, que será aquella obtenida durante las visitas a las empresas que posean problemas de programación de operaciones, por medio de la *observación*, como recolección de datos de los productos, distribución de máquinas, entre otros.

Se contará con una fuente de información *secundaria*, que será toda la información obtenida de textos, documentos, revistas, Internet y personal de la empresa.

1.6.3 TRATAMIENTO DE LA INFORMACIÓN.

La forma como se recolectarán los datos suficientes para el desarrollo de éste proyecto serán las tablas de datos, en donde se obtendrá información como: tiempos de operación de las máquinas, tipos de referencias de los productos,

entre otros. Al momento de entregar los datos obtenidos, se representarán por medios de gráficos de Gantt, en los cuales se podrán visualizar de manera más clara los resultados obtenidos.

CAPITULO 2

ANÁLISIS Y DISEÑO DEL MODELO

2.1 MODELADO DEL SISTEMA DE INFORMACIÓN

Para el desarrollo de la herramienta que facilitará la programación de las operaciones en las empresas, es necesario establecer los recursos que se tendrán en cuenta y los productos u órdenes que van a ser parte del modelo. Así como cuál es la ruta de cada una de estas órdenes, cómo se encuentran distribuidos los recursos en el sistema, entre otras.

Para modelar el sistema de producción seleccionado es necesario identificar, especificar y analizar todas las variables que van a ser tenidas en cuenta para el desarrollo de la herramienta como, por ejemplo, productos, máquinas, tiempos de ciclo, entre otros.

2.2 INGENIERIA DEL PRODUCTO

Para definir la ingeniería del producto, es necesario identificar la capacidad instalada que posee la empresa y los requerimientos necesarios para desarrollar el modelo. Se hace indispensable definir la arquitectura y la infraestructura que soportará el sistema.

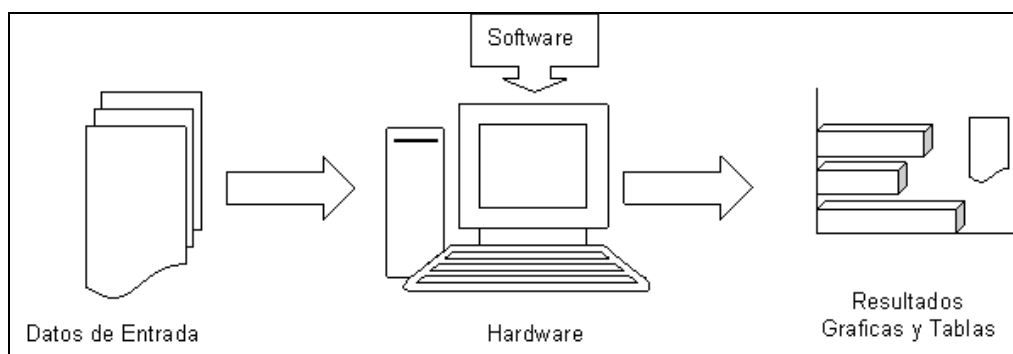
La arquitectura comprende tres componentes de sistemas distintos:

- ✓ **Software.** Este componente hace referencia a la forma cómo se corre el modelo que se va a desarrollar para la programación de las operaciones en la empresa, ayudando a organizar la producción de sus órdenes de producción, desde el momento en que éstas son ingresadas al sistema hasta el momento en que son terminadas de producir.

- ✓ **Hardware.** Para este componente es indispensable contar con un equipo que soporte las operaciones que ejecuta el modelo para la programación de las operaciones. Se requiere una memoria RAM con capacidad de 256MB, como requisito mínimo para un óptimo funcionamiento de la herramienta.

- ✓ **Datos.** Los datos que se ingresen en el modelo tendrán una gran importancia para el desarrollo de éste ya que representan la materia prima fundamental con la que se alimentará el modelo. Los datos serán ingresados por medio del mismo software, esto hace parte de la interfaz del usuario con el modelo. Los datos serán procesados por el modelo y serán arrojados y almacenados en tablas y gráficas que ayudarán en la visualización de los mismos.

Por otra parte, la infraestructura que soportará el sistema hace referencia a los requerimientos necesarios para desarrollar el modelo, es decir, la tecnología que unirá los componentes y la información empleada para correr el modelo. Para esto es necesario que el modelo brinde una interfaz fácil de utilizar por los usuarios de éste, para el ingreso de los datos y los resultados que proporcione el modelo.



Grafica 8. INGENIERÍA DEL PRODUCTO

Fuente: Elaboración Propia

2.3 INGENIERIA DE REQUISITOS

Esta parte del modelo facilita la comprensión de los requisitos del cliente, los cuales se desean cumplir en su totalidad por el modelo de programación de operaciones.

Dentro de estos se encuentran:

- Configuración del sistema productivo
- Configuración de los productos ó trabajos en el sistema
- Descripción de las órdenes de pedido
- Descripción de las órdenes de producción

2.3.1 Configuración del sistema productivo

En este paso se define todo lo referente a cómo se encuentra el sistema en donde se va implementar la heurística. Para poder vincular este campo se debe tener en cuenta:

- Definición de estaciones de trabajo.

Todos los lugares donde se realizan actividades diferentes que se encuentren en la parte productiva

- Definición de horarios laborales

Todas y cada una de las posibles jornadas las cuales pueda tener un recurso ó un operario

- Definición de recursos

Identificación de todos y cada uno de los recursos (máquinas y/ó vehículos) a operar, teniendo en cuenta la eficiencia, estación asociada ó posición en el sistema, costo y características requeridas del proceso.

- Definición de operarios

Identificación de todos y cada uno de los operarios que trabajan en la parte operativa, estación asociada, horario, costo, eficiencia y características de estos.

- Definición de grupos

Todos y cada uno de los posibles grupos de máquinas, operarios que pueden realizar tareas similares.

