

Jurado asignado:

Presidente: Dr. Carlos González Rivera
Primer Vocal: Dr. Julio Ernesto Solís Daun
Secretario: Dr. Bernardo Hernández Morales
Primer Suplente: M. En C. Rubén Vázquez Medrano
Segundo Suplente: M. En C. Antonio Huerta Cerdán

Sitio donde se desarrolló el tema: DTA de México S.A. De C.V.
Av. Río Churubusco no. 2099
Col Agrícola Oriental

Asesor: Dr. Luis Miguel Villegas Silva

Sustentante: I.Q.M. Octavio Peralta Alarcón

280796



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

A quienes disfrutaban las reuniones de esta comunidad.

A quien después de nueve años todavía dice que te extraña y que sabe que esta célula no es efímera.

A quien conocerá la experiencia del amor de Dios a través de ella.

A la familia.

AGRADECIMIENTOS

Debo agradecer en primer lugar al Dr. Luis Miguel Villegas Silva por haberme presentado la idea original del trabajo realizado, la aplicación de los sistemas expertos a los procesos metalúrgicos, por el apoyo, la paciencia y el interés mostrado para la realización de este proyecto. De la misma manera, deseo agradecer al M. en C. Oscar Jesús Rendón Gómez por habernos acompañado en la parte inicial del proyecto, durante los seminarios de programación lógica e inteligencia artificial.

RESUMEN

La aleación DOX es el nombre comercial de una aleación ternaria Ni-Co-B que es producida utilizando un proceso electrolítico único en el mundo por la empresa DTA de México. Utilizando este proceso se recubren aceros al carbón para producir aceros estructurales resistentes a la corrosión.

DTA de México pone pocas restricciones a sus potenciales compradores respecto a la cantidad y geometría de piezas a recubrir. Dependiendo de su geometría, se pueden formar arreglos que van de 2 a 400 piezas, o más, para procesar el material. El cálculo de las variables de operación tiene el problema de que la resistencia del baño y la densidad de corriente máxima son función de la geometría y el tipo de arreglos del lote de material que se introduce a la celda electrolítica. La variable a controlar para la electrólisis es el flujo de corriente; sin embargo, también se debe llevar un seguimiento de la resistencia del baño ya que durante el proceso se pueden presentar falsos contactos. Desafortunadamente, la complejidad de la geometría y la falta de información respecto a las reacciones anódicas y catódicas dificulta la elaboración de un modelo fenomenológico que sea factible económicamente y realizable a corto plazo. El diseño de los programas de producción, la estimación de las variables de operación y el control de proceso depende en gran medida de la habilidad y la experiencia del coordinador de producción y el ingeniero de proceso. Sin embargo, la cantidad de información que deben procesar generalmente hace que olviden datos u omitan detalles que permitirían obtener mejores soluciones a estos problemas.

Este problema es un buen candidato para la implementación de un sistema experto ya que requiere el uso de conocimiento heurístico para su solución. Si se necesita el uso de la intuición, el juicio, la inferencia lógica, la manipulación de grandes cantidades de información y el conocimiento de un experto para la solución de un problema podemos hablar de la necesidad de un sistema experto.

Utilizando la información de las hojas de control y la experiencia del experto se desarrolló un programa lógico que permite procesar la información del proceso para desarrollar planes de producción y calcular variables de operación. Para la implementación de este sistema se utilizó programación lógica, la técnica de programación por excelencia para los sistemas expertos. El modelo de control utiliza programación estructurada para desarrollar la interface con el proceso y el usuario (programación en ambiente Windows).

1. INTRODUCCIÓN	3
2. ANTECEDENTES	12
2.1 INTELIGENCIA ARTIFICIAL	12
2.1.1 <i>Historia de la inteligencia artificial</i>	13
2.1.2 <i>Ramas de la inteligencia artificial</i>	14
2.1.3 <i>Aplicaciones de la inteligencia artificial</i>	19
2.1.4 <i>Sistemas expertos</i>	20
2.2 DESARROLLO DE PROGRAMAS	24
2.2.1 <i>Programación estructurada</i>	26
2.2.2 <i>Programación orientada a objetos (POO)</i>	27
2.2.3 <i>Programación lógica</i>	27
2.3 PROCESOS ELECTROQUÍMICOS	38
2.3.1 <i>Leyes de Faraday</i>	39
2.3.2 <i>Eficiencia de corriente y voltaje</i>	40
2.3.2 <i>Termodinámica de procesos electroquímicos</i>	41
2.3.3 <i>Fenómenos de transporte</i>	42
3. DESARROLLO DEL MODELO	44
3.1 OBJETIVOS	44
3.2 METODOLOGÍA	45
3.3 DESCRIPCIÓN DEL PROBLEMA	46
3.3.1 <i>Planeación de producción</i>	46
3.3.2 <i>Cálculo de variables de operación</i>	51
3.3.3 <i>Control de proceso</i>	52
3.4 CONSTRUCCIÓN DEL MODELO	53
3.4.1 <i>Base de conocimientos</i>	54
3.4.2 <i>Construcción de nuevos conocimientos</i>	57
3.4.3 <i>Búsqueda de soluciones</i>	65
3.4.4 <i>Planeación de producción</i>	66
3.4.5 <i>Control de proceso</i>	69
4. VALIDACIÓN DEL MODELO	72
4.1 SOLICITUD DE MATERIAL	72
4.2 MATERIAL A PROCESAR	79
4.3 CÁLCULO DE RONDAS.	86
4.4 CÁLCULO DE VARIABLES DE OPERACIÓN	90
4.5 CONTROL DE PROCESO	93
5. ANÁLISIS DE RESULTADOS	94
5.1 CÁLCULOS	94
5.2 HOJAS DE CONTROL	94
5.3 CÁLCULO DE VARIABLES DE OPERACIÓN	96
6. CONCLUSIONES	98
APÉNDICE A. PROLOG	101

APÉNDICE B. CÁLCULO DE VARIABLES DE OPERACIÓN	106
APÉNDICE C. REGLAS PARA PLANEACIÓN DE PRODUCCIÓN	107
APÉNDICE E. HECHOS PARA PLANEACIÓN DE PRODUCCIÓN	113
BIBLIOGRAFÍA	117

CAPÍTULO 1

INTRODUCCIÓN

En la actualidad, la inteligencia artificial abarca una enorme cantidad de campos, desde áreas de propósito general como el razonamiento lógico, hasta tareas específicas como el ajedrez o el diagnóstico de enfermedades. Muchos científicos de las áreas más diversas empiezan a ver a la inteligencia artificial como una herramienta que facilita la sistematización y automatización del trabajo intelectual que cotidianamente desarrollan.

La sistematización y automatización del conocimiento, así como la reducción vertiginosa del costo y tamaño del hardware permiten implementar aplicaciones inteligentes en muchas ramas de la ciencia y la tecnología. La planeación, el diagnóstico de problemas y el control de proceso son sólo algunas de las áreas de la industria donde la inteligencia artificial puede ofrecer soluciones con bajo costo en hardware, el resto depende del trabajo interdisciplinario de expertos. Desafortunadamente, cuando hablamos de inteligencia artificial, la gente piensa inmediatamente en Terminator, en tecnologías que están fuera de nuestro alcance. Curiosamente en México¹ muchas industrias ya cuentan con el hardware suficiente para implementar sistemas expertos² pero el desconocimiento de esta rama de la ciencia, la poca cultura de trabajo interdisciplinario y la falta de interés por generar nuevos conocimientos les impide aprovechar correctamente esta herramienta.

El presente trabajo nace del interés por aplicar estas técnicas a la industria metalúrgica en México. Al iniciar la investigación, se presentó la oportunidad de trabajar en una planta piloto (DTA de México) que tiene parte de su proceso controlado por computadora. Durante año y medio, el autor trabajó como ingeniero de proceso y posteriormente como coordinador de producción en dicha empresa.

Aprovechando esta experiencia se sistematiza parte del conocimiento adquirido que representa una parte del trabajo cotidiano que se debe desarrollar en dicha empresa y posteriormente se hace una representación de este conocimiento

¹ El siguiente comentario parte de una apreciación o experiencia personal, fruto de la estancia en diferentes industrias como estudiante o profesionista y de pláticas con otros profesionistas del mismo ramo.

² La rama de la inteligencia artificial con mayor aplicación en el mercado.

para construir un sistema experto para ayudar en la toma de decisiones. Puesto que es una experiencia muy particular, es posible que las técnicas o conocimientos específicos no se apliquen en ningún otro lado³, pero el planteamiento del trabajo ilustra ampliamente la aplicación de los sistemas expertos en la industria nacional.

La aleación DOX es nombre comercial de una aleación ternaria Ni-Co-B que ha sido obtenida mediante dos tipos de procesos:

- Deposición por plasma químico
- Electrólisis

Densidad	$8.72 \frac{g}{cm^3}$
Dureza	28-35 RC

Tabla 1.1 Propiedades físicas de la aleación DOX

Ni	44-50%
Co	40-46%
B	3-5 %

Tabla 1.2. Composición química de la aleación DOX

Esta aleación es producida utilizando un proceso electrolítico único en el mundo⁴ por la empresa DTA de México. Utilizando este proceso se recubren aceros al carbón, Figura 1.1, para producir aceros estructurales resistentes a la corrosión, Figura 1.2.

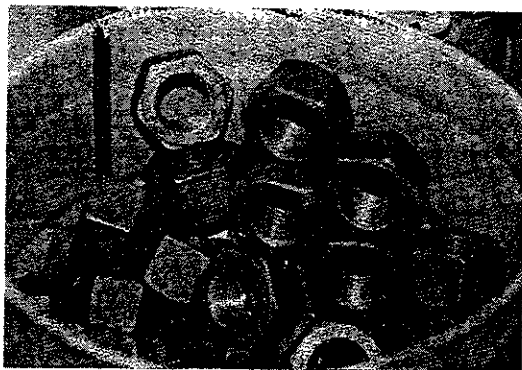


Figura 1.1. Tuercas sin recubrimiento

³ Incluso, aunque sea el mismo caso, es posible que exista una solución mejor.

⁴ Nos referimos a la composición química del electrolito, condiciones de operación y producto obtenido.

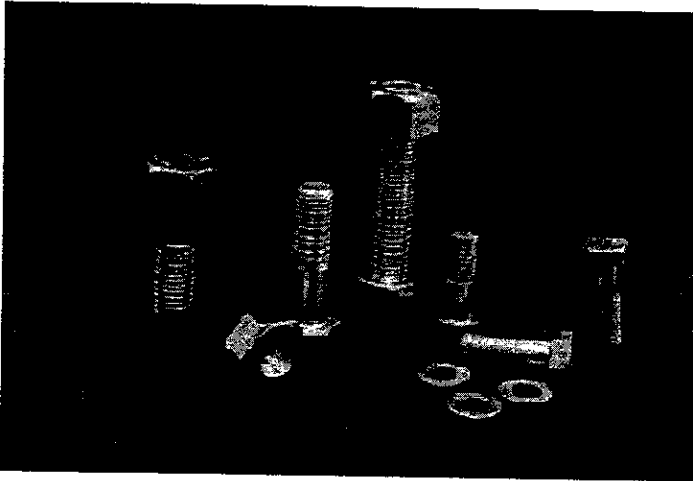


Figura 1.2 Aceros al carbón con recubrimiento DOX

DTA de México ha financiado proyectos de investigación en el Instituto de Física y la Facultad de Química (Departamento de Metalurgia) de la UNAM con el objetivo de caracterizar esta aleación y encontrar nuevas aplicaciones a su producto. Sin embargo, la termodinámica y cinética del proceso, y el mecanismo de formación de la microestructura, no han sido investigados con detalle.

El proceso de recubrimiento DOX puede dividirse en tres etapas fundamentales:

- Corte. DTA no fabrica los materiales a recubrir, sólo rectifica las dimensiones de las piezas.



Figura 1.3 Barras de acero sin cortar

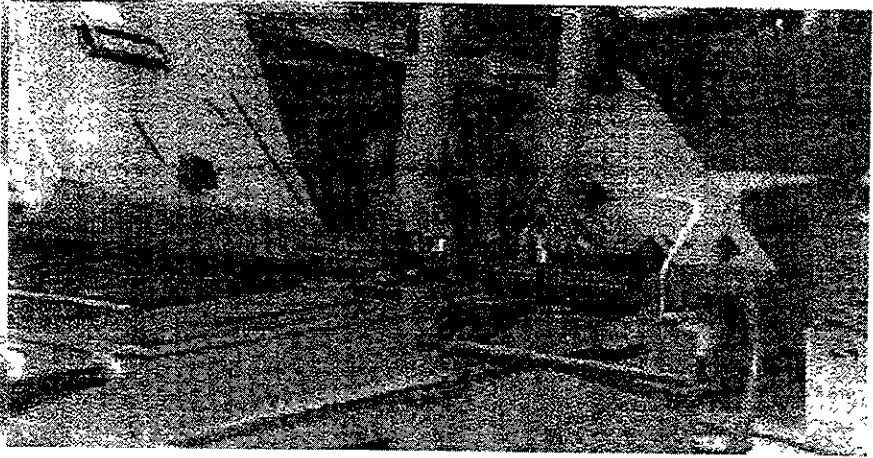


Figura 1.4 Corte de barras

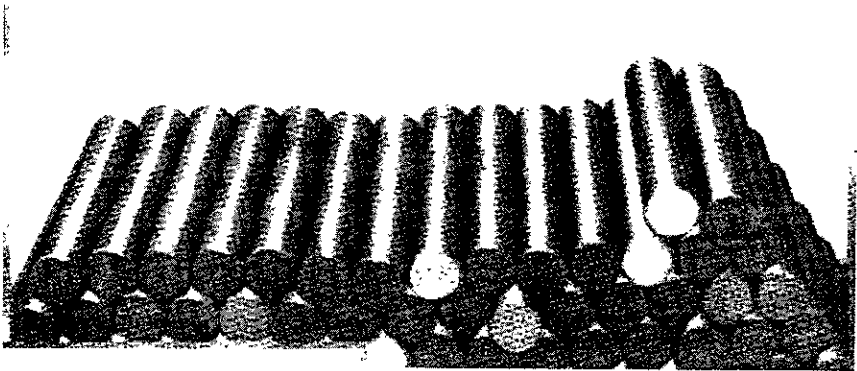


Figura 1.5 Espárragos

- *Preparación de superficie.* Toda superficie a recubrir debe estar limpia de grasa, polvos, óxidos y partículas que puedan perjudicar la adherencia del recubrimiento; utilizando procedimientos químicos o físicos, pueden realizarse varias operaciones para lograr este objetivo. Durante esta etapa también se puede efectuar un pulido de la superficie para mejorar la apariencia del producto terminado.
- *Aplicación del recubrimiento.* En esta etapa, Figura 1.6, se deben de controlar las siguientes variables de operación:

1. Corriente.
2. Tiempo de proceso.
3. Voltaje.
4. pH.
5. Composición química del electrolito.
6. Agitación.
7. Temperatura de la solución.

La corriente, el tiempo de proceso, el voltaje y el consumo de sales son calculados por el ingeniero de proceso. Aunque los cálculos pueden parecer rutinarios⁵, existen dos datos que sólo pueden ser decididos por el experto⁶: la densidad de corriente y la resistencia del baño. El pH es revisado periódicamente; la composición química de la solución es ajustada agregando las sales utilizadas durante el proceso, mediante el cálculo de consumo de sales, y es certificado haciendo análisis químico de la solución.

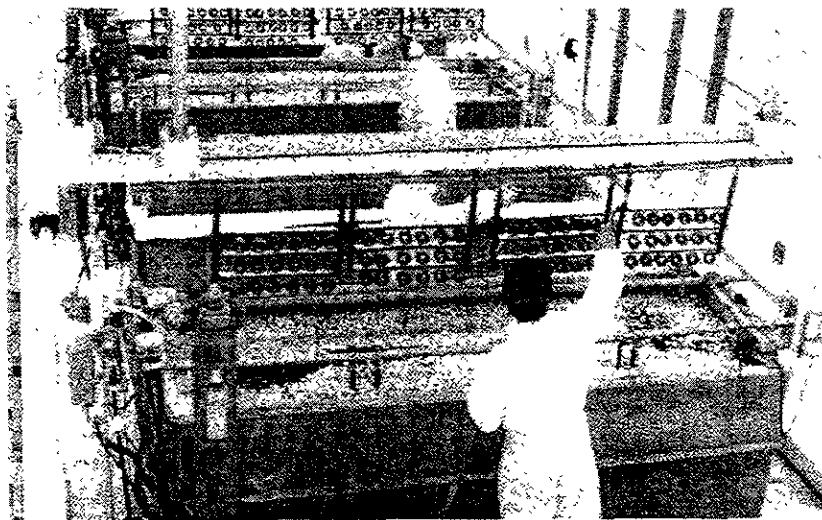


Figura 1.6 Proceso DOX

Uno de los productos de mayor demanda es el espárrago con tuercas, Figura 1.7. El espárrago es cortado según la dimensión solicitada antes de preparar la

⁵ Ver Apéndice B.

⁶ Se necesitan conocimientos básicos de ingeniería electroquímica, para comprender la influencia de estas variables de operación, y conocer el comportamiento de la solución para diferentes tipos de arreglos.

superficie para recubrimiento y las tuercas pasan directamente a la preparación de superficie. El proceso electroquímico es por lotes, cada lote a recubrir debe tener las siguientes características:

1. Para cada geometría, existe un número máximo de piezas que pueden ser introducidas en cada lote o ronda.
2. En una ronda se pueden introducir piezas de dimensiones diferentes, siempre y cuando sean del mismo tipo y su número máximo de piezas por ronda sea similar.

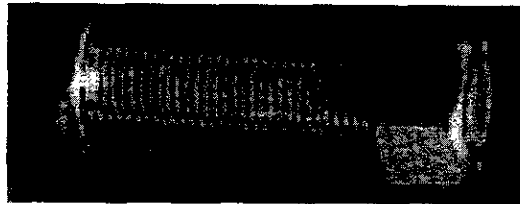


Figura 1.7 Espárrago con tuercas

En las figuras 1.8 y 1.9 se presentan los diagramas de bloque para producción de tuercas y espárragos.

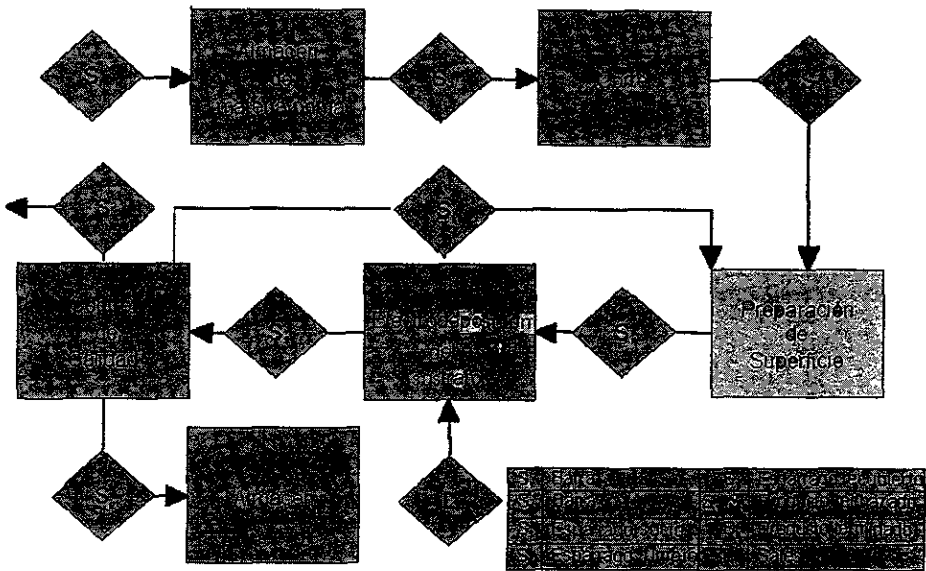


Figura 1.8 Diagrama de bloques para la fabricación de espárragos

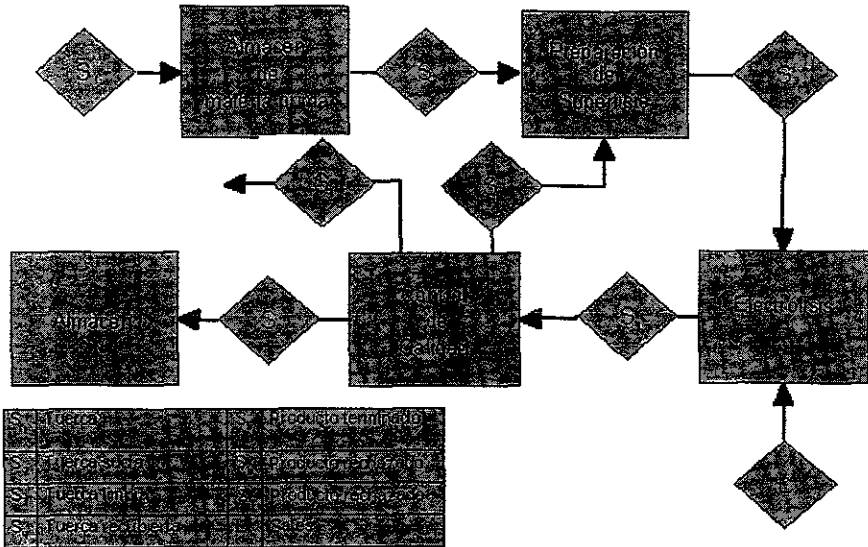


Figura 1.9 Diagrama de bloques para la fabricación de tuercas

DTA de México pone pocas restricciones a sus potenciales compradores respecto a la cantidad y geometría de piezas a recubrir. Por ejemplo, una solicitud de compra de material típica incluye:

Geometría	150 medidas diferentes
Cantidad mínima solicitada	1 pieza
Cantidad máxima solicitada	11,000 piezas
Número total de espárragos	70,000 piezas
Número total de tuercas	140,000 piezas
Diámetro menor	$\frac{3}{8}$ pulgadas
Diámetro mayor	$3\frac{3}{4}$ pulgadas
Longitud menor	$1\frac{1}{2}$ pulgadas
Longitud mayor	45 pulgadas

Tabla 1.3 Características de una solicitud típica de material

Dependiendo de su geometría, se pueden formar arreglos que van de 2 a 400 piezas o más para procesar el material. Definir el conjunto de rondas que se deben procesar implica la manipulación de una gran cantidad de datos, desde la solicitud de material hasta los inventarios de almacén. Es importante aclarar que cada programa de producción es único e irrepetible por lo que bien puede recibir el nombre de programación de proyecto. Para planificar se debe tener acceso a toda la información del proyecto, esto implica tener acceso al inventario de cada etapa. Al ejecutar el proyecto, se debe de llevar un control sobre el avance de la producción y el inventario de cada departamento

para saber si no ha fallado alguno de los pasos programados, lo que con frecuencia hace necesaria la reelaboración del proyecto.

