

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**



**FACULTAD DE CIENCIAS NATURALES Y MATEMÁTICAS  
DEPARTAMENTO DE MATEMÁTICAS**

**PROYECTO DE GRADUACIÓN**

**PREVIO A LA OBTENCIÓN DEL TÍTULO DE:  
“MAGÍSTER EN CONTROL DE OPERACIONES Y GESTIÓN LOGÍSTICA”**

**TEMA**

**ANÁLISIS Y PROPUESTA DE MEJORA DE LA PROGRAMACIÓN DE LA  
PRODUCCIÓN DE UNA LÍNEA DE PELETIZACIÓN DE ALIMENTO  
BALANCEADO MULTIESPECIE**

**AUTOR**

**LEO ALEXANDER TRUJILLO COLOMA**

**Guayaquil – Ecuador**

**AÑO**

**2016**

# DEDICATORIA

A DIOS,  
A MI ESPOSA,  
A MIS PADRES Y HERMANOS,  
A MIS AMIGOS Y FAMILIARES.

# AGRADECIMIENTO

A mi querida esposa: Alexandra, mi compañera de vida, por su cariño y apoyo incondicional durante todo este tiempo, sin ella, simplemente esto no hubiera sido posible.

A mis padres, hermanos y buenos amigos, mi eterno cariño y gratitud.

A mis profesores de la ESPOL, por su trasmisión del conocimiento sin egoísmo durante el desarrollo de esta maestría.

Al Ph.D. Fernando Sandoya, por su guía y consejo en la realización de esta tesis.

A mi querido colegio San José De La Salle, que en sus aulas aprendimos que con esfuerzo, lo imposible, solo toma un poco más de tiempo.

## DECLARACIÓN EXPRESA

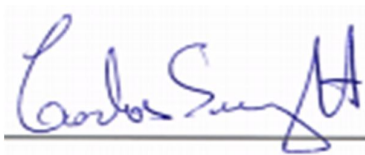
La responsabilidad por los hechos y doctrinas expuestas en este Proyecto de Graduación, me corresponde exclusivamente; el patrimonio intelectual del mismo, corresponde exclusivamente a la Facultad de Ciencias Naturales y Matemáticas, Departamento de Matemáticas de la Escuela Superior Politécnica del Litoral.



---

Ing. Leo Trujillo Coloma

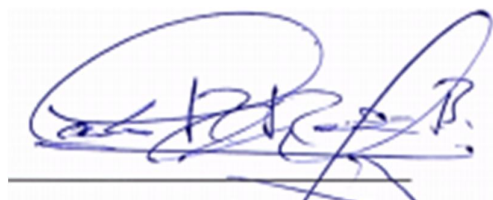
# TRIBUNAL DE GRADUACIÓN



Mg. Carlos Anibal Suarez  
**PRESIDENTE DEL TRIBUNAL**



Ph.D. Fernando Sandoya Sánchez  
**DIRECTOR DEL PROYECTO**



M.Sc. Carlos Martín Barreiro  
**VOCAL DEL TRIBUNAL**

# TABLA DE CONTENIDO

	Pág.
<b>CONTENIDO DE FIGURAS</b> .....	VII
<b>CONTENIDO DE TABLAS</b> .....	IX
<b>CAPÍTULO 1</b>	
1. Introducción.....	1
1.1 El paradigma de la producción.....	1
1.2 El rol e importancia de la programación de la producción.....	1
1.3 La función de la programación de la producción dentro de una empresa.....	3
1.4 Objetivos generales de la tesis.....	4
1.5 Objetivos específicos de la tesis.....	4
<b>CAPÍTULO 2</b>	
2. Marco teórico: el problema de la programación de la producción.....	5
2.1 Antecedentes.....	5
2.2 Descripción y complejidad del problema de la programación de la producción.....	7
2.2.1 Descripción del problema.....	7
2.2.2 Complejidad del problema.....	8
2.3 Notación y fundamento teórico.....	9
2.3.1 Campo $\alpha$ : Ambientes o entornos de la máquina.....	10
2.3.2 Campo $\beta$ : Características de procesamiento.....	12
2.3.3 Campo $\gamma$ : Objetivo a ser optimizado.....	14
2.4 Métodos heurísticos para resolver el problema de la programación de la producción.....	16
2.4.1 Heurísticas: Reglas básicas de programación.....	17
2.4.2 Heurísticas: Reglas compuestas de programación.....	20
2.5 LEKIN: Un sistema para la programación de la producción.....	26

### **CAPÍTULO 3**

3. Situación actual: el proceso de producción de una línea de peletización multiespecie.....	34
3.1 Antecedentes de la compañía y fábrica en estudio.....	34
3.2 Tipos de líneas de producción y selección para el análisis de la mejora de la programación.....	35
3.3 Descripción del proceso de producción de alimento peletizado.....	39
3.4 Situación actual del proceso de programación de la producción de la línea.....	41

### **CAPÍTULO 4**

4. Propuesta de mejora de la programación de la línea de producción.....	46
4.1 Programación actual de la línea de producción y sus resultados.....	46
4.2 Propuesta de un programa de producción con el uso del software LEKIN.....	51

### **CAPÍTULO 5**

5. Conclusiones y Recomendaciones.....	58
5.1 Conclusiones.....	58
5.2 Recomendaciones.....	60

<b>BIBLIOGRAFÍA.....</b>	<b>61</b>
--------------------------	-----------

<b>ANEXOS.....</b>	<b>62</b>
--------------------	-----------

## CONTENIDO DE FIGURAS

	<b>Pág.</b>
Figura 2.1	Operación estándar en la fundidora.....5
Figura 2.2	LEKIN: Flexible Job Shop Scheduling System.....28
Figura 2.3	Vista menú principal LEKIN.....28
Figura 2.4	Abriendo archivos LEKIN existentes.....29
Figura 2.5	Cuadro de diálogo #1.....30
Figura 2.6	Cuadro de diálogo #2.....30
Figura 2.7	Cuadro de diálogo #3.....31
Figura 2.8	Ventanas principales en LEKIN.....32
Figura 2.9	Comparación de resultados y objetivos.....33
Figura 3.1	Producción anual: fábrica alimentos balanceados.....34
Figura 3.2	División Salud Animal (SA).....36
Figura 3.3	División Acuicultura (AC).....36
Figura 3.4	División Consumo (CO).....36
Figura 3.5	Producción mensual promedio por división comercial.....37
Figura 3.6	Capacidad vs Producción mensual por división comercial.....37
Figura 3.7	% Uso capacidad de producción mes por división comercial.....38
Figura 3.8	Diagrama del proceso de producción de la línea SA.....39
Figura 3.9	Resumen de las 3 reglas empíricas actuales para la programación de la producción.....45
Figura 4.1	Parametrización en LEKIN de las máquinas.....46
Figura 4.2	Configuración en LEKIN de los trabajos.....47
Figura 4.3	Ingreso manual en LEKIN de la programación actual.....48
Figura 4.4	Secuencias de las máquinas del programa actual.....48
Figura 4.5	Diagrama de Gantt del programa actual.....49
Figura 4.6	Diagrama de resultados obtenidas del programa actual.....49
Figura 4.7	Resumen del desempeño del programa actual.....50
Figura 4.8	Vista de las heurísticas de programación en LEKIN.....51
Figura 4.9	Resultados de las propuestas de programas de producción.....52



Figura 4.10	Vista de la secuencia del programa elegido.....	53
Figura 4.11	Diagrama de Gantt del programa de producción elegido.....	54
Figura 4.12	Vista diagrama de los resultados del programa de producción propuesto.....	55
Figura 4.13	Diagrama de resultados programa propuesto vs actual.....	56

## CONTENIDO DE TABLAS

	<b>Pág.</b>
Tabla 2.1	Sistemas de programación comerciales y académicos.....27
Tabla 3.1	Listado de productos manufacturados en la línea SA.....42
Tabla 3.2	Listado de ratios de producción por proceso y producto.....43
Tabla 3.3	Listado de tiempos de operación por proceso y producto.....44
Tabla 4.1	Resultados del programa actual de producción.....50
Tabla 4.2	Matriz de ponderación de resultados de las propuestas.....52
Tabla 4.3	Selección de la mejor propuesta de programa de producción.....53
Tabla 4.4	Resultados del programa propuesto de producción.....55
Tabla 4.5	Comparación del programa actual vs programa propuesto.....56
Tabla 4.6	Estimación de los ingresos adicionales por el programa de producción propuesto.....57

# CAPÍTULO I

## INTRODUCCIÓN

### 1.1 El paradigma de la producción

Los sistemas actuales de producción en la sociedad moderna son sobresalientes. Estos sistemas forman la base para construir y mejorar la fortaleza y la vitalidad económica de un país. La tarea de desarrollar y operar los sistemas de producción crece en complejidad. Los cambios importantes en los productos, los procesos, las tecnologías de gestión, los conceptos y la cultura, dan como resultado: retos y necesidades cada vez mayores. Uno de los objetivos generales de esta tesis es ayudar al logro de estos retos de forma técnica y no de forma empírica, llamada a menudo: "experiencia".

Ecuador no es la excepción, en un mundo globalizado y cada vez más competitivo y desafiante entre las empresas de manufactura, las reglas o metodologías para programar la producción son cada vez más técnicos y menos empíricos o simplemente, basados en la experiencia y criterios de las personas asignadas a este rol.

### 1.2 El rol e importancia de la programación de la producción

La programación es un proceso de toma de decisiones que se utiliza de forma regular en muchas empresas manufactureras y de servicios también. Para el caso de este proyecto de tesis, se realizó en una empresa manufacturera ubicada en la ciudad de Guayaquil.

La programación es un proceso de asignación de recursos a las tareas durante períodos de tiempo determinados y su finalidad es optimizar uno o más objetivos establecidos previamente. Los recursos y las tareas de una organización pueden tomar muchas formas diferentes en función del tipo de organización.

Los recursos pueden ser: máquinas en un fábrica, pistas de aterrizaje en un aeropuerto, tripulaciones en un proyecto de construcción, las unidades de procesamiento en un entorno de computación, y así sucesivamente.

Las tareas pueden ser: operaciones en un proceso de producción, los despegues y aterrizajes en un aeropuerto, las etapas de un proyecto de construcción, la ejecución de programas en un ordenador, y así sucesivamente. Cada tarea puede tener un cierto nivel de prioridad o importancia, un tiempo de liberación o inicio y una fecha de vencimiento o entrega. Los objetivos también pueden tomar muchas formas diferentes: un objetivo puede ser la minimización del tiempo de finalización de la última tarea (el caso más común dada su importancia) y otra puede ser la minimización de la cantidad de tareas completadas después de sus respectivas fechas de vencimiento o entrega.

La programación de la producción, como un proceso de toma de decisiones, juega un rol importante dentro de la planeación de la cadena de suministro dentro de las organizaciones, especialmente en las empresas de manufactura.

La programación empezó a ser tomada en cuenta seriamente en las empresas de manufactura a inicios del siglo XX con el trabajo de Henry Gantt y otros pioneros de la Ingeniería Industrial. Sin embargo, pasó mucho tiempo para que saliera la primera publicación de programación en la literatura de Investigación de Operaciones e Ingeniería Industrial. Algunas de las primeras publicaciones aparecieron en la Investigación Naval Logística Trimestral a inicios de los años 1950 y contenía los resultados de W.E. Smith, S.M. Johnson y J.R. Jackson. Durante los años 1960 una cantidad significativa de trabajos fueron realizados en programación dinámica y programación entera para los problemas de programación. Luego en 1970 las investigaciones se enfocaban principalmente en la complejidad jerárquica de los problemas de programación.

En los años 1980 una gran cantidad de diferentes direcciones fueron tomadas en la academia e industria con un aumento en los estudios del problema de programación estocástico. Finalmente, con el surgimiento de las computadoras,

surgieron también los sistemas de programación de la producción, los mismos que actualmente siguen siendo utilizados.

### **1.3 La función de la programación de la producción dentro de una empresa**

La función de la programación en un sistema de producción o de servicio interactúa con muchas otras funciones. Estas interacciones son sistema-dependientes y pueden diferir sustancialmente de una situación a otra. A menudo, esto se realiza dentro de un sistema de información global de la empresa (ERP). Una fábrica moderna u organización de servicio, tiene un sistema de información elaborado que incluye una computadora central, bases de datos, computadoras personales con redes de área local, estaciones de trabajo y terminales de entrada de datos, los cuales están conectados a esta computador central, utilizados para extraer o recuperar datos de la base de datos o introducir nuevos datos. El software que controla tan elaborado sistema de información se denomina típicamente como un sistema de planificación de recursos empresariales (ERP por sus siglas en inglés: Enterprise Resource Planning). Existen un número de compañías de software que se especializa en el desarrollo de tales sistemas, por ejemplo: SAP R/3, JD Edwards y PeopleSoft. Dicho sistemas ERP juega un rol de autopista de información que atraviesa la empresa, en todos los niveles organizacionales, enlazando los sistemas de soporte de decisión.

La programación es frecuentemente realizada de forma interactiva con un sistema de apoyo de decisión instalado en la computadora personal y enlazada también al sistema ERP de la empresa. Terminales en lugares clave conectados al sistema ERP puede dar, a todos los departamentos, toda la información de la programación actual. Estos departamentos, a su vez, pueden proporcionar al sistema de programación con información actualizada.

Por supuesto, hay empresas en donde la comunicación entre la función de la programación de la producción y las otras entidades hacedoras de decisión

ocurre en reuniones o vía correo electrónico. La empresa en donde se realizó este proyecto de tesis, tiene esta configuración.

#### **1.4 Objetivos generales de la tesis**

Este proyecto de tesis, tiene como objetivo general, proponer un mejor programa de producción para una de las líneas de producción de la empresa en estudio con respecto a su situación actual, para definir dicha propuesta, se basará en el marco teórico y con el soporte de un software de programación de producción, llamado LEKIN, que se verá más adelante, su funcionamiento y características en detalle.

Se espera que la propuesta ayude a subir la capacidad de producción, por lo menos, en un 5% utilizando los mismos recursos humanos, maquinaria, tiempo y que ayude a cambiar los paradigmas que se tienen al momento de elaborar el programa de producción en la línea de alimento balanceado que rige en estos momentos.

#### **1.5 Objetivos específicos de la tesis**

Del mismo modo, este proyecto de tesis, espera cumplir los siguientes objetivos específicos, los mismos que se enumeran a continuación:

- Romper los paradigmas de las actuales reglas de programación que tiene la fábrica en estudio.
- Subir la productividad de la línea de producción.
- Mejorar el nivel de servicio de la fábrica.
- Implementar el software de programación de producción dentro de la empresa en estudio como parte de los procesos que tienen dentro de su planificación de la cadena de abastecimiento.

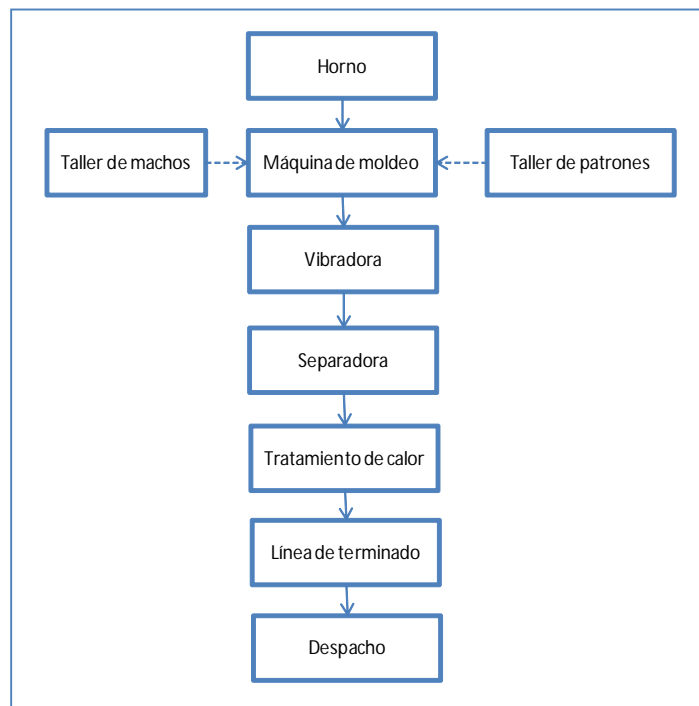
## CAPÍTULO II

### MARCO TEÓRICO: EL PROBLEMA DE LA PROGRAMACIÓN DE LA PRODUCCIÓN

#### 2.1 Antecedentes

Antes de iniciar con el marco teórico, se considera necesario empezar con un ejemplo introductorio del problema de la programación de la producción.

Fundidora SA produce diferentes moldes de metal para el sector automotriz. El proceso de producción es sencillo: es un fundido de arena, con partes moldeadas en máquinas automáticas. El proceso de producción de una fundidora usualmente consiste en las siguientes operaciones que se muestran en la Figura 2.1



**Figura 2.1 Operación estándar en la fundidora**

Fuente: Planeación Producción (Sipper Daniel) - Elaboración: Propia

Un horno proporciona metal líquido a una máquina de moldeo. La máquina usa un patrón para hacer los moldes de manera automática, a partir de una o más partes. Cuando el metal fundido entra a la máquina, fluye a través de

compuertas y rampas para llenar el molde y formar una parte. Si la parte tiene un espacio hueco en su interior, se colocan machos en los moldes antes de llenarlos. Los machos se hacen de arena u otro material en el taller de machos. La máquina de moldeo se abre y expulsa partes y arena, después repite el proceso. La arena y las partes moldeadas se colocan en un vibrador que quita la arena para usarla de nuevo y, también, separa muchas partes de las compuertas y rampas. Si la parte sigue pegada a una compuerta o a una rampa, se separa manualmente. En este punto, la parte puede ser enviada al cliente. Casi siempre se realizan operaciones de terminado en los moldes. Estas pueden incluir esmerilado, recubrimiento, tratamiento por calor, y varias pruebas como ultrasonido, calibración o inspección.

La fundidora produce más de 1,000 partes distintas para varios cientos de clientes. Una orden de un cliente, o un trabajo, especifica el número de partes idénticas que deben hacerse y una fecha de entrega para el trabajo. Cada parte tiene una hoja de ruta que contiene información de producción. Las hojas de ruta incluyen toda la información necesaria para hacer la parte, por ejemplo, contenido de aleación, patrón, ruta y requerimientos de mano de obra. Como un trabajo consiste en cierto número de las mismas partes, es sencillo obtener la información para el trabajo de la información de la parte.

Como puede inferirse del proceso de producción, la fundidora intenta minimizar el inventario de producto en proceso y el producto terminado. Es importante enviar el trabajo al cliente en la fecha de entrega, pero no debe producirse con demasiada anticipación. Esto da espacio de tiempo durante el cual debe producirse cada orden. Los trabajos con tiempos similares se unen en el programa de producción. Cada día se genera un programa de trabajos para un horizonte ondulado de 5 días. La planta se diseñó para que hubiera un flujo suave de productos a través del proceso sin inventario intermedio así, la línea de producción completa se maneja como una sola gran máquina. El taller de machos tiene una capacidad mayor que la línea; su programa está determinado por el programa de la fundidora. El problema de la fundidora es establecer qué



trabajo se hace en qué momento, es decir, programar la producción de la fundidora.

## 2.2 Descripción y complejidad del problema de la programación de la producción

### 2.2.1 Descripción del problema

Morton y Pentico (1993) afirman: *"programar es el proceso de organizar, elegir y dar tiempos al uso de recursos para llevar a cabo todas 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 entre las actividades y los recursos"*.

Volviendo al ejemplo de la fundidora, los trabajos son las actividades y las máquinas son los recursos. Entonces un programa de producción especifica el tiempo en el que comienza y termina cada trabajo en cada máquina, al igual que cualquier recurso adicional que se necesite. Una secuencia es un orden simple de trabajos; 3-1-2 significa que el trabajo 3 se hace primero, el trabajo 1 es el segundo y el trabajo 2 es el último. Si cada trabajo comienza tan pronto como es posible y se procesa sin interrupción para un tiempo dado de procesamiento, la secuencia determina los tiempos de inicio y terminación y, por lo tanto, determina la programación de la producción.

Determinar la "mejor" secuencia parece sencillo; sólo se enumera todas las secuencias y se elige la que optimiza alguna medida de desempeño previamente definida, esto realmente no es así y es la principal razón por la cual esta tesis cobra importancia e interés en el campo académico por todo el mundo y forma parte del pensum de muchas carreras como: ingeniería industrial, ingeniería logística, etc. Supongamos que tenemos 32 trabajos que procesar, el número de secuencias posibles es  $32! \approx 2.6 \times 10^{35}$  secuencias posibles a presentar como programa de producción para esta línea de producción. Supongamos que una computadora puede examinar 1000 millones de secuencias por segundo, con esta velocidad de procesamiento, tomaría **8.4**

x  $10^{15}$  siglos para examinar todas las posibles secuencias!. Ahora bien, supongamos que solo son 16 trabajos (la mitad de lo inicialmente indicado), tenemos  $16! \approx 2.1 \times 10^{13}$  secuencias posibles (más de 20 billones de programas).

Los recursos adicionales (mano de obra, materia prima, etc) y las dependencias entre trabajos (como los tiempos de preparación, conocidos como: set-ups) complican aún más el problema. Más adelante se comentará detalles sobre la complejidad de los problemas de programación. Esta explosión combinatoria muestra por qué es difícil resolver algunos problemas de programación de producción y la razón por la que su estudio se vuelve interesante.