- Distancias

Matrices de distancias (Manhattan) entre cada una de las estaciones

- Definición de restricciones

Todas y cada una de las posibles limitaciones que hagan que no se puedan utilizar unos recursos. Entre éstas se encuentran:

- Alistamiento de los recursos
- Recorridos o visitas que puede realizar un transporte
- Horarios laborales

- Definición de mantenimiento preventivo

Todas y cada una de las posibles operaciones a nivel preventivo que se tengan que realizar.

2.3.2 Configuración de los productos en el sistema

Aquí se introducen los requerimientos para elaborar un producto. Lo que se quiere con ello es elaborar un árbol de producto que represente qué es necesario para elaborar un producto. Este ítem se puede dividir en:

- Definición de productos

Aquí se definen todos y cada uno de los productos ó proyectos a ser elaborados por la empresa.

- Definición de piezas

Cada una de las piezas que componen un producto ó proyecto con sus respectivas características que afectan su procesamiento.

- Asociación pieza- pieza-recurso.

Éste es la parte más crítica al momento de introducir la información, aquí se define:

- Qué se requiere para realizar las uniones de las piezas.
- Se introducen cantidades de las piezas para elaborar otra pieza de un mayor nivel (parte).
- Recursos que requiere para hacer una parte con sus respectivos tiempos de procesamiento (según sus características).
- Precedencia de los procesos para llegar a un último, en caso de desarrollar varios procesos en la misma parte.

- Asociación parte- producto-recurso.

En éste se definen la asociación del nivel más alto del árbol, aquí se define qué partes se requieren para elaborar el producto.

- Definición de tipos de personalización de productos

Aquí se definen cada uno de los cambios que puede tener el producto según la demanda de los clientes. Estos cambios son de pequeñas características como lo son: colores, forma de entrega y accesorios adicionales entre otras.

2.3.3 Descripción de las órdenes de pedido

En esta parte se define qué quiere el cliente, este paso contiene:

- Definición de los clientes

Aquí se define y codifica cada cliente agregándole la importancia que tiene para el sistema al cumplirle con la fecha de entrega.

- Definición orden cliente

Se define la orden de pedido (cantidades y tipos de productos a llevar), así como su referencia y fecha esperada de entrega por parte del cliente.

- Definición producto cliente.

En éste paso se define las posibles personalizaciones que quiere el cliente de cada uno de los productos.

- Estado de la orden

Cómo se encuentra la orden en el sistema para ver si puede procesarla ó no. Estos posibles estados son: Activo, Stand by, Anulado y Despachado.

2.3.4 Descripción de las órdenes de producción

En este paso se introduce al sistema cómo producir las órdenes de pedido.

Esto lo hace el usuario a su gusto, generalmente agrupando los productos similares para la producción indiferentemente de la orden que provengan.

2.4 INGENIERIA DEL PROCESO

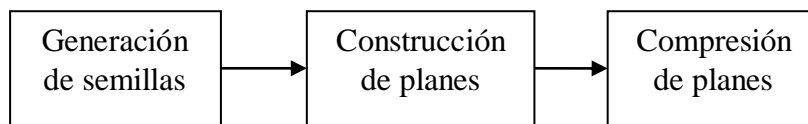
Para procesar toda la información, el software posee una serie de requisitos que él mismo utiliza para encontrar una solución buena. En el desarrollo del modelo se pueden definir una cantidad de variables para su interfaz de entrada. Consideremos la siguiente notación, para facilidad del lector se nombrarán nuevamente alguno de las notaciones ya empleadas:

- **OP_i** la orden de pedido i .
- **ST_i** el estado de la orden de pedido i (Activo, Stand by, Anulado, Despachado).
- **STM_r** el estado de un recurso r (Activo, Mantenimiento).
- **OR_j** la lista de productos que componen la orden de producción j .
- **U_{sj}** el producto s de la orden de producción j .
- **p** el número de órdenes de pedido.
- **J** el número de órdenes de producción.
- **H** una lista de K heurísticas de priorización de trabajos.
- **S_{usi}** , el número de unidades existentes en el inventario, del tipo de pieza u del producto s de la orden de pedido i
- **W_{usj}** lista de procesos que deben ser ejecutados para obtener una pieza u del producto s , para la orden de producción j .
- **TP_{tusj}** el tiempo de procesamiento para el proceso t , de pieza tipo u , del producto s , para la orden de producción j .
- **RQ_{tusj}** la lista de piezas requeridas para el proceso t , de piezas tipo u , del producto s , para la orden de producción j .
- **PR_{tusi}** la lista de procesos predecesores, del proceso t , de la pieza u del producto s , para la orden de producción j .
- **RS_{tusi}** la lista de grupos de recursos necesarios para realizar el proceso t , en la pieza u del producto s , para la orden de producción j .

- **WS_{tusj}** Una variable binaria que indica si se ha realizado ó no el proceso t , de la pieza u del producto s , para la orden de producción j .
- **$Temp$** una variable encargada de contabilizar el tiempo.
- **$Prog_j$** una lista de los procesos ejecutados por el recurso j indicando para cada proceso la fecha de inicio, fecha de finalización y órdenes a las cuales pertenece.
- **Inc** un valor entero de incremento, con el cual se modifica **$Temp$** .

2.4.1 PROCEDIMIENTO UTILIZADO

A continuación, se presentará el procedimiento algorítmico. Este se encuentra dividido en 3 etapas:



2.4.1.1 Etapa I. Generación de semillas.

Esta etapa tiene por objetivo la diversificación de la búsqueda por soluciones racionales, por medio de la aplicación de procedimientos heurísticos comunes. Como resultado, esta etapa produce un conjunto de priorizaciones de productos a fabricar. Estas se hacen tanto para priorizar el conjunto de órdenes así como también los trabajos que cada una contienen.

En el caso de priorizar ordenes, se pueden realizar de manera:

- Por mayor y menor tiempo de procesamiento:
Se suma los tiempos de procesamiento de todos los productos en una orden, la orden cuya suma de tiempos sea mayor ó menor tendrá prioridad.