El cálculo de las variables de operación tiene el problema de que la resistencia del baño y la densidad de corriente máxima son función de la geometría y el tipo de arreglos del lote de material que se introduce a la celda electrolítica. Tomando en cuenta la variedad de arreglos que se pueden definir, continuamente se tienen que estimar las variables de operación para cada ronda; además, después de procesar una ronda se tiene que revisar el material y decidir si la densidad de corriente tiene un valor óptimo o si se necesita modificar su valor. Desafortunadamente, la complejidad de la geometría y la falta de información respecto a las reacciones anódicas y catódicas dificulta la elaboración de un modelo fenomenológico que sea factible económicamente y realizable a corto plazo.

El diseño de los programas de producción y la estimación de las variables de operación depende en gran medida de la habilidad y experiencia del coordinador de producción y el ingeniero de proceso. Sin embargo, la cantidad de información que deben procesar generalmente hace que olviden datos u omitan detalles que permitirían obtener mejores soluciones a estos problemas; además, con frecuencia se modifican los programas de producción debido a:

- Cambios en las requisiciones de material.
- Incumplimientos de proveedores en tiempos de entrega.
- Descompostura de equipos.
- Errores humanos.
- Rechazo de material.
- Otros.

La variable a controlar para la electrólisis es el flujo de corriente. Sin embargo, también se realiza un seguimiento de la resistencia del baño ya que durante el proceso se pueden presentar falsos contactos. Esto provoca variaciones en el voltaje que pueden ser perceptibles o imperceptibles para el operador. Sin embargo, las variaciones en la resistencia tienen causas diferentes y, consecuentemente, soluciones diferentes. Debido a esto, es necesaria la utilización de un sistema de control inteligente que actúe racionalmente y no automáticamente.

Este problema es un buen candidato para la implementación de un sistema experto ya que requiere el uso de conocimiento heurístico para su solución. Si se necesita el uso de la intuición, el juicio, la inferencia lógica, la

manipulación de grandes cantidades de información y el conocimiento de un experto para la solución de un problema podemos hablar de la necesidad de un sistema experto.

Utilizando la información de las hojas de control y la experiencia del experto se desarrolló un programa lógico que permite procesar la información del proceso para desarrollar planes de producción, calcular variables de operación y solucionar problemas de proceso. Para la implementación de este sistema se utiliza programación lógica (Prolog), la técnica de programación por excelencia para los sistemas expertos, y se muestran los beneficios de la utilización de esta herramienta en la solución de este tipo de problemas. El modelo de control utiliza programación estructurada (lenguaje C) para desarrollar la interface con el proceso y el usuario (programación en ambiente Windows).

En el capítulo 2 se incluye una breve revisión de la literatura para presentar los conceptos más importantes que permiten entender los términos manejados en todo el trabajo.

En el capítulo 3 se hace una descripción más amplia del problema, y se presentan datos de proceso y reglas empíricas utilizadas por los expertos del proceso DOK. Este conocimiento se tradujo en un programa lógico que permite simular el cálculo de rondas de material y variables de operación.

En el capítulo 4 se presentan algunos ejemplos del tipo de cálculos que se pueden efectuar con este programa. El capítulo 5 el análisis de los resultados. Finalmente en el capítulo 6 se enuncian las conclusiones pertinentes y se incluyen también algunas recomendaciones para facilitar el desarrollo de futuros trabajos sobre este tema.

Capítulo 2

ANTECEDENTES

2.1 Inteligencia artificial

La historia de la inteligencia artificial (IA) ha estado caracterizada por opiniones contrapuestas y diversas apreciaciones. El problema de definir inteligencia artificial comienza con la definición misma de inteligencia del hombre, y de los aspectos éticos y morales de los propósitos de la IA. ¿Es la inteligencia una facultad particular del hombre? o ¿es el nombre de una colección de diversas habilidades no relacionadas entre sí? ¿qué sucede cuando el pensamiento ocurre? ¿qué es la intuición? ¿qué es la creatividad? ¿cómo se representa el conocimiento en la neurona de un ser vivo? ¿se puede diseñar una máquina inteligente? Hay quienes sugieren, incluso, cambiar el término inteligencia artificial por uno menos polémico tal como aplicaciones inteligentes a fin de referirse mas bien a la aplicación de hardware y software en la solución de problemas de decisión complejos.

Russel y Norving[1] agrupan las definiciones de IA en cuatro categorías:

Categoría	Definición
Sistemas que piensan como humanos	La tarea de lograr que las computadoras piensen... máquinas con mente, en su amplio sentido literal.
Sistemas que piensan racionalmente ⁷	El estudio de las facultades mentales mediante el uso de modelos computacionales.
Sistemas que actúan como humanos	El arte de crear máquinas con capacidad de realizar funciones que, efectuadas por personas, requieren inteligencia.
Sistemas que actúan racionalmente	La rama de la ciencia de la computación que se ocupa de la automatización de la conducta inteligente.

Tabla 2.1 Definiciones de inteligencia artificial.

Cada enfoque presenta objetivos, metas y metodologías particulares. Aunque cada corriente defiende su postura y ha criticado el trabajo de las otras, cada una de ellas ha hecho valiosas aportaciones al estudio y aplicación de la inteligencia artificial.

⁷ Se considera que un sistema es racional si hace lo correcto.

2.1.1 Historia de la inteligencia artificial

En 1950 Alan Turing definió la conducta inteligente como la capacidad de lograr eficiencia a nivel humano en todas las actividades de tipo cognoscitivo, a grado tal que sea suficiente para engañar a un evaluador. Turing propuso una prueba que consistía en que un humano debía interrogar a una computadora por medio de un teletipo; la prueba se consideraba aprobada si el evaluador era incapaz de determinar si una computadora o un humano era quien había respondido las preguntas en el otro extremo de la terminal. Turing también es el encargado de desarrollar la primera computadora moderna (1940) con fines militares. El nacimiento de las computadoras y la programación permiten a los científicos vislumbrar la posibilidad de desarrollar sistemas inteligentes; aunque el inicio es muy prometedor, y el avance de la ingeniería computacional es vertiginoso, los interesados debieron enfrentar serios obstáculos para el desarrollo de esta incipiente rama de la ciencia.

Entre los primeros trabajos de inteligencia artificial desarrollados destacan: i) trabajos con neuronas artificiales, McCulloch y Pitts (1943), ii) programas de ajedrez para computadora, Shannon (1950) y Turing (1953); y desarrollo de una computadora de redes neuronales, Minsky (1953). En 1956, McCarthy y otros organizan un taller para congregar a investigadores estadounidenses interesados en la teoría de los autómatas. En este taller destaca la presentación del programa Teórico Lógico desarrollado por Newell y Simon, utilizado para demostrar teoremas, y el consenso para definir este campo como inteligencia artificial.

En los siguientes 20 años se hicieron aportaciones importante en el campo de la programación: programas para resolver problemas matemáticos y para juegos por computadora. En 1958 se desarrolló el lenguaje de alto nivel Lisp que se convertiría en el lenguaje de programación dominante en inteligencia artificial; Lisp es el segundo lenguaje de programación más antiguo que todavía se utiliza. En estos años, McCarthy y otros inventan el tiempo compartido, como respuesta a la carencia y costo de los recursos de cómputo; esto les permite fundar Digital Equipment Company que se convertiría en el segundo fabricante más importante de computadoras del mundo gracias a sus minicomputadoras de tiempo compartido.

Sin embargo, conforme las investigaciones empezaron a abordar problemas más complejos o variados, los programas fracasaron al enfrentar el problema de la explosión combinatoria⁸; por ejemplo, el optimismo que trajo consigo la

⁸ Crecimiento acelerado del número de soluciones posibles provocado por el incremento de alguna de las propiedades que describen al sistema.

demostración automática de teoremas pronto se vio eclipsado cuando los investigadores fracasaron en la demostración de teoremas que implicaban más de una docena de condiciones. En el campo del procesamiento del lenguaje natural se presentó una situación similar al desarrollar programas de traducción. Los investigadores creyeron que todo se reduciría a sencillas transformaciones sintácticas apoyadas por la gramática de las dos lenguas en cuestión y el remplazo de palabras mediante un diccionario electrónico; sin embargo, la experiencia demostró que era necesario un conocimiento previo del tema o de las frases coloquiales para efectuar una traducción del discurso de interés. Esto provocó que en 1966 se cancelaran en Estados Unidos los patrocinios para los proyectos académicos de traducción, y en 1973 el gobierno británico canceló prácticamente todo el apoyo a las investigaciones sobre inteligencia artificial.

Para enfrentar estos problemas los investigadores desarrollan sistemas basados en el conocimiento y técnicas de programación heurística. Algunos de los primeros programas que utilizan estas técnicas son: Dendral (Buchanan y otros, 1969) para inferir estructuras moleculares a partir de la información proporcionada por un espectrómetro de masas; y Mycin (Buchanan y otros) para el diagnóstico de enfermedades sanguíneas. Aunque muchos ponen en tela de juicio la inteligencia de estos sistemas basados en el conocimiento, la eficacia que mostraron, como agentes racionales, permitió el desarrollo de la inteligencia artificial como una industria rentable y facilitó el resurgimiento de los apoyos financieros a proyectos académicos.

2.1.2 Ramas de la inteligencia artificial

Antes de considerar las aplicaciones mencionaremos seis campos fundamentales de investigación en inteligencia artificial:

1. *Representación del conocimiento.* Es una herramienta fundamental en la implementación de métodos desarrollados para inteligencia artificial. Las técnicas más populares para representar el conocimiento son los marcos, las redes semánticas y las reglas. El marco es una colección de atributos para describir un objeto, una clase de objetos, una situación, una acción, o un evento; el razonamiento mediante marcos está basado en la búsqueda de objetos que contienen la información requerida para resolver un problema determinado. Una red semántica está formada por dos nodos y un arco; cada nodo representa algún concepto y el arco representa la relación entre ambos conceptos; cualquier nodo puede estar relacionado con otros nodos llevando a la formación de una red de hechos. Las reglas dan un camino formal para representar recomendaciones, directivas, o estrategias; las reglas

condicionales (si/entonces) ligan la premisa con su conclusión asociada. Estas técnicas se pueden combinar para obtener las mejores características de cada una para enfrentar problemas.

2. *Estrategias de búsqueda.* Al resolver un problema debemos definir un conjunto de acciones y estados que debemos obtener para alcanzar un estado meta; al enfrentar el problema podemos obtener un conjunto n de soluciones, acciones y estados, que nos lleven a la meta deseada pero también podemos obtener un conjunto m de soluciones incorrectas, las cuales no nos llevan al estado deseado. Para obtener una solución al problema debemos plantear el conjunto de soluciones posibles y los métodos de búsqueda; todo método de búsqueda debe de contar con una prueba de meta, para certificar que la solución obtenida es correcta, y una función de costo, para poder decidir si una solución es mejor que otra. Es común representar al estado inicial (raíz) y al conjunto de estados (nodos) y acciones (ruta) de un problema mediante estructuras jerárquicas (árboles). Un árbol es una colección de elementos llamados nodos, uno de los cuales se distingue como raíz, junto con una relación de paternidad que impone una estructura jerárquica sobre los nodos. Formalmente, un árbol se puede definir de manera recursiva como sigue:

- Un solo nodo es, por sí mismo, un árbol. Este nodo es también la raíz de dicho árbol.
- Suponga que n es un nodo y que A_1, A_2, \dots, A_k son árboles con raíces n_1, n_2, \dots, n_k respectivamente. Se puede construir un nuevo árbol haciendo n el padre de los nodos n_1, n_2, \dots, n_k . En dicho árbol, n es la raíz y A_1, A_2, \dots, A_k son los subárboles de la raíz. Los nodos n_1, n_2, \dots, n_k reciben el nombre de hijos del nodo n .

Los nodos tienen las siguientes características:

- Si existe un camino de un nodo a a otro b , entonces a es un antecesor de b , y b es un descendiente de a . Por ejemplo en la Figura 2.1 los antecesores de *s2.1* son él mismo, *C2* y *Libro*. Observe que cada nodo es a la vez un antecesor y un descendiente de sí mismo.
- Un antecesor o un descendiente de un nodo que no sea él mismo recibe el nombre de antecesor propio o descendiente propio, respectivamente.
- En un árbol, la raíz es el único nodo que no tiene antecesores propios.
- Un nodo sin descendientes propios se denomina hoja.
- Un subárbol de un árbol es un nodo junto con todos sus descendientes.

- La altura de un nodo en un árbol es la cantidad de antecesores propios de ese nodo. En la Figura 2.1 el nodo *C1* tiene altura 1, *C2* altura 2 y el nodo *C3* altura 0. La altura del árbol es la altura de la raíz.
- La profundidad de un nodo es la longitud del camino único desde la raíz a ese nodo.

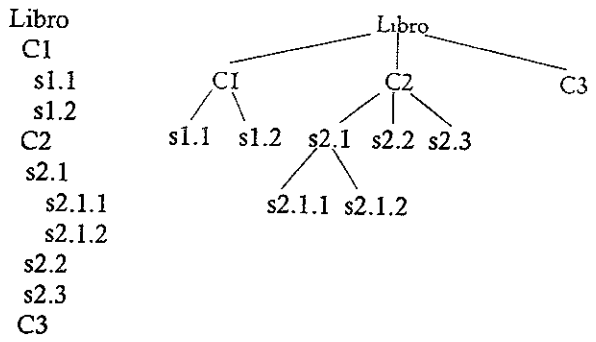


Figura 2.1 Un índice general y su representación como árbol. Utilizando estas nociones podemos definir dos tipos fundamentales de búsqueda:

- **Búsqueda por amplitud.** En este caso primero se expande el nodo raíz, y luego todos los nodos generados por éste; luego sus sucesores, y así sucesivamente. En general, todos los nodos que están en la profundidad *d* del árbol de búsqueda se expanden antes que los nodos que estén en la profundidad *d+1*. La búsqueda por amplitud es una estrategia sistemática, pues primero toma en cuenta todas las rutas de profundidad 1, luego las de profundidad 2, etc. En la Figura 2.2 se muestra el avance de la búsqueda efectuada en un árbol. En caso de haber solución, es seguro que ésta se encontrará; sin embargo, conforme aumenta la profundidad se incrementan considerablemente el tiempo de búsqueda y la memoria requerida de manera tal que la técnica es ineficiente.

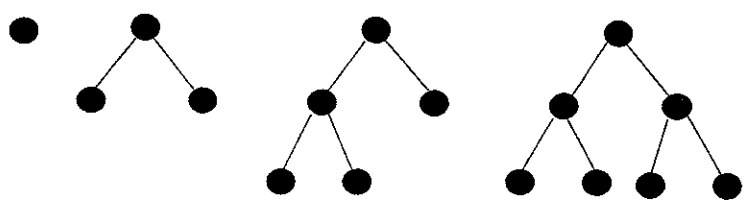


Figura 2.2 Árboles de búsqueda por amplitud después de la expansión de los nodos 0, 1, 2 y 3.

- **Búsqueda por profundidad.** En este caso, siempre se expande uno de los nodos hoja de mayor profundidad. Sólo si la búsqueda conduce a un callejón sin salida (un nodo sin meta), se reinicia la búsqueda y se expanden los nodos de niveles menos profundos. La búsqueda por profundidad utiliza poca memoria pero su desventaja es la posibilidad de que se quede estancada al avanzar por una ruta equivocada. En muchos problemas, los árboles de búsqueda son muy profundos o infinitos, por lo que una elección desafortunada de la ruta de búsqueda incrementará el tiempo utilizado para encontrar una solución. Para disminuir este problema se pueden desarrollar estrategias para detener la búsqueda antes de llegar al nodo hoja, si es que se puede concluir que la ruta seleccionada muy probablemente no nos lleva a un nodo meta.

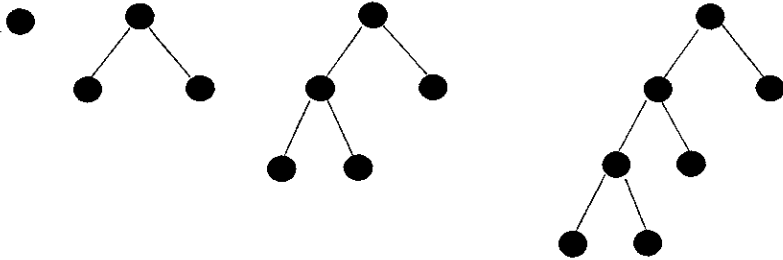


Figura 2.3 Árboles de búsqueda por profundidad.

3. **Heurística.** Se define como el conjunto de estrategias utilizadas para reducir el tiempo de búsqueda de una solución satisfactoria⁹. Estas estrategias se han desarrollado para enfrentar el problema de la explosión combinatoria y se pueden dividir en:

- Reducción del conjunto de soluciones, satisfactorias y no satisfactorias.
- Criterios para detener o continuar una búsqueda.
- Estrategias para seleccionar una ruta que probablemente sea satisfactoria.

La búsqueda heurística no garantiza la obtención de la solución óptima; de hecho, no garantiza la obtención de una solución satisfactoria, pero ofrece opciones correctas en la mayoría de los casos.

4. **Deducción.** Es cualquier procedimiento mediante el que se obtienen conclusiones; también se conoce como inferencia lógica. La inferencia es el

⁹ La cual no es necesariamente la solución óptima. Una solución puede no ser satisfactoria si no se llega al estado meta deseado o si el costo de la búsqueda de la solución es muy elevado.

procedimiento con el cual se obtienen nuevos conocimientos, a partir de otros previos. Nos esforzamos por diseñar procedimientos de inferencia válidos a través de los que se obtienen conclusiones verdaderas a partir de premisas verdaderas. Se dice que un procedimiento de inferencia es completo si mediante éste se pueden obtener todas las conclusiones verdaderas correspondiente a un conjunto de premisas. Es correcto si todas las deducciones que se logran son válidas.

5. *Estrategias de razonamiento.* La deducción es un método de razonamiento, sin embargo, en la mayoría de los casos hay preguntas a las que el sistema no puede dar respuestas categóricas, porque no cuenta con los conocimientos o la información suficiente o porque existe un conjunto de soluciones posibles; por lo tanto, el sistema debe ser capaz de trabajar bajo condiciones de incertidumbre. Los sistemas inteligentes deben ser capaces de llegar a una conclusión a partir de la información que se les proporciona, pero también deben ser capaces de tomar decisiones en caso de trabajar en condiciones de incertidumbre. La lógica matemática y la probabilidad son las herramientas básicas en el desarrollo de estrategias de razonamiento; en aplicaciones industriales también se han utilizado la lógica difusa y las redes neuronales como técnicas innovadoras en el razonamiento de datos de operación.
6. *Planeación.* Es el ordenamiento de acciones en tiempo y espacio para hacer posible la obtención de un estado final o meta. Esta se logra por la comparación del estado actual con el estado deseado y la predicción de la secuencia de acciones necesarias, tomando en cuenta las restricciones impuestas por: a) el ambiente, b) los recursos disponibles, y c) las consecuencias predecibles entre los estados y las acciones o entre los estados sucesivos. Generalmente se utiliza investigación de operaciones para determinar el mejor curso de acción; muy a menudo esta técnica utiliza casi exclusivamente procedimientos matemáticos para representar, por medio de un modelo, el plan de actividades. Sin embargo, existen importantes factores que no se pueden traducir directamente en términos de ecuaciones. Estos modelos deben considerarse como una herramienta para resumir un problema, en tal forma que se haga posible la identificación y evaluación sistemática de todas las alternativas de decisión. Después se llega a una solución seleccionando la alternativa que se juzgue más conveniente entre todas las opciones disponibles. Sin embargo, una gran cantidad de problemas reales tienen demasiadas relaciones para hacer posible una representación matemática adecuada. En otro sentido, aun cuando se puede formular un modelo matemático, éste puede ser demasiado complejo para resolverse. Esta situación ha conducido a la utilización de métodos heurísticos porque su

lógica esta basada en reglas o métodos prácticos que conllevan a la obtención de una buena solución. La ventaja de los métodos heurísticos es que normalmente implican un menor número de cálculos cuando se compara con algoritmos exactos. Así mismo, debido a que están basados en reglas prácticas, normalmente son más sencillos de explicar a los usuarios que no están orientados a las matemáticas. Los métodos heurísticos suelen emplearse para dos fines:

- Se pueden utilizar dentro del contexto de un algoritmo de optimización exacto, con el fin de aumentar la velocidad del proceso para alcanzar el nivel óptimo.
- Se utilizan simplemente para obtener una buena solución al problema. La solución resultante no tiene la garantía de ser óptima y, de hecho, su calidad en relación con el nivel óptimo real puede ser difícil de determinar.