### 2.2.2 Complejidad del problema

Lo siguiente es un análisis breve más que una descripción matemática precisa. Un algoritmo eficiente es aquel en el que el esfuerzo dedicado a un problema está acotado por un polinomio cuyo grado es el tamaño del problema (como el número de trabajos por ejemplo).

Un ejemplo sería el algoritmo para el árbol de expansión mínima, que se puede resolver a lo más en  $n^2$  iteraciones, donde  $n$  es el número de arcos. Se dice que el algoritmo es de orden  $n$  cuadrada [ $O(n^2)$ ].

Si el esfuerzo es exponencial [por ejemplo,  $O(2^n)$ ], el algoritmo no es eficiente. Un ejemplo sería el algoritmo de ramificación y acotamiento para variables binarias  $[0,1]$ , que puede requerir  $2^n$  nodos que deben explorarse.

El conjunto NP es el conjunto de todos los problemas que se puede resolver por enumeración total. El conjunto P es un subconjunto de NP que consiste en todos los problemas para los que se conocen algoritmos eficientes.

De nuevo, el árbol de expansión mínima es un ejemplo de un problema que pertenece a P. El conjunto NP duro también es un subconjunto de NP, pero estos problemas son los que se ha probado que son los más difíciles en NP (el problema del viajero se conoce como NP duro). De hecho, si alguien encontrará un algoritmo eficiente para cualquier problema en NP duro, ese algoritmo se

podría modificar para resolver todos los problemas en NP en tiempo polinomial, lo que haría al inventor rico y famoso.

Existen problemas en NP que en la actualidad no se pueden clasificar como pertenecientes a P o a NP duro; es decir, que su complejidad no se conoce, y se los denomina problemas abiertos.

Si un problema es NP duro, a menos que todos los problemas en NP se puedan resolver polinomialmente, deberá recurrir a algoritmos exponenciales (enumerativos) para obtener soluciones óptimas. Para ejemplos pequeños, o en algunos casos particulares, estos algoritmos enumerativos pueden ser aceptables. Conforme el problema crezca, la explosión combinatoria hará que sea imposible resolver el problema en un tiempo razonable. Otro enfoque es usar algoritmos heurísticos, que por lo general son buenos, pero no necesariamente proporcionan soluciones óptimas. Es común que los algoritmos heurísticos sean algoritmos polinomiales adaptados para una estructura de problema específica.

### 2.3 Notación y fundamento teórico

En todos los problemas de programación se considera el número de trabajos y el número de máquinas como una cantidad finita o limitada. El número de trabajos es denotado por "n" y el número de máquinas por "m". Usualmente, el sufijo "j" hace referencia a un trabajo mientras el sufijo "i" hace referencia a una máquina. Si un trabajo requiere un número de pasos o etapas de procesamiento u operaciones, entonces el par (i, j) hace referencia a la etapa de procesamiento u operación de un trabajo j en la máquina i.

**Tiempo de procesamiento ( $P_{ij}$ ):** representa el tiempo de procesamiento del trabajo j en la máquina i. El sufijo i es omitido si el tiempo de procesamiento del trabajo j no depende de la máquina o si el trabajo j es solo por ser procesado en una sola máquina.

**Fecha de liberación ( $r_j$ ):** la fecha de liberación  $r_j$  del trabajo j hace referencia al momento en que el trabajo j se encuentra listo para ingresar al sistema para

ser procesado. Este es el tiempo más pronto en que el trabajo puede empezar a ser procesado.

**Fecha de entrega ( $d_j$ ):** la fecha de entrega  $d_j$  del trabajo  $j$  representa el despacho comprometido o fecha de terminación. La culminación de un trabajo luego de su fecha de entrega es permitida, pero entonces una penalidad es generada.

**Ponderación ( $w_j$ ):** el peso  $w_j$  del trabajo  $j$  es básicamente un factor de prioridad o "peso", denotando la importancia del trabajo  $j$  con el resto de trabajos en el sistema. Por ejemplo: esta ponderación puede representar el costo actual de mantener el trabajo en el sistema. Este costo podría ser un costo de inventario. Acorde la información recopilada en la literatura disponible acerca de los problemas de programación de producción u operaciones (Scheduling en inglés) y utilizada para el desarrollo de esta tesis, tenemos que la notación para describir este tipo de problemas, se representa de la siguiente forma:

### **$\alpha$ | $\beta$ | $\gamma$**

El campo  **$\alpha$**  describe el ambiente o entorno de máquina.

El campo  **$\beta$**  describe o provee detalles de las características de procesamiento y restricciones.

El campo  **$\gamma$**  describe el objetivo a ser minimizado.

#### **2.3.1 Campo $\alpha$ : Ambientes o entornos de máquina**

Los posibles entornos de máquina indicados en este campo se muestran a continuación:

**Una sola máquina (1):** el caso de una sola máquina es el más simple de todos los posibles ambientes de máquina. Se tiene sólo una máquina y deben procesarse en ella todos los trabajos. La máquina puede procesar a lo más un trabajo a la vez. Una vez que un trabajo se ha procesado en la máquina, se termina.

**Máquinas idénticas paralelas (Pm):** en este entorno existen  $m$  idénticas máquinas en paralelo. El trabajo  $j$  requiere una simple operación y puede ser procesado en cualquiera de las  $m$  máquinas y una vez procesado, el trabajo queda terminado.

**Máquinas paralelas con diferentes velocidades (Qm):** en este entorno hay  $m$  máquinas en paralelo y cada una de ellas presenta diferentes velocidades. La velocidad de la máquina  $i$  es denotada por  $v_i$ . El tiempo  $p_{ij}$  que el trabajo  $j$  pasa en la máquina  $i$  es igual a  $p_j / v_i$

**Máquinas independientes en paralelo (Rm):** este entorno es una generalización más amplia del caso anterior. Hay  $m$  diferentes máquinas en paralelo. Máquina  $i$  puede procesar el trabajo  $j$  a una velocidad  $v_{ij}$ . El tiempo  $p_{ij}$  que el trabajo  $j$  pasa en la máquina  $i$  es igual a  $p_j / v_{ij}$ .

**Taller de flujo continuo (Fm):** conocido también como "Flow shop", tenemos en este entorno que hay  $m$  máquinas en serie. Cada trabajo tiene que ser procesado en cada una de las  $m$  máquinas sin excepción. Todos los trabajos tienen que seguir la misma ruta, es decir, los trabajos deben ser procesados primero en la máquina #1, luego en la máquina #2, etc. Luego de su culminación en una máquina, un trabajo se une a la cola para la siguiente máquina. Usualmente, todas las colas son asumidas que operan bajo la regla FIFO (First In First Out), esto significa que ningún trabajo se puede saltar a otro mientras espera en una cola.

**Taller de flujo continuo flexible (FFc):** este entorno es una generalización del taller de flujo continuo y el ambiente de máquinas paralelas. En vez de  $m$  máquinas en serie, hay  $c$  estaciones en serie y en cada estación está conformada por máquinas paralelas idénticas. Cada trabajo tiene que ser procesado primero en la estación #1, luego en la estación #2, etc.

**Taller de trabajo (Jm):** conocido también como "Job shop", en este entorno se tiene  $m$  máquinas y cada trabajo tiene su propia predeterminada ruta a seguir.

**Taller de trabajo flexible (FJc):** este entorno es una generalización del taller de trabajo y máquinas paralelas. En vez de  $m$  máquinas en serie, hay  $c$  centros de trabajo y cada centro de trabajo cuenta con un número de idénticas máquinas en paralelo. Cada trabajo tiene su propia ruta a seguir a través del taller.

**Taller abierto (Om):** conocido también como "Open shop", en este entorno existen  $m$  máquinas, cada trabajo tiene que ser procesado nuevamente en cada una de las  $m$  máquinas. Sin embargo, alguno de estos tiempos de procesamiento puede ser cero. No hay restricciones con respecto a la ruta de cada trabajo a través del entorno de máquina. El programador está permitido determinar una ruta para cada trabajo y diferentes trabajos tienen diferentes rutas de procesamiento.

### 2.3.2 Campo $\beta$ : Características de procesamiento

Las restricciones de procesamiento y las limitaciones especificadas en este campo pueden incluir múltiples entradas. Las posibles entradas son:

**Fechas de liberación ( $r_j$ ):** si este símbolo aparece en este campo, entonces el trabajo  $j$  no puede empezar su procesamiento antes de su tiempo de liberación  $r_j$ . Si  $r_j$  no aparece en este campo, el procesamiento del trabajo  $j$  puede iniciar en cualquier momento.

**Corte de procesamiento (prmp):** esta restricción implica que no es necesario mantener un trabajo en una máquina, una vez empezada, hasta su terminación. El programador está permitido de interrumpir el procesamiento de un trabajo (corte de procesamiento) en cualquier punto del tiempo y poner un diferente trabajo en dicha máquina.

**Restricción de precedencia (prec):** esta restricción aparece en un entorno de máquina sencilla o en una ambiente de máquina paralela, en donde solicita que uno o más trabajos tengan que ser terminados antes que otro trabajo sea permitido iniciar su procesamiento.

**Tiempos de preparación dependiente de la secuencia ( $s_{jk}$ ):** esta restricción indica que el tiempo de preparación de la máquina es dependiente de la secuencia incurrida entre el procesamiento del trabajo j y trabajo k.

**Familias de trabajos (fmls):** esta restricción indica que los “n” trabajos pertenecen, en este caso, a F diferentes familias de trabajos. Los trabajos de una misma familia pueden tener diferentes tiempo de procesamientos, pero ellos pueden ser procesados en una máquina uno después del otro sin requerir ninguna preparación entre ellos.

**Procesamiento de lotes [batch(b)]:** una máquina puede estar disponible para procesar un número de trabajos, digamos b, simultáneamente, eso es, la máquina puede procesar un lote de hasta b trabajos al mismo tiempo. Los tiempos de procesamiento de los trabajos en un lote pueda que no sean iguales y el lote entero es terminado solo cuando el último trabajo del lote haya sido completado.

**Averías (brkdown):** las averías en una máquina implican que la misma no puede estar continuamente disponibles. El periodo que una máquina no está disponible, se asume que está siendo reparada.

**Restricción de elección de máquina ( $M_j$ ):** esta restricción aparece en el ambiente de máquinas paralelas ( $P_m$ ). Cuando esta restricción está presente, no todas las m máquinas están en capacidad de procesar el trabajo j.

**Permutación (prmu):** una restricción que puede aparecer en los ambientes de taller de flujo continuo (flow shop) es que la cola en frente de cada máquina opere acorde la regla FIFO (first in first out). Esto implica que el orden (o permutación) en la cual los trabajos van a través de la primera máquina es mantenida en todo el sistema.

**Bloqueamiento (block):** El bloqueo es un fenómeno que puede ocurrir en en el entorno de taller de flujo continuo (flow shop). Si este entorno de máquina tiene un amortiguador limitado entre dos máquinas sucesivas, entonces puede

sucedan que cuando el amortiguador está lleno que precede no se le permite liberar un trabajo terminado. El bloqueo implica que el trabajo realizado ha de permanecer en la máquina precedente por prevención.

**Sin espera (nwt):** esta restricción es otro fenómeno que puede ocurrir en los entornos de taller de flujo continuo (flow shop). Los trabajos no están permitidos esperar entre dos máquinas sucesivas.

**Recirculación (rcrc):** esta restricción ocurre en un taller de trabajos o en un taller de trabajos flexible cuando un trabajo puede visitar una máquina o centro de trabajo más de una vez.

### 2.3.3 Campo $\gamma$ : Objetivo a ser optimizado

El mejor programa implica una medida de desempeño. Maximizar la ganancia o minimizar los costos son medidas obvias. Desafortunadamente, es difícil estimar los parámetros financieros que relacionen un programa asociando a costo o ganancia. Por otra parte, no se conocen algoritmos eficientes para optimizar la ganancia o el costo en modelos de programación de la producción. Se usan objetivos sustitutos para aproximar algunos costos relevantes.

El objetivo a ser minimizado es siempre una función de los tiempos de terminación de los trabajos, el cual, por supuesto, depende de la programación.

El **tiempo de terminación de la operación del trabajo  $j$  en la máquina  $i$**  es denominado por  $C_{ij}$ . El tiempo en la que el trabajo  $j$  deja el sistema (esto es, su tiempo de terminación en la última máquina el cual requiere procesamiento) es denominado por  $C_j$ . El objetivo puede además ser una función de la fecha de entrega.

El **retraso del trabajo  $j$**  es definido como  $L_j = C_j - d_j$ , el cual es positivo cuando el trabajo  $j$  es completado tarde y es negativo cuando este es completado tempranamente.

La **tardanza del trabajo  $j$**  es definido como  $T_j = \max (C_j - d_j, 0) = \max (L_j, 0)$



La diferencia entre la tardanza y el retraso radica en el hecho de que la tardanza nunca es negativa por su definición matemática.

La unidad de penalidad del trabajo  $j$  está definida de la siguiente manera:

$$U_j = \begin{cases} 1 & \text{si } t_j > d_j \\ 0 & \text{si } t_j \leq d_j \end{cases}$$

El retraso, la tardanza y la unidad de penalidad son tres básicas funciones de penalidad relacionadas a la fecha de entrega.

Dicho todo esto, ejemplos de funciones objetivo a ser minimizado son:

$C_{\max} = \max (C_1, \dots, C_n)$ , es el tiempo **máximo de finalización de todos los trabajos** o lapso (conocido también como **Makespan**), es equivalente al tiempo de terminación del último trabajo que deja el sistema. Un mínimo makespan implica una buena utilización de la máquina dado que implica una minimización del tiempo ocioso de la misma.

$L_{\max} = \max (L_1, \dots, L_n)$ , es el **retraso máximo** de todos los trabajos, este mide la peor violación a las fechas de entregas.

$T_{\max} = \max (T_1, \dots, T_n)$ , es la **tardanza máxima** de todos los trabajos.

$\sum_{j=1}^n w_j T_j$  = **tiempo de terminación ponderada total**, la suma de los tiempos de terminación ponderada de los  $n$  trabajos da una indicación del costo incurrido por el programa. La suma de los tiempos de terminación es en la literatura frecuentemente referida como el tiempo de flujo.

$\sum_{j=1}^n w_j (1 - e^{-rt_j})$  = **tiempo de terminación ponderada total descontada**, este es una más general función de costo que el caso anterior, aquí los costos son descontados a una tasa de  $r$ ,  $0 < r < 1$ , por unidad de tiempo. Esto es si el trabajo  $j$  no es completado por el tiempo  $t$ , un costo adicional  $w_j r e^{-rt} dt$  es incurrido sobre el periodo  $[t, t+dt]$ . Si el trabajo  $j$  es completado al tiempo  $t$ , el costo total incurrido sobre el periodo  $[0, t]$  es  $w_j (1 - e^{-rt})$ .

El valor de  $r$  es usualmente cercano a 0, digamos 0.1 o 10%

= **tardanza ponderada total**, este es un caso más general de la función de costo que el tiempo de terminación ponderada total.

= **número ponderado de trabajos tardíos**, esta función no es solo una medida de interés académico, este es frecuentemente un objetivo en la práctica dado que esta es una medida que puede ser calculada muy fácilmente.

## **2.4 Métodos heurísticos para resolver el problema de la programación de la producción.**

Algunos problemas de programación pueden ser formulados como programas lineales y son además, inherentemente fáciles; ellos pueden ser resueltos instantáneamente a través del uso de algoritmos eficientes. Otros problemas fáciles de programación pueden ser resueltos a través de algoritmos diferentes que son además eficientes; estos algoritmos son mencionados como algoritmos de tiempo polinomial. Que un problema pueda ser resuelto por un algoritmo eficiente implica que para muy grandes instancias de ese problema, con cientos o incluso miles de trabajo, pueda todavía ser resueltos en un relativo corto tiempo con ayuda de un computador. Sin embargo, hay muchos más problemas de programación que son intrínsecamente muy difíciles (NP – hard).

Ellos no pueden ser formulados como programación lineal y no hay simples reglas o algoritmos que encuentren una solución óptima en una limitada cantidad de tiempo computacional. Dado que en la práctica, no siempre está disponible una cantidad de tiempo computacional grande, uno usualmente se complace con una “aceptable” solución factible, que presumiblemente no se encuentre lejos de la solución óptima.

Esta sección describe una serie de procedimientos de uso general que son útiles para el tratamiento de problemas de programación en la práctica y que se pueden implementar con relativa facilidad en los sistemas de programación industrial. Es importante mencionar que todas las técnicas descritas son heurísticas que no garantizan una solución óptima; pero en cambio tienen como

objetivo encontrar soluciones razonablemente buenas en un tiempo relativamente corto. Las heurísticas tienden a ser bastante genéricas y se pueden adaptar fácilmente a una gran variedad de problemas de programación.

#### **2.4.1 Heurísticas: Reglas básicas de programación**

Una regla de programación es una instrucción que prioriza todos los trabajos que están esperando por ser procesados en una máquina. El esquema de priorización puede tomar en cuenta los atributos de los trabajos, los atributos de las máquinas así como el tiempo actual. La investigación en estas reglas de programación han sido activas por muchas décadas y muchas diferentes reglas han sido desarrolladas y estudiadas en la literatura. A continuación, se describe las reglas de programación más utilizadas:

**Regla del servicio en orden aleatorio (the service in random order rule - SIRO):** de acuerdo a esta regla de prioridad, siempre que una máquina es liberada, el siguiente trabajo es seleccionada en forma aleatoria. Con esta regla, no se intenta minimizar ningún objetivo en particular.

**Regla de primero la fecha más temprana de lanzamiento (the earliest release date first rule -ERD):** en este caso la regla es equivalente a la muy conocida regla de Primero en Llegar – Primero en Servir (First Come First Served rule - FCFS). Esta regla en un sentido minimiza la variación en los tiempos de espera de los trabajos en una máquina.

**Regla de primero la fecha de entrega más temprana (the earliest due date first rule -EDD):** en este caso, siempre que una máquina es liberada, el trabajo con la fecha de entrega mas pronta es la elegida para ser procesada. Esta regla tiende a minimizar el máximo retraso entre los trabajos que esperan procesamiento. Actualmente, en un ambiente de 1 máquina, con n trabajos disponibles al tiempo cero, la regla EDD hace minimizar el máximo retraso.

**Regla de primero la holgura mínima (the minimum slack first rule - MS):** esta regla es una variación de la regla EDD. Si una máquina es liberada

en el tiempo  $t$ , la holgura restante de cada trabajo en ese tiempo, definida como el  $\max(d_j - p_j - t, 0)$ , es registrada.

El trabajo con la mínima holgura es la siguiente en ser programada. Esta regla tiende a minimizar la fecha de entrega.

**Regla de primero el tiempo de procesamiento más corto ponderado (the weighted shortest processing time first rule - WSPT):** esta regla indica que cuando una máquina es liberada, el trabajo con el más alto ratio entre el peso ( $w_j$ ) sobre el tiempo de procesamiento ( $p_j$ ) será el siguiente en continuar. Los trabajos son ordenados en orden decrecientes de  $w_j/p_j$ . Esta regla tiende a minimizar la suma ponderada de los tiempos de terminación ( ). En un ambiente de una sola máquina, con  $n$  trabajos disponibles en el momento cero, la regla WSPT minimiza el . Cuando todos los pesos son iguales la regla WSPT se reduce a la regla SPT.

**Regla de primero el tiempo de procesamiento más pequeño (the shortest processing time first rule - SPT):** esta regla consiste en programar los trabajos en orden creciente de su tiempo de procesamiento  $p_j$ .

**Regla de primero el tiempo de procesamiento más largo (the longest processing time first rule - LPT):** esta regla ordena los trabajos en orden decreciente del tiempo de procesamiento. Cuando hay máquinas en paralelo, esta regla tiende a balancear la sobrecarga de trabajo de las máquinas. El razonamiento atrás de esto es el siguiente. Es ventajoso mantener los trabajos con corto tiempo de procesamiento para después debido que esos trabajos son útiles en el final para balancear la carga de trabajo. Después de que la asignación de trabajos a las máquinas ha sido determinado, los trabajos en cualquier máquina dada puede ser re-secuenciada sin afectar el balance de la carga de trabajo.

**Regla de primero el tiempo de preparación más corto (the shortest setup time first rule - SST):** siempre que una máquina se encuentre

liberada, esta regla selecciona para procesar el trabajo con el más corto tiempo de preparación (setup).

**Regla del radio crítico (critical ratio rule - CR):** esta regla programa los trabajos de acuerdo al radio del tiempo hasta el tiempo del vencimiento y el tiempo de procesamiento restante. Es un balance entre el EDD y LPT.