- Por fecha de entrega:
Se calcula el promedio de las fechas de entrega de los productos dentro de una orden de producción y tiene mayor prioridad la orden cuyo valor es menor.
- Por mayor o menor cantidad de procesos:
Se suma el número de procesos de todos los productos en una orden, la orden cuya suma sea mayor ó menor tendrá prioridad.
- Por importancia:
Se calcula el promedio de la importancia de los productos dentro de una orden de producción y tiene mayor prioridad la orden cuyo valor es mayor.

En caso de priorizar los trabajos dentro de una orden, se plantean las heurísticas SPT, EDD, LPT, $W(\text{orden})$, $W(\text{cliente})$, Máxima ó mínima cantidad de actividades para finalizar el producto.

La lógica para priorizar sería:

1. Se determina $H = \{H_1, H_2, H_3, \dots, H_k\}$. Actualmente el algoritmo trabaja con:
2. Para $k=1, 2, \dots, K$, hacemos
3. Aplicamos el procedimiento H_k a las ordenes de producción. Guardamos la priorización de las órdenes de producción y la definimos como el vector $SCHPro_k$
4. Aplicamos el procedimiento H_k a cada una de las órdenes de producción j para priorizar los productos existentes en ella. Guardamos la priorización de los productos y las almacenamos en $SCHPrd_{jk}$

Note que al final una semilla se encuentra definida por la pareja $SCHPro_k$ y $SCHPrd_{jk}$. $j=1, \dots, J$. El número total de semillas se encuentra definido por el número de órdenes de producción J y por la cantidad de heurísticas utilizadas K . La expresión para calcular éste valor, es la siguiente:

$$\text{Numero de semillas} = k * J^K$$

2.4.2 Etapa II. Construcción de planes.

Las etapas II y III, corresponden a dos algoritmos de programación, en la etapa II se da la construcción inicial, cargando en la maquinaria los trabajos según la ordenaciones establecidas en la etapa I, en la sección III, se aplica la heurística del margen de tolerancia. Ambos procedimientos funcionan como una sola herramienta para lograr la construcción de planes operativos confiables, toda semilla k generada es sometida a estas dos etapas.

1. Se identifica la orden de producción j con mayor prioridad, a la que le falte al menos un producto por ser programado.
2. Se hace $Temp = 0$, se identifica de entre OR_j , cuál es el producto s que no ha sido programado y que tiene la mayor importancia.
3. De entre la piezas de s que no han sido fabricadas y que ya puedan ser procesadas en el tiempo $Temp$ (que todas las piezas predecesoras ya hayan sido fabricadas), se escoge aleatoriamente una de ellas u . De entre los procesos de u que no han sido ejecutados y que ya puedan ser realizados en el tiempo $Temp$ (que se tengan las piezas requeridas en inventario y se hayan realizado los procesos predecesores), se escoge aleatoriamente uno de ellos t .

4. Se identifican las máquinas o recursos exigidos por RS_{tusi} en los que serán realizados los procesos. En el caso de tener varias máquinas que puedan realizar el proceso, se selecciona, de las disponibles, la que maneje una eficiencia más alta.

En caso de que los recursos no estén disponibles se hace $Temp = Temp + inc$ y se empieza nuevamente el paso 3 hasta que alguno de los recurso esté libre según $Prog_j$. El valor inc se deja a criterio del sistema, representa la mínima cantidad de tiempo que va a realizar un cambio el sistema.

5. Se actualizan los inventarios de las piezas involucradas en el punto 3. Fijémonos que S_{usj} solo se podrá actualizar cuando se hallan ejecutado todos los procesos asociados a u .
6. Se actualiza WS_{tusj} para marcar el proceso como ejecutado. (Note que esto habilita las condiciones de precedencia de otros procesos del mismo producto) y se actualiza $Prog_j$. Si todos los procesos relacionados con un producto, han sido terminados, se procede al paso 2. Si existen faltantes de procesos por programar del producto s se hace $Temp = 0$ y se va al paso 3. Si todos los productos de las orden de producción j , se va al paso 1. Si todas las órdenes han sido programadas, se calculan las medidas de desempeño, y se almacena el resultado.

2.4.3 Etapa III. Compresión de planes con el algoritmo del margen de tolerancia.

En este paso se aplicará la lógica del margen de tolerancia, fue diseñada por el ingeniero Francesco Narducci Marin, en su trabajo de tesis de maestría titulado “Programación de talleres intermitentes flexibles por medio de la heurística del margen de tolerancia”.

El margen de tolerancia, denominado MT_{ij} , se define como la capacidad que posee un trabajo j para retrasar su fecha de inicio en una máquina i . Con el

objetivo de permitir la reasignación de otros trabajos $j+k$ en la máquina i . Como resultado directo de la aplicación de MT_{ij} se obliga al programa a la eliminación de tiempos ociosos.

La búsqueda del mejor valor para MT_{ij} es el pilar fundamental del desempeño de HMT. Existe un MT_{ij} para cada una de las etapas de HMT. Lo anterior nos permite concentrar el mayor esfuerzo en la identificación de los valores MT_1 y MT_2 (Etapa 1 y 2 respectivamente), tarea que se simplifica a un procedimiento iterativo de incremento del valor MT_{ij} .

Para acotar el máximo valor permisible para MT_{ij} emplearemos la expresión

$$UC = E * \frac{MaxAcum}{N}$$

Donde:

E : Escalar de dimensionamiento

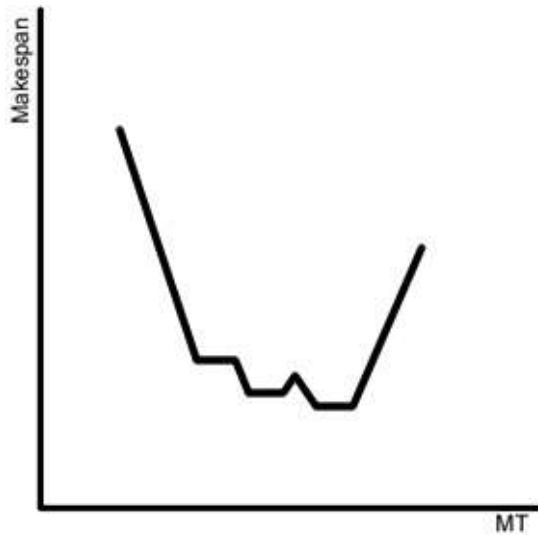
$$MaxAcum: Máx \sum_{j=1}^N P_{ij} . \forall i \leq M$$

N : Número de actividades

M : Número de estaciones

P_{ij} : Tiempo de procesamiento del trabajo j en la máquina i .