2.1.3 Aplicaciones de la inteligencia artificial

Podemos mencionar seis campos de aplicación:

1. *Procesamiento del lenguaje natural.* Involucra varias áreas de la lingüística tales como los sistemas de indagación de bases de datos, comprensión del texto, análisis gramatical y de estilo de texto, generación automática de texto, traducción por computadora, y análisis de lenguaje.
2. *Visión por computadora.* Se ocupa del análisis y comprensión de la imagen y del movimiento.
3. *Robótica.* Esta implica el control de robots para manipular objetos, movimiento de máquinas independientes, y el uso de sensores para guiar acciones.
4. *Planeación y solución de problemas.* Esto comprende aplicaciones tales como la división de metas de alto nivel en metas de nivel más bajo, determinación de acciones necesarias para alcanzar una meta, revisión de planes de acción basados en resultados intermedios y búsqueda enfocada a metas importantes.
5. *Aprendizaje.* Se encarga de investigar varias formas de aprendizaje incluyendo: aprendizaje por refuerzo, aprendizaje por ejemplo, aprendizaje a partir de la observación, aprendizaje basado en explicaciones, aprendizaje mediante el uso de información relevante.

6. *Sistemas expertos*. Estos se ocupan del procesamiento del conocimiento, el cual es diferente al procesamiento de datos; se desarrollan sistemas para solución de problemas con decisión compleja.

2.1.4 Sistemas expertos

Por definición, un sistema experto es un programa de computación que simula el proceso de pensamiento de un humano experto para resolver problemas de decisión complejos en un dominio específico [6]. Un sistema experto opera como un sistema interactivo que responde a preguntas, pregunta para clarificar, hace recomendaciones y ayuda al proceso de toma de decisiones.

La mayoría de los sistemas expertos caen dentro de una de las siguientes categorías:

- Interpretación e identificación
- Predicción
- Diagnóstico
- Diseño
- Planeación
- Monitoreo
- Depuración y ensayo
- Instrucción y educación
- Control

Los problemas que son buenos candidatos para la implementación de sistemas expertos son aquellos que requieren el uso de conocimiento heurístico para su solución. Si necesitamos el uso de la intuición, el juicio, la inferencia lógica y el conocimiento de un experto para la solución de un problema, podemos hablar de la necesidad de un sistema experto.

Los componentes principales de un sistema experto son:

- *Base de conocimiento*. Una representación declarativa de la experiencia, a menudo en reglas si/entonces.
- *Memoria de trabajo*. Los datos que son específicos para que un problema sea resuelto.
- *Máquina de inferencia*. El código que deriva las recomendaciones de la base de conocimientos y los datos del problema específico.

- *Comunicación con el usuario y el medio ambiente.* El código que controla el diálogo entre el usuario y el sistema, esta comunicación debe ser capaz de comprender, traducir y resolver el problema planteado por el usuario pero también debe ser capaz de explicar las razones por las que llegó a una conclusión o tomó una decisión. El sistema debe ser capaz de adquirir información del medio ambiente, consultar o cuestionar al usuario y a otros sistemas expertos.

Para entender el diseño de un sistema experto es necesario entender también los papeles principales de las personas que se relacionan con el sistema. Hay:

- *Experto del campo.* El individuo que comúnmente es experto en la solución de problemas del sistema que se intenta resolver.
- *Ingeniero del conocimiento.* El individuo que codifica el conocimiento del experto en forma declarativa para que puede ser usada por el sistema experto.
- *Usuario.* El individuo que consulta al sistema para obtener consejos que podrían haber sido proporcionados por un experto.

Los sistemas expertos son necesarios por las limitaciones asociadas con el proceso convencional de toma de decisiones de los humanos y del seguimiento de variables del problema en cuestión. Estas limitaciones incluyen:

- Los expertos humanos son muy escasos.
- Los humanos se cansan por el trabajo mental o físico.
- Los humanos olvidan detalles cruciales de un problema.
- Los humanos son inconsistentes en sus decisiones día a día.
- Los humanos tienen memoria de trabajo limitada.
- Los humanos son incapaces de comprender rápidamente grandes cantidades de datos.
- Los humanos son incapaces de retener grandes cantidades de datos en memoria.
- Los humanos son lentos para recordar información almacenada en memoria.
- Los humanos son propensos a prejuzgar deliberada o inadvertidamente sus acciones.
- Los humanos pueden deliberadamente eludir responsabilidades en las decisiones.
- Los humanos descansan, desaparecen y mueren.

Consecuentemente, los beneficios obtenidos por la utilización de sistemas expertos son:

- Incrementan la probabilidad, frecuencia y consistencia de la toma de decisiones buenas.
- Ayudan a distribuir el conocimiento humano.
- Permiten tomar decisiones en tiempo real, a bajo costo y a nivel de experto utilizando un operador.
- Permiten la utilización de la mayoría de los datos disponibles.
- Permiten la objetividad ante el peso de la evidencia sin la predisposición emocional y sin la apreciación del usuario.
- Permiten dinamismo a través de la modulariad de la estructura.
- Liberan la mente y el tiempo de los expertos humanos para capacitarse ellos o para concentrarce en actividades más creativas; por ejemplo, generar nuevos conocimientos.
- Estimulan la investigación en áreas refinadas de un problema.

Sistemas expertos en procesos metalúrgicos

Planeación y producción

Los sistemas expertos han mostrado gran versatilidad en el desarrollo de planes de producción debido a que las plantas no pueden depender de un solo producto (una siderúrgica puede producir mas de 2000 grados diferentes de acero) y de que deben responder a: i) cambios en composición y existencia de materia prima; ii) cambios en el ritmo y la prioridad de producción; y iii) paros técnicos. Los ingenieros industriales deben responder al problema del diseño de planes de producción tomando en cuenta las limitaciones materiales, técnicas, económicas y humanas; la explosión combinatoria y la necesidad de reducir el error humano y el tiempo de respuesta en la toma de decisiones hacen atractiva la aplicación de sistemas expertos.

Control de proceso u operación

Para ejemplificar la aplicación de los sistemas expertos en procesos metalúrgicos presentaremos cinco casos:

1. *Sistema experto para planta de procesamiento de minerales usando redes neuronales*[2,Bradford]. El hidrociclón es uno de los equipos mas ampliamente usado en el procesamiento de minerales. Esta herramienta es relativamente fácil de operar pero su modelado fenomenológico es extremadamente complejo; por eso se utilizan los modelos de control

empíricos calibrados para cada caso particular. Haciendo un estudio de la influencia de las variables de operación se llega a la conclusión de que los factores que afectan en gran medida el control de la operación son: la presión del hidrociclón, la fracción de sólidos alimentada y la distribución del tamaño de partícula (normal, grueso y fino). Usando un modelo de redes neuronales se desarrolló un sistema de control; el modelo mostró ser tan bueno como otros modelos ya desarrollados para simular la operación; además, al utilizar datos de tamaño de partícula con dispersión mostró capacidad para analizar los datos y controlar correctamente la operación.

2. *Sistema experto para alto horno*[2,Choi y otros]. El alto horno consta de una gran cantidad de variables de operación que deben ser constantemente analizadas, algunas de estas variables de operación tienen un efecto relativamente lento en el sistema; por ejemplo, los materiales que son distribuidos y cargados en la parte superior del horno tardan de 4 a 8 horas en descender a la zona inferior del horno. Por ello muchas veces el seguimiento de estas variables de operación pierde continuidad debido a la gran cantidad de variables de operación y a los cambios de turno. Debido a esto se han implementado sistemas expertos que: i) procesan la diferentes variables de operación; ii) analizan el comportamiento de estas variables de operación a largo plazo; iii) definen el estado actual del horno; iv) toman acciones correctivas y preventivas. Para el procesamiento de las variables de operación se utilizan métodos estadísticos y para el análisis de estos datos y la toma de decisiones se utilizan reglas de inferencia basadas en los conocimientos proporcionados por los expertos.
3. *Sistema experto híbrido para alto horno*[3]. Este sistema propone la utilización de dos tipos de conocimiento, un modelo matemático y reglas de experiencia. El modelo matemático propuesto fue el modelo TS para control térmico, modelo utilizado durante 10 años para predecir las condiciones de operación¹⁰ del horno. Sin embargo, existen estados donde la utilización de este modelo no es confiable; basados en la experiencia los operadores desarrollaron reglas para poder definir en que situaciones no era conveniente utilizar el modelo. Con base en estos conocimientos se desarrolló un sistema para analizar las variables de operación y decidir si el estado del horno es adecuado para utilizar el modelo matemático o si se utilizan las reglas de experiencia¹¹.

¹⁰Fundamentalmente flujo, temperatura y composición de soplo y combustibles secundarios (gas natural, combustóleo, carbón en polvo, etc.).

¹¹También conocidas como reglas empíricas.

4. *Sistema experto híbrido para convertidor*[2,Yoshida y otros]. Este modelo utiliza balances de materia y energía para predecir la combinación de arrabio, pellet, chatarra, volumen de oxígeno y formadores de escoria para obtener acero con composición y temperatura controlada. Sin embargo, el modelo no predice exactamente el estado final del acero, por lo que es necesario hacer mediciones de la temperatura y composición química del acero antes de vaciar; el ajuste final de la composición química y la temperatura se hace utilizando reglas empíricas. Respecto a la composición de la escoria se hace un seguimiento del consumo de oxígeno en el sistema analizando durante el proceso los gases de salida y la temperatura del acero; pero también se utilizan reglas basadas en los conocimientos de los operadores para estimar el estado de la escoria y poder determinar el camino a seguir en caso de que sea necesario efectuar operaciones adicionales para controlar el contenido de fósforo y manganeso.
5. *Sistema experto para la inspección superficial automática de aceros planos* [2,Haataja y otros]. Utiliza tecnología de inspección de máquina basada en la revisión de imagen en línea, el procesamiento muy rápido de imagen y sofisticados algoritmos de clasificación basados en conocimientos para defectos de superficie. Este sistema se usa para control de calidad en varias etapas de manufactura donde la inspección visual es requerida. Tradicionalmente los inspectores humanos hacen el trabajo debido a la carencia de sistemas automatizados; sin embargo, existen muchos problemas relacionados con la inspección humana, por ejemplo, el criterio de inspección difiere entre las personas, la ejecución varía con el transcurso del tiempo, y la dificultad de revisar productos con alta velocidad de producción (300 m/min). Esto ha motivado la implementación y desarrollo de sistemas automatizados de inspección visual.

2.2 Desarrollo de programas de cómputo

En general, un programa de cómputo no es más que una descripción abstracta de un procedimiento o fenómeno que existe o sucede en el mundo real. Frecuentemente un programa imita un comportamiento o acción humana; otras veces simula un fenómeno físico. Sin embargo, la relación entre abstracción y lenguaje de programación es doble: por un lado se utiliza el lenguaje de programación para escribir un programa que es una abstracción del mundo real; por otro lado se utiliza el lenguaje de programación para describir de un modo abstracto el comportamiento físico de la computadora que se está utilizando. El arte de la programación es el método por el que se describirá a una computadora (mediante un lenguaje de programación) un fenómeno, una acción, un

comportamiento o una idea. La construcción de programas requiere el cumplimiento de numerosas características. Entre ellas destacan las siguientes:

- *Eficiencia.* La eficiencia del programa es su capacidad para hacer un buen uso de los recursos que manipula; en especial, es importante que se ejecute con la mayor rapidez posible.
- *Movilidad.* Es la facilidad con la que un programa puede ser transportado a diferentes sistemas físicos o lógicos.
- *Verificación.* Un programa debe de ser capaz de soportar los procedimientos de validación y de aceptar pruebas o ensayos de programas.
- *Integridad.* La integridad es la capacidad de un programa para proteger sus propios componentes contra los procesos que no tengan derecho de acceso.
- *Facilidad de utilización.* Un programa es fácil de utilizar si se puede comunicar con él de manera cómoda.
- *Corrección.* Capacidad de realizar exactamente las tareas definidas por su especificación.
- *Robustez.* Capacidad de funcionar incluso en situaciones anormales.
- *Adaptabilidad.* Facilidad que tienen los productos de adaptarse a cambios en su especificación.
- *Reutilización.* Capacidad de los productos para ser reutilizados, en su totalidad o en parte, en nuevas aplicaciones.
- *Compatibilidad.* Facilidad de los productos para ser combinados con otros.

Sin embargo, esto eleva la complejidad y tiempo de desarrollo de un programa haciendo necesaria una evaluación del costo del tiempo de programación respecto al costo de ejecución del programa, de modo que el costo a optimizar sea la escritura del programa. En cambio, cuando se presenta un problema cuya solución se va a utilizar muchas veces, el costo de ejecución del programa puede superar en mucho al de escritura, en especial si en la mayor parte de las ejecuciones se tienen datos iniciales de gran tamaño. Entonces, es más ventajoso, desde el punto de vista económico, realizar un algoritmo complejo siempre que el tiempo de ejecución del programa resultante sea significativamente menor que el de un programa más simple.

Aunque todos los programas pueden ser escritos en términos de solo tres estructuras de control (estructura de secuencia, estructura de selección y estructura de repetición) existen diferentes estilos y lenguajes de programación que nos permiten enfrentar un mismo problema desde diferentes

enfoques; pero también es posible combinar las estrategias y los lenguajes para resolver un problema común. Los principales estilos de programación son:

- Programación estructurada
- Programación orientada a objetos
- Programación lógica

Estas técnicas se han desarrollado y perfeccionado, reduciendo el tamaño de los programas y reutilizando códigos, fundamentalmente para: i) mejorar la eficiencia, integridad y robustez; y ii) disminuir el tiempo utilizado para el diseño, desarrollo y mantenimiento del programa.

2.2.1 Programación estructurada

Se emplea desde principio de la década de los setenta. Propone la eliminación de las transferencias de control (sentencia GOTO) e introduce el concepto de diseño descendente; el diseño descendente propone la división del problema en problemas más pequeños (;Divide y vencerás!) conocidos como funciones (subrutinas, subprogramas o procedimientos) que realizan tareas menos complejas. Un programa estructurado se construye rompiendo el programa en funciones; sin embargo, durante la elaboración del programa se puede presentar la necesidad de añadir funciones adicionales en los niveles superiores, o la conveniencia de consolidar ciertas funciones de nivel superior y diseño ascendente. También se introduce el concepto de abstracción que se puede definir como la capacidad para examinar algo sin preocuparse de sus datos internos (caja negra). En un programa estructurado es suficiente conocer que un procedimiento dado realiza una tarea específica. El cómo realiza esta tarea no es importante, sino conocer cómo se utiliza correctamente la función y lo que hace.

En un programa estructurado las estructuras de datos son tan importantes como las operaciones realizadas sobre ellas. Esto se hace más evidente a medida que crece un programa en tamaño. Los tipos de datos se procesan en muchas formas dentro de un programa estructurado, y cuando se producen cambios en esos tipos, las modificaciones se deben hacer en cada posición que actúa sobre ellos dentro del programa. Esta tarea puede ser frustrante y consumir un tiempo considerable en programas con millones de líneas de código y centenares de funciones. En un programa estructurado, los datos locales se ocultan dentro de funciones y los datos compartidos se pasan como argumentos; las estructuras de datos utilizadas en estos programas son con frecuencia globales o se pasan explícitamente como parámetros. Sin embargo, es común sugerir la eliminación

de variables globales y el uso de la mayor cantidad de variables locales posibles.

2.2.2 Programación orientada a objetos (POO)

Un programa orientado a objetos se compone solamente de objetos. Un objeto es una encapsulación genérica de datos y de los procedimientos para manipularlos. Cuando se ejecuta un programa orientado a objetos, los objetos están recibiendo, interpretando y respondiendo a mensajes de otros objetos. La POO puede considerarse como una extensión natural de la programación estructurada en un intento de potenciar los conceptos de modularidad y reutilización del código. Para reutilizar el código se propone un mecanismo para compartir automáticamente los procedimientos y datos de un tipo genérico de objeto (definido como clase o clase base) con otro objeto (definido como subclase o clase derivada) quien hereda estas características y agrega nuevos procedimientos y datos al nuevo tipo de objeto que pueden ser proporcionados por otra clase base (herencia múltiple). Otra característica importante de los lenguajes orientados a objetos es su capacidad para implementar múltiples formas de un mismo método (polimorfismo), dependiendo cada una de ellas de la clase sobre la que se realice la implementación. Esto permite acceder a una variedad de métodos distintos, todos con el mismo nombre, utilizando exactamente el mismo medio de acceso.

2.2.3 Programación lógica

Un programa lógico es un conjunto finito de axiomas y reglas que definen relaciones entre objetos. La programación lógica esta basada en la lógica de primer orden, resuelve problemas aplicando técnicas originalmente desarrolladas para probar teoremas lógicos. La programación lógica utiliza un sistema o principios de razonamiento basado en reglas de inferencia cuyo objetivo es permitirnos razonar correctamente. Sin embargo, estas reglas no son de gran utilidad si no sabemos como aplicarlas a nuestro lenguaje, como el español. Cualquier lenguaje hablado es muy complicado y, además, es ambiguo; para enfrentar este problema es necesario traducir el conocimiento explicado utilizando nuestro lenguaje coloquial a un lenguaje formal libre de ambigüedades. Los matemáticos han desarrollado un lenguaje basado en términos y predicados, conocido como lenguaje para lógica o cálculo de predicados que permite representar el conocimiento de una manera muy parecida a nuestro lenguaje común. La programación lógica utiliza una versión simplificada del cálculo de predicados conocida como forma clausal, la cual utiliza únicamente cláusulas de Horn.

En la programación lógica describimos el conocimiento que queremos representar utilizando nuestro lenguaje cotidiano y utilizamos el lenguaje del cálculo de predicados para expresarlo o declararlo de manera explícita pero libre de ambigüedades. Decimos que el conocimiento representado usando este camino es declarativo porque es contenido en declaraciones acerca del mundo. Tales enunciados son almacenados en estructuras simbólicas que son utilizadas por los procedimientos que usan este conocimiento; a este tipo de programación también se le conoce como programación declarativa. Una diferencia fundamental entre este tipo de programación y las otras dos es que la programación estructurada y la programación orientada utilizan una secuencia de operaciones o procedimientos ejecutada por el programa de manera tal que el conocimiento utilizado es implícito porque está contenido en el conjunto de procedimientos que usa éste; decimos que son técnicas de programación procedimentales.

Términos y predicados

Básicamente, un enunciado atómico consta de un sujeto y un predicado. Por ejemplo:

1. Ana está embarazada.

- Sujeto: Ana; predicado: está embarazada.

a este enunciado se le puede asignar un valor de verdad (verdadero o falso). Sin embargo, hay expresiones que tienen la forma de un enunciado, pero no son enunciados porque no son ni verdaderos ni falsos.

1. Ella está embarazada.

- Sujeto: Ella; predicado: está embarazada.

2. X es un número primo.

- Sujeto: X; predicado: es un número primo.

Si nosotros no sabemos quien es "ella" en el primer enunciado, o que "x" es en el segundo, no podemos determinar en que caso esta expresión es verdadera o falsa. Tales expresiones serán llamadas **formas o expresiones proposicionales**, porque si el pronombre es remplazado por un nombre y la variable es remplazada por una constante, éstas se convertirán en enunciados.

Los sujetos de cada uno de los enunciados presentados (Ana, ella, X) son ejemplos de términos. Un término es una expresión que es un nombre, una constante o una función, o que resulta en un nombre, constante o función cuando las variables o pronombres son remplazados por éstos. Se utiliza la palabra predicado para indicar una propiedad o relación. Los predicados presentados anteriormente son de grado 1; esto es, sólo se aplican a un término. Otros predicados pueden aplicarse a dos o más términos; por ejemplo,

"es mamá de". Formalmente, en matemáticas, predicados, funciones y términos son definidos como:

Definición: Un predicado o relación P de n lugares es un conjunto de eneadas ordenadas.

Si $n > 1$, el dominio de un predicado P de n lugares, R , es el conjunto de todas las eneadas $n-1$ $\langle u_1, u_2, \dots, u_{n-1} \rangle$ tales que para alguna v , $\langle u_1, u_2, \dots, u_{n-1}, v \rangle \in R$.

Definición: f es una función de n variables si f es una relación de $n+1$ lugares tales que para toda $\langle u_1, u_2, \dots, u_n \rangle$ en el dominio de f , si $\langle u_1, u_2, \dots, u_{n-1}, v_1 \rangle$ y $\langle u_1, u_2, \dots, u_{n-1}, v_2 \rangle$ son ambos elementos de f entonces $v_1 = v_2$.

Esto es, cada elemento v tiene asociada una y solo una eneada n $\langle u_1, u_2, \dots, u_n \rangle$; esto implica que toda función es una relación pero no viceversa.

Generalmente, escribimos $f(u_1, u_2, \dots, u_n) = v$ en lugar de $\langle u_1, u_2, \dots, u_n, v \rangle \in f$, donde f es llamado el símbolo de función.

Definición: Términos son definidos como:

1. Constantes y variables son términos.
2. Si f es un símbolo de función de n variables y t_1, t_2, \dots, t_n son términos, entonces $f(t_1, t_2, \dots, t_n)$ es un término.
3. Otras expresiones no son términos.