**Regla del costo aparente de la tardanza con tiempos de preparación (Apparent Tardiness Cost with Setups rule - ATCS):** esta regla es una combinación de la regla WSPT, MS y SST en un sencillo indicador de jerarquización o ranking. Bajo esta regla, los trabajos son secuenciados de uno en uno, es decir, cada vez que la máquina esta liberada, un indicador es calculado para cada trabajo pendiente. Hay  $n$  trabajos y una máquina sencilla. Los trabajos están sujetos a la secuencia dependiente de los tiempos de preparación  $S_{jk}$ . El objetivo es minimizar la suma de la tardanza ponderada. La prioridad de cualquier trabajo  $j$  depende del trabajo una vez completado cuando la maquina esta liberada. La regla calcula el indicador del trabajo  $j$  una vez culminado el trabajo  $l$  en el tiempo  $t$  como se muestra:

$$(\theta_1, \theta_2) = \frac{1}{K_1} \exp\left(-\frac{\max(t - S_{jk}, 0)}{\bar{S}}\right) \exp\left(-\frac{K_2}{t - S_{jk}}\right)$$

Donde  $\bar{S}$  es el promedio de los tiempos de preparación (setups) de los trabajos pendientes a ser programados,  $K_1$  es el parámetro de escala relacionado a la fecha de entrega y  $K_2$  es el parámetro de escala relacionado al tiempo de preparación. Notar que los parámetros de escala son cantidades no dimensionales, haciendo independientes a ellos de las unidades usadas para expresar cantidades varias. Los 2 parámetros de escala,  $K_1$  y  $K_2$  pueden ser consideradas como funciones de 3 factores:

- (i) el factor de pequeñez de la fecha de entrega  $\theta_1$
- (ii) el factor de rango de la fecha de entrega  $\theta_2$
- (iii) el factor de severidad del tiempo de preparación  $\theta_3 = \bar{S} / t$

El factor  $\theta_1$  es definido como:  $\theta_1 = 1 - \frac{K_1}{t - S_{jk}}$ , donde  $\bar{S}$  es el promedio de las fechas de terminación. Valores de  $\theta_1$  cercanos a 1 indica que las fechas de

culminación son ajustados, valores cercanos a 0 indica en cambio que las fechas son holgadas.

El factor  $\theta_2$  es definido como:  $\theta_2 = \frac{\text{rango de fechas}}{\text{tiempo de preparación}}$ , un alto valor de  $\theta_2$  indica un rango grande de las fechas de entrega, mientras que un valor bajo indica que existe un rango estrecho de las fechas de entrega.

Estas estadísticas no son fáciles de determinar, incluso con una máquina sencilla el tiempo máximo de culminación del último trabajo (makespan =  $C_{\max}$ ) es dependiente de la programación a causa de los tiempos de preparación. Antes de calcular los factores  $\theta_1$  y  $\theta_2$  el makespan tiene que ser estimado. Un simple estimado para el makespan en una maquina sencilla puede ser:

$$= \sum t_{ij} + \dots$$

Esta cantidad es apropiada para estimar el makespan como el programa final que tomará ventaja de los tiempos de preparación que son más bajos que el promedio. Las definiciones de  $\theta_1$  y  $\theta_2$  tienen que ser modificadas reemplazando el makespan con su estimación.

Un experimental estudio de la regla ATCS, ha sugerido algunas guías para la selección de los 2 parámetros  $K_1$  y  $K_2$ , las siguientes reglas pueden ser usadas para calcular los valores para  $K_1$  y  $K_2$

$$K_1 = 4.5 + \theta_2 \quad (\text{para } \theta_2 \leq 0.5)$$

$$K_1 = 6 - 2\theta_2 \quad (\text{para } \theta_2 \geq 0.5)$$

$$K_2 = \theta_1 / (2 \cdot \sqrt{\theta})$$

#### 2.4.2 Heurísticas: Reglas compuestas de programación

Las reglas básicas de programación son útiles cuando uno intenta encontrar un razonablemente buen programa con respecto a un sencillo objetivo tal como el makespan, la suma de los tiempos de terminación o el máximo retraso. Sin embargo, en la práctica los objetivos son frecuentemente muy complicados.

Un objetivo realista puede ser una combinación de una gran cantidad de objetivos y estos pueden además ser una función del tiempo o una función del conjunto de trabajos esperando por ser procesados. Secuenciar los trabajos en

base de 1 o 2 parámetros conduzca a un aceptable programa. En esta sección se revisará la literatura de las heurísticas más conocidas.

**La heurística del cuello de botella cambiante (the shifting bottleneck heuristic):**  $M$  denota el conjunto de todas las  $m$  máquinas. En la descripción de una iteración de la heurística, esta es asumida que la iteraciones previas de una selección de arcos disyuntivos ya ha sido fijado para un subconjunto  $M_0$  de máquinas. Por lo que para cada una de las máquinas en  $M_0$  una secuencia de operaciones ya ha sido determinada.

Una iteración determina cual máquina en  $M - M_0$  tiene que ser incluida para el siguiente conjunto  $M_0$ . La secuencia en la cual las operaciones en esta máquina tienen que ser procesada es además generada en esta iteración. En orden para seleccionar la máquina para ser incluida en el siguiente en  $M_0$ , se hace un intento para determinar que una de las máquinas aún por ser programados causaría en uno u otro sentido la interrupción severa. Para determinar esto, el grafo dirigido original se modificó mediante la supresión de todos los arcos disyuntivos de las máquinas que aún no se ha programado y conservar solo los arcos disyuntivos relevantes de las máquinas en el conjunto  $M_0$ . (uno desde cada par), llamar este grafo  $G''$ .

Eliminando todos los arcos disyuntivos de una máquina específica implica que todas las operaciones en esta máquina, que originalmente iban a ser hecho en esta máquina, una tras otra, ahora se pueden hacer en paralelo (como si la máquina tiene capacidad infinita, o equivalentemente, cada uno de estas operaciones tiene la máquina por sí mismo). El grafo  $G$  tiene uno o más caminos críticos que determinan el makespan correspondiente. Llame a este makespan  $C_{\max}(M_0)$ . Supongamos que la operación  $(i,j)$ ,  $i \in \{M - M_0\}$ , tiene que ser procesado en una ventana de tiempo para el cual la fecha de liberación y la fecha de terminación son determinados por el camino crítico en  $G''$ , la fecha de liberación es igual al camino más largo en  $G''$  desde el recurso  $U$  al nodo  $(i,j)$  y la fecha de terminación es igual a  $C_{\max}(M_0)$ , menos el camino más largo desde el nodo  $(i,j)$  al sumidero mas  $p_{ij}$ . Considerar cada de una de las máquinas en  $M$

–  $M_0$  como un separado problema del tipo  $1 \mid r_j \mid L_{\max}$ . El mínimo  $L_{\max}$  del problema de máquina sencilla corresponde a la máquina  $i$  es denotado por  $L_{\max}(i)$  y es una medida de la criticidad de la máquina  $i$ .

Después de resolver todos estos problemas de máquinas sencillas, la máquina con el más largo retraso máximo es elegida. Entre las máquinas restantes, esta máquina, es en un sentido la más crítica o el "cuello de botella" (bottleneck) y además la una por ser incluida en la siguiente en  $M_0$ . Etiqueta de esta máquina  $k$ , llama a su máximo retraso  $L_{\max}(k)$  y se programa está acorde a la óptima solución obtenida por el problema de la máquina sencilla asociado con esta máquina. Si los arcos disyuntivos que especifica la secuencia de operaciones en la máquina  $k$  son insertados en el grafo  $G''$ , entonces el makespan de la programación parcial actual por  $L_{\max}(k)$ , esto es:

$$C_{\max}(M_0 \cup k) \geq C_{\max}(M_0) + L_{\max}(k)$$

Antes de empezar la siguiente iteración y determinar la siguiente máquina por ser programada, un paso adicional tiene que ser hecho dentro de la actual iteración. En este paso adicional todas las máquinas en el original conjunto  $M_0$  son reprogramadas en orden para ver si el makespan puede ser reducido. Eso es, una máquina, llamada máquina  $l$ , es sacado del conjunto  $M_0$  y un grafo  $G''$  es construido por modificando el grafo  $G''$  a través de la inclusión de los arcos disyuntivos que especifica la secuencia de operaciones en la máquina  $k$  y la exclusión de los arcos disyuntivos asociados con la máquina  $l$ .

La máquina  $l$  es reprogramada por la solución correspondiente al problema  $1 \mid r_j \mid L_{\max}$  con la fecha de liberación y terminación determinada por la ruta crítica en el grafo  $G''$ . Reprogramando cada una de las máquinas en el original conjunto  $M_0$  completa la iteración.

En la siguiente iteración el procedimiento entero es repetido y otra máquina es agregada al actual conjunto  $M_0 \cup k$ .

**La heurística de la búsqueda local (Local Search):** los algoritmos de recocido simulado y búsqueda tabú que se mencionaran a continuación en



forma más detallada, estos se consideran como procedimientos de búsqueda local. Esta heurística se apalanca en la idea de que una solución puede ser mejorada mediante cambios pequeños y luego revisando la respuesta. Sea un conjunto  $P$  de soluciones posibles cuya bondad es dada por una función de costo  $c: P \rightarrow R$ , que corresponde a la función que uno desea optimizar (se asume que la optimización es de minimización). Se define que para solución  $x \in P$  un vecindario  $N(x)$  donde cada solución contenida en  $N(x)$  es denominada vecino y puede ser alcanzada mediante un desplazamiento unitario. Así pues, una ejecución de un algoritmo de búsqueda local determina un camino por el que cada solución visitada es vecina de la inmediatamente anterior.

**La heurística del recocido simulado (Simulated Annealing):** El recocido simulado es un proceso de búsqueda que tiene su origen en los campos de la ciencia de los materiales y la física. Esto fue desarrollado como un modelo de simulación para describir el proceso físico del recocido de la materia condensada. El procedimiento del recocido simulado va a través de un número de iteraciones. En la iteración  $k$  del procedimiento, hay un programa actual  $S_k$  tan bueno como un mejor programa que se encontrará más adelante,  $S_0$ . Para una simple máquina estos programas son secuencias (permutaciones) de los trabajos. Dado que  $G(S_k)$  y  $G(S_0)$  denotan los valores correspondientes de la función objetivo. Note que  $G(S_k) \geq G(S_0)$ .

El valor del mejor programa obtenido,  $G(S_0)$ , es frecuentemente referido como el criterio de aspiración. El algoritmo, en su búsqueda para un programa óptimo, se mueve de un programa a otro. En la iteración  $k$ , una búsqueda por un nuevo programa es conducido dentro del vecindario de  $S_k$ .

Primero, un llamado programa candidato, llamado  $S_c$ , es seleccionado desde el vecindario. Esta selección de un programa candidato puede ser hecho de forma aleatoria o en un organizado, posiblemente secuencial, forma. Si  $G(S_c) < G(S_k)$ , un movimiento es realizado, parametrizando  $S_{k+1} = S_c$ . Si  $G(S_c) < G(S_0)$ , entonces  $S_0$  es colocado igual a  $S_c$ .

Sin embargo, si  $G(S_c) \geq G(S_k)$ , un movimiento a  $S_c$  es realizado solo con la probabilidad:

$$P(S_k, S_c) = \exp\left(\frac{f(S_k) - f(S_c)}{\beta_k}\right);$$

con una probabilidad  $1 - P(S_k, S_c)$  el programa  $S_c$  es rechazado a favor del actual programa, colocando  $S_{k+1} = S_k$ . El programa  $S_0$  no hace cambios cuando este es mejor que el programa  $S_c$ . Dado que  $\beta_1 \geq \beta_2 \geq \beta_3 \geq \dots > 0$  son controles de parámetros referidos como parámetros de enfriamiento o temperaturas (en analogía con el proceso de recocido mencionado arriba). Frecuentemente  $\beta_k$  es elegido para ser un  $a^k$  para algún  $a$  entre 0 y 1.

De la descripción anterior del procedimiento de recocido simulado es evidente se permite que se mueve a soluciones peores. La razón por permitir estos movimientos es dar al procedimiento la oportunidad de alejarse de un mínimo local y encontrar una solución mejor más adelante. Desde  $\beta_k$  disminuye con  $k$ , la probabilidad de aceptación para un movimiento no mejora, es menor en iteraciones posteriores del proceso de búsqueda. La definición de la probabilidad de aceptación también asegura que si un vecino es significativamente peor, su probabilidad de aceptación es muy baja y un movimiento es poco probable hacerse. Varios criterios de parada se utilizan para este procedimiento. Una manera es dejar que el procedimiento de ejecución para un número predefinido de iteraciones. Otra es la de permitir que el procedimiento de ejecución hasta que no se ha logrado mejorar durante un predeterminado número de iteraciones.

**La heurística de la búsqueda Tabu (Tabu search):** esta heurística es en muchos aspectos similar al recocido simulado debido a que también se mueve de una lista a otra con el siguiente programa de ser peor que posiblemente el anterior. Para cada programa, un vecindario se define como en el simulado recocido. La búsqueda dentro de la vecindad por un candidato potencial para trasladarse a nuevo es un problema del diseño. Al igual que en el recocido simulado, esto se puede hacer al azar o de manera organizada. La diferencia básica en la búsqueda tabu y el recocido simulado se encuentra en el mecanismo que se utiliza para la aprobación de un programa candidato. En la búsqueda tabú del mecanismo es no probabilística, sino de una naturaleza

determinística. En cualquier etapa del proceso de una lista tabú de mutaciones, el cual el procedimiento no está permitido hacer, se mantiene. Una mutación en la lista tabú puede ser, por ejemplo, un par de trabajos que no puede ser intercambiado. La lista tabú tiene un número fijo de entradas (por lo general entre 5 y 9), que depende de la aplicación. Cada vez que un movimiento se realiza a través de una cierta mutación en el programa actual, la mutación inversa se introduce en la parte superior de la lista tabú; todas las otras entradas en la lista tabú son empujadas hacia abajo una posición y el fondo se elimina la entrada. La mutación inversa se pone en la lista tabú para evitar volver a un mínimo local que se ha visitado antes. Actualmente, en el momento que una mutación inversa que es tabú podría tener lugar a un nuevo programa, no visitado antes, eso es mejor que cualquiera generado hasta el momento. Esto puede suceder cuando la mutación está cerca de la parte inferior de la lista tabú y un número de movimientos ya se han realizado desde la mutación se introduce en la lista.

**La heurística de algoritmos genéticos (Genetic Algorithms):** los algoritmos genéticos son más generales y abstractas que el caso del recocido simulado y la búsqueda tabú. El recocido simulado y búsqueda tabú pueden, en cierto modo, verse como casos especiales de los algoritmos genéticos. Los algoritmos genéticos, cuando se aplica a la programación, mira a las secuencias o los programas como individuos o como miembros de una población. Cada individuo se caracteriza por su salud. La salud de un individuo se mide por el valor asociado de la función objetivo. El procedimiento funciona de forma iterativa, y cada iteración se conoce como una generación.

La población de una generación se compone de los sobrevivientes de la generación anterior, además de los nuevos programas, es decir, los descendientes (hijos) de la generación anterior. El tamaño de la población por lo general se mantiene constante de una generación a la siguiente. La descendencia se genera a través de la reproducción y mutación de las personas que formaron parte de la generación anterior (los padres). Las personas a veces se denominan también como cromosomas. En un entorno multi-máquina

de un cromosoma puede consistir en sub-cromosomas, cada uno que contiene la información sobre la secuencia de trabajo en una máquina. Una mutación en un cromosoma padre puede ser equivalente a un intercambio de pares adyacentes en la secuencia correspondiente. En cada generación los individuos más aptos reproducen mientras que en el ajuste mueren menos. Los procesos de nacimiento, defunción y reproducción que determinan la composición de la próxima generación pueden ser complejos, y por lo general dependerá de los niveles de condición física de los individuos de la actual generación.

Un algoritmo genético, como un proceso de búsqueda, difiere en un aspecto importante del recocido simulado y búsqueda tabú. En cada paso iterativo un número de diferentes horarios se genera y se llevó a la siguiente etapa. En recocido simulado y búsqueda tabú sólo un único programa continua desde una iteración a la siguiente. Por lo tanto recocido simulado y búsqueda tabú se pueden considerar como casos especiales de algoritmos genéticos con un tamaño de población que es igual a 1.

Este esquema de diversificación es una característica importante de los algoritmos genéticos. En los algoritmos genéticos el concepto de vecindad tampoco se basa en un simple programa, sino en varios programas. El diseño de la vecindad de la actual población de programas se basa en las técnicas más generales que las utilizadas en el recocido simulado y búsqueda tabú. Un nuevo programa puede ser generado mediante la combinación de partes de diferentes programas de la población actual. Un mecanismo que crea un nuevo programa de este tipo se refiere a menudo como un operador de cruce.

## **2.5 LEKIN: Un sistema para la programación de la producción**

Como se comentó en el marco teórico, debe existir un balance entre la calidad de la solución y el tiempo computacional requerido para definir el mejor programa de producción. En las últimas 3 décadas, cientos de sistemas de programación de la producción han sido desarrollados en todo el mundo. Estos desarrollos han tomado lugar tanto para la industria como para la academia en

universidades en varios países alrededor del mundo, la siguiente tabla muestra una pequeña referencia de dichos sistemas de programación de la producción:

SISTEMAS	COMPAÑÍA - UNIVERSIDAD	CREADOR
Cyberplan	Cybertec	<a href="http://www.cybertec.it">www.cybertec.it</a>
SKEP	DynaSys Group	<a href="http://www.adaptasolutions.com">www.adaptasolutions.com</a>
Production Scheduler	i2 Technologies, Inc.	<a href="http://www.i2.com">www.i2.com</a>
Quintig Scheduler	Quintiq	<a href="http://www.quintiq.com">www.quintiq.com</a>
APO	SAP AG	<a href="http://www.sap.com">www.sap.com</a>
SSA Manufacturing Scheduling	SSA Global	<a href="http://www.ssaglobal.com">www.ssaglobal.com</a>
LEKIN	New York University	Pinedo (1998)
OPIS	Carnegie-Mellon University	Smith (1994)
TORSCHÉ	Czech Technical University	Stibor (2006)
TOSCA	University of Edinburgh	Beck (2003)
TTA	Universidad Católica de Chile	Nussbaum and Parra (1993)

**Tabla 2.1 Sistemas de programación comerciales y académicos**

Fuente: Scheduling (Pinedo Michael) - Elaboración: Propia

El sistema LEKIN es un software educacional para la programación de la producción desarrollado por el profesor Michael Pinedo de la Universidad de New York y otros colaboradores. Esta versión puede obtenerse gratuitamente desde la página de la universidad o a través de los libros en donde el profesor Pinedo ha sido autor o colaborador.



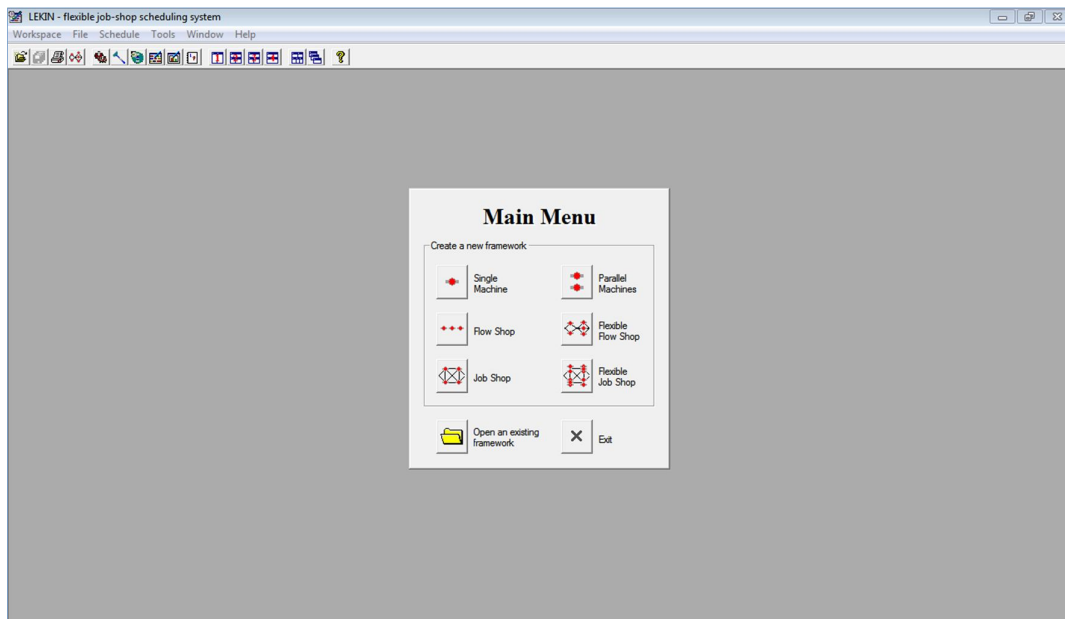
**Figura 2.2 LEKIN: Flexible Job Shop Scheduling System**

Fuente: LEKIN - Elaboración: Propia

LEKIN contiene un número de reglas y heurísticas, fue diseñado para permitir al usuario probar sus propias heurísticas y luego compararlas con las que vienen ya incorporadas en el software. El sistema puede manejar un número de diferentes entornos de máquina, como se enumera a continuación:

- Máquina única (Single machine)
- Máquinas en paralelo (Parallel machines)
- Taller de producción continuo (Flow Shop)
- Taller de producción continuo flexible (Flexible flow shop)
- Taller de trabajos (Job Shop)
- Taller de trabajos flexible (Flexible job shop)

Además, este programa está habilitado para considerar tiempos de preparación de máquinas (set-ups) dependientes de la secuencia para todos los ambientes de máquina mencionados anteriormente. Esta versión académica tiene una capacidad de hasta 50 trabajos, 20 estaciones de trabajo y 100 máquinas. Cuando LEKIN está corriendo por primera vez, una página de “bienvenida” se muestra, cerrando esta, aparece el menú principal (ver figura 2.3)



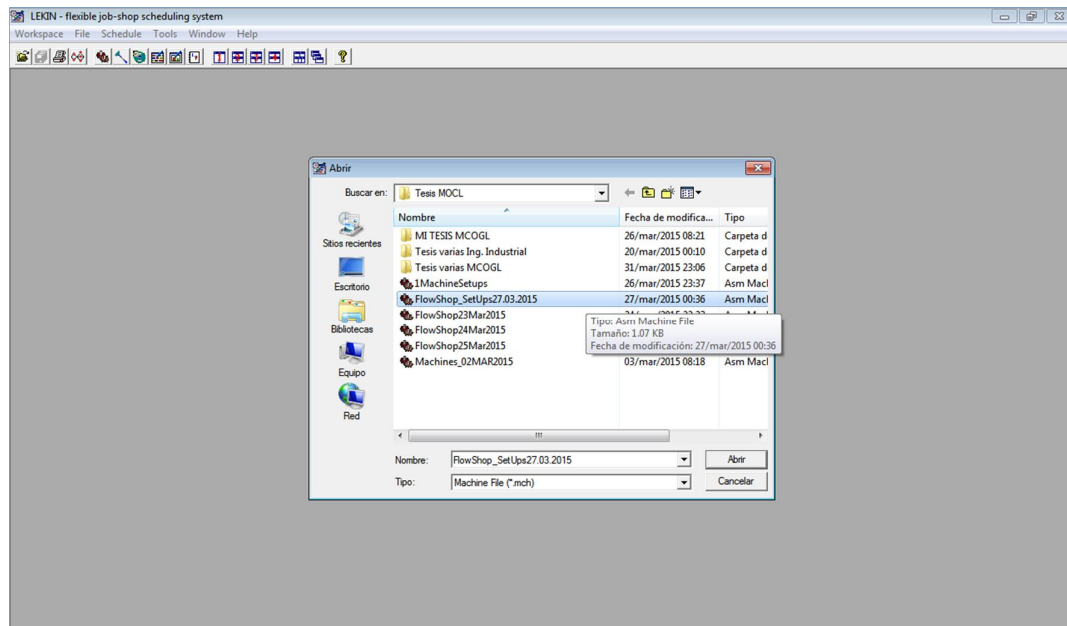
**Figura 2.3 Vista menú principal LEKIN**

Fuente: LEKIN - Elaboración: Propia

El menú principal permite al usuario seleccionar el ambiente de máquina que le se ajuste al entorno real que necesite analizar. Cuando el usuario selecciona un

tipo de ambiente de máquina, debe ingresar todos los datos de las máquinas necesarios y luego los datos de los trabajos manualmente. Sin embargo, el usuario además tiene la opción de abrir un archivo existente (ver figura 2.4). Un archivo existente contiene datos con respecto al entorno de máquina y al conjunto de trabajos.