La cualidad que le permite a MT utilizar un método iterativo es su comportamiento previsible, el cual puede ser observado en la **grafica 1**. El eje horizontal representa los valores que adquiere MT y el eje vertical su lapso asociado. El aumento del valor asignado a MT produce una mejor medida de desempeño del lapso, este comportamiento continúa con pequeñas fluctuaciones crecientes y decrecientes. Finalmente MT sobrepasa de manera continuada la tolerancia de las fechas de inicio de las actividades, lo cual se ve reflejado en un incremento considerable en el lapso. Se selecciona el MT que presenta un menor lapso asociado. En caso de empate, se selecciona el menor de los MT, esto con el fin de reducir cualquier efecto del desplazamiento de las fechas de inicio de las actividades.



Grafica 1

Las iteraciones corresponden a la verificación del efecto sobre el lapso de asignar un valor particular a MT. Debido a que el número de iteraciones se encuentra definido por la cota superior de MT, es posible llegar a desarrollar o evaluar soluciones, en un número definido de ciclos computacionales. A continuación, procedemos a definir el algoritmo.

Definición del algoritmo MT

Definimos

N : Número de actividades

M : Número de estaciones

MT : cantidad de tiempo que se está dispuesto a prolongar el inicio de un trabajo.

M_i : Número de máquinas tipo i

S : Valor con el que se incrementa MT

UC : Cota superior para MT

l_{ki} : Instancia k de la estación i .

1. Obtenemos una **lista de prioridad inicial**, aplicando una heurística. Nótese que ignoramos la existencia de instancias de máquinas de tipo j
2. Definimos el valor de MT
3. inicializamos la variable **identificadora**, en un valor de 1, este contador apunta al tipo de estación en la que estamos asignando las tareas
4. Si el trabajo j indicado a ser programado por la **lista de prioridad inicial** no puede ser programado, nos dirigimos al paso 7. Si el trabajo puede ser programado, nos dirigimos al paso siguiente
5. Una vez programado el trabajo j , verificamos en orden ascendente la **lista de prioridad inicial** con el fin de determinar cuál o cuáles de los trabajos $j+k$ que serán realizados en la estación tipo i , se encuentra en capacidad de ser programado. Si $j+k$ puede ser posicionado entre la fecha de finalización del penúltimo trabajo programado $j-1$ y de la fecha de inicialización del último trabajo programado j incrementada en MT unidades, es posible redefinir el orden del programa inicial y situar el trabajo $j+k$ que está en capacidad de ser programado entre el penúltimo y el último trabajo programado j . Nótese que $j+k$ se convierte en $j-1$. Repetimos el paso 5 hasta recorrer toda la **lista de prioridad inicial**
6. Si todos los trabajos han sido reasignados, comparamos el nuevo lapso que nos entrega la **nueva lista de prioridad** con el lapso de la **lista de prioridad inicial**. Si el lapso de la lista nueva es menor que el de la lista inicial, reemplazamos la lista inicial con la nueva y nos dirigimos al paso 3, si no establecemos la **lista de prioridad inicial** como la de mejor rendimiento para el valor dado de MT , damos por terminado el procedimiento
7. Hacemos **identificador = identificador + 1** Si $\text{identificador} \leq M$ nos dirigimos al paso 4, sino hacemos **identificador = 1** y nos dirigimos al paso 4

El anterior procedimiento tan solo genera la mejor lista de prioridad para un valor dado de MT . De igual forma es posible obtener un total de $\frac{UC}{s}$ listas de prioridades de entre las cuales se selecciona el MT de la que presente un menor lapso.

Con la determinación de la lista de prioridad, podemos enfocarnos en la asignación de tareas en las instancias de las estaciones. HMT puede ser aplicado a esta asignación mediante el siguiente algoritmo.

1. Definimos el valor de MT
2. Inicializamos la variable **identificador**, en un valor de 1, este contador apunta al tipo de estación en la que estamos asignando las tareas
3. Identificamos de entre las k instancias de la estación apuntada por **identificador**, cuál es la que posee mayor disponibilidad y la denominamos I
4. Si el trabajo j indicado a ser programado por la **lista de prioridad inicial** no puede ser programado en la instancia apuntada por **identificador**, nos dirigimos al paso 7. Si el trabajo puede ser programado, nos dirigimos al paso siguiente
5. Una vez programado el trabajo j en la instancia I , verificamos en orden ascendente la **lista de prioridad inicial** con el fin de determinar cuál o cuáles de los trabajos $j+k$ que serán realizados en la estación tipo i instancia I , se encuentra en capacidad de ser programado. Si $j+k$ puede ser posicionado entre la fecha de finalización del penúltimo trabajo programado $j-1$ y de la fecha de inicialización del último trabajo programado j incrementada en MT unidades, es posible redefinir el orden del programa inicial y situar el trabajo $j+k$ que está en capacidad de ser programado entre el penúltimo y el último trabajo programado j . Note que $j+k$ se convierte en $j-1$. Repetimos el paso 5 hasta recorrer toda la **lista de prioridad inicial**

6. Si todos los trabajos han sido reasignados, definimos como el **programa final la asignación** recién lograda y damos por terminado el procedimiento
7. Hacemos **identificador = identificador+1**. Si el valor del identificador es $\leq M$ nos dirigimos al paso 4, si no **identificador = 1** y nos dirigimos al paso 3

De una manera similar a la etapa 1 es posible evaluar el desempeño de la etapa 2, con base en la utilización de diversos valores para **MT**.