En esta sección, escribimos $P(t_1, t_2, \dots, t_n)$ para representar una relación o predicado P con n lugares, y t_1, t_2, \dots, t_n términos, donde P es el símbolo del predicado. Para construir el símbolo del predicado P , utilizaremos las palabras del predicado obtenidas de un enunciado unidas por el carácter "_". Por ejemplo:

- Ana está embarazada.
 Término: Ana.
 Predicado: está embarazada.
 Símbolo de predicado: está embarazada.
 Traducción: está embarazada (Ana).

- Samuel quiere a Ana.
 Términos: $\langle \text{Samuel}, \text{Ana} \rangle$.
 Predicado: quiere a.
 Símbolo de predicado: quiere_a .
 Traducción: $\text{quiere}_a(\text{Samuel}, \text{Ana})$.
- Teraj es padre de Abraham.
 Términos: $\langle \text{Teraj}, \text{Abraham} \rangle$.
 Predicado: es padre de.
 Símbolo de predicado: es_padre_de .
 Traducción: $\text{es_padre_de}(\text{Teraj}, \text{Abraham})$.

Utilizaremos esta notación porque es muy parecida a la utilizada por el lenguaje de programación Prolog¹², y porque es más explícita que la utilizada por el lenguaje formal¹³ del cálculo de predicados.

Hechos

Un hecho está dado por un predicado o relación entre términos particulares llamados constantes y es una fórmula atómica o primitiva. Un conjunto finito de hechos constituye un programa, esta es la forma más simple de un programa lógico. Por ejemplo, los enunciados:

1. Teraj es padre de Abraham.
2. Teraj es padre de Najor.
3. Teraj es padre de Harán.
4. Abraham es padre de Isaac.
5. Harán es padre de Lot.
6. Harán es padre de Milka.
7. Harán es padre de Yiska.
8. Saray es madre de Isaac.
9. Teraj es hombre.
10. Abraham es hombre.
11. Lot es hombre.
12. Najor es hombre.

¹²Ver Apéndice A.

¹³Cuyas traducciones hubieran sido: Ea para "Ana está embarazada"; Qsa para "Samuel quiere a Ana"; y Pta para "Teraj es padre de Abraham".

13. Harán es hombre.
14. Isaac es hombre.
15. Saray es mujer.
16. Milka es mujer.
17. Yiska es mujer.

Los cuales pueden ser traducidos para formar el conjunto finito B de hechos:

```
{es_padre_de(Teraj,Abraham),
es_padre_de(Teraj,Najor),
es_padre_de(Teraj,Haran),
es_padre_de(Abraham,Isaac),
es_padre_de(Haran,Lot),
es_padre_de(Haran,Milka),
es_padre_de(Haran,Yiska),
es_madre_(Saray,Isaac),
es_hombre(Teraj),
es_hombre(Abraham),
es_hombre(Lot),
es_hombre(Najor),
es_hombre(Haran),
es_hombre(Isaac),
es_mujer(Saray),
es_mujer(Milka),
es_mujer(Yiska),
}
```

Regla

La regla es un enunciado¹⁴ de la forma :

$$A \leftarrow B_1 \wedge B_2 \wedge \dots \wedge B_n.$$

Donde $n \geq 0$. A es un enunciado atómico que es conocido como la consecuencia ó conclusión de la regla, la eneada de enunciados $\langle B_1, B_2, \dots, B_n \rangle$ y el símbolo de conjunción \wedge forman un enunciado que es conocido como el antecedente ó hipótesis. Note que un hecho es precisamente un caso especial de una regla donde $n=0$ y sus términos son constantes. Las reglas pueden ser vistas de dos maneras. Por un lado, es un medio para expresar preguntas más complejas en

¹⁴Un enunciado atómico es un enunciado. De la misma manera, Si ϕ y ψ son enunciados, entonces $(\neg\phi)$, $(\phi \wedge \psi)$, $(\phi \vee \psi)$, $(\phi \rightarrow \psi)$, y $(\phi \leftrightarrow \psi)$ también lo son.

términos de preguntas simples. El segundo enfoque viene de la interpretación de la regla como un enunciado; el símbolo \leftarrow es usado para denotar la implicación ó condición inversa.

La traducción normal del condicional inverso \leftarrow es "... si...". Sin embargo, existen otras expresiones que pueden significar lo mismo; por ejemplo: "si..., entonces...", "una condición necesaria para... es que...". Por ejemplo, si tenemos el enunciado "Abraham es hijo de Teraj si Teraj es padre de Abraham y Abraham es hombre" tiene tres enunciados atómicos:

1. Abraham es hijo de Teraj.

Traducción: $es_hijo_de(Abraham, Teraj)$.

2. Teraj es padre de Abraham.

Traducción: $es_padre_de(Teraj, Abraham)$.

3. Abraham es hombre.

Traducción: $es_hombre(Abraham)$.

Además, los enunciados atómicos dos y tres están unidos por la conjunción "y", conector \wedge , de manera tal que forman el antecedente del enunciado condicional. La traducción del enunciado sería:

$es_hijo_de(Abraham, Teraj) \leftarrow es_padre_de(Teraj, Abraham) \wedge es_hombre(Abraham)$

Podemos hacer una generalización de esta regla con el enunciado "Existen un par de términos (X, Y) tal que X es hijo de Y si Y es padre de X y X es hombre. Donde X y Y pertenecen al conjunto T de términos {Teraj, Abraham, Najor, Harán, Isaac, Lot, Milka, Yiska, Saray}". El conjunto T se obtiene de los enunciados 1 a 7 o de revisar el conjunto B . La traducción de este enunciado es:

$es_hijo_de(X, Y) \leftarrow es_padre_de(Y, X) \wedge es_hombre(X)$

Si no sabemos quién o qué términos pueden formar al par (X, Y) entonces el enunciado anterior no se puede definir como una regla y decimos que es una **forma o expresión proposicional**.

Unificación

El corazón de los programas lógicos es el algoritmo de unificación. La unificación es la base de la mayoría de los trabajos en deducción automatizada, y el uso de la inferencia lógica en inteligencia artificial. La unificación es el proceso de determinar si dos enunciados o expresiones pueden ser iguales.

Definición: Una sustitución es un conjunto finito de pares de la forma $x_i=t_i$, donde x_i es una variable, t_i es un término, $x_i \neq x_j$, para toda $i \neq j$, y x_i no ocurre en t_j para ningún i y j .

Definición: A es una instancia de B si hay una sustitución θ tal que $A=B\theta$.

Un ejemplo de una sustitución consta del par $\{x=Isaac\}$. Las sustituciones pueden ser aplicadas a expresiones proposicionales para crear nuevos enunciados. El resultado de aplicar una sustitución θ a una expresión proposicional A, denotado por $A\theta$, es la relación¹⁵ obtenida por reemplazar toda ocurrencia x por t en A, para todo par $x=t$ en θ . El resultado de aplicar $\{x=Isaac\}$ a la expresión `es_padre_de(abraham,x)` es la instancia `es_padre_de(Abraham,Isaac)`.

Definición: Un conjunto de expresiones proposicionales $\{A_1, A_2, \dots, A_n\}$ es unificable si y solo si hay una sustitución θ , llamada sustitución de unificación, que hace que las relaciones sean idénticas.

Por ejemplo, para las relaciones `añadir([1,2,3],[3,4],L)` y `añadir([X|Xs], Ys, [X|Zs])` la sustitución de unificación es $\{X=1, Xs=[2,3], Ys=[3,4], L=[1|Zs]\}$. Su instancia común, determinada por la sustitución de unificación, es `añadir([1,2,3], [3,4], [1|Zs])` quedando unificadas las dos relaciones.

Preguntas

Es el medio para obtener información en un programa lógico. Con esta estructura se indaga sobre la existencia de una relación entre constantes. Por ejemplo, la pregunta `es_padre_de(Abraham, Isaac)?` investiga si existe la relación "es_padre_de" entre los términos `Abraham` e `Isaac`; si el hecho existe en la base de datos del programa, la respuesta a la pregunta será afirmativa. Si una pregunta se contesta con una negación, esto significa realmente que no se puede probar porque el hecho no está en la base de conocimientos o no se puede deducir de ella. Al responder a una pregunta un programa está determinando si la pregunta es una consecuencia o conclusión lógica.

Al preguntar podemos utilizar variables que representan al conjunto de constantes acerca del cual el conocimiento esta siendo expresado. Una pregunta que contiene una variable responde si hay un valor para la variable que hace a

¹⁵Esta relación puede ser un enunciado o una expresión proposicional.

la pregunta una conclusión lógica del programa. Por ejemplo, una variable sirve como un camino de expresión para la pregunta `es_padre_de(Abraham,X)?`, cuya respuesta es `X=Isaac` representa al único miembro del conjunto¹⁶ de personas cuyo padre es Abraham {Isaac}. Si hacemos la pregunta `es_padre_de(TeraJ,X)?` La respuesta X podría ser algún miembro del conjunto {Abraham, Najor, Haran}.

El intérprete y los programas lógicos

Definición: Un programa lógico es un conjunto finito de reglas.

Definición: Un objetivo existencialmente cuantificado G es una consecuencia lógica de un programa P si hay una cláusula P con una instancia básica $A \leftarrow B_1 \wedge B_2 \wedge \dots \wedge B_n$, $n \geq 0$, tal que $B_1 \wedge B_2 \wedge \dots \wedge B_n$ son consecuencias lógicas de P y A es una instancia de G.

Operacionalmente, las respuestas a preguntas reflejan la definición de conclusiones lógicas. Para probar un objetivo A, con un programa P, buscamos una regla $A_1 \leftarrow B_1 \wedge B_2 \wedge \dots \wedge B_n$ en P, y buscamos una sustitución θ tal que $A = A_1\theta$ y se encuentra $B_i\theta$ para $1 \leq i \leq n$, probando cada $B_i\theta$. Si realizamos una pregunta, será necesario efectuar una serie de cálculos para encontrar la solución a la pregunta, el sistema encargado para efectuar estos cálculos se define como intérprete abstracto.

Definición: El objetivo actual de cualquier estado de un cálculo es llamado resolvente.

Definición: Una traza del intérprete es la sucesión de resolventes producida durante el cálculo mediante las elecciones hechas.

Definición: Una reducción de la base de un objetivo G de un programa P es el remplazo de G por el cuerpo de una instancia base de una regla en P, dicha cabeza es idéntica al objetivo buscado.

La reducción es la etapa computacional básica en la programación lógica. El objetivo reemplazado en la reducción es reducido, y son derivados nuevos objetivos. Por ejemplo:

¹⁶ Según la tradición bíblica, el conjunto de personas cuyo padre es Abraham es {Ismael, Isaac} pero, como no incluimos la sentencia "Abraham es padre de Ismael", el conjunto de personas cuyo padre es Abraham corresponde a {Isaac} para este caso.

```

/* Programa de árbol familiar bíblico */
es_padre_de(Teraj,Abraham) .
es_padre_de(Teraj,Najor) .
es_padre_de(Teraj,Haran) .
es_padre_de(Abraham,Isaac) .
es_padre_de(Haran,Lot) .
es_padre_de(Haran,Milka) .
es_padre_de(Haran,Yiska) .
es_madre_(Saray,Isaac) .
es_hombre(Teraj) .
es_hombre(Abraham) .
es_hombre(Lot) .
es_hombre(Najor) .
es_hombre(Haran) .
es_hombre(Isaac) .
es_mujer(Saray) .
es_mujer(Milka) .
es_mujer(Yiska) .

es_hijo_de(X,Y)←es_padre_de(Y,X)∧es_hombre(X) .

```

Si nosotros preguntamos `es_hijo_de(lot,haran)?`, el intérprete generará la siguiente traza para buscar la solución:

Entrada: `es_hijo_de(lot,haran)?` y programa del árbol familiar bíblico.

Resolvente: `es_hijo_de(lot,haran)` .

Al buscar `es_hijo_de(lot,haran)` .

Encontramos: `es_hijo_de(lot,haran)←es_padre_de(haran,lot)∧es_hombre(lot)` .

El nuevo resolvente es: `es_padre_de(haran,lot),es_hombre(haran)?`

Al buscar `es_padre_de(haran,lot)` .

Encontramos: `es_padre_de(haran,lot) /*verdadero*/`

Al buscar: `es_hombre(lot)`

Encontramos: `es_hombre(lot) /*verdadero*/`

No hay nuevos resolventes.

El interprete contestará afirmativamente, pero si preguntamos `es_hijo_de(Saray,Isaac)?` el interprete contestará no¹⁷ ya que en el programa no está presente la regla `es_hijo_de(X,Y)←es_madre_de(Y,X)∧es_hombre(X)`. El conjunto de reglas con el mismo predicado en la cabeza es llamado procedimiento.

El modelo de Herbrand

Cuando nosotros desarrollamos un programa lógico lo hacemos pensando en un dominio del mundo real. Sin embargo, la computadora no opera directamente sobre este dominio, su trabajo esta restringido a su propio dominio interno.

Definición. Sea P un programa lógico. El universo de Herbrand de P , denotado $U(P)$, es el conjunto de todos los términos meta que pueden ser formados de las constantes y las relaciones que aparecen en P .

En el programa del árbol familiar bíblico (P), el universo de Herbrand $U(P)$ es el conjunto de todas las constantes que aparecen en el programa $\{Tera_j, Abraham, Najor, Harán, Isaac, Lot, Milka, Yiska, Saray\}$. Si no hay funciones, el universo de Herbrand es finito.

Definición. La base de Herbrand, denotada como $B(P)$, es el conjunto de todos los objetivos meta que pueden ser formados de los predicados en P y los términos en el universo de Herbrand. La base de Herbrand es finita si el universo de Herbrand lo es.

Para P existen cinco símbolos de predicado $\{es_padre_de, es_madre_de, es_hombre, es_mujer, es_hijo_de\}$. La base de Herbrand $B(P)$ incluye 81 metas para cada relación con dos términos $\{es_padre_de, es_madre_de, es_hijo\}$ y 9 metas para las relaciones con un término $\{es_hombre, es_mujer\}$.

`es_padre_de(Tera_j,Tera_j),`
`es_padre_de(Tera_j,aAraham),`
`es_padre_de(Tera_j,Najor),`

¹⁷La "negación" en programación lógica se suele definir como "negación por falla", es decir, algo es falso si no puede encontrarse. Así, al preguntar `es_hijo_de(Saray,Isaac)?`, el programa al carecer de información da como respuesta no. Este concepto de negación tiene sus limitaciones, ya que no responde a todo el concepto de negación lógica y también hay otras alternativas (otras definiciones), pero el concepto y aplicación de la negación en programación lógica sigue siendo tema de discusión.

```

es_padre_de(Teraj,Haran) ,
es_padre_de(Teraj,Isaac) ,
es_padre_de(Teraj,Lot) ,
es_padre_de(Teraj,Milka) ,
es_padre_de(Teraj,Yiska) ,
es_padre_de(Teraj,Saray) ,
es_padre_de(Abraham,Abraham) ,
es_padre_de(Abraham,Teraj) ,
es_padre_de(Abraham,...
es_madre_de(Teraj,Teraj) ,
es_madre_de(Teraj,Abraham) ,
es_madre_de(.....
es_hombre(Teraj) ,
es_hombre(Abraham) ,
es_hombre(...
es_mujer(Saray) }

```

Definición: una interpretación para un programa lógico es un subconjunto¹⁸ de la base de Herbrand.

Una interpretación asigna verdad o falsedad a los elementos de la base de Herbrand. Un objetivo en $\mathbf{B(P)}$ es verdadero con respecto a la interpretación si es miembro de éste, falso en cualquier otro caso.

Definición: una interpretación de \mathbf{I} es un modelo para un programa lógico si para cada instancia meta de un predicado en el programa $A \leftarrow B_1 \wedge \dots \wedge B_n$, A está en \mathbf{I} si $B_1 \wedge \dots \wedge B_n$ están en \mathbf{I} .

La intersección de dos modelos para un programa lógico \mathbf{P} es también un modelo. Esta propiedad permite la intersección de todos los modelos.

Definición: El modelo obtenido como la intersección de todos los modelos es conocido como el modelo mínimo y denotado $\mathbf{M(P)}$. El modelo mínimo es el significado afirmativo de un programa lógico.

Para el programa del árbol familiar bíblico $\mathbf{M(P)}$ es

¹⁸Los subconjuntos de la base de Herbrand son interpretaciones para un programa lógico, pero no tienen porque ser las únicas interpretaciones para un programa.

```

{es_padre_de(teraj,abraham),
es_padre_de(teraj,najor),
es_padre_de(teraj,haran),
es_padre_de(abraham,isaac),
es_padre_de(haran,lot),
es_padre_de(haran,milka),
es_padre_de(haran,yiska),
es_madre_de(saray,isaac),
es_hombre(teraj),
es_hombre(abraham),
es_hombre(lot),
es_hombre(najor),
es_hombre(haran),
es_hombre(isaac),
es_hombre(haran),
es_mujer(saray),
es_mujer(milka),
es_mujer(yiska),
es_hijo_de(abraham,teraj),
es_hijo_de(najor,teraj),
es_hijo_de(haran,teraj),
es_hijo_de(isaac,abraham),
es_hijo_de(lot,haran)}

```

2.3 Procesos electroquímicos

Definición: Un proceso unitario electroquímico es una operación unitaria con reacción química que puede producir trabajo eléctrico en el entorno.

Este trabajo eléctrico, W_{el} , es menor o igual que la disminución en la energía de libre de la reacción

$$W_{el} \leq -\Delta G$$

Cuando el sistema suministra trabajo eléctrico al entorno, se dice que tenemos una proceso galvánico. Si, en cambio, el sistema consume trabajo eléctrico de una fuente de energía externa, almacenando como consecuencia energía química, se dice que tenemos un proceso electrolítico.

Todo proceso electroquímico esta compuesto cuando menos por dos conductores electrónicos (electrodos) y un conductor iónico (electrolito) unidos por un conductor externo para dar continuidad al circuito. En este sistema la

corriente fluye a través del electrolito mediante reacciones de oxidación-reducción y la migración iónica a través de la solución. La corriente positiva pasa hacia el electrolito a través de reacciones de oxidación en la interfase de uno de los electrodos y el electrolito, dicho electrodo recibe el nombre de ánodo. Las reacciones de reducción se presentan en la interfase del otro electrodo y el electrolito, permitiendo el flujo de la corriente positiva hacia el electrodo, definido como cátodo. En general, podemos decir que los aniones (iones negativos) migran hacia el ánodo y los cationes (iones positivo) hacia el cátodo.

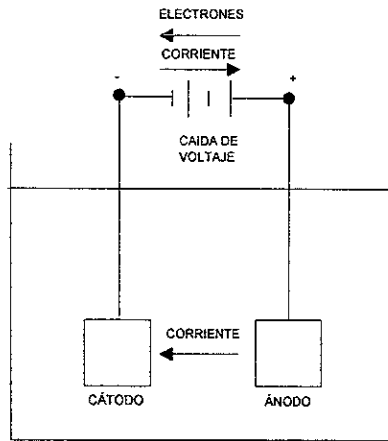


Figura 2.4 Celda electroquímica

2.3.1 Leyes de Faraday

La relación entre la carga transmitida y la cantidad de substancia oxidada o reducida en un electrodo fue cuantificada por Faraday en 1830. Sus ideas se resumen en dos leyes correspondientes a los procesos electroquímicos:

- La cantidad de producto formado es directamente proporcional a la carga transmitida.
- Para una cantidad específica de carga transmitida, la masa de los productos formados es proporcional a los pesos equivalentes de los productos.

Estos dos principios se expresan en una ecuación

$$n_j = \frac{I_j t}{z_j F},$$

donde:

n_j , son las moles producidas por una corriente I_j .

t es el tiempo transcurrido.

z, es el número de electrones asociados con la producción de una molécula de j (equivalente).

F es la constante de Faraday (96487 $\frac{C}{equivalente}$).

2.3.2 Eficiencia de corriente y voltaje

Para un número m de reacciones que ocurren en un electrodo tenemos

$$\sum_{j=1}^m n_j = \frac{t}{F} \sum_{j=1}^m \frac{I_j}{z_j},$$

donde:

$\sum_{j=1}^m n_j$ es el total de moles producidas y

$\sum_{j=1}^m \frac{I_j}{z_j}$ es la corriente total I .

Definimos un término llamado la eficiencia de corriente, ε_j , para la producción de una especie dada j cuando otras reacciones ocurren, como la relación de la corriente producida j respecto a la corriente total.

$$\varepsilon_j = \frac{I_j}{I} .$$

Para calcular la eficiencia total de la energía necesitamos determinar también la eficiencia del voltaje. En un proceso electrolítico se requiere aplicar un voltaje mas grande que el predicho teóricamente; se define esta eficiencia de voltaje, ε_v , como

$$\varepsilon_v = \frac{\text{voltaje teórico}}{\text{caída de voltaje real}} .$$

El voltaje teórico se define como:

$$E_T = E_0 + \eta_c - \eta_a - IR_e,$$

donde:

E_0 es el potencial reversible de reacción.

η_c es el sobrepotencial por concentración.

η_a es el sobrepotencial por activación.

IR_e es la caída óhmica en la celda asociada con la resistencia eléctrica interna. Esta resistencia es debida al electrolito, los materiales de los electrodos y las conexiones eléctricas.

Usando las definiciones de eficiencia de corriente y voltaje podemos obtener la eficiencia de energía ε_e del producto de las eficiencias de corriente y voltaje:

$$\varepsilon_e = \varepsilon_c \varepsilon_v .$$

2.3.2 Termodinámica de procesos electroquímicos

Con el análisis termodinámico de un proceso electroquímico podemos determinar la dirección y magnitud de las fuerzas impulsoras para reacciones electroquímicas potenciales. Una reacción que no está termodinámicamente favorecida no ocurre en un grado significativo y no necesita ser analizada con mayor detalle. Una reacción termodinámicamente favorecida no necesariamente ocurre a una velocidad significativa. Para realizar el análisis de un proceso, es necesario el conocimiento adicional de las condiciones de transporte de masa y de la cinética del proceso.