El usuario puede abrir un archivo existente, hacer cambios en el archivo y trabajar luego con el archivo modificado. Al finalizar la sesión el usuario puede grabar el archivo modificado con otro nombre. Si el usuario requiere ingresar un conjunto de datos que es completamente nuevo, debe primero seleccionar un entorno de máquina, y luego, un cuadro de dialogo se muestra e indica donde tiene que ingresar la información básica: el número de de estaciones de trabajo y el número de trabajos por ser programados (Ver figura 2.5)

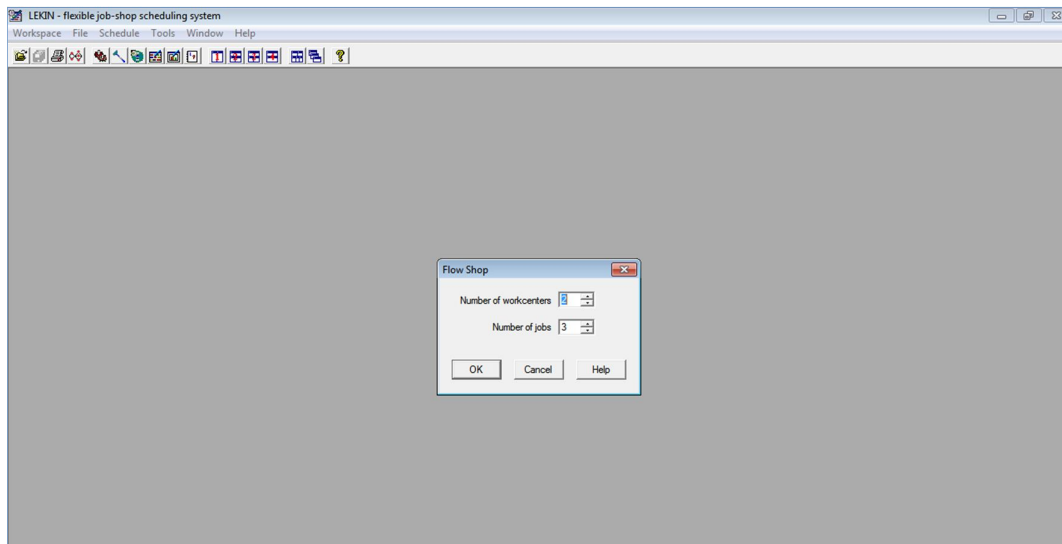


**Figura 2.4 Abriendo archivos LEKIN existentes**

Fuente: LEKIN - Elaboración: Propia

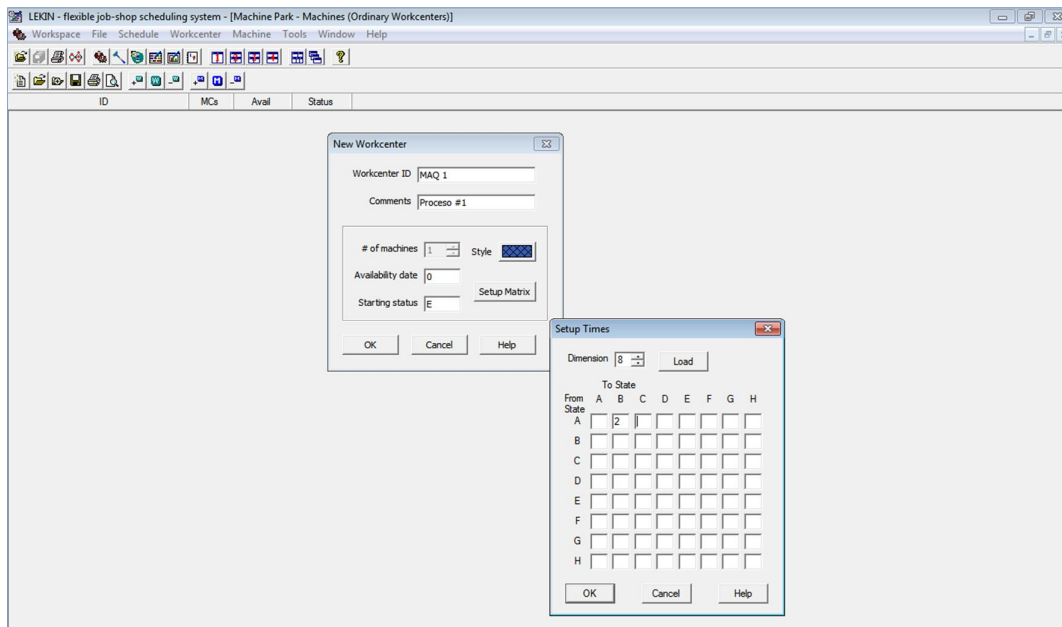
Luego de que el usuario ha hecho esto, un segundo cuadro de dialogo aparece en la pantalla y allí se debe ingresar más información sobre la estación de trabajo: su nombre, su disponibilidad, y los detalles requeridos para determinar los tiempos de preparación (set-ups) en cada máquina (Ver figura 2.6).

En el caso de los tiempos de preparación, es importante recalcar que la matriz de tiempos de preparación, su tamaño debe ser igual al número de trabajos del problema más 1, esto es porque se debe considerar el tiempo inicial ( $t=0$ ). Luego esta información debe concordar al momento de parametrizar los trabajos que se van a procesar en cada máquina (ver figura 2.7).



**Figura 2.5 Cuadro de diálogo #1**

Fuente: LEKIN - Elaboración: Propia

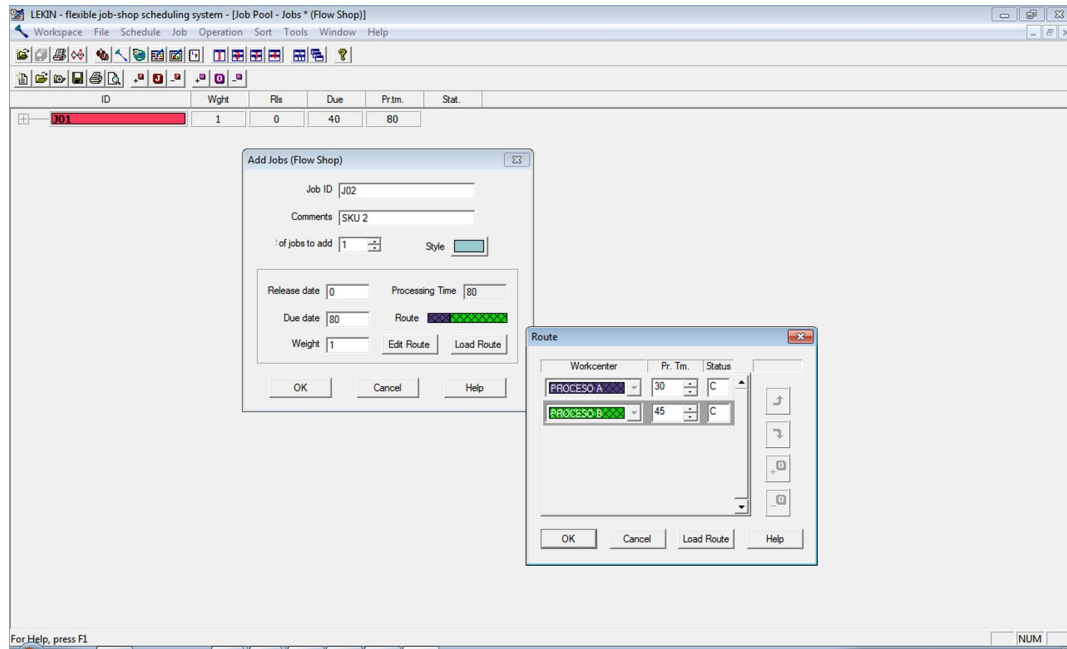


**Figura 2.6 Cuadro de diálogo #2**

Fuente: LEKIN - Elaboración: Propia



En el tercer cuadro de diálogo: el usuario tiene que ingresar la información detallada con respecto a los trabajos: fecha de liberación, fecha de terminación, pesos, rutas y tiempos de procesamiento de las diferentes operaciones.

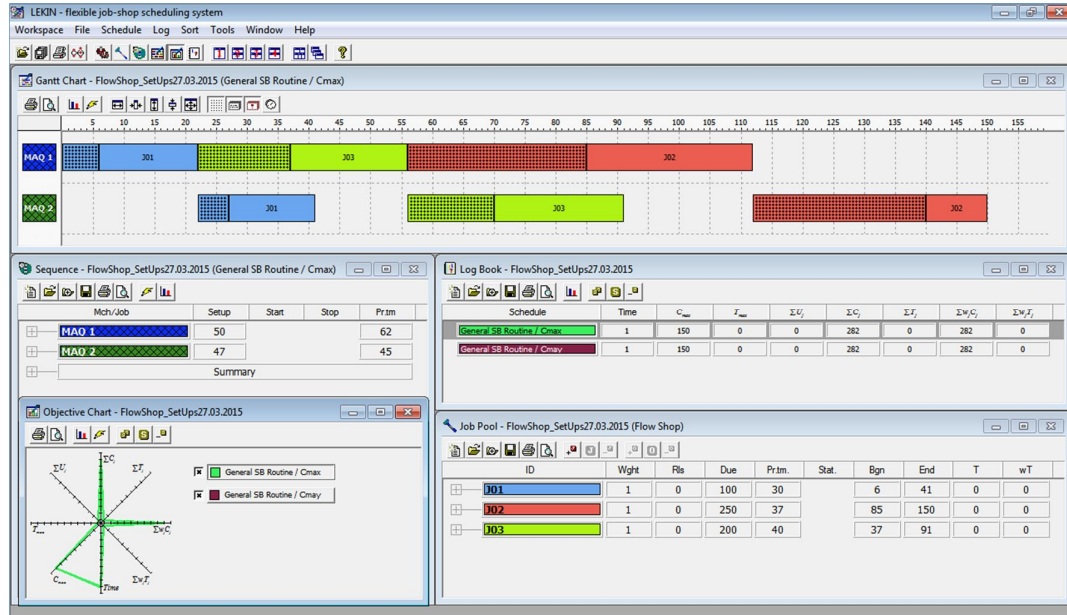


**Figura 2.7 Cuadro de diálogo #3**

Fuente: LEKIN - Elaboración: Propia

Luego de que todos los datos han sido ingresados en el sistema LEKIN, estas ventanas aparecen simultáneamente (Ver figura 2.8):

- Ventana del parque de máquinas
- Ventana del número de trabajos
- Ventana de secuencias
- Ventana del diagrama de Gantt
- Ventana del cuadro objetivo (una vez que se ha ejecutado una heurística de programación)
- Ventana de resumen de todos los programas de producción realizados.



**Figura 2.8 Ventanas principales en LEKIN**

Fuente: LEKIN - Elaboración: Propia

El sistema LEKIN contiene un número de algoritmos para una gran cantidad de ambientes de máquinas y función objetivos, todas ellas, fueron detalladas en el marco teórico al inicio de este capítulo. Estos algoritmos incluyen:

- (i) Reglas básicas de programación.
- (ii) Heurísticas del tipo de cuello de botella desplazante.
- (iii) Técnicas de búsqueda local
- (iv) Técnicas de método de descomposición (SB – SL)

Adicionalmente, existe la opción de ingresar manualmente un programa de producción con la secuencia que el usuario defina previamente y así poder comparar con las heurísticas que LEKIN tiene incorporado, esta opción será de mucha ayuda al momento de analizar la situación actual de la línea de producción versus la propuesta de mejora del programa de producción.

Finalmente, una vez que el usuario haya corrido algunas heurísticas, LEKIN muestra el resultado comparativo de las diferentes heurísticas realizadas, de esta forma, es fácil determinar que programa de producción es mejor o cumple los objetivos esperados por el usuario (Ver figura 2.9)

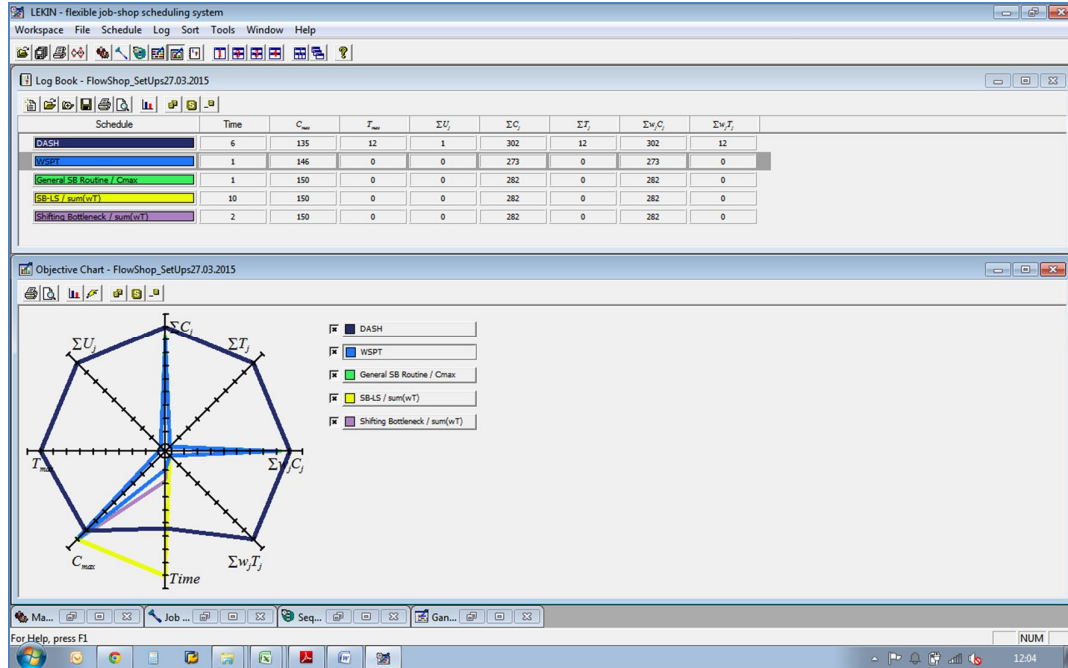


Figura 2.9 Comparación de resultados y objetivos

Fuente: LEKIN - Elaboración: Propia

## CAPÍTULO III

### SITUACIÓN ACTUAL: EL PROCESO DE PRODUCCIÓN DE UNA LÍNEA DE PELETIZACIÓN MULTIESPECIE

#### 3.1 Antecedentes de la compañía y fábrica en estudio

La fábrica en estudio forma parte de una compañía con más de 40 años de presencia en el mercado ecuatoriano, líder en el sector agroindustrial, con una facturación de 250 millones de dólares anuales y ubicada entre las 50 empresas más grandes del país (fuente: Revista Vistazo edición anual 2015).

La compañía cuenta con 1,100 colaboradores aproximadamente, 160 agencias ubicadas alrededor de todo el país y posee actualmente 5 fábricas, donde se manufactura y abastece los siguientes tipos de productos terminados:

1. alimentos balanceados para diferentes especies de animales,
2. agroquímicos (herbicidas, fungicidas e insecticidas),
3. semillas para diferentes especies
4. fertilizantes (simples y compuestos).

El desarrollo de esta tesis, se realiza en la fábrica de alimentos balanceados, que es la más grande, en personal, área de construcción y es la que ha mostrado un crecimiento sostenido en los últimos años (Ver figura 3.1)

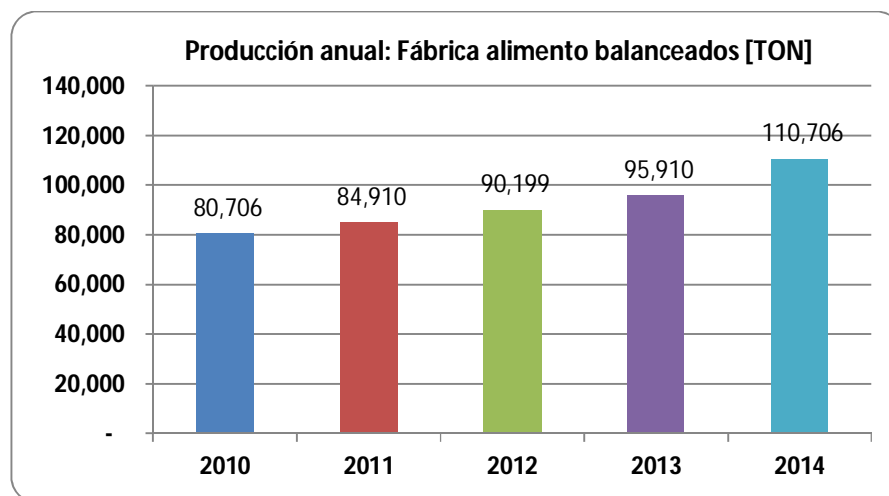


Figura 3.1 Producción anual: fábrica alimentos balanceados

Fuente: Estadísticas fábrica en estudio - Elaboración: Propia

Adicionalmente, esta fábrica, es la más reciente en formar parte de la compañía pues inicialmente formaba parte de otra administración, los mismos que fueron pioneros en el Ecuador desde 1979 en la elaboración de alimento balanceado para diferentes animales, especialmente: pollos, cerdo, ganado y camarón.

Desde el 2002, la fábrica en estudio tuvo un acercamiento con la compañía en mención, a la cual daba servicios de maquila de todo el portafolio de alimento balanceado que la compañía comercializaba en aquella época (alimento para pollos y camarones) para atender a sus clientes directos y sus agencias en todo el Ecuador. Con el pasar de los años, la relación se hizo más estrecha hasta que en el 2010, la compañía adquiere el 100% del paquete accionario a los dueños fundadores de la fábrica en estudio y desde ese momento, forma parte de la compañía en su totalidad tanto el personal administrativo, operativo e instalaciones productivas.

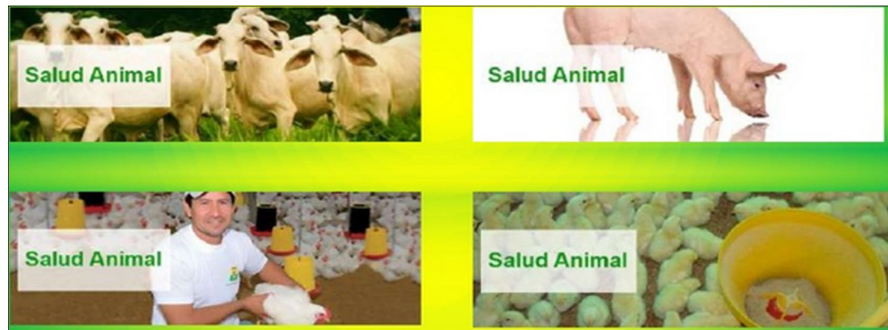
A raíz de la compra total de la fábrica, tanto el personal administrativo como el personal operativo se ha venido renovando y alineando a la filosofía que la compañía tiene con el resto de fábricas que posee, sin embargo: el conocimiento y decisiones inherentes al proceso de manufactura de alimentos balanceados (incluyendo la programación de la producción) se ha mantenido de la misma forma como lo manejaba los dueños fundadores de la fábrica. Este "paradigma" es uno de los objetivos específicos que esta tesis pretende cambiar pues al momento, la programación de la producción se la viene realizando del mismo modo o, dicho de otra manera, con el mismo criterio que la anterior administración lo realizaba.