Si el objetivo consiste en utilizar HMT como un método de optimización, posterior a la aplicación de un procedimiento de búsqueda local, sólo sería necesaria la aplicación de la segunda etapa sobre una solución generada por cualquier otro método. Nótese que en el peor de los casos HMT garantiza al menos una solución activa.

El número de ciclos computacionales requeridos NCCR, se encuentra definido por la siguiente expresión:

$$NCCR = 2 * M * \left(\frac{\left(\frac{E * \left(\frac{\text{Máx} \sum_{j=1}^N P_{ij}, \forall i \leq M}{N} \right)}{S} \right) * (N^2 * M)}{S} \right) + (N^2 * M)$$

2.4.4 Consideración de programación reactiva, por cambio en el estado del sistema.

El algoritmo anteriormente descrito ha sido desarrollado para combatir la complejidad derivada del problema de reprogramar las actividades, cada vez que sucedan eventos que modifican el estado del sistema. Estos eventos pueden ser categorizados en algunos de estos tipos:

- Ingreso de una nueva orden de pedido.
- Alteración de los estados de las órdenes de pedido.
- Cambios imprevistos en los estados de los recursos.
- Falla en las proyecciones de reabastecimientos de materia prima.

En cada uno de los casos planteados, la solución que se aplica se basa en la siguiente metodología.

1. Identificación del o de los factores que alteran el estado del sistema.
2. Actualización del estado del sistema, lo cual incluye identificación del estado real de progreso de los procesos a la fecha y estado real de los niveles de inventario.
3. Reformulación del programa operativo, por medio del procedimiento expuesto para asignación de recursos.

Esta problemática requiere la construcción de una estructura informática e interface que sea capaz de manejar el flujo de información y las actividades derivadas de la reprogramación.

2.5 MODELADO DEL SISTEMA

Por último, para observar y analizar todos los componentes que abarcará el modelo resulta de gran ayuda un **modelado del sistema**, éste consiste en poner todos los requisitos del cliente y establecer formas de llevarlos a cabo.

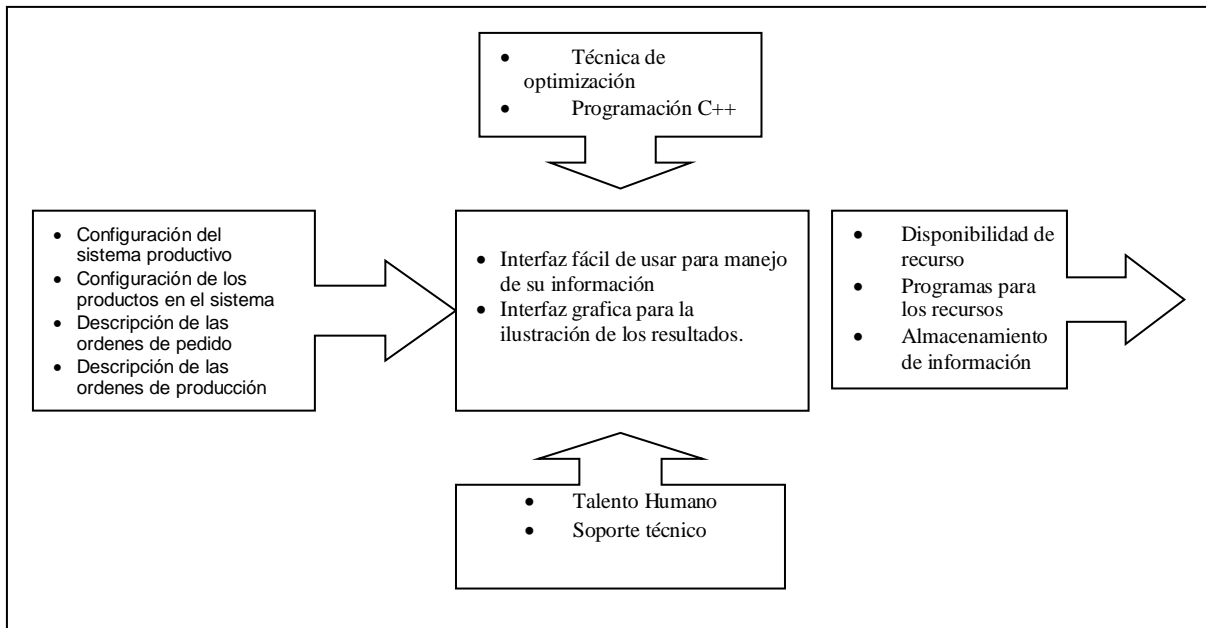
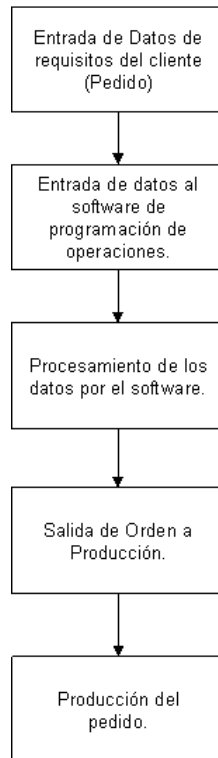


Ilustración 1 MODELADO DEL SISTEMA

Por último es necesario verificar la validez de los **requisitos y gestionarlos**, es decir, observar si realmente la empresa carece de los requisitos aquí planteados o ya los posee, de igual manera conseguirlos y plasmarlos en el modelo.

2.6 MODELADO FUNCIONAL Y FLUJO DE INFORMACIÓN

El modelado funcional de la información se esquematiza de tal manera que primero se tenga en cuenta la información obtenida de los clientes de la empresa, como son los pedidos que éstos realizan, luego los datos son procesados por el modelo de programación de operaciones y, por último, el software genera un programa de producción que podría ponerse en marcha en el sistema de producción.