Para realizar el análisis necesitamos relacionar el potencial termodinámico con variables de estado. Usualmente tratamos a los procesos electroquímicos como sistemas que operan a temperatura y presión constante. Las condiciones de espontaneidad y equilibrio para estos procesos son:

1. Condición de espontaneidad: $\Delta G + W_a < 0$.
2. Condición de equilibrio: $\Delta G + W_a = 0$.

Donde:

ΔG es la energía libre.

W_a es el trabajo asociado con la transformación.

En este tipo de procesos el trabajo más importante es de tipo eléctrico.

Definición: El trabajo eléctrico en un circuito es el número de cargas multiplicado por la diferencia máxima de potencial.

Para una reacción química, el número total de cargas es igual al número de equivalentes multiplicado por la constante de Faraday. Se tiene entonces que:

$$W_e = zF\Delta E .$$

Si despreciamos cualquier otro tipo de trabajo, y suponemos que el proceso es reversible, tenemos que:

$$\begin{aligned} \Delta G + W_a &= \Delta G + W_e = \Delta G + zF\Delta E = 0 \\ -\Delta G &= zF\Delta E. \end{aligned}$$

Esta relación nos permite analizar el equilibrio químico de estos sistemas utilizando mediciones eléctricas.

2.3.3 Fenómenos de transporte

Una descripción general de un sistema electroquímico toma en cuenta los siguientes efectos: flujo de masa, conservación de materia, flujo de corriente, electroneutralidad, cinética de electrodo y flujo de fluidos.

Flujo de masa. Existen fuerzas impulsoras que permiten la transferencia de masa en los sistemas; por ejemplo: gradientes de potencial químico, de presión, térmicos y campos de fuerza. Para representar estas relaciones podemos hacer uso de la teorías de Onsager [25].

1. Los procesos irreversibles se llevan a cabo cerca del equilibrio y las variables termodinámicas se pueden usar para describir el sistema.
2. Para un sistema cercano al equilibrio, las relaciones lineales entre los flujos y fuerzas se consideran válidas y están dadas por

$$N_i = \sum_{j=1}^n L_{ij} X_j,$$

donde:

L_{ij} es un coeficiente fenomenológico.

X_j es la fuerza impulsora.

3. Los coeficientes fenomenológicos son simétricos; por ejemplo,

$$L_{ij} = L_{ji}.$$

Una descripción común para sistemas electroquímicos se puede plantear haciendo algunas suposiciones. Por ejemplo, suponiendo que las concentraciones de las especies electroactivas son extremadamente pequeñas; en este tratamiento sólo se consideran las interacciones entre el soluto y el solventes, mientras que las interacciones soluto-soluto se desprecian. Bajo estas condiciones, a temperatura y presión constantes, podemos describir el flujo molar de una especie como

$$N_i = -z_i u_i F c_i + \underbrace{\nabla \phi}_{\text{potencial eléctrico}} - \underbrace{\nabla D_i c_i}_{\text{potencial químico}} + c_i v$$

donde:

N_i es el flujo molar.

z_i es la carga del ion.

u_i es la movilidad iónica.

c_i es la concentración de la especie.

$\nabla \phi$ es el gradiente de potencial.

D_i^{19} es la difusividad.

v es la velocidad del fluido en el baño.

Densidad de corriente. La corriente surge del movimiento de todas las especies cargadas

$$i = F \sum_{j=1}^n z_j N_j.$$

Balace de materia. El balance en el electrolito se puede expresar como

$$\frac{\partial c_i}{\partial t} = \nabla \cdot N_i + R_i,$$

donde R_i representa la rapidez de producción de masa i .

Electroneutralidad. Debido a que las fuerzas eléctricas entre las especies cargadas son tan grandes, no puede ocurrir una separación de carga significativa. Desviaciones de la electroneutralidad solo se observan en las interfaces electrodo-electrolito²⁰, la cual puede ser modelada separadamente. En el baño, la suposición de neutralidad es válida:

$$\sum_{i=1}^n z_i c_i = 0.$$

Las cuatro ecuaciones anteriores forman la base para el modelado del transporte de masa en soluciones electrolíticas. Para hacer una simulación, necesitamos agregar la descripción del campo de velocidades v de la solución. Esta descripción requiere una solución separada obtenida de las ecuaciones de dinámica de fluidos. Es importante conocer la termodinámica y la cinética de los electrodos, así como posibles reacciones en el baño, para poder plantear las ecuaciones de transporte de masa.

¹⁹La fuerza impulsora para la difusión es el potencial químico, y no la concentración como se muestra en la ecuación. Sin embargo, es difícil determinar en forma experimental el gradiente de potencial químico, mientras que el gradiente de concentración por lo general puede medirse fácilmente; es común utilizar el gradiente de concentración y el coeficiente de difusión para representar esta fuerza impulsora.

²⁰Región de doble capa.

CAPÍTULO 3

DESARROLLO DEL MODELO

3.1 Objetivos

Objetivo general:

Desarrollar un sistema experto para planeación de producción y control de proceso DOX.

Justificación:

1. El proceso DOX es único en el mundo, por lo que es necesario sistematizar, compilar y unificar los conocimientos adquiridos durante 4 años de operación.
2. La planta puede producir una gran variedad de productos; por ejemplo:

Pieza	Diámetro(pulgadas)	Largo(pulgadas)
Tuercas	$\frac{1}{8}, \frac{3}{16}, \dots, \frac{1}{2}, \frac{5}{8}, \dots, 2, 2\frac{1}{4}, 2\frac{1}{2}, 3, 3\frac{1}{2}, 4$	
Rondana plana	$\frac{1}{8}, \frac{3}{16}, \dots, \frac{1}{2}, \frac{5}{8}, \dots, 2$	
Rondana de presión	$\frac{1}{8}, \frac{3}{16}, \dots, \frac{1}{2}, \frac{5}{8}, \dots, 2$	
Tornillo	$\frac{1}{8}, \frac{3}{16}, \dots, \frac{1}{2}, \frac{5}{8}, \dots, 2$	$\frac{1}{2}, \frac{3}{4}, \dots, 45$
Espárrago	$\frac{3}{8}, \frac{1}{2}, \frac{5}{8}, \dots, 2, 2\frac{1}{4}, 2\frac{1}{2}, 3, 3\frac{1}{2}, 4$	$2, 2\frac{1}{4}, \dots, 45$

Tabla 3.1 Materiales que se recubren con aleación DOX.

debido al número de combinaciones posibles, la planeación de la producción se convierte en un problema de explosión combinatoria que puede enfrentarse con técnicas de búsqueda y planeación propias de la inteligencia artificial y los sistemas expertos.

3. El desarrollo de un modelo que permita estimar la densidad de corriente y la resistencia de la celda permitiría utilizar un operador técnico en vez de un ingeniero para controlar el proceso.
4. Automatización de proceso.

Objetivos específicos:

1. **Cálculo de rondas.** Utilizando una solicitud o requisición de material, los inventarios de materia prima y producto terminado y las reglas de inferencia utilizadas por el coordinador de producción, el sistema experto será capaz de:

- Calcular la cantidad de barras y tuercas que se deben comprar para completar una requisición de material.
 - Calcular el tipo y número de rondas necesarias para procesar un conjunto de partidas y el tiempo mínimo de producción.
 - Seleccionar un conjunto de partidas que puedan ser procesadas de manera tal que existan suficientes tuercas y barras para procesar la cantidad de piezas solicitada en la requisición.
2. **Estimación de variables de operación.** Utilizando la información proporcionada por los departamentos de control de proceso y control de calidad, el sistema experto será capaz de proponer un valor correcto para la resistencia de la celda y la densidad de corriente.
3. **Control de proceso.** Utilizando un sistema de adquisición de datos y salida de voltaje, el experto será capaz de controlar automáticamente la intensidad de corriente. Simultáneamente, el controlador evaluará la resistencia de la celda y, si concluye que posiblemente existe un corto circuito, suspenderá el control automático e iniciará un diálogo con el usuario para determinar las acciones necesarias para corregir el problema; al corregir la causa del problema, el experto permitirá que el control automático siga controlando el proceso.

Ventajas:

Obtención de los beneficios de los sistemas expertos.

Restricciones:

No se analiza la termodinámica ni la cinética del proceso. Por lo que no se puede construir ningún modelo fenomenológico útil para la construcción de un sistema experto híbrido.

3.2 Metodología

Se dividirá la construcción del sistema experto en dos etapas:

1. **Descripción del problema.** Se hará una descripción del problema y de las reglas utilizadas por el experto humano para la solución de éste.
2. **Representación del conocimiento²¹.** Se utilizarán dos lenguajes de programación para representar el conocimiento de manera tal que éste pueda ser manipulado por la computadora. Los lenguajes de programación utilizados son:

²¹Se sugiere tener un conocimiento básico del lenguaje de programación Prolog para leer esta sección. En el Apéndice A se presenta un resumen de la sintaxis e instrucciones de este lenguaje.

- **Programación lógica.** Se construirán estructuras proposicionales para representar la información del sistema, documentada en hojas de control y procedimientos operativos, y se desarrollan reglas para manipular dicha información y hacer cálculos de rondas y estimación de variables de operación.
- **Programación estructurada.** Se desarrollará una programa que permita obtener información del proceso a través de la adquisición de datos y diálogos con el usuario para hacer tomas de decisión para controlar el proceso, modificando las variables de operación y dando órdenes al operador.

3.3 Descripción del problema

3.3.1 Planeación de producción

Cálculo de rondas

Definición: Ronda es el conjunto de piezas que se introducen a la celda electroquímica para recibir un recubrimiento de aleación DOX.

Con base en la experiencia acumulada, se han distinguido los siguientes tipos de arreglos²²:

pieza	Diámetro (plg)	Longitud (plg)	Piezas máximas por ronda
espárrago		$35 \leq L \leq 55$	4
espárrago		$25 \leq L < 35$	6
espárrago		$20 \leq L \leq 25$	8
espárrago	$\phi = 2\frac{1}{2}$	$L \leq 19\frac{1}{2}$	16
espárrago	$\phi = 2\frac{1}{4}$	$14 \leq L \leq 19\frac{1}{2}$	16
espárrago	$\phi = 2$	$16 \leq L \leq 19\frac{1}{2}$	16
espárrago	$\phi = 2$	$L \leq 16$	24
espárrago	$1\frac{1}{2} \leq \phi < 2$	$L \leq 19\frac{1}{2}$	24
espárrago	$\phi < 1\frac{5}{8}$	$10 \leq L \leq 19\frac{1}{2}$	40
espárrago	$\phi < 1\frac{5}{8}$	$6\frac{3}{4} \leq L < 10$	80
espárrago	$\phi < 1\frac{5}{8}$	$4\frac{3}{4} \leq L < 6\frac{3}{4}$	120
espárrago	$\phi < 1\frac{5}{8}$	$3\frac{3}{4} \leq L < 4\frac{3}{4}$	160
espárrago	$\phi < \frac{7}{8}$	$L < 3\frac{3}{4}$	240
tuerca	$1\frac{3}{4} \leq \phi \leq 2\frac{1}{2}$		64
tuerca	$1\frac{3}{8} \leq \phi < 1\frac{3}{4}$		144
tuerca	$1 \leq \phi < 1\frac{1}{8}$		208
tuerca	$\frac{3}{8} \leq \phi < 1$		400

Tabla 3.2 Reglas para calcular el número máximo de piezas por ronda.

²²No se presentan todas las combinaciones posibles porque dificultarían la explicación del problema.

Existen más reglas que permiten diseñar arreglos para aprovechar mejor el espacio disponible e introducir una mayor cantidad de piezas por ronda, por ejemplo:

- *Regla para optimización de espacio:* Existe 40 o más piezas de un espárrago E_1 con $\phi_1 < 1\frac{5}{8}$ y tiene una longitud tal que $10 < L_1 < 13$ y existen 40 o más piezas de otro espárrago E_2 tal que $\phi_1 < 1\frac{5}{8}$ y $6\frac{3}{4} \leq L_2 \leq 10$ de manera tal que $L_1 + L_2 < 19\frac{1}{2}$, entonces se puede construir una ronda de 40 piezas de E_1 y 40 piezas de E_2 .

Otros factores importantes que deben ser considerados son:

- *Existencia de materia prima.*
- *Empaque del material.* Todo espárrago debe ser empacado con su par de tuercas, esto implica procesar el material en un orden tal que, al salir de control de calidad, pueda ser preparado para embarque.
- *Preparación de superficie.* El tiempo de preparación de superficie para tuercas es el factor limitante en la producción del material²³.

Diámetro	$\frac{\text{piezas}}{\text{hora}}$	Diámetro	$\frac{\text{piezas}}{\text{hora}}$
$2\frac{1}{2}$	32	$1\frac{1}{4}$	50
$2\frac{1}{4}$	32	$1\frac{1}{8}$	50
2	32	1	55
$1\frac{7}{8}$	40	$\frac{7}{8}$	120
$1\frac{3}{4}$	40	$\frac{3}{4}$	150
$1\frac{5}{8}$	40	$\frac{5}{8}$	175
$1\frac{1}{2}$	45	$\frac{1}{2}$	200
$1\frac{3}{8}$	45	$\frac{3}{8}$	200

Tabla 3.3 Tiempo de preparación de superficie.

Existe un dispositivo para preparar la superficie de tuercas con diámetro $1 \leq \phi \leq 2\frac{1}{2}$ y dos para diámetros $\frac{3}{8} \leq \phi \leq \frac{7}{8}$. El tiempo utilizado para el cambio de diámetro en la preparación de superficie es de 1 hora para $1 \leq \phi \leq 2\frac{1}{2}$ y 30 minutos para $\frac{3}{8} \leq \phi \leq \frac{7}{8}$.

- *Dispositivo para ronda.* Los sistemas donde se introducen las piezas a la electrólisis tienen una vida útil de 10 rondas.
- *Tiempo de proceso.* El tiempo que dura la electrólisis es diferente para cada tipo de arreglo y el tiempo utilizado para el cambio de dispositivo es de 13' 45''.

²³ Nos referimos exclusivamente al caso de espárragos con tuerca.

Requisición de material

Una solicitud de compra de material típica consta de la siguiente descripción de material.

LOTE	PARTIDA	CANTIDAD	DESCRIPCIÓN	RECUBRIMIENTO	DIAMETRO	LARGO	FECHA
alfa	1	192	jes	dox	3/8	9 1/2	28/02/98
alfa	2	30	jes	dox	1/2	2	28/02/98
alfa	3	262	jes	dox	1/2	2 1/2	28/02/98
alfa	4	3600	jes	dox	1/2	3	28/02/98
alfa	5	780	jes	dox	1/2	3 1/2	28/02/98
alfa	6	156	jes	dox	1/2	4	28/02/98
alfa	7	36	jes	dox	1/2	4 1/2	28/02/98
alfa	8	24	jes	dox	1/2	6	28/02/98
alfa	9	56	jes	dox	5/8	2 1/2	28/02/98
alfa	10	546	jes	dox	5/8	3	28/02/98
alfa	11	5156	jes	dox	5/8	3 1/2	28/02/98
alfa	12	24	jes	dox	5/8	3 3/4	28/02/98
alfa	13	10274	jes	dox	5/8	4	28/02/98
alfa	14	4108	jes	dox	5/8	4 1/2	28/02/98
alfa	15	620	jes	dox	5/8	5	28/02/98
alfa	16	264	jes	dox	5/8	5 1/2	28/02/98
alfa	17	32	jes	dox	5/8	6	28/02/98
alfa	18	212	jes	dox	5/8	6 1/2	28/02/98
alfa	19	112	jes	dox	5/8	7	28/02/98
alfa	20	48	jes	dox	5/8	7 1/2	28/02/98
alfa	21	8	jes	dox	5/8	8 1/2	28/02/98
alfa	22	128	jes	dox	5/8	9 1/2	28/02/98
alfa	23	56	jes	dox	3/4	2 1/2	28/02/98
alfa	24	28	jes	dox	3/4	3	28/02/98
alfa	25	88	jes	dox	3/4	3 1/2	28/02/98
alfa	26	2156	jes	dox	3/4	4	28/02/98
alfa	27	5066	jes	dox	3/4	4 1/2	28/02/98
alfa	28	32	jes	dox	3/4	4 3/4	28/02/98
alfa	29	3426	jes	dox	3/4	5	28/02/98
alfa	30	144	jes	dox	3/4	5 1/4	28/02/98
alfa	31	3278	jes	dox	3/4	5 1/2	28/02/98
alfa	32	256	jes	dox	3/4	6	28/02/98
alfa	33	56	jes	dox	3/4	6 1/2	28/02/98
alfa	34	100	jes	dox	3/4	7	28/02/98
alfa	35	140	jes	dox	3/4	7 1/2	28/02/98
alfa	36	24	jes	dox	3/4	8	28/02/98
alfa	37	32	jes	dox	3/4	8 1/2	28/02/98
alfa	38	8	jes	dox	3/4	9	28/02/98
alfa	39	36	jes	dox	3/4	9 1/2	28/02/98
alfa	40	8	jes	dox	3/4	10	28/02/98
alfa	41	32	jes	dox	3/4	10 1/2	28/02/98
alfa	42	8	jes	dox	3/4	12	28/02/98
alfa	43	112	jes	dox	7/8	4 1/2	28/02/98
alfa	44	1672	jes	dox	7/8	5	28/02/98
alfa	45	964	jes	dox	7/8	5 1/2	28/02/98

LOTE	PARTIDA	CANTIDAD	DESCRIPCIÓN	RECUBRIMIENTO	DIÁMETRO	LARGO	FECHA
alfa	46	1732	jes	dox	7/8	6	28/02/98
alfa	47	1464	jes	dox	7/8	6 1/2	28/02/98
alfa	48	72	jes	dox	7/8	7	28/02/98
alfa	49	120	jes	dox	7/8	7 1/2	28/02/98
alfa	50	48	jes	dox	7/8	8	28/02/98
alfa	51	96	jes	dox	7/8	8 1/2	28/02/98
alfa	52	84	jes	dox	7/8	9	28/02/98
alfa	53	16	jes	dox	7/8	10	28/02/98
alfa	54	12	jes	dox	7/8	11	28/02/98
alfa	55	16	jes	dox	7/8	12	28/02/98
alfa	56	12	jes	dox	7/8	12 1/2	28/02/98
alfa	57	24	jes	dox	7/8	13	28/02/98
alfa	58	48	jes	dox	7/8	20	28/02/98
alfa	59	64	jes	dox	1	5	28/02/98
alfa	60	100	jes	dox	1	5 1/2	28/02/98
alfa	61	480	jes	dox	1	6	28/02/98
alfa	62	28	jes	dox	1	6 1/2	28/02/98
alfa	63	3520	jes	dox	1	7	28/02/98
alfa	64	304	jes	dox	1	7 1/2	28/02/98
alfa	65	16	jes	dox	1	9 1/2	28/02/98
alfa	66	64	jes	dox	1	10	28/02/98
alfa	67	36	jes	dox	1	12 1/2	28/02/98
alfa	68	12	jes	dox	1	13	28/02/98
alfa	69	20	jes	dox	1 1/8	5	28/02/98
alfa	70	48	jes	dox	1 1/8	5 1/2	28/02/98
alfa	71	144	jes	dox	1 1/8	6	28/02/98
alfa	72	432	jes	dox	1 1/8	6 1/2	28/02/98
alfa	73	384	jes	dox	1 1/8	7	28/02/98
alfa	74	148	jes	dox	1 1/8	7 1/2	28/02/98
alfa	75	468	jes	dox	1 1/8	8	28/02/98
alfa	76	48	jes	dox	1 1/8	8 1/2	28/02/98
alfa	77	28	jes	dox	1 1/8	9 1/2	28/02/98
alfa	78	12	jes	dox	1 1/8	11	28/02/98
alfa	79	16	jes	dox	1 1/8	14	28/02/98
alfa	80	32	jes	dox	1 1/8	21	28/02/98
alfa	81	180	jes	dox	1 1/4	7	28/02/98
alfa	82	1520	jes	dox	1 1/4	7 1/2	28/02/98
alfa	83	1020	jes	dox	1 1/4	8	28/02/98
alfa	84	368	jes	dox	1 1/4	8 1/2	28/02/98
alfa	85	1736	jes	dox	1 1/4	9	28/02/98
alfa	86	260	jes	dox	1 1/4	9 1/2	28/02/98
alfa	87	80	jes	dox	1 1/4	10	28/02/98
alfa	88	80	jes	dox	1 1/4	11	28/02/98
alfa	89	120	jes	dox	1 1/4	12 1/2	28/02/98
alfa	90	60	jes	dox	1 1/4	15	28/02/98
alfa	91	64	jes	dox	1 1/4	18	28/02/98
alfa	92	20	jes	dox	1 1/4	18 1/2	28/02/98
alfa	93	20	jes	dox	1 1/4	19 1/2	28/02/98
alfa	94	20	jes	dox	1 1/4	21	28/02/98
alfa	95	84	jes	dox	1 3/8	9 1/2	28/02/98
alfa	96	80	jes	dox	1 3/8	10	28/02/98
alfa	97	160	jes	dox	1 3/8	11	28/02/98
alfa	98	160	jes	dox	1 1/2	9	28/02/98