### **3.2 Tipos de líneas de producción y selección para el análisis de la mejora de la programación**

La fábrica en estudio, actualmente atiende a 3 divisiones comerciales de las 7 que posee la compañía, estas divisiones comerciales atendidas son:

**División Salud Animal (SA):** esta división atiende a todo el sector agropecuario con un portafolio de alimento balanceado para las etapas de

crianza de las siguientes especies: pollos, cerdos, ganado, tilapia y truchas (ver figura 3.2)



**Figura 3.2 División Salud Animal (SA)**

Fuente: Página internet compañía en estudio - Elaboración: Propia

**División Acuicultura (AC):** esta división comercial atiende a todo el sector acuícola, concentrado principalmente en el sector camaronero, ofertando alimentos balanceado para todas las etapas de crianza de esta especie (ver figura 3.3)



**Figura 3.3 División Acuicultura (AC)**

Fuente: Página internet compañía en estudio - Elaboración: Propia

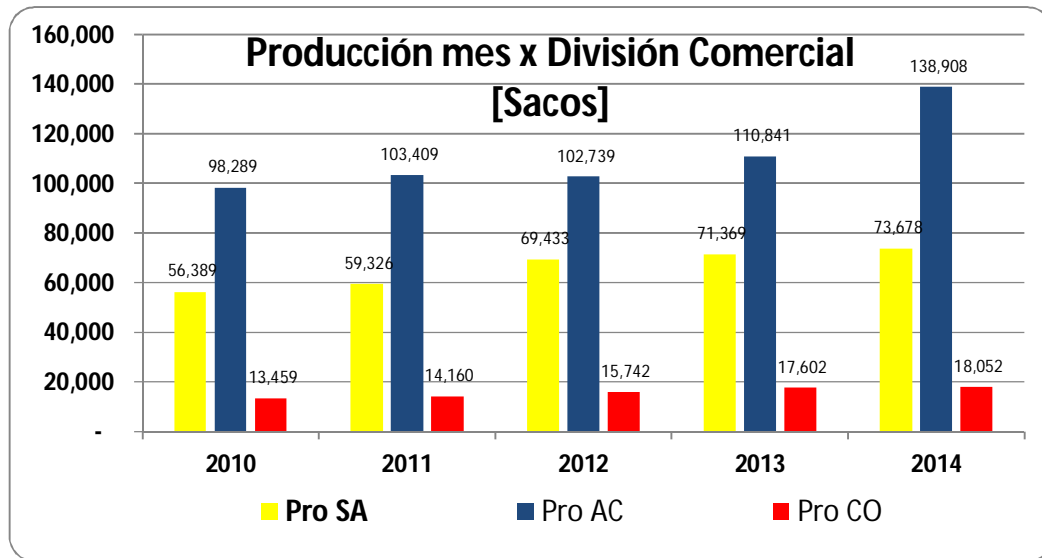
**División Consumo (CO):** esta división comercial atiende al sector de consumo masivo, enfocado en la alimentación para las mascotas: perros y gatos específicamente (ver figura 3.4)



**Figura 3.4 División Consumo (CO)**

Fuente: Página internet compañía en estudio - Elaboración: Propia

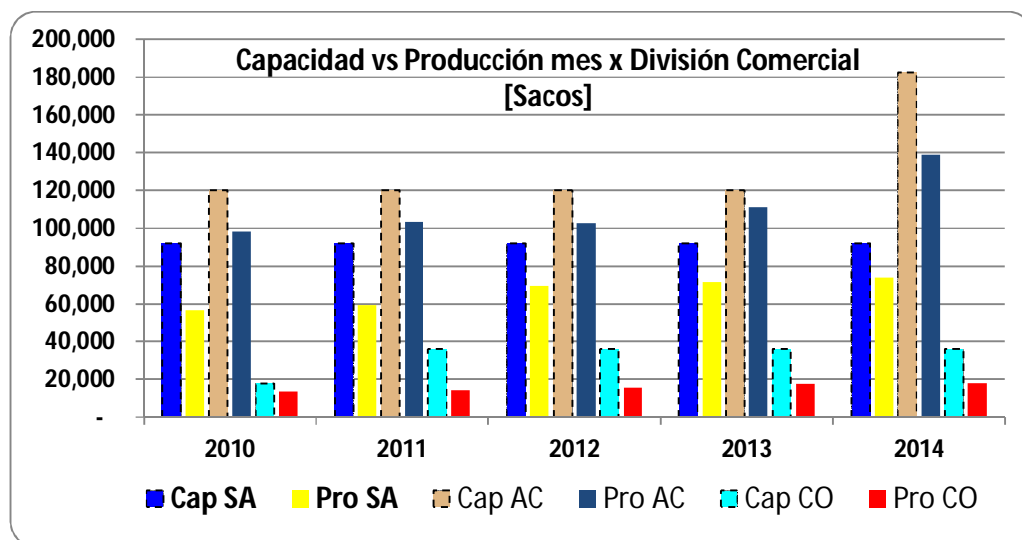
Los productos de estas 3 divisiones comerciales componen el volumen de total de producción de la fábrica en estudio y que en los últimos años ha tenido el siguiente volumen promedio mensual por división comercial en sacos de 40 kilogramos (ver figura 3.5)



**Figura 3.5 Producción mensual promedio por División Comercial**

Fuente: Estadísticas fábrica en estudio - Elaboración: Propia

Como se puede apreciar, las 3 divisiones comerciales presentan un crecimiento año tras año y esto se refleja en el volumen total de producción mostrado inicialmente. Ahora se contrastará el volumen versus la capacidad instalada para cada división comercial en sacos de 40 kilos (ver figura 3.6)

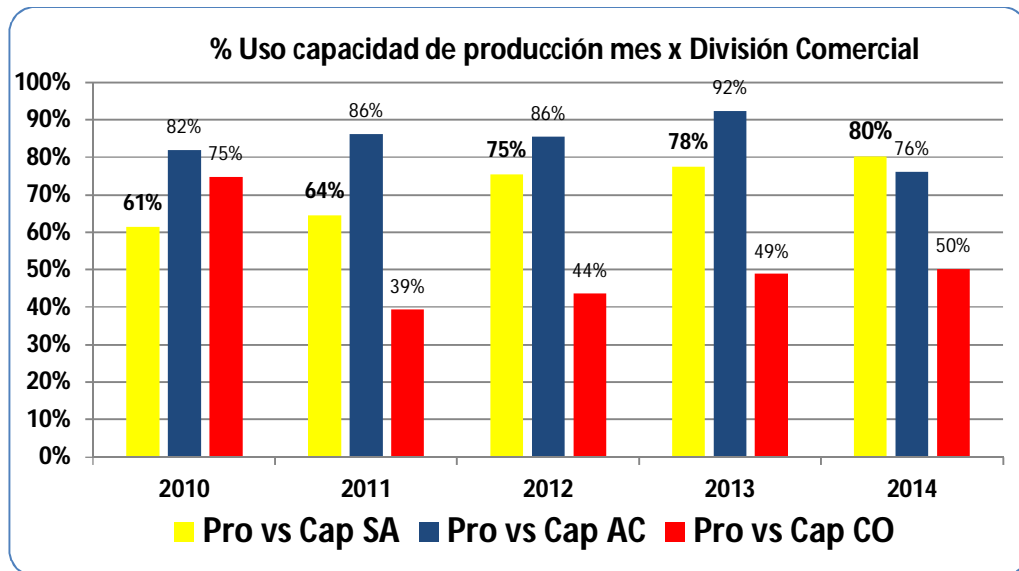


**Figura 3.6 Capacidad vs Producción mensual por División Comercial**

Fuente: Estadísticas fábrica en estudio - Elaboración: Propia

Se puede apreciar, que se ha tenido capacidad suficiente para poder atender la demanda en cada año en todas las divisiones comerciales, especialmente en la División Consumo, que a partir del año 2011, arrancó con una nueva línea de producción de mayor capacidad y del mismo modo para la División Acuicultura, que en el año 2014, arrancó una nueva línea de producción para la elaboración de alimento peletizado para camarón.

Con respecto a la División Salud Animal, la capacidad actual de producción de la línea ha sido mayor a la demanda y por ende sus niveles de producción fueron suficientes para atender las necesidades de esta división, se aprecia que cada año esa holgura se ha venido reduciendo, esto se aprecia mejor en el siguiente cuadro de % de ocupación de las líneas de producción por división comercial (ver figura 3.7)



**Figura 3.7 % Uso capacidad de producción mes por División Comercial**

Fuente: Estadísticas fábrica en estudio - Elaboración: Propia

Como se puede apreciar en la figura 3.7, desde el año 2014, el porcentaje de capacidad de producción de la línea de producción de la División Salud Animal, es la más alta y no se tiene, al menos para este año 2015 y 2016, la adquisición o repotenciación de la línea de producción, lo que implica que cada hora disponible de producción debe ser aprovechada de la mejor manera para no



tener problemas de abastecimiento de los productos que conforman esta división comercial en el corto y mediano plazo.

Adicionalmente, esta división comercial, cada vez, ha aumentado su portafolio de SKU, lo que implica un mayor tiempo dedicado a la preparación de máquinas (set-ups), mermando aún más las horas disponibles para la producción de alimento. Por estos motivos, se toma la decisión de analizar la programación de producción de esta línea de producción de alimento balanceado.

### 3.3 Descripción del proceso de producción de alimento peletizado

En la sección anterior, se justificó el motivo de la selección de la línea de producción de alimento peletizado que abastece de productos terminados a la División Salud Animal. A continuación, se detallará en forma breve, el proceso de producción, desde la mezcla de las materias primas hasta el envasado de los sacos de alimento peletizado (sacos de 40Kg).

La peletización es una técnica de procesamiento que convierte una mezcla de ingredientes molidos finamente en aglomerados de diferentes tamaños (pelets). En la figura 3.8 se muestra un diagrama esquemático de todo el flujo de la línea de producción, la misma se la puede globalizar como una sola gran máquina pero para efectos de desarrollo de esta tesis, se desglosará en sus principales 4 subprocesos, los mismos que más adelante serán considerados en la elaboración del programa de producción, más detalle en el Anexo A

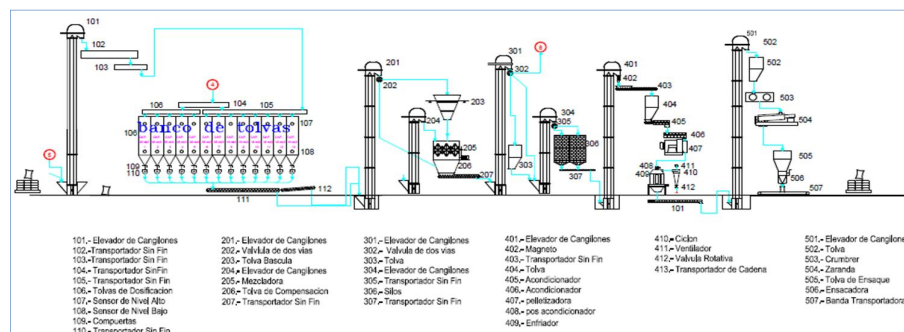


Figura 3.8 Diagrama del proceso de producción de la línea SA

Fuente: Fábrica en estudio - Elaboración: Propia

**1.- El proceso de la molienda:** Es el primer procesamiento que sufren las materias primas, conocidas como los macro ingredientes, a través de molinos de martillos, se reduce la granulometría de los mismos y son depositados en tolvas de forma independiente hasta la siguiente proceso.

**2.- El Proceso de mezclado:** una vez que las materias primas han sido molidas finamente, estas se dosifican, según la fórmula nutricional, dentro de un mezclador de tambor, que tiene una capacidad de 2,000 Kg por lote (batch) y por un período de 300 segundos aproximadamente con el objetivo de lograr una mezcla homogénea tanto de los macro ingredientes molidos, los aditivos líquidos y el núcleo formado por vitaminas y minerales.

**3.- El proceso de peletización:** este es el proceso principal de la línea de producción, aquí las materias primas son sometidas a un proceso físico de temperatura y presión, de forma resumida este proceso abarca 3 etapas: acondicionamiento, compresión y enfriamiento del pelet.

El acondicionamiento: consiste en la preparación de la masa húmeda por medio de vapor y calor, la calidad del vapor es vital en esta etapa.

La compresión: o peletización, aquí la masa es sometida a una presión y temperatura por medio de un rodillo, donde se obliga a pasar por un dado de diámetro específico y luego cortado al tamaño requerido.

El enfriamiento: una vez peletizado, el pelet es sometido a un enfriamiento con el objetivo de reducir la humedad y calor.

**4.- El proceso de envasado:** finalmente, una vez que el pelet esta enfriado y duro, es llevado por medio de un elevador de cangilones hacia una tolva de ensaque, la misma que tiene una balanza calibrada para pesar principalmente sacos de 40Kg +/- un margen de tolerancia aprobado por el departamento de calidad (+/- 0.5% del peso objetivo), adicionalmente la ensacadora posee un par de brazos mecánicos que sostiene el saco laminado mientras se está llenando con producto, una vez llenado, los sacos laminados son cosidos e identificados por medio de marbetes de cartulina donde se detalla: el número de lote, fecha de producción y fecha de caducidad (45 días luego de haber sido

manufacturado). Finalmente, por medio de una banda transportadora y mano de obra, el saco es ubicado en pallets de madera, cuya unidad de carga es de 40 sacos por pallet.

### 3.4 Situación actual del proceso de programación de la producción de la línea

Como se indicó en secciones anteriores, hace pocos años, la fábrica en estudio fue adquirida en su totalidad por la compañía, la misma que no tenía experiencia previa en la producción de alimentos balanceados de ninguna especie dado que todos estos eran maquilados en fábricas externas.

Al adquirir el 100% del paquete accionario de la fábrica, la compañía adquirió también, el conocimiento y experiencia empírica de todos los trabajadores que laboran allí, y como era de esperarse, el criterio para elaborar la programación de la producción de la línea para la división de salud animal, no fue la excepción. A continuación, se detallará todos los productos terminados que esta línea de producción elabora, sus volúmenes semanales promedio (demanda), las características inherentes al proceso como: tamaño del pelet, tiempos de preparación de máquinas (setups) y demás información relevante.

Primeramente, se detallará los productos terminados que actualmente están activos y son producidos en esta línea de producción (ver tabla 3.1)

#	Categoría	Sub categoría	Código	Descripción	Peso [Kg / PT]	Parada [Kg/Batch]	Matriz [mm]	Und x Parada
1	Balanceado	Cerdos	CINI	CERDOS INICIAL 18%	40	2000	4.00	50
2	Balanceado	Cerdos	CGES	CERDAS GESTANTES	40	2000	5.00	50
3	Balanceado	Cerdos	CPF1	CERDOS PRE-INICIAL FASE 1	20	2000	3.20	100
4	Balanceado	Cerdos	CPF2	CERDOS PRE-INICIAL FASE 2	20	2000	3.20	100
5	Balanceado	Cerdos	CPF3	CERDOS PRE-INICIAL FASE 3	20	2000	3.20	100
6	Balanceado	Cerdos	CACB	CERDO ACABADO 14%	40	2000	5.00	50
7	Balanceado	Cerdos	CACM	CERDOS ACABADO MEDICADO	40	2000	5.00	50
8	Balanceado	Cerdos	CFIN	CERDO FINAL 16%	40	2000	5.00	50
9	Balanceado	Cerdos	CLAC	CERDO LACTANCIA PLUS	40	2000	5.00	50
10	Balanceado	Ganado	GL18	GANADO LECHERO 18% COSTA	40	2000	8.00	50
11	Balanceado	Ganado	GTER	GANADO TERNERAS	40	2000	5.00	50
12	Balanceado	Ganado	GL14	GANADO LECHERO 14%	40	2000	8.00	50
13	Balanceado	Ganado	G19R	GANADO LECHERO 19% CON RUMENSIN	40	2000	5.00	50

14	Balanceado	Ganado	GTIN	GANADO TERN.INIC.19%	40	2000	5.00	50
15	Balanceado	Ganado	GP14	GANADO 14% POLVO	40	2000	0.00	50
16	Balanceado	Pollo	PPRE	BROILER PRE-INICIO COSTA 23%	40	2000	3.20	50
17	Balanceado	Pollo	PICO	BROILER INICIO COSTA 21%	40	2000	4.00	50
18	Balanceado	Pollo	PFCO	BROILER FINAL COSTA 19%	40	2000	4.00	50
19	Balanceado	Tilapia	TI24	FISHPAC 24% 20KLS.	20	2000	4.70	100
20	Balanceado	Tilapia	TI28	FISHPAC 28% 20KLS.	20	2000	4.70	100
21	Balanceado	Tilapia	TI32	FISHPAC 32% 20KLS.	20	2000	3.00	100
22	Balanceado	Tilapia	TI38	FISHPAC 38% 20KLS.	20	2000	2.38	100
23	Balanceado	Trucha	TR46	TRUCHA CRECIMIENTO 46% 20 KG	20	2000	2.38	100
24	Balanceado	Trucha	TR40	TRUCHA ENGORDE 40% 20 KG	20	2000	4.70	100

**Tabla 3.1 Listado de productos manufacturados en la línea SA**

Fuente: Fábrica en estudio - Elaboración: Propia

En la tabla mostrada se puede apreciar que actualmente tenemos lo siguiente:

- 9 productos para cerdos,
- 6 productos para ganado,
- 3 productos para pollos,
- 4 productos para tilapia y
- 2 productos para trucha.

Lo que da un total de 24 productos que actualmente esta línea de producción elabora semanalmente. Se muestra también la presentación que tiene cada SKU, casi todos son sacos de 40Kg pero tenemos algunos productos cuyo formato de comercialización son sacos de 20Kg.

La línea de producción trabaja por lotes o batchs de producción de 2000 kilogramos, debido a que la capacidad máxima de la mezcladora es de 2000 kilogramos, esto quiere decir que los programas de producción deben ser múltiplos de esta capacidad, en otras palabras, para los productos de presentaciones de 40Kg, los planes de producción son múltiplos de 50 unidades (50 sacos x 40Kg = 2,000 Kg que es el batch o lote de producción), en el caso de los SKU´s de formatos de 20Kg corresponden a 100 unidades cada lote de producción. Como se mencionó, el proceso de la línea de producción se compone de 4 grandes subprocesos, a continuación se detalla los ratios estándar para cada uno de ellos por cada producto (ver tabla 3.2)

#	Categoría	Sub categoría	Código	Descripción	Ratio Molienda [Ton/hora]	Ratio Mezclado [Ton/hora]	Ratio Peletizadora [Ton/hora]	Ratio Envasado [Ton/hora]
1	Balanceado	Cerdos	CINI	CERDOS INICIAL 18%	14	11	6.58	11
2	Balanceado	Cerdos	CGES	CERDAS GESTANTES	14	13	6.07	11
3	Balanceado	Cerdos	CPF1	CERDOS PRE-INICIAL FASE 1	13	11	1.90	11
4	Balanceado	Cerdos	CPF2	CERDOS PRE-INICIAL FASE 2	14	11	2.14	10
5	Balanceado	Cerdos	CPF3	CERDOS PRE-INICIAL FASE 3	14	12	2.88	10
6	Balanceado	Cerdos	CACB	CERDO ACABADO 14%	14	12	7.12	10
7	Balanceado	Cerdos	CACM	CERDOS ACABADO MEDICADO	15	11	6.96	10
8	Balanceado	Cerdos	CFIN	CERDO FINAL 16%	15	13	6.41	10
9	Balanceado	Cerdos	CLAC	CERDO LACTANCIA PLUS	13	13	6.41	11
10	Balanceado	Ganado	GL18	GANADO LECHERO 18% COSTA	15	11	6.60	10
11	Balanceado	Ganado	GTER	GANADO TERNERAS	15	12	5.65	11
12	Balanceado	Ganado	GL14	GANADO LECHERO 14%	15	13	7.01	10
13	Balanceado	Ganado	G19R	GANADO LECHERO 19% CON RUMENSIN	13	13	6.00	12
14	Balanceado	Ganado	GTIN	GANADO TERN.INIC.19%	14	12	7.45	11
15	Balanceado	Ganado	GP14	GANADO 14% POLVO	13	12	2.00	12
16	Balanceado	Pollo	PPRE	BROILER PRE-INICIO COSTA 23%	15	13	7.60	11
17	Balanceado	Pollo	PICO	BROILER INICIO COSTA 21%	15	13	7.60	12
18	Balanceado	Pollo	PFCO	BROILER FINAL COSTA 19%	15	11	7.85	12
19	Balanceado	Tilapia	TI24	FISHPAC 24% 20KLS.	14	13	2.86	11
20	Balanceado	Tilapia	TI28	FISHPAC 28% 20KLS.	13	11	1.85	11
21	Balanceado	Tilapia	TI32	FISHPAC 32% 20KLS.	14	13	1.85	11
22	Balanceado	Tilapia	TI38	FISHPAC 38% 20KLS.	13	11	1.85	10
23	Balanceado	Trucha	TR46	TRUCHA CRECIMIENTO 46% 20 KG	13	11	2.86	11
24	Balanceado	Trucha	TR40	TRUCHA ENGORDE 40% 20 KG	14	11	2.86	11

**Tabla 3.2 Listado de ratios de producción por proceso y producto**

Fuente: Fábrica en estudio - Elaboración: Propia

A continuación, se muestran los volúmenes de producción que actualmente se manejan por cada producto que se manufacturan en la línea de producción (ver tabla 3.3) y los tiempos promedio para cada una de las operaciones de los 4 procesos acorde los ratios mostrados en la tabla anterior, con esta información se elaborará el programa de producción que actualmente rige en la fábrica en estudio, adicionalmente, se levantó la información de los tiempos de preparación (set-ups) para el subproceso de peletización, dado que es el cuello de botella de toda la línea y sus tiempos son representativos, el detalle se muestra en el Anexo B.