Grafica 10. FLUJO DE INFORMACIÓN

Fuente: Elaboración Propia

CAPITULO 3

ANALISIS DE RESULTADOS

El algoritmo propuesto fue codificado utilizando el lenguaje de programación C++ bajo la herramienta C++ builder, con una base de datos de Microsoft Access. En éste documento se plantea la comparación con otros modelos, debido a varias situaciones las cuales son:

- **Dificultad de generar un modelo matemático asociado.**
- **Falta de software con que considere toda la complejidad del problema.**
- **Ausencia de datos históricos para la comparación.**

A continuación se desarrollara una breve descripción de cada uno de estos problemas.

1. Dificultad de generar un modelo matemático asociado.

Se encontraron diversos modelos matemáticos, ninguno de estos fue lo suficientemente completo ó robusto para poder realizar una comparación detallad. En muchos casos los modelos omitían los diferentes problemas asociados como son los del tipo de ensamblaje y de reprocesamiento, buscaban sólo una solución, no tenían en cuenta las restricciones que maneja el modelo planteado en esta investigación, como son de jornadas laborales y reactividad entre otras.

Algunos de los modelos encontrados que se pueden considerar más cercanos al problema planteado son:

- Planteado por. M.C. GOMES, A.P. BARBOSA-PO´ VO y A.Q. NOVAIS. En el artículo titulado “Optimal scheduling for flexible job shop operation” publicado en International Journal of Production Research. En él se plantea un modelo matemático capaz de generar un óptimo para un sistema tipo Job shop flexible con reproceso, pero no toma en cuenta la parte de ensamble.
- Planteado por: Stephen C. Graves; Bruce W. Lamar en el artículo titulado, “An Integer Programming Procedure for Assembly System Design Problems”, plantean un modelo para ensamble pero no cuenta con la parte de la estructura tipo Job shop flexible.
- Planteado por: Deepu Philip, en el paper titulado “Scheduling Reentrant Flexible Job Shops With Sequence Dependent Setup Times” tampoco cuenta con la parte de ensamble.

2. Falta de software del mismo tipo para la comparación.

En muchos casos no se pudo encontrar una versión de un software que maneje las mismas características para poder realizar una comparación en detalle con nuestro modelo. Se cree que en Colombia no se ha generado ninguna

herramienta con las capacidades de manejar una solución a los problemas que planteamos. Ésta fue la idea principal por cual se generó este proyecto, para este caso se comparara con el software lekin® el cual está disponible web, éste ofrece la posibilidad de programar operaciones basado en distintas heurística.

3. Ausencia de datos históricos para la comparación.

En ninguna de las dos empresas donde se está implementando la herramienta, COTECMAR y SUPER BRIX, se cuenta con datos históricos para poder realizar posibles comparaciones de los obtenidos en la realidad con los que arrojaría el modelo. Ésta es la razón para implementar esta herramienta. Sin embargo, por conversaciones obtenidas con personal de estas empresas, se cree que ésta es un buen método para evaluar los estados en los cuales ellos se encuentran y formular posibles opciones de mejora, viendo los resultados históricos a través de la plataforma.

3.1 RESULTADOS OBTENIDOS

3.1.1 Comparación con la realidad

Para poder demostrar el poder que alcanza a tener el modelo ya codificado, se le ilustrará un ejemplo en el cual se muestran las ventajas que se tiene al momento de realizar las programaciones utilizando este método de optimización planteado en esta tesis, se puede resolver cuales son las mejoras obtenidas.

El siguiente es un ejemplo basado en la reparación de un buque de carga. A continuación se muestra un diagrama de PERT, en el cual se visualizan todos los procesos a seguir para dicha reparación.

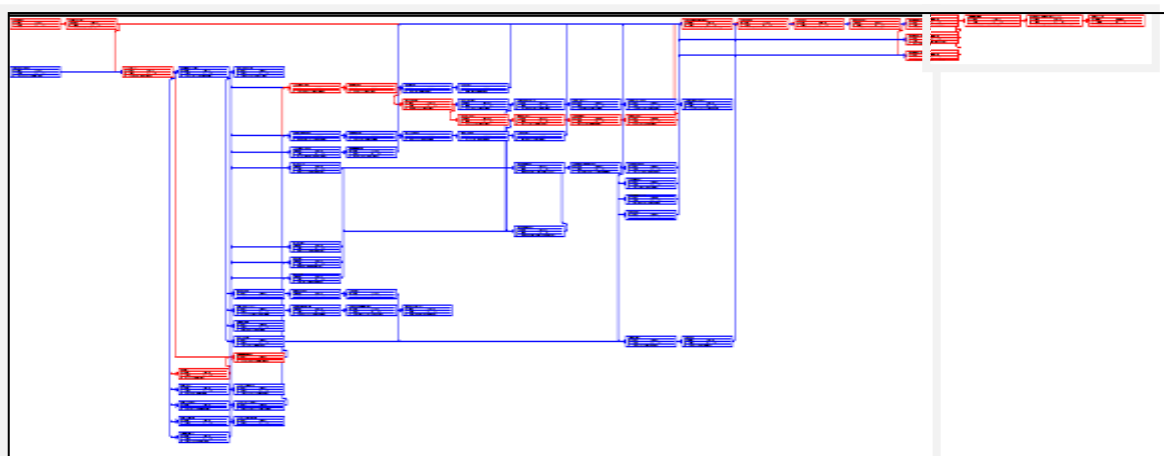


Figura 7: Gráfica de programación para reparación de Buque

Para el desarrollo de este ejercicio se contó con sesenta y nueve procesos diferentes los cuales pueden tener tiempos y ó recursos totalmente diferentes en cada uno, viendo en rojo los pasos críticos.

Para cada uno de estos procesos se asoció a uno ó más recursos y/ó operarios relacionados a esta labor, donde estos los pueden realizar a manera de opción, es decir si se tiene dos máquinas (A y B) para el desarrollo del proceso donde requiere una sola de las dos, la manera de programarlo es seleccionar uno solo de acuerdo a la conveniencia.