LOTE	PARTIDA	CANTIDAD	DESCRIPCIÓN	RECUBRIMIENTO	DIÁMETRO	LARGO	FECHA
alfa	99	480	jes	dox	1 1/2	9 1/2	28/02/98
alfa	100	712	jes	dox	1 1/2	10	28/02/98
alfa	101	172	jes	dox	1 1/2	10 1/2	28/02/98
alfa	102	96	jes	dox	1 1/2	11	28/02/98
alfa	103	20	jes	dox	1 1/2	11 1/2	28/02/98
alfa	104	152	jes	dox	1 1/2	12	28/02/98
alfa	105	36	jes	dox	1 1/2	12 1/2	28/02/98
alfa	106	20	jes	dox	1 1/2	18 1/2	28/02/98
alfa	107	60	jes	dox	1 1/2	23	28/02/98
alfa	108	20	jes	dox	1 5/8	9 1/2	28/02/98
alfa	109	20	jes	dox	1 5/8	11	28/02/98
alfa	110	428	jes	dox	1 5/8	11 1/2	28/02/98
alfa	111	480	jes	dox	1 5/8	12	28/02/98
alfa	112	248	jes	dox	1 5/8	12 1/2	28/02/98
alfa	113	40	jes	dox	1 5/8	13	28/02/98
alfa	114	120	jes	dox	1 5/8	13 1/2	28/02/98
alfa	115	24	jes	dox	1 5/8	14 1/2	28/02/98
alfa	116	16	jes	dox	1 5/8	18 1/2	28/02/98
alfa	117	20	jes	dox	1 5/8	20 1/2	28/02/98
alfa	118	48	jes	dox	1 5/8	26 1/2	28/02/98
alfa	119	28	jes	dox	1 3/4	12	28/02/98
alfa	120	32	jes	dox	1 3/4	13	28/02/98
alfa	121	16	jes	dox	1 3/4	18 1/2	28/02/98
alfa	122	16	jes	dox	1 3/4	21	28/02/98
alfa	123	32	jes	dox	1 3/4	22	28/02/98
alfa	124	504	jes	dox	1 7/8	13 1/2	28/02/98
alfa	125	168	jes	dox	1 7/8	14	28/02/98
alfa	126	48	jes	dox	1 7/8	14 1/2	28/02/98
alfa	127	72	jes	dox	1 7/8	15	28/02/98
alfa	128	24	jes	dox	1 7/8	15 1/2	28/02/98
alfa	129	24	jes	dox	1 7/8	16	28/02/98
alfa	130	24	jes	dox	1 7/8	17	28/02/98
alfa	131	16	jes	dox	1 7/8	21	28/02/98
alfa	132	24	jes	dox	1 7/8	30 1/2	28/02/98
alfa	133	24	jes	dox	1 7/8	31	28/02/98
alfa	134	32	jes	dox	2	13 1/2	28/02/98
alfa	135	20	jes	dox	2	19 1/2	28/02/98
alfa	136	40	jes	dox	2	20	28/02/98
alfa	137	80	jes	dox	2	20 1/2	28/02/98
alfa	138	60	jes	dox	2	21	28/02/98
alfa	139	20	jes	dox	2	21 1/2	28/02/98
alfa	140	112	jes	dox	2 1/2	16	28/02/98
alfa	141	672	jes	dox	2 1/2	16 1/2	28/02/98
alfa	142	28	jes	dox	2 1/2	17	28/02/98
alfa	143	28	jes	dox	2 1/2	18	28/02/98
alfa	144	56	jes	dox	2 1/2	18 1/2	28/02/98
alfa	145	56	jes	dox	2 1/2	19	28/02/98
alfa	146	16	jes	dox	3 3/4	32	28/02/98

Tabla 3.4 Requisición de material

Donde j_{es}^{24} representa a un espárrago con dos tuercas, lo que nos da un total de 66054 espárragos y 132108 tuercas.

Control y modificación del proyecto

Es importante aclarar que cada programa de producción es único e irrepetible por lo que bien puede recibir el nombre de programación de proyecto. Para planificar se debe tener acceso a toda la información del proyecto, esto implica tener acceso al inventario de cada etapa. Al ejecutar el proyecto se debe de llevar un control sobre el avance de la producción y el inventario de cada departamento para saber si no ha fallado alguno de los pasos programados, lo que con frecuencia hace necesaria la reelaboración del proyecto.

3.3.2 Cálculo de variables de operación

Reglas para calcular variables de operación:

1. La caída de voltaje total deberá ser menor a 5 V.
2. Existe una densidad de corriente $\left(\frac{A}{cm^2}\right)$ máxima para cada arreglo. Si se excede esta densidad se provocan defectos superficiales de las piezas procesadas. Si no se conoce el valor, se utiliza $j = 0.022 \frac{A}{cm^2}$ como primera aproximación.
3. Si el valor conocido de la densidad de corriente no corresponde al valor máximo, y no se viola la regla 1, se puede incrementar su valor según la relación:

$$j = j_o + 0.002 \left(\frac{A}{cm^2}\right)$$

4. Si se ha producido un defecto superficial se debe reducir la densidad de corriente según las siguientes reglas:

Regla	Ecuación	Acciones adicionales
Si la caída de presión del sistema de circulación es $\Delta P \geq 100 \frac{lb_f}{in^2}$	$j = j_o - 0.004 \left(\frac{A}{cm^2}\right)$	Cambiar los filtros del sistema de circulación
Si j_{max} no existe.	$j = j_o - 0.001 \left(\frac{A}{cm^2}\right)$	Si no se presentan defectos superficiales utilizando j entonces $j = j_{max}$
Si j_{max} existe.	$j = j \left(\frac{A}{cm^2}\right)$	Revisar la preparación de superficie.

Tabla 3.5 Reglas para defectos superficiales.

²⁴ j_{es} = juego de espárragos.

5. Existe una resistencia para cada arreglo, pero ésta puede variar dependiendo de la vida del dispositivo. Si no se conoce la resistencia del baño, se puede utilizar la siguiente gráfica.

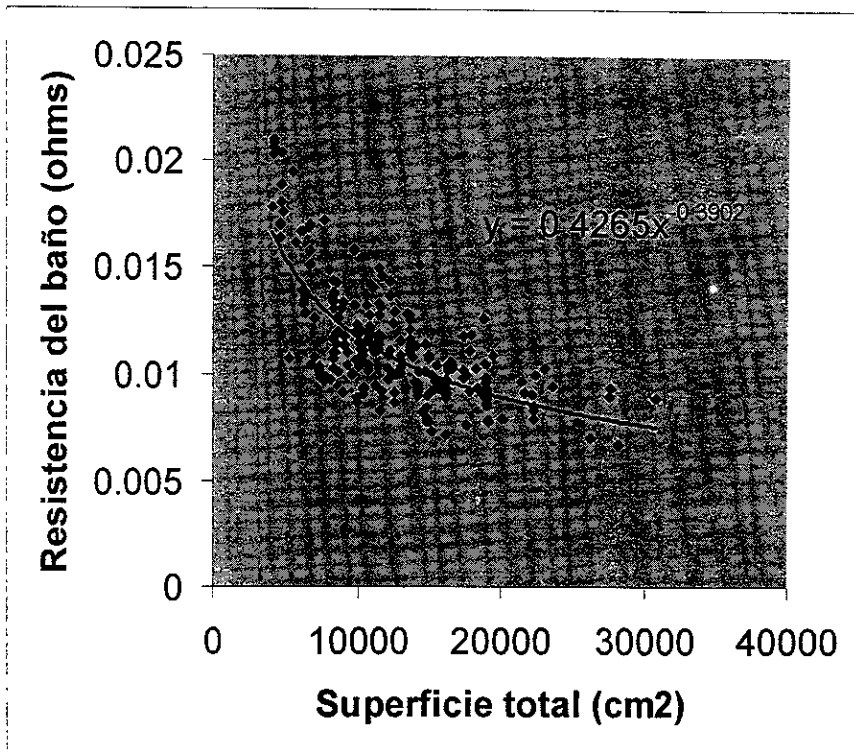


Figura 3.2 Resistencia de celda contra superficie total del cátodo.

Como podemos ver en la gráfica anterior, existe una relación entre la superficie a recubrir y la resistencia del baño. Desafortunadamente, la geometría de las piezas dificulta la construcción de un modelo fenomenológico²⁵ que permita estimar la resistencia del baño y la densidad de corriente óptima.

3.3.3 Control de proceso

La variable a controlar para la electrólisis es el flujo de corriente. Sin embargo, también se debe llevar un seguimiento de la resistencia del baño ya que durante el proceso se pueden presentar falsos contactos. Esto provoca

²⁵Además, no se conoce completamente la termodinámica y cinética del proceso.

variaciones en el voltaje que pueden ser perceptibles o imperceptibles para el operador. No obstante, las variaciones en la resistencia tienen causas diferentes y, en consecuencia, soluciones diferentes.

Desviación	Causa	Acción correctiva
5% al 10%	Falso contacto en barra catódica.	Revisar y reparar contactos eléctricos en la barra catódica.
	Si no existe falso contacto en la barra catódica, probablemente el cátodo necesita mantenimiento o el valor de control de la resistencia es incorrecto.	1. Sustituir el valor de control de resistencia por el valor promedio registrado por el adquisidor de datos. 2. Revisar y reparar el cátodo al terminar el proceso.
	Si no existe falso contacto en la barra catódica, el cátodo está en buen estado y el problema se ha repetido durante cuatro rondas consecutivas entonces el ánodo necesita mantenimiento.	Programar un mantenimiento general a la celda.
10% al 20%	Falso contacto en barra catódica.	Revisar y reparar contactos eléctricos en la barra catódica.
	Si no existe falso contacto en la barra catódica, probablemente el cátodo (dispositivo) necesita mantenimiento.	Parar el proceso para revisar y reparar el cátodo.
	Si no existe falso contacto en la barra catódica y el cátodo está en buen estado, entonces probablemente el valor de control de la resistencia es incorrecto.	Sustituir el valor de control de resistencia por el valor promedio registrado por el adquisidor de datos.
	Si no existe falso contacto en la barra catódica, el cátodo está en buen estado y el problema se ha repetido durante cuatro rondas consecutivas entonces el ánodo necesita mantenimiento.	Dar mantenimiento general a la celda.
Mayor al 20%	Falso contacto	Parar proceso y reparar el cátodo.

Tabla 3.6 Causas y acciones correctivas para variaciones de resistencia.

$$\text{desviación} = \frac{R_0 - R}{R_0} \times 100,$$

donde:

R_0 es el valor de control para la resistencia de la celda.

R es el valor real de la resistencia de la celda.

3.4 Construcción del modelo

Como ha quedado claro en la descripción del problema, hay muchas experiencias que pueden ser sistematizadas e incluidas en un sistema experto. En vista de que no se tiene un modelo termodinámico o cinético, el sistema experto


constará exclusivamente de la experiencia previa que se tiene sobre la operación del proceso.

3.4.1 Base de conocimientos

Los técnicos de proceso deben llevar un registro diario para el control del material en las diferentes etapas; esta información debe ser compilada y analizada por el coordinador de producción y el ingeniero de proceso. Si revisamos los formatos de las hojas de control de cada departamento podemos construir expresiones proposicionales para formar una base de conocimientos según la siguiente estructura:

hoja de control (variables de control)

Por ejemplo, en la etapa de recubrimiento se utiliza la siguiente hoja de control:



DTA DE MÉXICO S.A. DE C.V.

PROCESO DOX

2 / 7 / 97
Día Mes Año

FORMATO
PO 20 20 05

RONDA	TIPO	MICRAS	CANTIDAD	PIEZAS	MEDIDAS	GANCHO	AREA TOTAL	AMPERES	DENSIDAD	SALE
1	dox	6	80	esp	1¼x8	80	18633	447	0.024	7:35

Figura 3.2 Hoja de control de proceso DOX

para hacer la siguiente afirmación:

“La ronda número 1 que contenía 80 espárragos con recubrimiento DOX, con diámetro de 1¼ pulgadas y largo de 8 pulgadas, y donde se utilizaron 80 ganchos, cuya superficie total es de 18633 centímetros cuadrados, que fue recubierta con una capa de aleación DOX, cuyo espesor fue 6 micras, utilizando una intensidad de corriente de 447 amperes y densidad de corriente de 0.024 $\frac{A}{cm^2}$ salió de proceso el día 2 del mes 7 del año 1997 a las 7 horas con 35 minutos.”

Esta afirmación tiene los siguientes términos:

Variable	Constante
Número de ronda	1
Estado de material	DOX
Espesor	6
Cantidad de piezas	80
Tipo de piezas	esp
Diámetro	1¼
Largo	8
Número de ganchos	4
Superficie total	18633
Intensidad de corriente	447
Densidad de corriente	0.024
Hora de salida	7
Minuto de salida	35
Día de salida	2
Mes de salida	7
Año de salida	97

Tabla 3.7 Términos de la hoja de control de proceso.

Si definimos a **proceso** como el símbolo de predicado que relaciona a los términos {Numero_de_ronda, Estado_de_material, Espesor, Cantidad_de_piezas, Tipo_de_material, Diámetro, Largo, Código_del_gancho, Superficie_total, Intensidad_de_corriente, Densidad_de_corriente, Hora_de_salida, Minuto_de_salida, Día_de_salida, Mes_de_salida, Año_de_salida}, la expresión proposicional puede presentar la siguiente estructura:

proceso(Numero_de_ronda, Estado_de_material, Espesor, Cantidad_de_piezas, Tipo_de_material, Diámetro, Largo, Código_del_gancho, Superficie_total, Intensidad_de_corriente, Densidad_de_corriente, Hora_de_salida, Minuto_de_salida, Día_de_salida, Mes_de_salida, Año_de_salida).

El enunciado con los datos que se muestran en la Figura 3.2 es:

proceso(1, dox, 6, 80, esp, 1.25, 8, 4, 18633, 447, 0.024, 7, 35, 2, 7, 97).

Sin embargo, puede resultar complicado, por ejemplo, recordar que significa el número 35. Para solucionar este problema es conveniente utilizar descripciones o funciones que nos relacionen a varios de los términos; por ejemplo, podemos agrupar a *Día_de_salida*, *Mes_de_salida* y *Año_de_salida* en una sola función cuyo símbolo sea *fecha*. Usando esta estrategia, podemos definir la siguiente estructura:

proceso(ronda, cantidad, material, medidas, datos de proceso, fecha de salida, hora de salida)

donde: *ronda*, *cantidad*, *material*, *medidas*, *datos de proceso*, *fecha de salida* y *hora de salida* son términos.

<i>ronda</i>	<i>ronda (Numero de)</i>
<i>cantidad</i>	<i>cantidad (De piezas)</i>
<i>material</i>	<i>material (Tipo de, Estado)</i>
<i>medidas</i>	<i>medidas (Diámetro, Largo)</i>
<i>datos de proceso</i>	<i>datos (Espesor, Resistencia, Densidad)</i>
<i>fecha de salida</i>	<i>fecha (Año, Mes, Día)</i>
<i>Hora de salida</i>	<i>hora (Hora, Minuto)</i>

Tabla 3.8 Términos para la relación proceso.

Si comparamos la hoja de control de proceso con la relación **proceso**, vemos que faltan algunos datos (gancho, área total y amperes). Estos datos se omiten porque pueden ser deducidos o calculados utilizando reglas²⁶ o fórmulas²⁷ y los demás datos de la hoja de control de proceso.

Esta expresión proposicional pueden ser utilizada para construir reglas que analicen esta información y que nos permita generar nuevos conocimientos. Otras expresiones y términos que se desarrollaron utilizando hojas de control fueron:

<i>materia prima</i>	<i>materia prima (lote, cantidad, material, ubicación, operación, fecha)</i>
<i>corte de espárragos</i>	<i>corte de espárragos (lote, cantidad, material, medidas, operación, fecha)</i>
<i>preparación de superficie</i>	<i>preparación de superficie (lote, cantidad, material, medidas, operación, fecha)</i>
<i>solicitud de corte</i>	<i>solicitud de corte (lote, cantidad, material, medidas, fecha)</i>
<i>solicitud de producción</i>	<i>solicitud de producción (lote, cantidad, material, medidas, fecha)</i>
<i>solicitud de material</i>	<i>solicitud de material (lote, cantidad, Material, medidas, fecha)</i>
<i>control de calidad</i>	<i>control de calidad (ronda, cantidad, material, medidas, observaciones, fecha)</i>
<i>almacen</i>	<i>almacen (lote, ubicación, cantidad, material, medidas, operación, fecha)</i>
<i>requisición</i>	<i>requisición (lote, cantidad, material, medidas, fecha)</i>
<i>embarque</i>	<i>embarque (lote, ubicación, cantidad, material, medidas, fecha)</i>
<i>lote</i>	<i>lote (Nombre del, Número de partida)</i>
<i>ubicación</i>	<i>ubicación (Contenedor, Número de contenedor, Coordenada)</i>
<i>operación</i>	<i>operación (entrada) operación (salida) operación (producción)</i>

Tabla 3.9 Expresiones y términos para hojas de control.

²⁶ El número de ganchos por espárrago es dos.

²⁷ Para el cálculo de la corriente (amperes) y el área total véase el Apéndice B.

Utilizando los datos proporcionados por las hojas de control se desarrollan balances de material para llevar el inventario que ser archivado en un nueva expresión.

existencia(Etapa,Lote,Cantidad,Material,Medidas)

3.4.2 Construcción de nuevos conocimientos

Requisición de material

Si revisamos la Tabla 3.4 vemos que los términos {Lote, Partida, Cantidad, Cantidad, Recubrimiento, Diametro, Largo, Dia_de_entrega, Mes_de_entrega, Año_de_entrega} se utilizan para describir la relación requisición, la palabra requisición puede ser utilizada para construir el símbolo *requisicion*²⁸. Si agrupamos los términos en funciones o descripciones, que también son términos, podemos construir una estructura proposicional de la forma:

requisicion(lote(N,P),cantidad(C),material(jes,dox),medidas(D,L),fecha(A,M,D))

Donde el juego de espárragos (jes) está formado por un espárrago de diámetro D y largo L, y dos tuercas de diámetro D. Para planear la producción se debe desglosar el conjunto de piezas, ya que no se pueden procesar juntos los espárragos y las tuercas. Esto se puede representar utilizando las siguientes reglas:

- "Existe una requisición con código L y cantidad C de espárragos con recubrimiento dox con medidas M y fecha de entrega F ²⁹ existe una requisición con código L y cantidad C de juegos de espárragos con recubrimiento dox con medidas M y una fecha de entrega F."

Podemos deducir que la conclusión de la regla tiene la estructura:

requisicion(L,cantidad(C),material(esparrago,dox),M,F),

y el antecedente o hipótesis tiene la estructura:

requisicion(L,cantidad(C),material(jes,dox),M,F).

La estructura proposicional de la regla es:

requisicion(L,cantidad(C),material(esparrago,dox),M,F) ←

requisicion(L,cantidad(C),material(jes,dox),M,F).

²⁸Se utiliza la palabra sin acento porque el lenguaje de programación no permite la utilización de estos caracteres, véase Apéndice A.

²⁹Recuerde que la traducción normal del condicional inverso ← es "...sí...", véase sección 2.2.3.

En el lenguaje de programación Prolog, véase Apéndice A, el símbolo \leftarrow se representa con el operador ":-" de manera que, usando la sintaxis del Prolog, la estructura anterior se escribe como:

```
requisicion(L,cantidad(C),material(esparrago,dox),M,F):-
    requisicion(L,cantidad(C),material(jes,dox),M,F).
```

- "Existe una requisición con código L y cantidad C de tuercas con recubrimiento dox con diámetro D y fecha de entrega F si existe una requisición con código L y cantidad C_jes de juegos de espárragos con recubrimiento dox con diámetro D y una fecha de entrega F, y C es igual a dos veces C_jes."

Podemos deducir que la conclusión de la regla tiene la estructura:

```
requisicion(L,cantidad(C),material(tuerca,dox),medidas(D,_),F),
```

y el antecedente o hipótesis tiene dos estructuras atómicas:

1. `requisicion(L,cantidad(C),material(jes,dox),medidas(D,_),F)`.
2. `C=2*C_jes`.

La estructura de la regla es:

```
requisicion(L,cantidad(C),material(tuerca,dox),medidas(D,_),F) ←
    requisicion(L,cantidad(C_jes),material(jes,dox),medidas(D,_),F) ∧ (C=2*C).
```

Su traducción a Prolog sería:

```
requisicion(L,cantidad(C),material(tuerca,dox),medidas(D,_),F):-
    requisicion(L,cantidad(C_jes),material(jes,dox),medidas(D,_),F),
    C is C_jes*2.
```

Donde "_" se conoce como la variable anónima. Se utiliza para casos donde no nos interesa conocer el valor del término, vea que en el enunciado jamás se menciona o se requiere la longitud del material.