Código	Descripción	Plan Semanal Producción [UND]	Plan Semanal Producción [KG]	Molienda [Horas]	Mezclado [Horas]	Peletizadora [Horas]	Envasado [Horas]
CINI	CERDOS INICIAL 18%	300	12,000	1	2	2	2
CGES	CERDAS GESTANTES	250	10,000	1	1	2	1
CPF1	CERDOS PRE-INICIAL FASE 1	200	4,000	1	1	3	1
CPF2	CERDOS PRE-INICIAL FASE 2	300	6,000	1	1	3	1
CPF3	CERDOS PRE-INICIAL FASE 3	400	8,000	1	1	3	1
CACB	CERDO ACABADO 14%	1,000	40,000	3	4	6	4
CACM	CERDOS ACABADO MEDICADO	200	8,000	1	1	2	1
CFIN	CERDO FINAL 16%	1,300	52,000	4	4	9	6
CLAC	CERDO LACTANCIA PLUS	250	10,000	1	1	2	1
GL18	GANADO LECHERO 18% COSTA	500	20,000	2	2	4	2
GTER	GANADO TERNERAS	250	10,000	1	1	2	1
GL14	GANADO LECHERO 14%	900	36,000	3	3	6	4
G19R	GANADO LECHERO 19% CON RUMENSIN	250	10,000	1	1	2	1
GTIN	GANADO TERN.INIC.19%	250	10,000	1	1	2	1
GP14	GANADO 14% POLVO	200	8,000	1	1	4	1
PPRE	BROILER PRE-INICIO COSTA 23%	500	20,000	2	2	3	2
PICO	BROILER INICIO COSTA 21%	5,000	200,000	14	16	27	17
PFCO	BROILER FINAL COSTA 19%	5,100	204,000	14	19	26	17
TI24	FISHPAC 24% 20KLS.	200	4,000	1	1	2	1
TI28	FISHPAC 28% 20KLS.	200	4,000	1	1	3	1
TI32	FISHPAC 32% 20KLS.	300	6,000	1	1	4	1
TI38	FISHPAC 38% 20KLS.	100	2,000	1	1	2	1
TR46	TRUCHA CRECIMIENTO 46% 20 KG	100	2,000	1	1	1	1
TR40	TRUCHA ENGORDE 40% 20 KG	100	2,000	1	1	1	1

**Tabla 3.3 Listado de tiempos de operación por proceso y producto**

Fuente: Fábrica en estudio - Elaboración: Propia

Como se mencionó en secciones anteriores, la compañía hace pocos años adquirió en su totalidad el 100% del paquete accionario de la fábrica en estudio, con dicha adquisición, la compañía adquirió toda el conocimiento y experiencia en la elaboración de alimentos balanceados, tanto el proceso de elaboración, los controles de calidad, las fórmulas nutricionales de cada producto y también las reglas para elaborar los programas de producción.

Como se definió previamente, se centrará el análisis en la línea de producción de alimento peletizado que abastece a la división salud animal, esta línea de producción tiene la particularidad que elabora actualmente alimentos para 5 tipos diferentes de animales: ganado, cerdo, pollo, tilapia y truchas.

Actualmente, tenemos 3 reglas empíricas para elaborar la programación de la producción en la línea de producción para salud animal:

1. **Regla de la agrupación:** esta primera regla indica que la producción debe ser en agrupamientos o bloques de especie, es decir, todos los productos de alimentos cerdo, todos los de pollo, todos los de ganado, etc.
2. **Regla del color:** esta segunda regla indica que la producción debe ir desde los colores más oscuros hacia los más claros.
3. **Regla del tamaño:** esta tercera regla indica que la producción debe ir desde los productos de mayor diámetro de pelet [mm] hacia los de menor diámetro de pelet [mm].

Resumiendo estas 3 reglas empíricas que actualmente son usadas para la programación de la producción de los productos de la división Salud Animal, tenemos lo siguiente (ver figura 3.9):



**Figura 3.9 Resumen de las 3 reglas empíricas actuales para la programación de la producción**

Fuente: Fábrica en estudio - Elaboración: Propia

La secuencia que mejor se ajusta a estas 3 reglas empíricas es la que se muestra en la figura 3.9, donde se muestra que: un programa de producción en esta línea de producción debe empezar con los alimentos para ganado, luego para tilapia, continua con los alimentos para trucha, continua con los alimentos para cerdo y finalmente termina la producción con los alimentos para pollos.

En el siguiente capítulo IV, se revisará en detalle la situación actual de la programación de la producción de la línea de salud animal y con la ayuda del software LEKIN, se contrastará este programa de producción y sobretodo, se analizarán los resultados obtenidos versus otras reglas de programación, este será el punto de partida y referencia a analizar versus un programa de producción propuesto que sea mejor, acorde el objetivo principal de esta tesis.

## CAPÍTULO IV

### PROPUESTA DE MEJORA DE LA PROGRAMACIÓN DE LA LÍNEA DE PRODUCCIÓN

#### 4.1 Programación actual de la línea de producción y sus resultados.

En el capítulo anterior, se mostraron los productos, los tiempos de procesamiento, los tiempos de preparación de máquinas, sus volúmenes de producción y las reglas empíricas utilizadas actualmente para la programación de la producción. Toda esta información es ingresada en el software LEKIN con el objetivo de cuantificar el desempeño de la situación actual.

En la figura 4.1 se muestra el ingreso de los 4 subprocessos: molienda, mezclado, peletización y envasado, y para el caso del proceso de peletización, se adiciona la matriz de tiempos de preparación de cada uno de los productos.

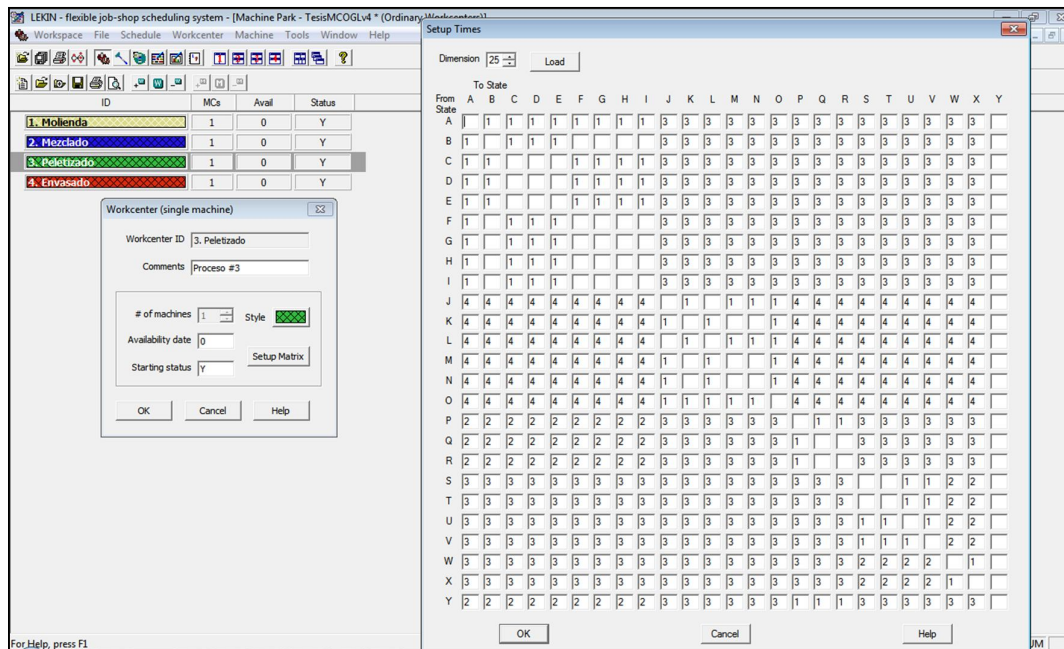
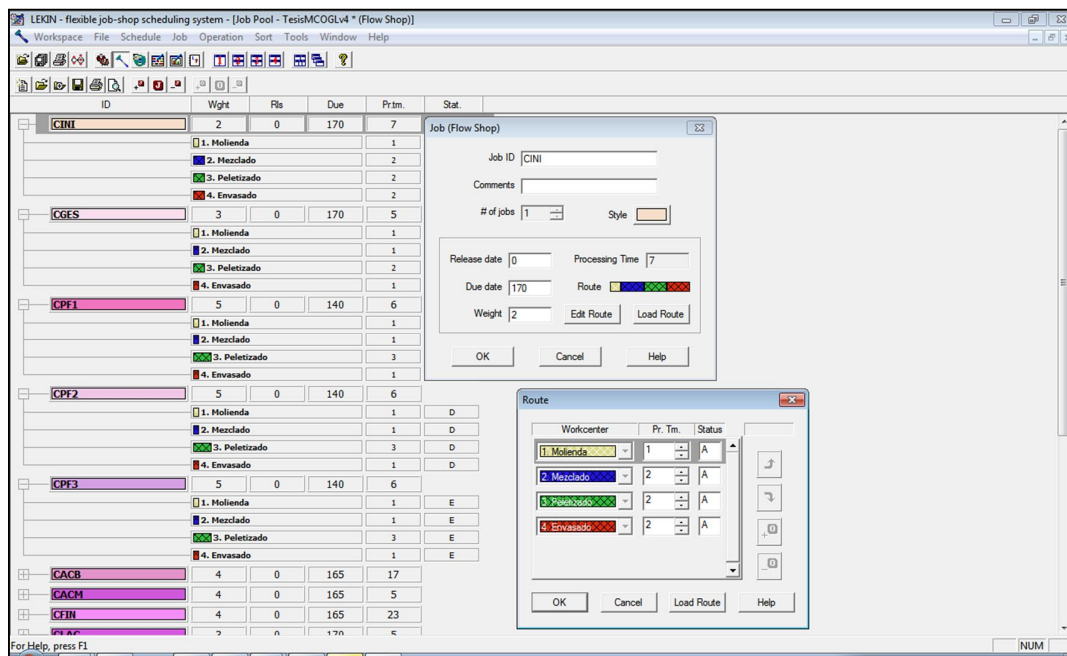


Figura 4.1 Parametrización en LEKIN de las máquinas

Fuente: LEKIN - Elaboración: Propia



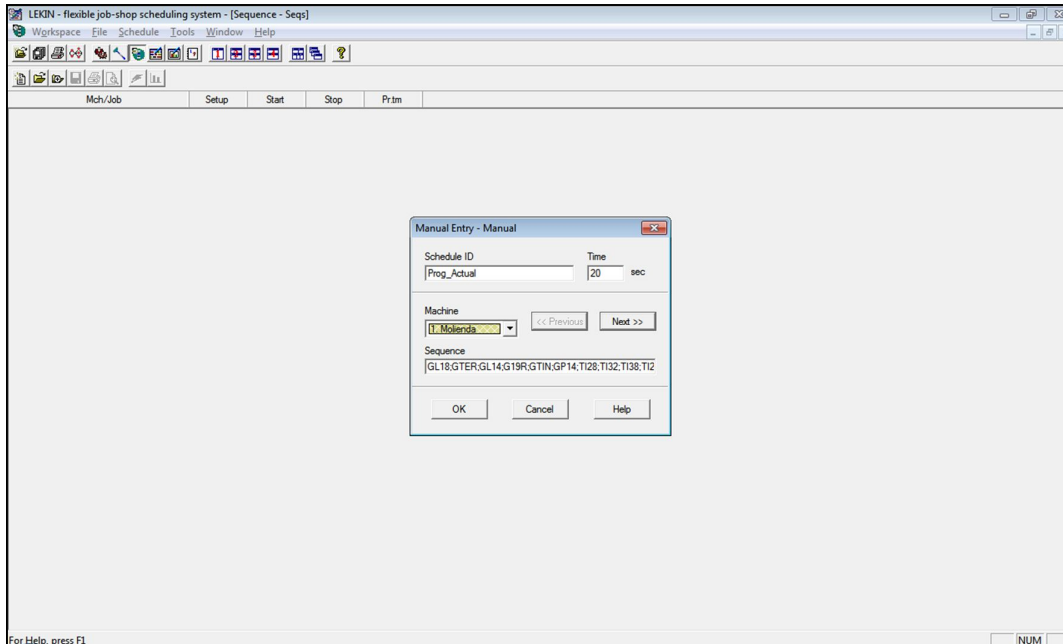
Terminado de ingresar los datos de las 4 máquinas (para realizar los 4 subprocesos de toda la línea de producción), continuamos con los datos de cada uno de los trabajos o productos terminados (ver figura 4.2): sus tiempos de operación, sus tiempos de liberación, sus tiempos de terminación y además se asocia el campo llamado estatus, este campo sirve para poder unir los tiempos de preparación de cada producto con las respectivas matrices de set-ups que se parametrizó en la sección anterior (máquinas), este campo es de gran importancia pues el tiempo de culminación del último trabajo (makespan) de un programa de producción cualquiera es dependiente de los tiempos de preparación asociados a dicho programa.



**Figura 4.2 Configuración en LEKIN de los trabajos**

Fuente: LEKIN - Elaboración: Propia

Una vez ingresado toda la información de las máquinas y de los trabajos a procesar en LEKIN, definiremos el programa de producción acorde la situación actual, para esto utilizaremos la función de ingreso manual de programación de la producción como se muestra en la figura 4.3, en ella se aprecia como se respeta las 3 reglas empíricas que actualmente se considera en la fábrica en estudio para elaborar el programa de producción para cada uno de los 4 subprocesos que conforman la línea de producción.



**Figura 4.3 Ingreso manual en LEKIN de la programación actual**

Fuente: LEKIN - Elaboración: Propia

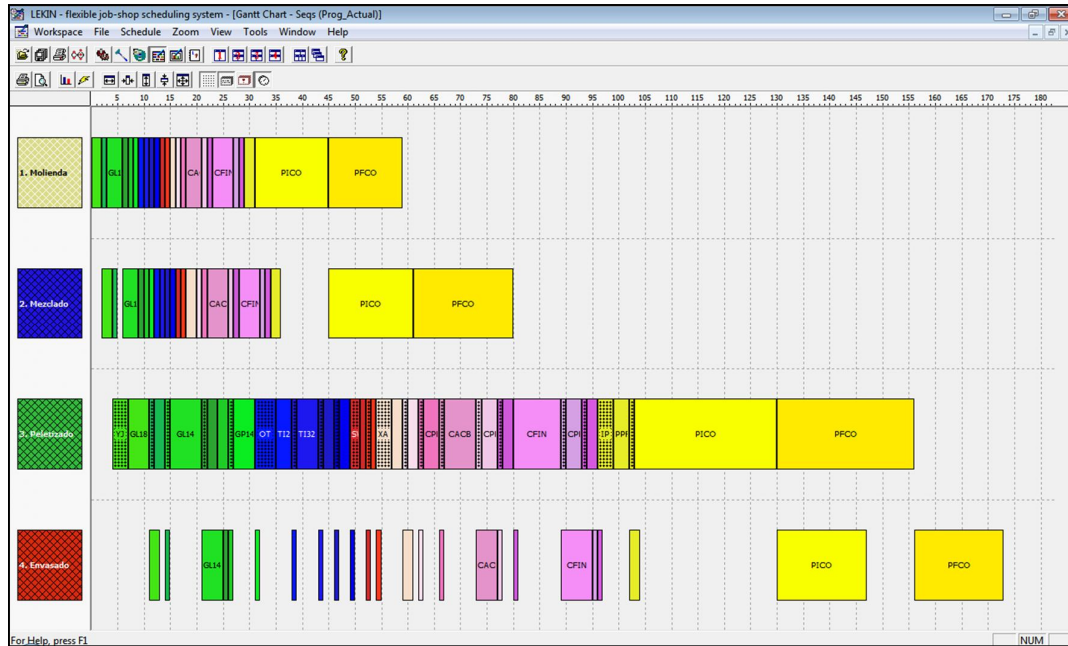
Una vez ingresada las reglas empíricas actuales, el programa LEKIN muestra los siguientes resultados:

Mch/Job	Setup	Start	Stop	Prm
1. Molienda	0			59
2. Mezclado	0			68
3. Pelletizado	31			121
4. Envasado	0			70
GL18	0	11	13	2
GT18	0	14	15	1
GT14	0	21	25	4
GT19R	0	25	26	1
GTIN	0	26	27	1
GP14	0	31	32	1
TI28	0	38	39	1
TI32	0	43	44	1
TI38	0	45	47	1
TI14	0	49	50	1
TR46	0	52	53	1
GT18	0	54	55	1
GTIN	0	59	61	2
CG45	0	62	63	1
GP14	0	66	67	1
GCACB	0	73	77	4
CFP2	0	77	78	1
GCACB	0	80	81	1
GTIN	0	89	95	6
CFP3	0	95	96	1
GLAC	0	96	97	1
PPR2	0	102	104	2

**Figura 4.4 Secuencias de las máquinas del programa actual**

Fuente: LEKIN - Elaboración: Propia

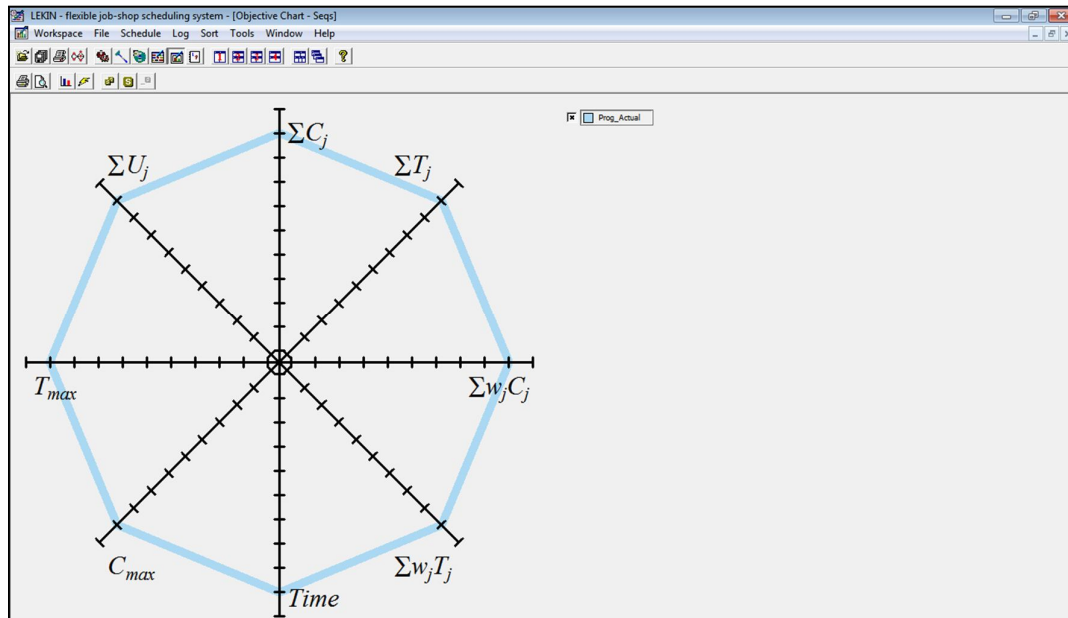
En la figura 4.4 se muestra la secuencia por cada máquina y sus tiempos de parametrización correspondientes.



**Figura 4.5 Diagrama de Gantt del programa actual**

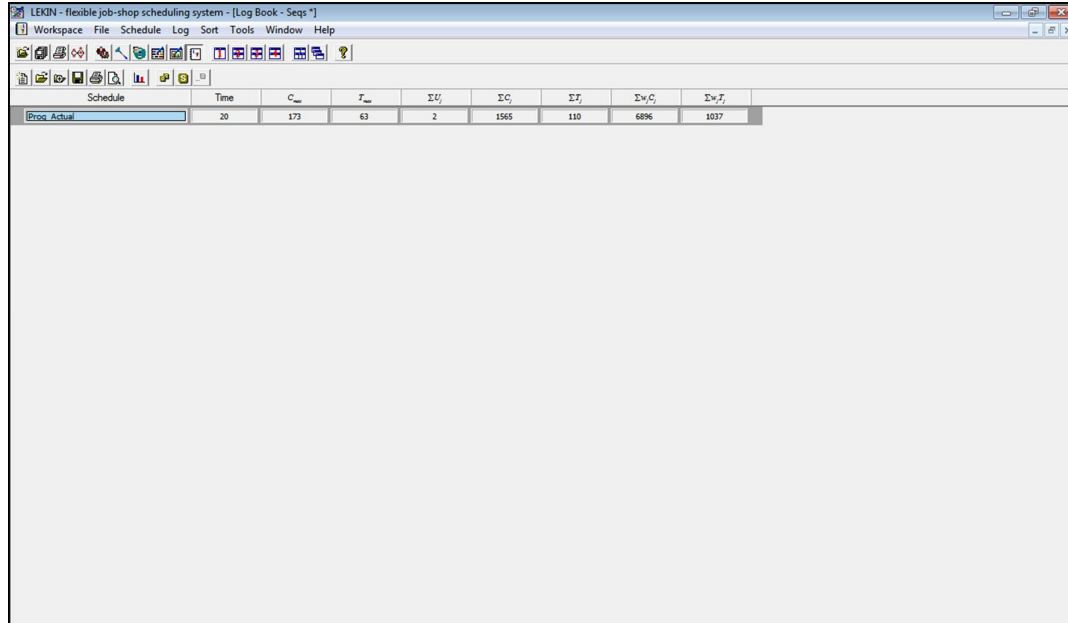
Fuente: LEKIN - Elaboración: Propia

La figura 4.5, detalla cómo se cumplen las 3 reglas empíricas en el programa actual de producción que actualmente rige en la fábrica en estudio.