De manera similar, cada operario y/ó maquina posee un horario laboral en el cual él puede trabajar, un trabajo no se puede desarrollar si no se ha cumplido con su predecesor si existiese el caso. Además, los procesos pueden requerir material que fue creado en proceso anterior, es decir, una precedencia.

Esta reparación del buque fue un proyecto real, el cual sucedió el año anterior. Una vez almacenados todos los datos que se presentaron en el capítulo anterior en la herramienta, se obtuvo el mismo resultado que ellos tenían, valga aclarar que para éste proyecto se contaba con la disponibilidad de todos los recursos, por lo cual el tiempo de ejecución iba a ser el mismo.

La respuesta obtenida fue validada por expertos en el proceso y personas que estuvieron en el desarrollo del proyecto, lo que soporta la idea de que los resultados ó salidas generadas por el modelo son confiables.

3.1.2 Comparación Con Otros Modelos

Se realizaron comparaciones de resultados con el software LEKIN, el cual genera soluciones meta heurísticas a los problemas. LEKIN es uno de los software de libre distribución más usados, en donde el máximo nivel de complejidad que puede manejar son los problemas tipo, Job-shop flexible, omitiendo muchas de las características que tiene el modelo planteado en este trabajo.

Las pruebas, se realizaron tomando en cuenta varios escenarios aleatorios de producción en un sistema tipo Job Shop Flexible, manejando una cantidad de 2, 5, 10 y 20 estaciones, en los cuales se procesa entre 10 y 20 trabajos.

Para el cálculo de la cantidad de máquinas por estación se manejó un valor aleatorio que va de 1 a 5 máquinas en cada estación, para el tiempo de procesamiento se manejó, de manera análoga, un valor que iba de 5 a 30 unidades de tiempo, para la importancia de la orden se empleó un número aleatorio que iba de la siguientes forma:

$$T_p = \text{Int}((50 + \text{SUM}(P_i)) * \text{Rnd} + (\text{SUM}(P_i) + 20))$$

Esto representa: la suma de los tiempos de procesamiento de cada uno de los trabajos más cincuenta, multiplicado por un número aleatorio, más la suma de los tiempos de procesamiento, más veinte unidades de tiempo más.

Para obtener los resultados se corrió la heurística planteada en éste documento (heurística de la semilla) como si fuese una sola orden de producción por cada trabajo, incluyendo como semillas las heurísticas: SPT, EDD, LPT, las demás heurísticas expuestas, (W(orden), W(cliente), Máxima ó mínima cantidad de actividades para finalizar el producto) tendrían el mismo peso en los trabajos y no afectaría el resultado.

Para el caso de LEKIN, se corrieron los mismos escenarios trabajando con las heurísticas (EDD, LPT, SPT, y utilizando shifting bottleneck aplicado a los objetivos de (CMAX, SUM C, WT).

De la heurística de la semilla y los resultados arrojados por cada uno de las heurísticas de LEKIN solo se almacenó el mejor resultado de cada una de los posibles objetivos a tener. Los resultados obtenidos fueron tabulados y mostrados a continuación:

# Trabajos	# Estaciones	MAKESPAN		MAX TAR		Num Trab Tar		FLUJO T		TARDANZA T	
		H Semilla	Lekin	H Semilla	Lekin	H Semilla	Lekin	H Semilla	Lekin	H Semilla	Lekin
2	5	53	53	0	0	0	0	198	198	0	0
2	10	174	174	41	66	4	3	890	880	84	84
5	5	117	117	0	0	0	0	505	505	0	0
5	10	215	215	0	0	0	0	1627	1627	0	0
10	5	222	222	0	0	0	0	979	979	0	0
10	10	250	250	0	0	0	0	2104	2137	0	0
20	5	379	379	165	232	9	10	4250	4631	751	1184
20	10	377	377	34	34	5	5	5531	5531	68	68

Tabla 1 Resultados Obtenidos

En la tabla anterior pudimos observar la diferencia que existe entre resolver utilizando la heurística de la semilla completa versus los resultados de LEKIN. En el caso de makespan, los dos arrojan el mismo resultado. En el caso de la máxima tardanza ganó la heurística de la semilla por 2 de los casos. En el caso de los trabajos tardíos y flujo hay un empate mientras que en la tardanza gana la heurística de la semilla.

CONCLUSIONES

Hemos llegado a diversas conclusiones con respecto a los diferentes temas que se manejaron a lo largo de este proyecto. Se puede resaltar que la tecnología de apoyo es algo muy importante para el progreso de cualquier sector empresarial. Esto se alcanza a observar claramente en el momento de realizar la programación de los trabajos (Scheduling) en donde, para programar las actividades, hay una gran cantidad de posibles soluciones que se pueden tomar en cuenta, pero encontrar la mejor es algo que lleva mucho análisis. De esto se pueden encargar diversas herramientas, como es el caso de los software de apoyo en la toma de decisiones, mejor conocidas por sus nombre en inglés, “decisión support systems (DSS)”, cuya función es buscar una solución buena, si no es la óptima, a los distintos problemas que día a día se presentan.

El mundo progresa de la misma manera como progresa nuestro conocimiento y nuestra tecnología, que cada día hacen la vida más fácil y mejor. Las empresas de todo el mundo están entrando en razón, al momento de aprobar la consecución de software de apoyo y tecnología apropiada para solucionar cada uno de sus problemas. Se observa cómo desde los grandes eventos a nivel mundial hasta los más pequeños, necesitan de esta tecnología y en cada momento dejan de preocuparse por problemas que éstos solucionan y se enfocan en otros problemas difíciles de solucionar.

Tanto SUPER BRIX, como COTECMAR son empresas regionales de talla internacional, que han implementado nuevas tecnologías desde la automatización de algunos de sus procesos de producción, hasta gestionar para la compra de software, programación y control de la producción. Estas empresas, en los últimos años, han ido expandiendo su mercado. Problemas como la incertidumbre de la demanda y la no organización adecuada de su información, hacen que éstas recurran a este tipo de software de apoyo. Los software facilitan a las empresas la labor de cumplir con cada uno de los objetivos propuestos por ésta, así como cumplir con las estrategias corporativas.