Normalmente existe mas de una partida (P) de material con el mismo diámetro. Para calcular la cantidad total o global de tuercas por medida que se necesitan para el lote N se deben buscar todas las partidas (P) con el mismo diámetro y hacer la sumatoria.

```
requisicion_de_tuercas(N):-
    requisicion(lote(N,P),C,material(tuerca,E),M,F),
    sumatoria(lote(N,P),C,material(tuerca,E),M,F),
    fail.
```

```
sumatoria(lote(N,_),cantidad(C),material(T,E),M,F):-
    retract(requisicion(lote(N,total),cantidad(C_vieja),
        material(T,E),M,F)),
    C_nueva is C_vieja+C,
    asserta(requisicion(lote(N,total),cantidad(C_nueva),
        material(T,E),M,F)),!.
```

```
sumatoria(lote(N,_),cantidad(C),material(T,E),M,F):-
    C_nueva is C,
    asserta(requisicion(lote(N,total),cantidad(C_nueva),
        Material(T,E),M,F)),!.
```

Modificación de inventario

Al revisar la existencia de material, se asigna el material de reserva como material perteneciente al lote.

```
modificar_existencia(N):-
    requisicion(lote(N,P),cantidad(C_req),material(esparrago,dox),Dim,_),
    retract(existencia(almacen,lote(reserva,total),cantidad(C_inv),
        material(esparrago,dox),Dim)),
    modificacion(C_req,C_inv,lote(N,P),material(esparrago,dox),
        Dim,almacen),fail.
```

```
modificar_existencia(N):-
    requisicion(lote(N,total),cantidad(C_req),material(tuerca,dox),Dim,_),
    retract(existencia(almacen,lote(reserva,total),cantidad(C_inv),
        material(tuerca,dox),Dim)),
    modificacion(C_req,C_inv,lote(N,total),material(tuerca,dox),Dim,almacen),
    fail.
```

```
modificacion(C_req,C_inv,lote(N,P),M,Dim,Etapa):-
    C_req>=C_inv,!,
    asserta(existencia(Etapa,lote(N,P),cantidad(C_inv),M,Dim)).
```

```
modificacion(C_req,C_inv,lote(N,P),M,Dim,Etapa):-
    asserta(existencia(Etapa,lote(N,P),cantidad(C_req),M,Dim)),
    C is C_inv-C_req,
    asserta(existencia(Etapa,lote(reserva,total),cantidad(C),M,Dim)).
```


La primer regla del procedimiento `modificar_existencia` busca un espárrago de la requisición P del lote N y, si la encuentra, busca en el `almacen` un material de reserva, `lote(reserva,total)`, con las mismas características, M y Dim; si existe este material, lo borra de la base de conocimientos y agrega un nuevo conocimiento utilizando la regla modificación. La instrucción `fail` hace fracasar el objetivo y el interprete lógico busca una nueva requisición P' y trata de cumplir el objetivo; se repite esta operación hasta que revisa todos los hechos relacionados con requisición. Sin embargo, este objetivo siempre fracasa y, aunque encuentra todas las soluciones, busca un nuevo resolvente y encuentra la segunda regla y trata de hacer verdadero el objetivo repitiendo la misma operación solo que ahora para tuercas.

Orden de producción

El material a procesar se calcula restando a la requisición la existencia en almacén. Esta operación se puede representar utilizando la siguiente regla:

```
orden_de_produccion(L,cantidad(C),material(T,dox),Dim,F):-
    requisicion(L,cantidad(C_requisicion),material(T,dox),Dim,F),
    revisar_existencia(almacen,L,cantidad(C_almacen),material(T,dox),Dim),
    C_requisicion>C_almacen,
    C is C_requisicion-C_almacen.
```

```
revisar_existencia(Etapa,L,C,M,Dim):-
    existencia(Etapa,lote(N,_),C,M,Dim),!
```

```
revisar_existencia(_,_ ,cantidad(0),_ ,_).
```

Se debe utilizar la regla `revisar_existencia` porque si no existe material en el almacén, el resolvente:

```
existencia(Etapa,lote(N,_),cantidad(C),material(T,E),Medidas).
```

fracasa. Al ocurrir esto, el intérprete lógico busca un nuevo resolvente y encuentra:

```
revisar_existencia(_ ,_ ,cantidad(0),_ ,_).
```

asignando el valor de 0 a la variable `C_almacen`.

Requisición de barras

Se utilizan barras de 144 pulgadas de largo para cortar los espárragos solicitados a la medida. Primero se calcula la cantidad de espárragos que se pueden cortar por barra, dividiendo la longitud de la barra entre la longitud del espárrago más un octavo de pulgada (material que se pierde en el corte). Después se divide la cantidad de espárragos a producir por partida entre la cantidad de espárragos por barra. Finalmente se suman todas las barras del mismo diámetro para obtener el consumo de barras.

requisicion_de_barras(N):-

```
orden_de_produccion(lote(N,P),cantidad(C),
material(esparrago,dox),medidas(D,L),F),
Espárragos_por_barra is integer(144/(L+0.125)),
Barras_por_partida is C/Espárragos_por_barra+1,
sumatoria(lote(N,P),cantidad(Barras_por_partida),
material(barras,virgen),medidas(D,144),F),
fail.
```

Solicitud de material

Para estimar la cantidad de material que se debe comprar se hacen los siguientes balances:

- Barras: $\text{Barras_faltantes} = \text{Barras_requisición} - \text{Barras_existencia}$
- Tuercas: $\text{Tuercas_faltantes} = \text{Tuercas_a_producir} - \text{Tuercas_existencia}$

solicitud_de_material(N,tuerca,Dim,cantidad(C_solicitud),F_solicitud):-

```
date(F_solicitud),
orden_de_produccion(lote(N,total),cantidad(C_produccion),
material(tuerca,dox),Dim,_),
revisar_existencia(materia_prima,lote(N,total),
cantidad(C_existencia),material(tuerca,virgen),Dim),
C_produccion > C_existencia,
C_solicitud is C_produccion - C_existencia.
```

solicitud_de_material(N,barra,Dim,cantidad(C_solicitud),F_solicitud):-

```
date(F_solicitud),
requisicion(lote(N,total),cantidad(C_requisicion),
material(barra,virgen),Dim,_),
```

```

revisar_existencia(materia_prima,lote(N,total),
cantidad(C_existencia),material(barra,virgen),Dim),
C_requisicion>C_existencia,
C_solicitud is C_requisicion-C_existencia.

```

Piezas por rondas

Regla para calcular el número máximo de piezas por ronda: Las piezas máximas por ronda de una pieza x_i con diámetro ϕ_i y longitud L_i es C_i si $\phi_{menor} \leq \phi_i \leq \phi_{mayor}$ y $L_{menor} \leq L_i \leq L_{mayor}$.

Su representación, utilizando programación lógica y la Tabla 3.2, es:

```

/*piezas_máximas_por_ronda(Tipo,medidas(D,L),Cantidad)*/
piezas_máximas_por_ronda(espárrago,medidas(D,L),4):-L<55,L>35.
piezas_máximas_por_ronda(espárrago,medidas(D,L),6):-L<35,L>25.
piezas_máximas_por_ronda(espárrago,medidas(D,L),8):-L<25,L>19.5.
piezas_máximas_por_ronda(espárrago,medidas(D,L),16):-D=2.5,L<19.5.
piezas_máximas_por_ronda(espárrago,medidas(D,L),16):-D=2.25,L<19.5,L>14.
piezas_máximas_por_ronda(espárrago,medidas(D,L),16):-D=2,L<19.5,L>16.
piezas_máximas_por_ronda(espárrago,medidas(D,L),24):-D=2,L<16.
piezas_máximas_por_ronda(espárrago,medidas(D,L),24):-D<2,D>1.5,L<19.5.
piezas_máximas_por_ronda(espárrago,medidas(D,L),40):-D<1.625,L<19.5,L>10.
piezas_máximas_por_ronda(espárrago,medidas(D,L),80):-D<1.625,L<10,L>6.75.
piezas_máximas_por_ronda(espárrago,medidas(D,L),120):-D<1.375,L<6.75,L>4.75.
piezas_máximas_por_ronda(espárrago,medidas(D,L),160):-D<1.375,L<4.75,L>3.75.
piezas_máximas_por_ronda(espárrago,medidas(D,L),240):-D<0.875,L<3.75.
piezas_máximas_por_ronda(tuerca,medidas(D,_),64):-D<2.5,D>1.75.
piezas_máximas_por_ronda(tuerca,medidas(D,_),144):-D<1.75,D>1.375.
piezas_máximas_por_ronda(tuerca,medidas(D,_),208):-D<1.375,D>1.
piezas_máximas_por_ronda(tuerca,medidas(D,_),400):-D<1,D>0.5.

```

Si queremos preguntar el número de piezas máximas por ronda para una tuerca con diámetro de 1.5 le diremos al intérprete lógico:

```
piezas_máximas_por_ronda(tuerca,medidas(1.5,_),N)?
```

El intérprete tratará de unificar el objetivo genera la siguiente traza:

Resolvente: $\text{piezas_m\acute{a}ximas_por_ronda}(\text{tuerca}, \text{medidas}(1.5, _), N)?$
 Al buscar: $\text{piezas_m\acute{a}ximas_por_ronda}(\text{tuerca}, \text{medidas}(1.5, _), N)?$
 Encontramos: $\text{piezas_m\acute{a}ximas_por_ronda}(\text{tuerca}, \text{medidas}(1.5, _), 64) :-$
 $1.5 \leq 2.5, 1.5 \geq 1.875.$
 El nuevo resolvente es: $1.5 \leq 2.5, 1.5 \geq 1.75.$
 Al evaluar: $1.5 \leq 2.5$ el objetivo se cumple.
 Al evaluar: $1.5 \geq 1.75$ el objetivo fracasa y es necesario buscar un nuevo resolvente.
 El nuevo resolvente es $\text{piezas_m\acute{a}ximas_por_ronda}(\text{tuerca}, \text{medidas}(1.5, _), N)?.$
 Al buscar: $\text{piezas_m\acute{a}ximas_por_ronda}(\text{tuerca}, \text{medidas}(1.5, _), N)?$
 Encontramos: $\text{piezas_m\acute{a}ximas_por_ronda}(\text{tuerca}, \text{medidas}(1.5, _), 144) :-$
 $1.5 < 1.75, 1.5 \geq 1.375.$
 El nuevo resolvente es: $1.5 < 1.75, 1.5 \geq 1.375.$
 Al evaluar: $1.5 < 1.75$ el objetivo se cumple.
 Al evaluar: $1.5 \geq 1.375$ el objetivo se cumple. El cual es verdadero.
 El intérprete lógico asigna el valor de 144 a N.

Cálculo de rondas

Utilizando el conjunto de reglas para calcular el número máximo de piezas por ronda podemos definir:

Regla para calcular una ronda con un solo tipo de material: Existe una ronda de C_i piezas de material X_i si existen N_i piezas del material X_i de manera tal que N_i sea mayor o igual que C_i . Donde C_i es el número de piezas máximas por ronda.

descripción_de_ronda([cantidad(C)], material(T,E), [medidas(D,L)]) :-
orden_de_producción(_, cantidad(N), material(T,E), medidas(D,L), _),
piezas_máximas_por_ronda(T, medidas(D,L), C), $N \geq C$.

Regla para calcular una ronda con varios tipos de material: Existe una ronda de materiales $\{x_1, x_2, \dots, x_n\}$ si la cantidad de piezas máximas por ronda C_{\max} es igual para todos y $N_i < C_{\max}$, donde N_i es el número de piezas de material x_i ; además, $\sum_{i=1}^{n-1} N_i < C_{\max}$ y la cantidad de piezas en la ronda es $C_i = N_i$ para toda $i < N$

$$C_n = C_{\max} - \sum_{i=1}^{n-1} N_i.$$

```

descripcion_de_ronda(X,material(T,E),Z):-
  orden_de_produccion(_,cantidad(C),material(T,E),medidas(D,L),_),
  piezas_maximas_por_ronda(T,medidas(D,L),C_max),C<C_max,
  ronda([cantidad(C)],material(T,E),[medidas(D,L)],C_max,C,X,Z).

ronda(Cs,material(T,E),Ds,C_max,Sum1,X,Z):-
  C_max>Sum1,
  orden_de_produccion(_,cantidad(C),material(T,E),medidas(D,L),_),
  piezas_maximas_por_ronda(T,medidas(D,L),C_max),C<C_max,
  not(member(medidas(D,L),Ds)),acumulacion(Sum2,Sum1,C_max,C,N),
  ronda([cantidad(N)|Cs],material(T,E),[medidas(D,L)|Ds],C_max,Sum2,X,Z).

ronda(Cs,_,Ds,C_max,C_max,Cs,Ds).

acumulacion(C_acum2,C_acum1,C_max,C,N):-
  C_acum2 is C_acum1+C,C_acum2=<C_max,!,N is C.

acumulacion(C_max,C_acum1,C_max,_,N2):-!,N2 is C_max-C_acum1.

```

Además, se pueden hacer otras combinaciones para aprovechar el espacio disponible y meter una mayor cantidad de piezas por ronda, por ejemplo:

Regla para optimización de espacio: Se puede construir una ronda de 40 piezas de E_1 y 40 piezas de E_2 si existen 40 o más piezas de un espárrago E_1 con $\phi_1 < 1\frac{5}{8}$ y tiene longitud tal que $10 \leq L_1 \leq 13$ y existen 40 o más piezas de otro espárrago E_2 tal que $\phi_2 < 1\frac{5}{8}$ y $6\frac{3}{4} \leq L_2 < 10$ de manera tal que $L_1 + L_2 < 19\frac{1}{2}$.

```

descripcion_de_ronda([cantidad(40),cantidad(40)],material(esparrago,dox),
  [medidas(D1,L1),medidas(D2,L2)]):-
  orden_de_produccion(_,cantidad(C1),material(esparrago,dox),
  medidas(D1,L1),_),
  piezas_maximas_por_ronda(esparrago,medidas(D1,L1),40),C1>=40,L1<=13.5,
  orden_de_produccion(_,cantidad(C2),material(esparrago,dox),
  medidas(D2,L2),_),
  piezas_maximas_por_ronda(esparrago,medidas(D2,L2),80),
  C2>=40,L2+L1 <=19.5.

```

Es importante resaltar que se utiliza 13 como longitud máxima para E_1 en vez de utilizar 19½. Aunque para nosotros parece obvio, ya que inconscientemente tendemos a despreciar cualquier espárrago con una longitud mayor a trece, para un programa resulta importante ya que le evitamos la necesidad de comparar un espárrago E_1 con $L_1 > 13$ y con n espárragos E_2 para finalmente descartarlos porque la condición $L_1 + L_2 < 19½$ no se cumple.

Al efectuar un cálculo de rondas, el coordinador de producción optimiza primero el espacio y después construye rondas con un solo tipo de material. De manera similar debemos indicarle al intérprete lógico que debe optimizar primero; esto se reduce a colocar primero la regla para optimización de espacio y después la regla para calcular rondas con un solo tipo de material (búsqueda descendente).

3.4.3 Búsqueda de soluciones

Cálculo de variables de operación

Utilizando las reglas para cálculo de variables de operación y las expresiones proposicionales de la base de conocimientos podemos definir el siguiente procedimiento:

```
variables_de_operacion(A,B,C,R,Den):-
    proceso(R,A,B,C,datos_de_proceso(_,R,Den1),F,_),
    control_de_calida(R,A,B,C,con_defectos,F),!,
    Den is Den1-0.001.
```

```
variables_de_operacion(A,B,C,R,Den):-
    proceso(R,A,B,C,datos_de_proceso(_,R,Den1),F,_),
    control_de_calida(R,A,B,C,sin_defectos,F),!,
    Den is Den1+0.002.
```

```
variables_de_operacion(A,material(T,_),[C|_],R,0.022):-
    !,superficie_total(A,material(T,_),C,0,Area),
    log(Area,Ln),
    Prod is -0.3902*Ln,
    exp(Prod,Exp),
    R is 0.4265*Exp.
```

```
superficie_total([],_,[],Sum,Sum).
```

```

superficie_total([cantidad(C)|Cs],material(esparrago,_),[medidas(D,L)|Ms],
Sum,Area):-
Sum1 is C*L*D*D/4*3.1416*2.54*2.54+Sum,
superficie_total(Cs,material(esparrago,_),Ms,Sum1,Area).

```

La estructura del procedimiento supone que la base de conocimiento está ordenada de manera descendente de la fecha más reciente a la fecha más antigua. Por ejemplo, suponga que usted pide la resistencia R y la densidad de corriente j de un material X ; al buscar la solución, encuentra una j_0 con defecto de superficie y utiliza la ecuación $j=j_0-0.001$ para decidir la nueva densidad de corriente. Sin embargo, al procesar el material, control de calidad informa que el material presentó defectos superficiales; al iniciar una nueva búsqueda, y si ya se actualizó la base de conocimientos, se iniciará la búsqueda y se encontrará j con defecto superficial definiendo una nueva densidad de corriente j_1 a partir de j . Aunque se puede desarrollar un algoritmo para determinar cual es la menor densidad con defecto superficial en toda la base de conocimientos, es preferible ahorrarse el algoritmo y la búsqueda ordenando adecuadamente la base de conocimientos.

3.4.4 Planeación de producción

Estimación del tiempo de producción

Para estimar el tiempo que se tardará en procesar el material, se utiliza la siguiente relación:

$$\text{Tiempo de producción} = N \text{ rondas} \times \frac{\text{turno}}{9 \text{ rondas}} \times \frac{\text{semanas}}{5 \text{ turno}}$$

Con este cálculo se puede estimar si se terminará a tiempo la requisición de material³⁰. Para estimar las rondas que deben ser procesadas, utilizaremos³¹ una simulación con los siguientes pasos:

1. Seleccionar una ronda.
2. Declarar la ronda como material procesado, agregarlo a la existencia de material en almacén.
3. Repetir la operación hasta procesar todo el material.

³⁰Suponiendo que no hay rechazo de material, no hay retrasos por materia prima, no hay paros no programados en la línea de producción, etc.

³¹Omito la explicación de las estrategias que utiliza el coordinador de producción para calcular el número de rondas, pero supongo que es evidente lo tedioso del procedimiento.

```
tiempo_de_produccion(N):-
    tiempo_de_produccion(N,ronda(0)).
```

```
tiempo_de_produccion(N,ronda(No)):-
    N1 is No+1,descripcion_de_ronda(A,B,C),
    assertz(ronda_programada(ronda(N1),A,B,C)),
    simular_existencia(N,A,B,C),tiempo_de_produccion(N,ronda(N1)).
```

```
tiempo_de_produccion(N,ronda(No)):-
    !,write('Lote:'),write(N),nl,write('No. de rondas:'),write(No),nl,
    T is integer(No/45)+1,
    write('Tiempo estimado:'),write(T),write(' semanas'),nl,
    tell(rondas),listing(ronda_programada/4),told.
```

```
simular_existencia(_,[],_,[]):-!.
```

```
simular_existencia(N,[cantidad(C1)|Cs],M,[Dim|Dims]):-
    revisar_existencia(almacen,lote(N,P),cantidad(C0),M,Dim),
    ((C0>0)->retract(existencia(almacen,lote(N,P),cantidad(C0),M,Dim));true)
    C2 is C1+C0,asserta(existencia(almacen,lote(N,P),cantidad(C2),M,Dim)),
    nl,simular_existencia(N,Cs,M,Dims).
```

Órdenes de producción

Para desarrollar las órdenes de producción para corte y preparación de superficie se siguen los siguientes pasos:

- Seleccionar un conjunto de partidas que puedan ser procesadas de manera tal que existan suficientes tuercas y barras para procesar la cantidad de piezas solicitada en la requisición.
- Calcular las tuercas totales a procesar y ordenar la preparación de superficie.

```
materia_a_procesar(N,PL,TcaL):-!,
    materia_a_procesar(N,4,[],[],PL,TcaL).
```

```
materia_a_procesar(N,D,Ps,Ts,PL,TcaL):-
    D1 is D-0.125,D1>=0.375,
    seleccionar(N,[],0,0,T_acum,D1,P),
    append(P,Ps,Ps1),T_acum>0,
    materia_a_procesar(N,D1,Ps1,[preparar(tuerca,D1,T_acum)|Ts],PL,TcaL).
```



```

material_a_procesar(N,D,Ps,Ts,PL,TcaL):-
    D1 is D-0.125,D1>=0.375,
    material_a_procesar(N,D1,Ps,Ts,PL,TcaL).

material_a_procesar(_,_ ,Ps,Ts,Ps,Ts):-!.

seleccionar(N,Ps,B_acum1,T_acum1,T,D,PL):-
    orden_de_produccion(lote(N,P),cantidad(C),material(esparrago,dox),
    medidas(D,L),_),
    nonmember(P,Ps),
    Esparragos_por_barra is integer(144 / (L+0.125)),
    Barras_por_partida is C//Esparragos_por_barra+1,
    B_acum2 is B_acum1+Barras_por_partida,
    revisar_existencia(materia_prima,lote(N,total),
    cantidad(C_barras_existencia),material(barra,virgen),medidas(D,_)),
    C_barras_existencia>B_acum2,
    requisicion(lote(N,P),cantidad(C_esp_partida),
    material(esparrago,dox),_ ,_),
    revisar_existencia(almacen,lote(N,total),cantidad(C_al_tca),
    material(tuerca,dox),medidas(D,_)),
    revisar_existencia(materia_prima,lote(N,total),cantidad(C_mp_tca),
    material(tuerca,virgen),medidas(D,_)),
    C_total_tca is C_al_tca+C_mp_tca,
    T_acum2 is T_acum1+C_esp_partida*2,
    C_total_tca>T_acum2,
    seleccionar(N,[P|Ps],B_acum2,T_acum2,T,D,PL).

seleccionar(_ ,Ps,_ ,T_acum,T_acum,D,Ps):-!.

```

Donde PL es la lista de partidas y TcaL es la lista de tuercas que deben de ser procesadas.

- Calcular las rondas de espárragos a procesar, ésta se convierte en la orden de corte y preparación de superficie para espárragos.
- La lista de tuercas (TcaL) se convierte en la orden de producción para preparación de superficie de tuercas. Su preparación se efectúa en orden descendente, del mayor al menor diámetro, y se aprovechan los dos sistemas para preparación de tuercas ($1 \leq \phi \leq 2\frac{1}{2}$ y $\phi < 1$).

3.4.5 Control de proceso

El control de proceso se puede efectuar utilizando un modelo de control automático convencional para regular la corriente. El programa debe suministrar al modelo de control los valores de control para el proceso:

Figura 3.3 Cuadro de diálogo para datos de control de proceso.