**Figura 4.6 Diagrama de resultados obtenidos del programa actual**

Fuente: LEKIN - Elaboración: Propia



**Figura 4.7 Resumen del desempeño del programa actual**

Fuente: LEKIN - Elaboración: Propia

En la figura 4.7 se muestra los resultados de los indicadores de desempeño que tiene el actual programa de producción, para apreciarlos de mejor manera, se describe mejor en la siguiente tabla

Resultados	Criterio	Volumen	Time	Cmax	Tmax	$\Sigma U_j$	$\Sigma C_j$	$\Sigma T_j$	$\Sigma w_j C_j$	$\Sigma w_j T_j$
Programa Actual	Reglas empíricas fábrica en estudio	17,200	20	173	63	2	1,565	110	6,896	1,037

**Tabla 4.1 Resultados del programa actual de producción**

Fuente: LEKIN - Elaboración: Propia

Ahora viene la parte interesante, el análisis de los resultados obtenidos del actual programa de producción: se puede apreciar que el makespan ( $C_{max}$ ) es de 173 horas, es decir, el último trabajo en salir (que es un alimento para pollos) termina en la hora 173, es decir, una corrida de 7 días para una producción de 17,200 sacos de 40Kg.

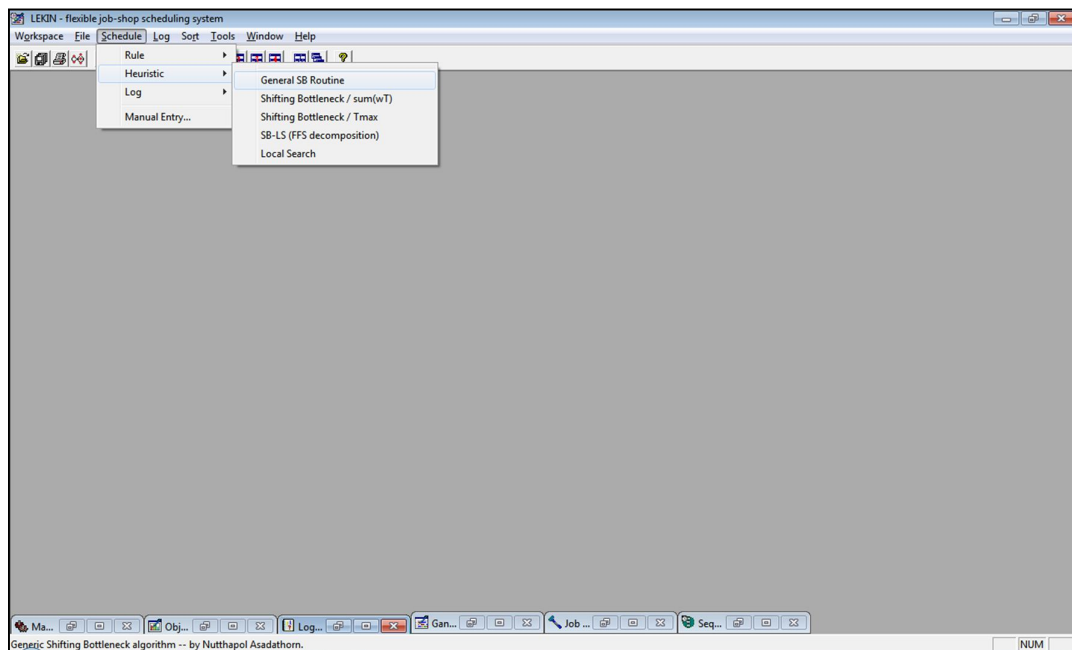
La tardanza máxima ( $T_{max}$ ) es de 63 horas, es decir, un trabajo tuvo 63 horas tarde (tiempo de terminación vs el tiempo de entrega).

La sumatoria de penalidades ( $\Sigma U_j$ ) es de 2, es decir, existen dos trabajos que su terminación fue después de su fecha de entrega.

Se considera que estos son los 3 principales indicadores que aportan más valor y serán los determinantes para decidir si la propuesta es mejor o no que la situación actual.

## 4.2 Propuesta de un programa de producción con el uso del software LEKIN

En esta sección, se detalla la propuesta de un mejor programa de producción por medio de la herramienta LEKIN, como se mencionó en el capítulo II, LEKIN posee algunas reglas básicas y compuestas de programación (ver figura 4.8)



**Figura 4.8 Vista de las heurísticas de programación en LEKIN**

Fuente: LEKIN - Elaboración: Propia

Una vez que se ejecutan todas las reglas y heurísticas que posee LEKIN, podemos ver, de forma resumida, los resultados de cada una de ellas, incluyendo la situación actual analizada en la sección anterior (ver figura 4.9)

Ahora tenemos que analizar y decidir, si alguna de las propuestas planteadas es mejor que la situación actual y cuantificar su desempeño versus la situación actual. Como se mencionó en la sección anterior: de los indicadores de desempeño que muestra LEKIN, vamos a enfocarnos en 3: el makespan ( $C_{max}$ ),

la tardanza máxima ( $T_{max}$ ) y la sumatoria de penalidades ( $\sum U_j$ ), en todos ellos, se busca su minimización.

Schedule	Time	$C_{max}$	$T_{max}$	$\sum U_j$	$\sum C_j$	$\sum T_j$	$\sum w_j C_j$	$\sum w_j T_j$
FCFS	1	151	35	2	1811	54	5080	505
General SB Routine / Cmax	3	160	2	1	2598	2	5927	18
WSPT	1	161	19	2	1984	22	4659	201
Prog Actual	20	173	63	2	1565	110	6896	1037
EDD	1	175	0	0	3186	0	8107	0
SPT	1	175	75	2	1307	113	5456	1092
MS	1	179	0	0	3240	0	8182	0
SB-LS / sum(wT)	6	179	0	0	3150	0	8082	0
DASH	8	180	29	5	3045	67	8228	343
Shifting Bottleneck / sum(wT)	2	186	66	4	2130	132	6919	972
ATCS (0.5,0.5)	1	189	24	1	3617	24	9496	240
LPT	1	209	31	12	3981	184	10731	823
CR	1	221	24	14	4196	147	10858	511

**Figura 4.9 Resultados de las propuestas de programas de producción**

Fuente: LEKIN - Elaboración: Propia

En la figura 4.9 se muestra las medidas de desempeño de todas las propuestas de programas de producción por LEKIN, para elegir la mejor, utilizaremos una matriz de ponderación, en donde se da un peso a los 3 indicadores mencionados anteriormente y se muestra en la tabla 4.2

Propuestas Programas Producción (Heurísticas)	Time	Cmax	Tmax	$\sum U_j$	$\sum C_j$	$\sum T_j$	$\sum w_j C_j$	$\sum w_j T_j$	0%	40%	30%	30%	0%	0%	0%	0%
FCFS	1	151	35	2	1,811	54	5,080	505								
General SB Routine / Cmax	3	160	2	1	2,598	2	5,927	18								
WSPT	1	161	19	2	1,984	22	4,659	201								
EDD	1	175	0	0	3,186	0	8,107	0								
SPT	1	175	75	2	1,307	113	5,456	1,092								
MS	1	179	0	0	3,240	0	8,182	0								
SB-LS / sum (wT)	6	179	0	0	3,150	0	8,082	0								
DASH	8	180	29	5	3,045	67	8,228	343								
Shifting Bottleneck / sum (wT)	2	186	66	4	2,130	132	6,919	972								
ATCS	1	189	24	1	3,617	24	9,496	240								
LPT	1	209	31	12	3,981	184	10,731	823								
CR	1	221	24	14	4,196	147	10,858	511								

**Tabla 4.2 Matriz de ponderación de resultados de las propuestas**

Fuente: LEKIN - Elaboración: Propia

Como se aprecia en la tabla 4.2, solo tres de los ocho indicadores tienen un peso %, de tal forma, que la propuesta de LEKIN con el valor total más

pequeño (debido a que se busca la minimización de los 3 indicadores), será el programa de producción seleccionado, el resultado se muestra en la tabla 4.3

Propuestas Programas Producción Heurística:	Time	Cmax	Tmax	$\Sigma U_j$	$\Sigma C_j$	$\Sigma T_j$	$\Sigma w_j C_j$	$\Sigma w_j T_j$	TOTAL
FCFS	0	60	11	1	0	0	0	0	72
<b>General SB Routine / Cmax</b>	<b>0</b>	<b>64</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>65</b>
WSPT	0	64	6	1	0	0	0	0	71
EDD	0	70	0	0	0	0	0	0	70
SPT	0	70	23	1	0	0	0	0	93
MS	0	72	0	0	0	0	0	0	72
SB-LS / sum (wT)	0	72	0	0	0	0	0	0	72
DASH	0	72	9	2	0	0	0	0	82
Shifting Bottleneck / sum (wT)	0	74	20	1	0	0	0	0	95
ATCS	0	76	7	0	0	0	0	0	83
LPT	0	84	9	4	0	0	0	0	97
CR	0	88	7	4	0	0	0	0	100

Tabla 4.3 Selección de la mejor propuesta de programa de producción

Fuente: LEKIN - Elaboración: Propia

Acorde la matriz, el programa de producción seleccionado es por la heurística de la rutina del cuello de botella desplazante (General SB Routine /C<sub>max</sub>).

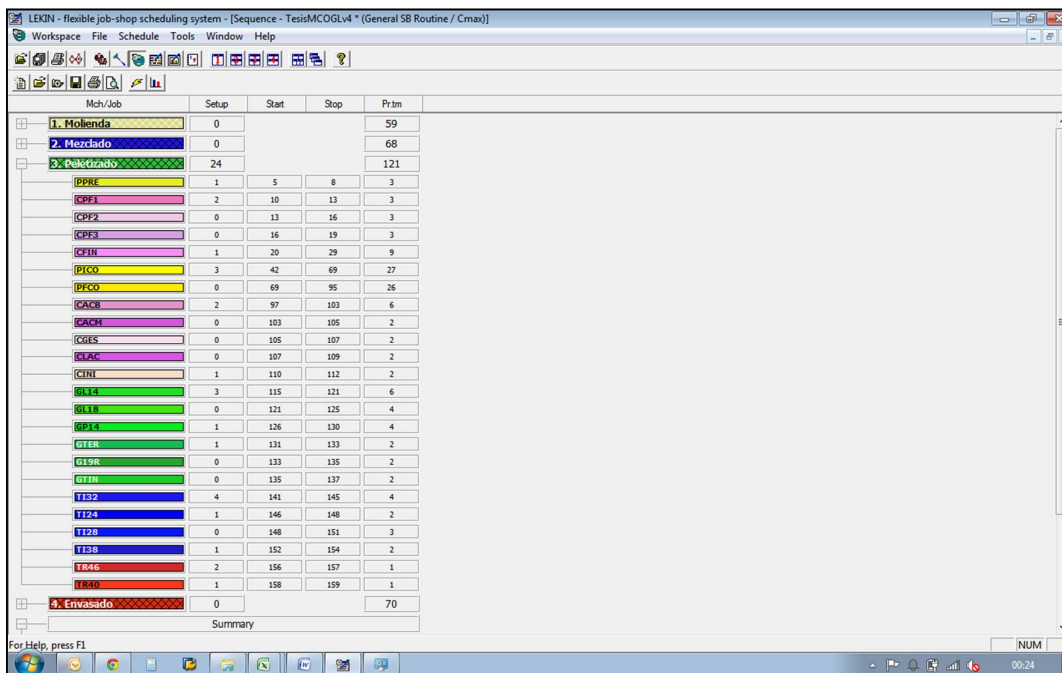
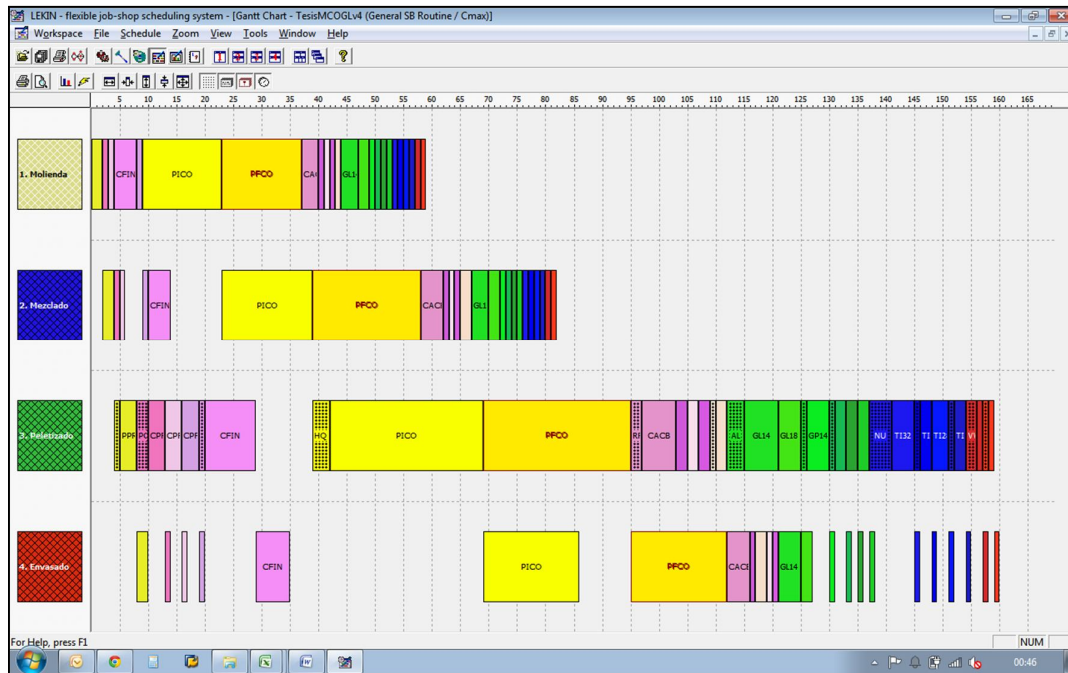


Figura 4.10 Vista de la secuencia del programa elegido

Fuente: LEKIN - Elaboración: Propia

En la figura 4.10 se muestra la secuencia de los trabajos que el programa seleccionado propone (pollo → cerdo → pollo → cerdo → ganado → tilapia → trucha), el cual es muy diferente a la situación actual (ganado → tilapia → trucha → cerdo → pollo).

El programa de producción completo, para cada subprocesso de la línea de producción, se muestra a continuación (ver figura 4.11)

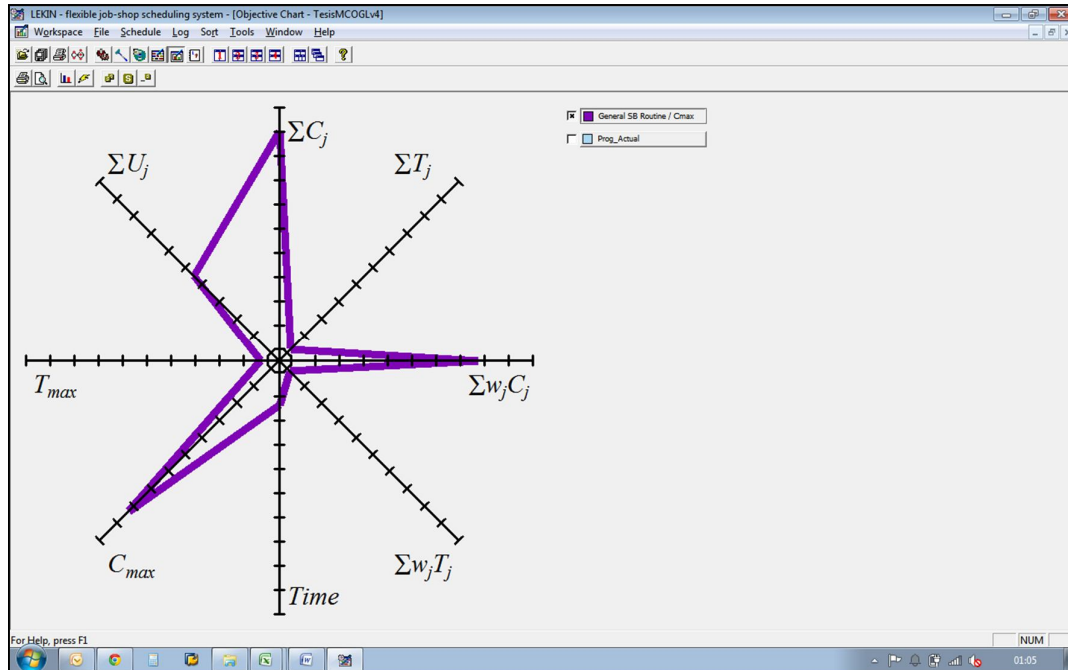


**Figura 4.11 Diagrama de Gantt del programa de producción elegido**

Fuente: LEKIN - Elaboración: Propia

En la figura 4.12, se muestra el diagrama de resultados del programa de producción propuesto elegido, a simple vista se aprecia que los principales indicadores, seleccionados anteriormente, muestran una mejora versus el programa actual de producción, más adelante se hará la comparación entre los 2 programas de producción.





**Figura 4.12 Vista diagrama de los resultados del programa de producción propuesto**

Fuente: LEKIN - Elaboración: Propia

En la figura 4.9 se mostró el resultado de los indicadores de desempeño que tiene el programa propuesto elegido de producción, para apreciarlos de mejor manera, se describe en la siguiente tabla 4.4

Resultados	Criterio	Volumen	Time	Cmax	Tmax	$\Sigma U_j$	$\Sigma C_j$	$\Sigma T_j$	$\Sigma w_j C_j$	$\Sigma w_j T_j$
Propuesta LEKIN	Heurística Cuello de Botella Desplazante	17,200	3	160	2	1	2,598	2	5,927	18

**Tabla 4.4 Resultados del programa propuesto de producción**

Fuente: LEKIN - Elaboración: Propia

Se puede apreciar que el makespan ( $C_{max}$ ) es de 160 horas, es decir, el último trabajo en salir (un alimento para truchas) termina en la hora 160, es decir, 6.6 días de producción para una producción de 17,200 sacos de 40Kg.

La tardanza máxima ( $T_{max}$ ) es de 2 horas, es decir, un trabajo tuvo 2 horas tarde (tiempo de terminación vs el tiempo de entrega).

La sumatoria de penalidades ( $\Sigma U_j$ ) es de 1, es decir, hubo 1 trabajo que su terminación fue después de su fecha de entrega.