A través de este modelo en conjunto con la aplicación que fue desarrollada, se puede realizar programas de producción en sistemas de fabricación y ensamble, que faciliten las labores diarias y de rutina en una empresa, mejorando sus procesos y haciéndolos cada vez más sencillos buscando diferentes tipos de objetivos.

Esto da pie a cuestionar por qué las empresas no poseen estos sistemas de apoyo. Se cuestiona también si no es cierto, ó no sería mejor acción dejar trabajos rutinarios a las máquinas y preocuparnos por la toma de decisiones en acciones inesperadas que se presentan en el día a día, por esto es conveniente tener el apoyo de la tecnología, aprender a usarla a nuestro favor, para de esta forma mejorar nuestras herramientas y realizar una buena toma de decisiones.

Algunos de los beneficios que cabe destacar dentro de un adecuado plan de producción es el mayor control de las variables de producción, como son la materia prima, inventario, mano de obra, entre otras, permitiendo ofrecer a esta compañía mejores productos y servicios a menores precios, ya que no se incurrirán en costos por mala administración.

RECOMENDACIONES

Al concluir este proyecto se puede recomendar a la empresa apoyarse en recursos tecnológicos, como son los software de apoyo para la solución de los problemas. También se recomienda a las empresas de cualquier tipo, adquirir un sistema de pronósticos de la demanda para disminuir un poco la incertidumbre al momento de producir, debido a que ésta afecta de gran forma la manera cómo se va a programar la producción y si estos pronósticos son bastante alejados de la realidad, puede incurrir en gran cantidad de gastos derivados de las variables de producción.

Es indispensable en un futuro que la empresa adquiera un sistema de información que permita conocer más a fondo los requerimientos del cliente, así como también un sistema de planeación de la producción que conduzca a un programa que se acerque al mejor resultado, para que la empresa esté mejor preparada frente a las incertidumbres del día a día.

FUTURAS INVESTIGACIONES

El resultado de este trabajo abre diferentes caminos, los cuales pueden ampliar las investigaciones al momento de realizar programación de operaciones y toma de decisiones, éstas podrían ser:

- Manejar la posibilidad de manejar opciones de construcción, es decir qué pasaría si el usuario tuviera varias opciones de materia prima al momento de realizar un producto.

- Tener en cuenta restricciones de bloqueo de producción para manejar posibles restricciones.

BIBLIOGRAFIA

- [1] Ching-Jong Liao; Chii-Tsuen You. An Improved Formulation for the Job-Shop Scheduling Problem. The Journal of the Operational Research Society, Vol. 43, No. 11. (Nov., 1992), pp. 1047-1054.
- [2] Deepu Philip. Scheduling reentrant flexible job shops with sequence dependent setup times. Tesis de Maestria en ingeniería industrial
- [3] Dorf, R Sistemas Modernos de Control, adison-wesley. Iberoamericana Segunda edición, 1984.
- [4] Ballestín G, Francisco; nuevos métodos de resolución del problema de secuenciación de proyectos con recursos limitados. Tesis doctoral Universidad de Valencia. 2002
- [5] Esther Álvarez. Francisco Zubillaga. Análisis Comparativo de Métodos de Secuenciación para la Programación Dinámica de Operaciones. VIII Congreso de Ingeniería de Organización. Leganés, 9 y 10 de septiembre de 2004.
- [6] M.c. Gomes, A.p. Barbosa-póvo y A.q. Novais. Optimal scheduling for flexible job shop operation paper publicado en International Journal of Production Research.
- [7] Morton, T. Pentico W. Heuristic scheduling systems: with applications to productions systems and project management. Jonh Wiley and sons. 1993.
- [8] Narducci Marin Francesco. Programación de talleres intermitentes flexibles por medio de la heurística del margen de tolerancia. Tesis de maestria en ingeniería industrial 2005.
- [9] Pinedo M. Chao X. Operation of Scheduling with applications in manufacturing and services. Irwing- Mc Graw Hill. 1999.
- [10] Sipper, D. Y Bulfin R. Planeación y Control de la Producción. 1998 Mc Graw Hill.
- [11] Stephen C. Graves; Bruce W. Lamar en el paper titulado, "An Integer Programming Procedure for Assembly System Design Problems" Operations Research, Vol. 31, No. 3. (May - Jun., 1983), pp. 522-545.
- [12] CAMPBELL, Kevin. Bret, Robert. FedEx Generates Bid Lines Using Simulated Annealing. Revista INTERFACES Volume 27, Number 2,

Mar/Apr 1997 pp:1-16 disponible en internet:
<<http://www.interfaces.smeal.psu.edu/pdf/v27n2a1.pdf>>

- [13] WEINTRAUB, Andres. EPSTEIN, Rafael. A Truck Scheduling System Improves Efficiency in the Forest Industries. Revista INTERFACES Volume 26, Number 4, Jul/Aug 1996 pp:1-12 disponible en internet:
<<http://www.interfaces.smeal.psu.edu/pdf/v26n4a1.pdf>>

- [14] HILLIARD, Michael. LIU, Cheng. BUSCH, Ingrid. HARRISON, Glen. Scheduling the Operation Desert Storm Airlift: An Advanced Automated Scheduling Support System. Revista INTERFACES Volume 22, Number 1, Jan/Feb 1992 pp:131-146. disponible en internet.
<<http://www.interfaces.smeal.psu.edu/pdf/v22n1a7.pdf>>

- [15] PRISTSKER, A.A.B. HAAP w. Multiproject scheduling with limited resources: a Zero-One programming approach, Management Science 16. P 93-16

- [16] Slowinski, R. Multiobjective project Scheduling under multiple category resource Constrains, Advance in project scheduling , Elsevier 1989. P 151-167