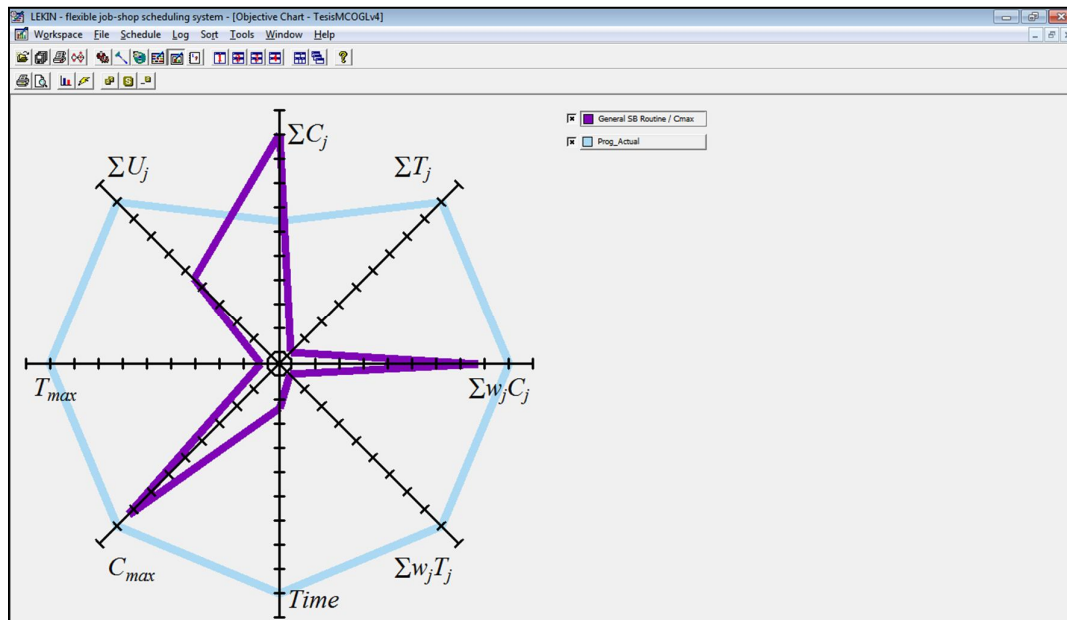
Dado estos resultados, el siguiente paso, es compararlos con la situación actual y de esta forma, ver si el programa de producción propuesto es mejor, esto se resumen en la tabla 4.5

Escenarios de programación	Criterio	Volumen	Time	Cmax	Tmax	$\Sigma U_j$	$\Sigma C_j$	$\Sigma T_j$	$\Sigma w_j C_j$	$\Sigma w_j T_j$
Situación Actual	Reglas empíricas fábrica en estudio	17,200	20	173	63	2	1,565	110	6,896	1,037
Propuesta LEKIN	Heurística Cuello de Botella Desplazante	17,200	3	160	2	1	2,598	2	5,927	18
Variación	Propuesta vs Actual	0	-17	-13	-61	-1	1,033	-108	-969	-1,019
%	Propuesta vs Actual	0%	-85%	-8%	-97%	-50%	66%	-98%	-14%	-98%

**Tabla 4.5 Comparación del programa actual vs programa propuesto**

Fuente: LEKIN - Elaboración: Propia

En la tabla 4.5 se aprecia que los 3 principales indicadores muestran una reducción; lo que da una pauta de que el programa de producción propuesto plantea una mejor utilización de la línea de producción ( $C_{max}$ ) y un mejor nivel de servicio para el área comercial al reducirse el número de trabajos con tardanza y penalidades ( $T_{max}$  y  $\Sigma U_j$ ). En la figura 4.13 se aprecia gráficamente, los indicadores entre los 2 programas de producción.



**Figura 4.13 Diagrama de resultados programa propuesto vs actual**

Fuente: LEKIN - Elaboración: Propia

Finalmente, esta propuesta de programa de producción representa un incremento en la capacidad global de la línea de producción, aproximadamente en un 9% utilizando los mismos recursos (personal, tiempo y maquinaria), mejorando el nivel de servicio para la división comercial salud animal. En la tabla 4.6 se muestra la estimación monetaria que esta propuesta generaría como ingreso adicional a la compañía al permitir subir la capacidad de producción en la línea de producción con los mismos recursos.

Unidad	Descripción	Valor	Comentarios
Horas	Capacidad extra de producción	13	Equivale a un turno de producción aproximadamente
Sacos x 40Kg	Producción extra semanal	1,625	Ratio promedio: 5 Tons x hora
\$ x Semana	Ingreso semanal adicional	\$ 35,750	PVP saco promedio: 22 \$/saco x 40Kg
<b>\$ x Año</b>	<b>Ingreso anual adicional</b>	<b>\$ 1,859,000</b>	<b>\$1.9MM más de ingresos x aplicar la propuesta</b>

**Tabla 4.6 Estimación de los ingresos adicionales por el programa de producción propuesto**

Fuente: Estadísticas fábrica en estudio - Elaboración: Propia

## CAPÍTULO V

### CONCLUSIONES Y RECOMENDACIONES

#### 5.1 Conclusiones:

- Se cumplió con el objetivo general de esta tesis que consistía en proponer un programa de producción que sea mejor que el programa actual y adicionalmente se cumplió con el resultado esperado pues el aumento de la capacidad es del 9% versus el 5% que era el valor mínimo esperado.
- Se cumplieron con todos los objetivos específicos planteados al inicio de esta tesis, superando las expectativas planteadas.
- Se aplicó correctamente, los criterios mostrados en el marco teórico y se implementó adecuadamente el uso del software LEKIN como herramienta para el desarrollo de la tesis y la consecución de los objetivos planteados.
- El programa de producción propuesto, genera un aumento de la capacidad de producción de la línea (9% aproximadamente), lo que se traduce en un mayor abastecimiento hacia el área comercial, lo que representa en mayores ingresos para la división comercial salud animal, esto equivale a 1,625 sacos más de alimento balanceado, lo que equivale a \$35,750 dólares semanales de potencial venta adicional y que anualmente representa \$1.9 millones de ingreso potencial por ventas en esta división comercial.
- La heurística del cuello de botella desplazante, presentó la mejor propuesta de programa de producción con respecto al resto de programas de producción que la herramienta LEKIN mostró como posibles opciones, la elección se basó en una matriz de ponderación de 3 factores de desempeño, donde la propuesta con la menor sumatoria total, fue la propuesta elegida.

- El programa de producción propuesto rompe los paradigmas actuales que se venían utilizando al momento de elaborar el programa de producción de la línea y que se resumía en 3 reglas.
- El programa de producción propuesto presenta una mejora versus la programación actual en los 3 indicadores de desempeño para un programa de producción más importantes definidos para la elaboración de esta tesis.
- El makespan ( $C_{max}$ ) se reduce en un 8% versus la situación actual, esto implica una mejor utilización de la línea de producción pues se necesita menos tiempo para cumplir el mismo volumen de producción que lo realizaba la programación actual, por ende, la línea sube su capacidad semanal de producción de alimento balanceado con los mismos recursos.
- La tardanza máxima ( $T_{max}$ ) se reduce en 97% versus la situación actual, esto implica una mejora en el nivel de servicio para la área comercial.
- La sumatoria de penalidades ( $\sum U_j$ ) se reduce en 50% versus la situación actual, esto representa también una mejora en el nivel de servicio hacia el área comercial.
- La implementación y configuración de la herramienta LEKIN dentro de la empresa en estudio, fue clave para la consecución de los objetivos planteados en esta tesis, ahora este software es parte de las herramientas del departamento de Planificación de la empresa y es parte del proceso de programación de la producción.
- Finalmente, por los resultados obtenidos, fue factible convencer a la Gerencia de Planta e implementar el programa de producción propuesto de manera inmediata.

## 5.2 Recomendaciones:

- Dado los beneficios obtenidos por implementar la programación de la producción a través de soporte matemático y automatizado mediante un software para este fin, se recomienda replicar esta práctica, en las otras líneas de producción que posee la fábrica en estudio y además, en el resto de fábricas que pertenecen a la compañía.
- Periódicamente, se debe validar que el modelo desarrollado en LEKIN este alineado con la realidad en el proceso de producción, dado los cambios que pudieran haber en la línea de producción.
- De igual forma, periódicamente, se debe validar o actualizar los datos maestros asociados a la parametrización de las máquinas (procesos) y trabajos (productos), como por ejemplo: ratios de producción por producto y por proceso, tiempos de preparación por cambio de producto, todo esto con el objetivo de que la parametrización en LEKIN sea lo más adherida a la realidad del proceso de producción. Se puede aprovechar reclutando estudiantes de ingeniería como pasantes para dicho propósito.

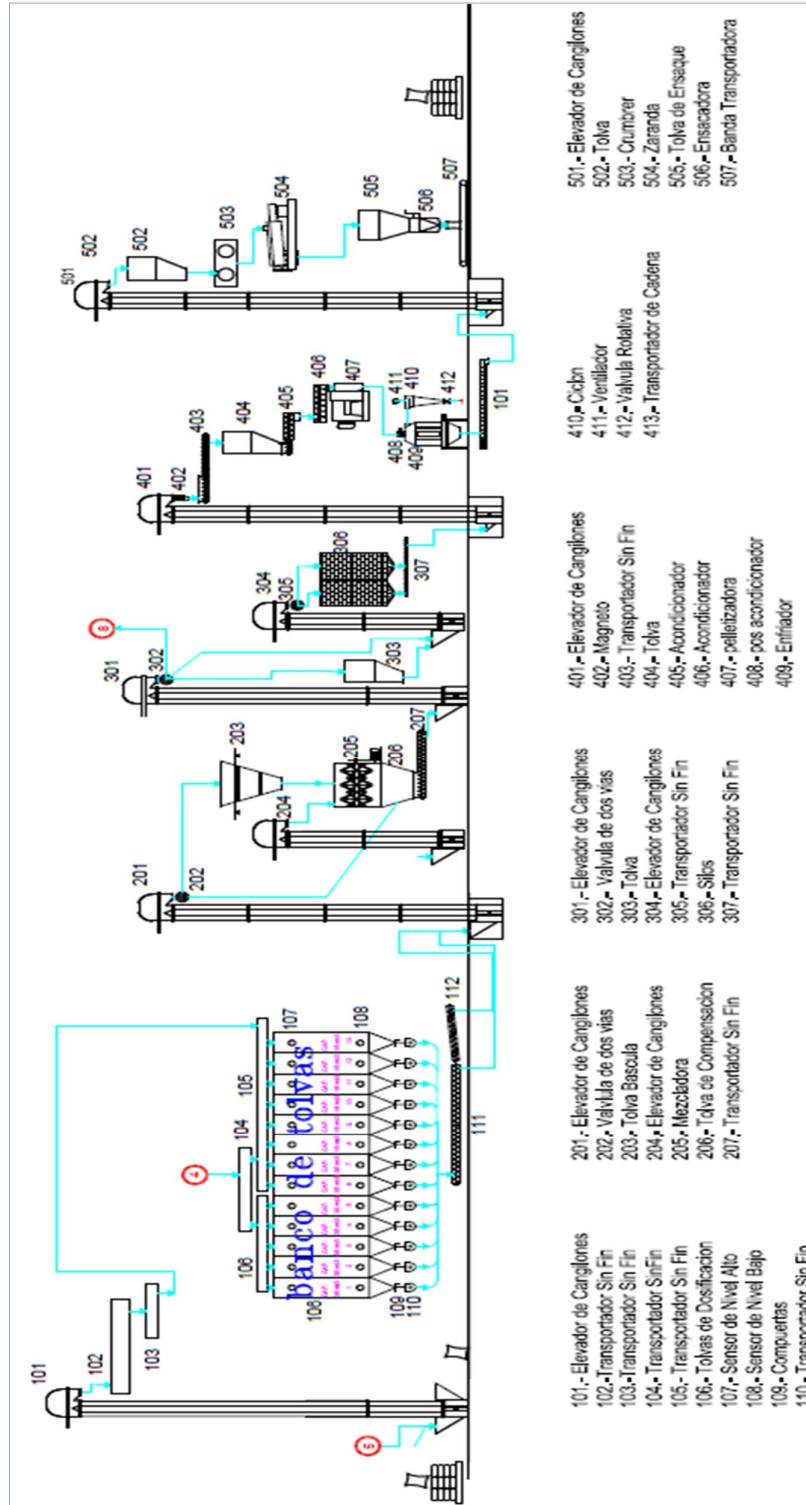
## BIBLIOGRAFÍA

1. Pinedo Michael, Scheduling: Theory, Algorithms, and Systems, Springer, USA – 2008, página 1 a la página 234
2. Pinedo Michael, Planning and Scheduling in Manufacturing and Services, Springer, USA – 2005, página 1 a la página 311
3. Sipper Daniel, Bulfin Robert, Planeación y control de la producción, McGraw-Hill, USA – 1998, página 1 a la página 472
4. Ballou Ronald, Logística: Administración de la cadena de suministro, Pearson, México – 2004, página 62 a la página 130
5. Chase Richard, Administración de producción y operaciones, McGraw-Hill, Colombia – 2000, página 678 a la página 712
6. Chopra Sunil, Administración de la cadena de suministro: Estrategia, Planeación y Operación, Pearson, México – 2008, página 218 a la página 238
7. Heizer Jay, Dirección de la producción y de operaciones, Pearson, España – 2008, página 201 a la página 248

## ANEXOS



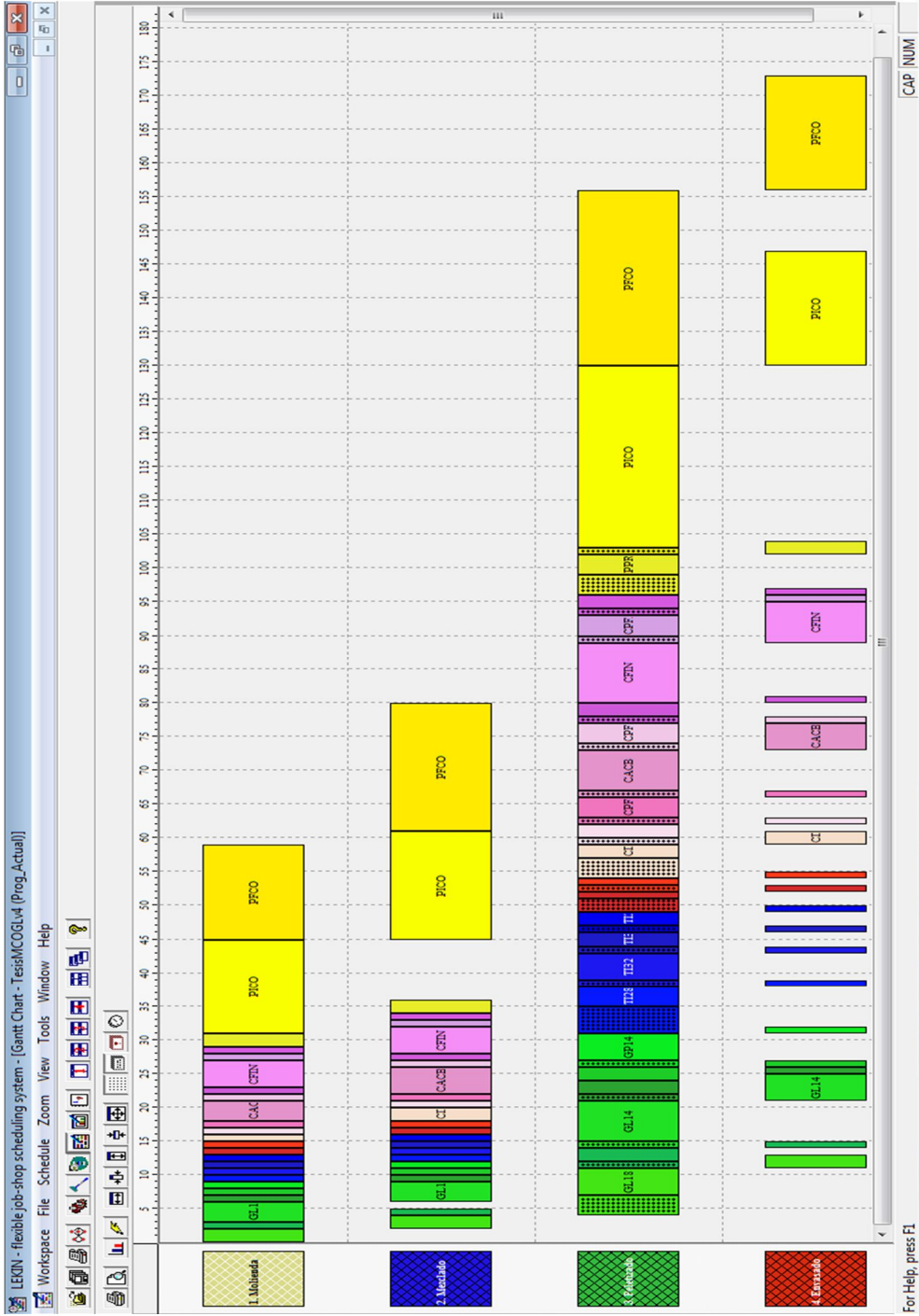
# ANEXO A



## ANEXO B

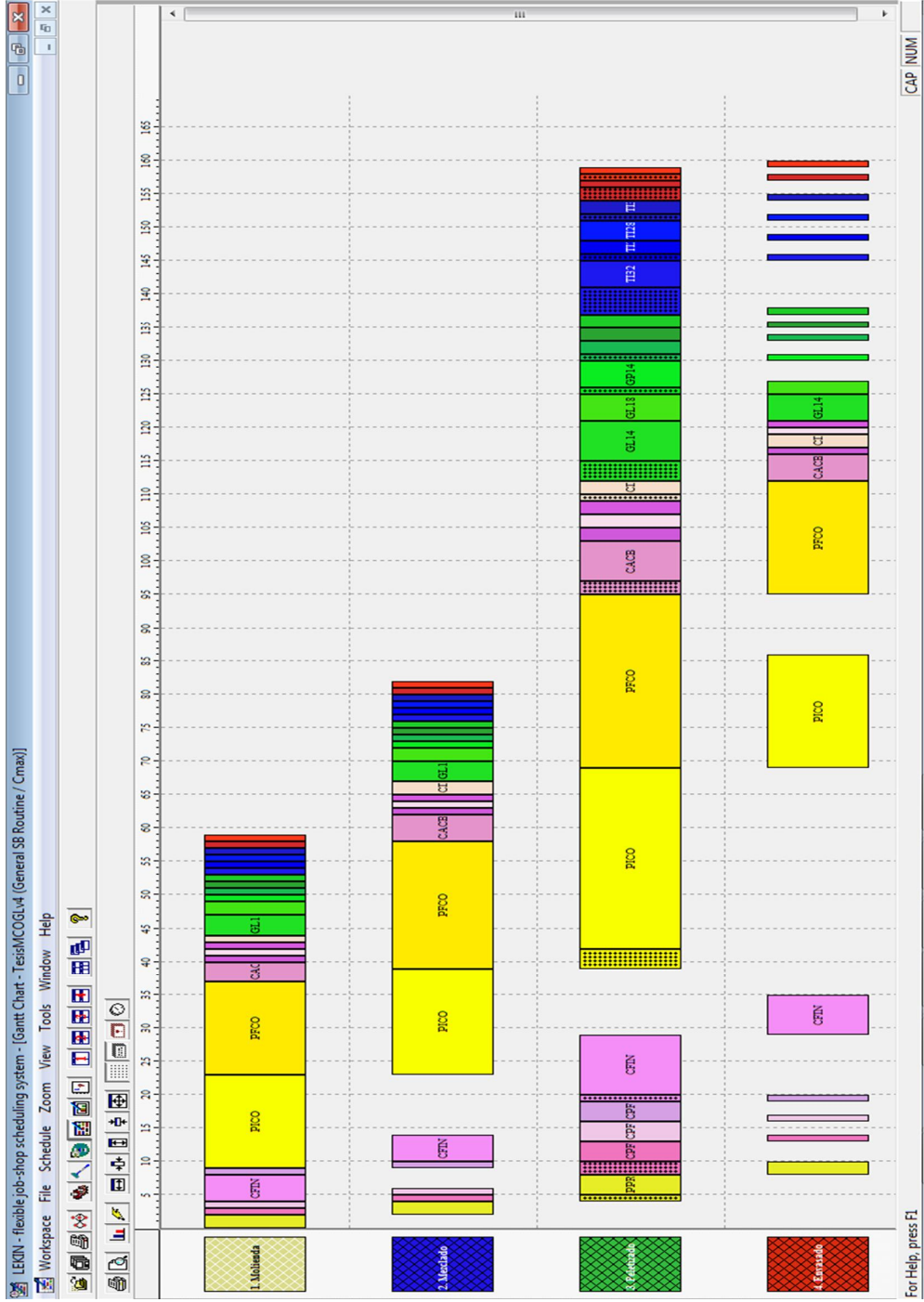
MATRIZ SET-UP [Horas]	LEKIN	MATRIZ																							
		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
	Matriz [mm]	4	5	3.2	3.2	3.2	5	5	5	5	8	5	8	5	5	0	3.2	4	4	4.7	4.7	3	2.38	2.38	4.7
	TRABAJO	CINI	CGES	CPF1	CPF2	CPF3	CACB	CACM	CFIN	CLAC	GL18	GTER	GL14	G19R	GTIN	GP14	PPRE	PICO	PFCO	T124	T128	T132	T138	TR46	TR40
A	4	0	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
B	5	1	0	1	1	1	0	0	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
C	3.2	1	1	0	0	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
D	3.2	1	1	0	0	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
E	3.2	1	1	0	0	0	1	1	1	1	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
F	5	1	0	1	1	1	0	0	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
G	5	1	0	1	1	1	0	0	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
H	5	1	0	1	1	1	0	0	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
I	5	1	0	1	1	1	0	0	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
J	8	4	4	4	4	4	4	4	4	4	0	1	0	1	1	1	4	4	4	4	4	4	4	4	4
K	5	4	4	4	4	4	4	4	4	4	1	0	1	0	0	1	4	4	4	4	4	4	4	4	4
L	8	4	4	4	4	4	4	4	4	4	0	1	0	1	1	1	4	4	4	4	4	4	4	4	4
M	5	4	4	4	4	4	4	4	4	4	1	0	1	0	0	1	4	4	4	4	4	4	4	4	4
N	5	4	4	4	4	4	4	4	4	4	1	0	1	0	0	1	4	4	4	4	4	4	4	4	4
O	0	4	4	4	4	4	4	4	4	4	1	1	1	1	1	0	4	4	4	4	4	4	4	4	4
P	3.2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	0	1	1	3	3	3	3	3	3
Q	4	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	1	0	0	3	3	3	3	3	3
R	4	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	1	0	0	3	3	3	3	3	3
S	4.7	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	0	0	1	1	2	2
T	4.7	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	0	0	1	1	2	2
U	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	1	1	0	1	2	2
V	2.38	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	1	1	1	0	2	2
W	2.38	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	2	2	2	0	1
X	4.7	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	2	2	2	1	0

# ANEXO C



For Help, press F1

# ANEXO D



# ANEXO E