Con los datos de densidad de corriente, área y espesor, el programa calcula la intensidad de corriente y el tiempo de Proceso.

Figura 3.4 Control de proceso.

Durante la operación, el programa debe leer el valor de la intensidad de corriente y la resistencia de la celda, o en su defecto el valor de la caída de voltaje y aplicar la ley de ohm para estimar el valor de la resistencia. Si la desviación excede al 5%, entonces se inicia un diálogo con el usuario para determinar las causas de la desviación, Tabla 3.6. Simultáneamente, el programa debe seguir leyendo el valor de la resistencia.



Figura 3.5 Ventanas de diálogo para control de resistencia.

Si el programa determina que la desviación ha disminuido a un valor menor al 5%, entonces el programa cancela automáticamente la ventana de diálogo y reinicia el control automático. Si el operador concluye que no hay un falso contacto y decide continuar la operación, el programa preguntará si se debe modificar el valor de control y cambiarlo por el valor real de operación; al modificar este valor, el procedimiento de control de proceso debe utilizar este valor como el nuevo punto de control.



Figura 3.6 Ventana de diálogo para modificar la resistencia de control. El programa para el modelo de control utiliza las siguientes herramientas:

1. Comunicación con puertos.
2. Programación multitarea y multisubproceso.
3. Estructuras de control.
4. Estructuras de repetición.
5. Comunicación con el usuario.

Aunque ya existen en el mercado algunas versiones de lenguajes para programación lógica para ambiente windows, un lenguaje como C (programación estructurada) o C++ (programación orientada a objetos) tiene mejores herramientas³² para desarrollar este programa. Además, el programa no necesita una base de conocimientos y la toma de decisiones no es un problema complejo.

³²De hecho, algunas versiones de Prolog sugieren que toda la programación Windows se desarrolle en C y se utilice, incorporando al intérprete dentro del programa, su máquina de inferencia para el manejo de datos y la toma de decisiones complejas.

CAPÍTULO 4

VALIDACIÓN DEL MODELO

Para validar el sistema experto se dedujo una solución para la requisición de la Tabla 3.4 y se comparó con la respuesta del sistema. En el Apéndice E se presenta el inventario de material utilizado para obtener la solución. Se utilizó la versión 3.3 de Quintus Prolog como intérprete.

4.1 Solicitud de material

Para estimar la solicitud de compra podemos utilizar una hoja de cálculo, véase Figura 4.1.

G2		=SI(F2=0,0,ENTERO((F2)/ENTERO(144/(C2+0.125)))+1)					
	PARTIDA	DIÁMETRO	LARGO	REQUISICIÓN	EXISTENCIA	FALTANTE	BARRAS
2	1	3/8	9 1/2	192	64	128	10
3	2	1/2	2	30	0	30	1
4	3	1/2	2 1/2	262	179	83	2
5	4	1/2	3	3600	450	3150	69

Figura 4.1 Hoja de cálculo para solicitud de material.

Donde "REQUISICIÓN" es la cantidad de espárragos de la requisición, "EXISTENCIA" es la cantidad de espárragos con la misma dimensión que están disponibles en el almacén, "FALTANTE" es la diferencia entre la cantidad requerida y la cantidad existente, y "BARRAS" es el número de barras que se deben cortar para producir el material requerido. El cálculo del número de barras se hace con la fórmula " $=SI(Fi=0, 0, ENTERO((Fi)/ENTERO(144/(Ci+0.125)))+1$ ", véase Figura 4.1. Para obtener esta información se utiliza la requisición alfa, Tabla 3.4, y el inventario de material, Apéndice E. El resultado se presenta en la siguiente tabla:

PARTIDA	DIÁMETRO	LARGO	REQUISICIÓN	EXISTENCIA	FALTANTE	BARRAS
1	3/8	9 1/2	192	64	128	10
2	1/2	2	30	0	30	1
3	1/2	2 1/2	262	179	83	2
4	1/2	3	3600	450	3150	69
5	1/2	3 1/2	780	108	672	18
6	1/2	4	156	185	0	0
7	1/2	4 1/2	36	0	36	2
8	1/2	6	24	0	24	2

PARTIDA	DIÁMETRO	LARGO	REQUISIÓN	EXISTENCIA	FALTANTE	BARRAS
9	5/8	2 1/2	56	0	56	2
10	5/8	3	546	0	546	12
11	5/8	3 1/2	5156	0	5156	133
12	5/8	3 3/4	24	0	24	1
13	5/8	4	10274	4038	6236	184
14	5/8	4 1/2	4108	86	4022	130
15	5/8	5	620	0	620	23
16	5/8	5 1/2	264	0	264	11
17	5/8	6	32	0	32	2
18	5/8	6 1/2	212	0	212	11
19	5/8	7	112	14	98	5
20	5/8	7 1/2	48	0	48	3
21	5/8	8 1/2	8	0	8	1
22	5/8	9 1/2	128	45	83	6
23	3/4	2 1/2	56	14	42	1
24	3/4	3	28	9	19	1
25	3/4	3 1/2	88	0	88	3
26	3/4	4	2156	0	2156	64
27	3/4	4 1/2	5066	700	4366	141
28	3/4	4 3/4	32	200	0	0
29	3/4	5	3426	0	3426	123
30	3/4	5 1/4	144	0	144	6
31	3/4	5 1/2	3278	99	3179	128
32	3/4	6	256	450	0	0
33	3/4	6 1/2	56	130	0	0
34	3/4	7	100	0	100	6
35	3/4	7 1/2	140	2	138	8
36	3/4	8	24	0	24	2
37	3/4	8 1/2	32	0	32	3
38	3/4	9	8	0	8	1
39	3/4	9 1/2	36	0	36	3
40	3/4	10	8	0	8	1
41	3/4	10 1/2	32	0	32	3
42	3/4	12	8	0	8	1
43	7/8	4 1/2	112	0	112	4
44	7/8	5	1672	228	1444	52
45	7/8	5 1/2	964	0	964	39
46	7/8	6	1732	127	1605	70
47	7/8	6 1/2	1464	113	1351	65
48	7/8	7	72	0	72	4
49	7/8	7 1/2	120	0	120	7
50	7/8	8	48	47	1	1
51	7/8	8 1/2	96	0	96	7
52	7/8	9	84	0	84	6
53	7/8	10	16	0	16	2
54	7/8	11	12	0	12	2
55	7/8	12	16	0	16	2
56	7/8	12 1/2	12	0	12	2
57	7/8	13	24	0	24	3
58	7/8	20	48	0	48	7
59	1	5	64	0	64	3
60	1	5 1/2	100	74	26	2

PARTIDA	DIÁMETRO	LARGO	REQUISIÓN	EXISTENCIA	FALTANTE	BARRAS
61	1	6	480	0	480	21
62	1	6 1/2	28	10	18	1
63	1	7	3520	147	3373	169
64	1	7 1/2	304	17	287	16
65	1	9 1/2	16	0	16	2
66	1	10	84	0	84	5
67	1	12 1/2	36	0	36	4
68	1	13	12	24	0	0
69	1 1/8	5	20	5	15	1
70	1 1/8	5 1/2	48	0	48	2
71	1 1/8	6	144	14	130	6
72	1 1/8	6 1/2	432	0	432	21
73	1 1/8	7	384	15	369	19
74	1 1/8	7 1/2	148	404	0	0
75	1 1/8	8	468	80	388	23
76	1 1/8	8 1/2	48	0	48	4
77	1 1/8	9 1/2	28	28	0	0
78	1 1/8	11	12	0	12	2
79	1 1/8	14	16	30	0	0
80	1 1/8	21	32	143	0	0
81	1 1/4	7	180	40	140	8
82	1 1/4	7 1/2	1520	127	1393	78
83	1 1/4	8	1020	29	991	59
84	1 1/4	8 1/2	368	0	368	24
85	1 1/4	9	1736	113	1623	109
86	1 1/4	9 1/2	260	10	250	18
87	1 1/4	10	80	0	80	6
88	1 1/4	11	80	11	69	6
89	1 1/4	12 1/2	120	0	120	11
90	1 1/4	15	60	18	42	5
91	1 1/4	18	64	37	27	4
92	1 1/4	18 1/2	20	10	10	2
93	1 1/4	19 1/2	20	285	0	0
94	1 1/4	21	20	0	20	4
95	1 3/8	9 1/2	84	34	50	4
96	1 3/8	10	80	24	56	5
97	1 3/8	11	160	69	91	8
98	1 1/2	9	160	8	152	11
99	1 1/2	9 1/2	480	33	447	32
100	1 1/2	10	712	52	660	48
101	1 1/2	10 1/2	172	41	131	11
102	1 1/2	11	96	22	74	7
103	1 1/2	11 1/2	20	15	5	1
104	1 1/2	12	152	5	147	14
105	1 1/2	12 1/2	36	0	36	4
106	1 1/2	18 1/2	20	18	2	1
107	1 1/2	23	60	1	59	10
108	1 5/8	9 1/2	20	0	20	2
109	1 5/8	11	20	46	0	0
110	1 5/8	11 1/2	428	6	422	36
111	1 5/8	12	480	23	457	42
112	1 5/8	12 1/2	248	0	248	23

PARTIDA	DIÁMETRO	LARGO	REQUISION	EXISTENCIA	FALTANTE	BARRAS
113	1 5/8	13	40	0	40	5
114	1 5/8	13 1/2	120	48	72	8
115	1 5/8	14 1/2	24	20	4	1
116	1 5/8	18 1/2	16	9	7	2
117	1 5/8	20 1/2	20	28	0	0
118	1 5/8	26 1/2	48	0	48	10
119	1 3/4	12	28	0	28	3
120	1 3/4	13	32	0	32	4
121	1 3/4	18 1/2	16	0	16	3
122	1 3/4	21	16	0	16	3
123	1 3/4	22	32	0	32	6
124	1 7/8	13 1/2	504	30	474	48
125	1 7/8	14	168	11	157	16
126	1 7/8	14 1/2	48	47	1	1
127	1 7/8	15	72	40	32	4
128	1 7/8	15 1/2	24	5	19	3
129	1 7/8	16	24	1	23	3
130	1 7/8	17	24	0	24	4
131	1 7/8	21	16	15	1	1
132	1 7/8	30 1/2	24	0	24	7
133	1 7/8	31	24	0	24	7
134	2	13 1/2	32	0	32	4
135	2	19 1/2	20	0	20	3
136	2	20	40	0	40	6
137	2	20 1/2	80	23	57	10
138	2	21	60	0	60	11
139	2	21 1/2	20	0	20	4
140	2 1/2	16	112	37	75	10
141	2 1/2	16 1/2	672	0	672	85
142	2 1/2	17	28	0	28	4
143	2 1/2	18	28	0	28	5
144	2 1/2	18 1/2	56	0	56	9
145	2 1/2	19	56	0	56	9
146	3 3/4	32	16	0	16	5

Tabla 4.1 Cálculo de Barras para la requisición alfa.

Para calcular la solicitud de compra de barras es necesario sumar la cantidad de barras de todas las partidas con el mismo diámetro; por ejemplo:

PARTIDA	DIÁMETRO	BARRAS
2	1/2	1
3	1/2	2
4	1/2	69
5	1/2	18
6	1/2	0
7	1/2	2
8	1/2	2

Barras totales: 94

Tabla 4.2 Barras totales para espárragos de 1/2 pulgada.

Del inventario de material vemos que la cantidad de barras en el almacén es de 160 y concluimos que no necesitamos comprar más barras porque la cantidad de barras en almacén es mayor que la cantidad de barras necesarias para procesar el material faltante. El resumen del cálculo para todas las barras es:

DIÁMETRO	REQUISICIÓN	EXISTENCIA	SOLICITUD
3/8	10	6	4
1/2	94	160	0
5/8	524	334	190
3/4	495	314	181
7/8	273	112	161
1	223	101	122
1 1/8	78	118	0
1 1/4	334	145	189
1 3/8	17	48	0
1 1/2	139	65	74
1 5/8	129	9	120
1 3/4	19	0	19
1 7/8	94	34	60
2	38	23	15
2 1/2	122	1	121
3 3/4	5	0	5

Tabla 4.3 Solicitud de barras para la requisición alfa.

Para calcular la solicitud de compra de tuercas se debe sumar la cantidad de espárragos por partida y multiplicarlo por dos, a esta cantidad se le debe restar el inventario de almacén, incluyendo las tuercas con recubrimiento y sin recubrimiento. Por ejemplo:

PARTIDA	DIÁMETRO	REQUISICIÓN
2	1/2	30
3	1/2	262
4	1/2	3600
5	1/2	780
6	1/2	156
7	1/2	36
8	1/2	24

Total: 4888

Tuercas totales: 9776

Tabla 4.4 Tuercas totales para espárragos de 1/2 pulgada.

Del inventario de material tenemos que hay 2870 tuercas con recubrimiento y no hay tuercas sin recubrimiento de manera tal que la cantidad de tuercas que se deben comprar es 6906 (9776-2870). La solicitud de compra de tuercas para todas los diámetros es:

DIAMETRO	REQUISICION	EXISTENCIA	EXISTENCIA	SOLICITUD
3/8	384	5000	0	0
1/2	9776	2870	0	6906
5/8	43176	1640	0	44536
3/4	29948	45	0	29903
7/8	12984	0	2600	10384
1	9248	4847	0	4401
1 1/8	3560	1356	3500	0
1 1/4	11096	1315	6750	3031
1 3/8	648	124	1060	0
1 1/2	3816	141	6900	0
1 5/8	2928	1636	120	1172
1 3/4	248	600	570	0
1 7/8	1856	445	0	1411
2	504	18	585	0
2 1/2	1904	103	2736	0
3 3/4	32	0	0	32

CON RECUBRIMIENTO SIN RECUBRIMIENTO

Tabla 4.5 Solicitud de compra de tuercas para la requisición alfa.

La respuesta del intérprete

Para poder hacer preguntas al intérprete es necesario suministrar los hechos y reglas correspondientes.

```

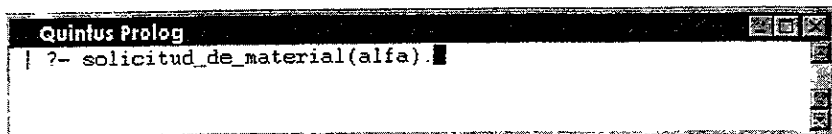
Quintus Prolog
| ?- [requisicion,almacen,experto]
% compiling file C:/experto/requisicion.pl
% requisicion.pl compiled in module user, 0.160 sec 18,956 bytes
% compiling file C:/experto/almacen.pl
% almacen.pl compiled in module user, 0.110 sec 15,440 bytes
% compiling file C:/experto/experto.pl
% experto.pl compiled in module user, 0.220 sec 14,724 bytes

yes
| ?-

```

Figura 4.2 El intérprete lógico.

Si le preguntamos al intérprete por la materia prima, espárragos y tuercas, que se deben de comparar para cumplir con el lote alfa, Figura 4.3,



```

Quintus Prolog
| ?- solicitud_de_material(alfa).

```

Figura 4.3 Preguntando al intérprete.

su respuesta será:

```

Quintus Prolog
| ?- solicitud_de_material(alfa).
Lote alfa
Fecha de solicitud: date(100,5,15)
Comprar.
Material          Diametro          Cantidad
barra             3 75             5
barra             2.5             121
barra             2               15
barra             1.875           60
barra             1.75            19
barra             1 625           120
barra             1.5             74
barra             1.25            189
barra             1               122
barra             0 875           161
barra             0 75            181
barra             0.625           190
barra             0.375           4
tuerca            3.75            32
tuerca            1.875           1411
tuerca            1.625           1172
tuerca            1.25            3031
tuerca            1               4401
tuerca            0.875           10384
tuerca            0.75            29498
tuerca            0.625           41536
tuerca            0 5             6906

yes
| ?-

```

Figura 4.4 La respuesta del intérprete.

Si comparamos los resultados presentados en las Tablas 4.3 y 4.5 con los resultados presentados en la Figura 4.4 tenemos:

DIAMETRO	SOLICITUD DE TUERCAS		SOLICITUD DE BARRAS	
	SOLUCIÓN	INTERPRETE	SOLUCIÓN	INTERPRETE
3/8	0	0	4	4
1/2	6906	6906	0	0
5/8	41536	41536	190	190
3/4	29498	29498	181	181
7/8	10384	10384	161	161
1	4401	4401	122	122
1 1/8	0	0	0	0
1 1/4	3031	3031	189	189
1 3/8	0	0	0	0
1 1/2	0	0	74	74
1 5/8	1172	1172	120	120
1 3/4	0	0	19	19
1 7/8	1411	1411	60	60
2	0	0	15	15
2 1/2	0	0	121	121
3 3/4	32	32	5	5

Tabla 4.6 Validación de la solución para solicitud de material.

Donde las columnas con título "SOLUCIÓN" corresponde a los valores de las Tablas 4.4 y 4.5 y las columnas título "INTÉRPRETE" corresponde a los valores calculados por el intérprete para el mismo caso. Las celdas que están vacías corresponden a los casos donde la solicitud es cero, en los demás casos vemos que el valor de los dos cálculos es el mismo. De esta tabla podemos concluir que el procedimiento `solicitud_de_material(N)` es correcto.

Del Apéndice C tenemos que el procedimiento `solicitud_de_material(N)` es:

```
solicitud_de_material(N):-
    requisicion_de_tuercas(N),
    modificar_existencia(N),
    modificar_barras(N),
    modificar_tuercas(N),
    solicitud_de_material(N,tuerca),
    solicitud_de_material(N,barra),
    write('Lote '),write(N),nl,
    requisicion(lote(N,comprar),_,_,_,F_solicitud),
    write('Fecha de solicitud: '),write(F_solicitud),nl,
    write('Comprar: '),nl,
    write('Material          Diametro          Cantidad'),nl,
    desplegar(N,comprar).
```

Para que la conclusión (`solicitud_de_material(N)`) sea verdadera, el argumento tiene que ser verdadero. Para el caso en que $N=\alpha$, concluimos que son verdaderos los procedimientos:

- `requisicion_de_tuercas(N)`,
- `modificar_existencia(N)`,
- `modificar_barras(N)`,
- `modificar_tuercas(N)`,
- `solicitud_de_material(N,M)`,
- `desplegar(N,X)`.

Suponemos que estos procedimientos son correctos para cualquier caso en donde el material a procesar sean juegos de espárragos, salvo que alguien presente un contraejemplo que demuestre lo contrario. De la misma manera podemos concluir que son válidos los procedimientos:

- `sumatoria(L,C,M,D,F)`.
- `modificacion(C1,C2,L,M,D,E)`.
- `orden_de_produccion(L,C,M,D,F)`.
- `revisar_existencia(E,L,C,M,D)`.

4.2 Material a procesar

Solo se resolvieron cuatro casos particulares que se consideró que son representativos de este tipo de cálculos:

1. Hay suficientes barras y tuercas en el almacén para procesar y empacar todas las partidas del mismo diámetro.
2. No hay suficientes tuercas en el almacén para procesar y empacar todas las partidas del mismo diámetro.
3. No hay suficientes barras en el almacén para procesar y empacar todas las partidas del mismo diámetro.

**ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA**

4. No hay suficiente material para procesar y empacar una sola partida de un diámetro específico.

Caso 1. Diámetro: $1\frac{1}{8}$ pulgadas.

De las Tablas 4.3 y 4.4 podemos obtener la siguiente información:

- Diámetro: $1\frac{1}{8}$ pulgadas.
- Requisición de barras: 78.
- Barras en almacén: 118.
- Requisición de tuercas: 3560.
- Tuercas con recubrimiento en almacén: 1356.
- Tuercas sin recubrimiento en almacén: 3500.

Como la cantidad de barras y tuercas en almacen es mayor que la cantidad de barras y tuercas requeridas se concluye que se pueden procesar y empacar todas las partidas con este diámetro.

De la Tabla 4.1 se obtiene la siguiente información:

PARTIDA	DIÁMETRO	LARGO	REQUISIÓN	EXISTENCIA	FALTANTE	BARRAS
69	1 1/8	5	20	5	15	1
70	1 1/8	5 1/2	48	0	48	2
71	1 1/8	6	144	14	130	6
72	1 1/8	6 1/2	432	0	432	21
73	1 1/8	7	384	15	369	19
74	1 1/8	7 1/2	148	404	0	0
75	1 1/8	8	468	80	388	23
76	1 1/8	8 1/2	48	0	48	4
77	1 1/8	9 1/2	28	28	0	0
78	1 1/8	11	12	0	12	2
79	1 1/8	14	16	30	0	0
80	1 1/8	21	32	143	0	0

Tabla 4.7 Cálculo de barras para requisiciones con diámetro de $1\frac{1}{8}$ pulgadas.

De la Tabla 4.7 se obtiene el conjunto de partidas con diámetro $1\frac{1}{8}$ pulgadas es {69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80} donde el subconjunto {74, 77, 79, 80} integra a las partidas de espárragos que solo deben ser empacadas, y el subconjunto {69, 70, 71, 72, 73, 75, 76, 78} agrupa a las partidas de espárragos que deben ser procesadas y empacadas. La cantidad de tuercas a procesar (2204) se calcula de la diferencia entre las tuercas totales necesarias para la requisición (3560) y la cantidad de tuercas que ya tiene recubrimiento (1356).

Caso 2. Diámetro: $\frac{1}{2}$ pulgada.

De las Tablas 4.3 y 4.4 podemos obtener la siguiente información:

- Diámetro: $\frac{1}{2}$ pulgada.
- Requisición de barras: 94.
- Barras en almacén: 160.
- Requisición de tuercas: 9776.
- Tuercas con recubrimiento en almacén: 2870.
- Tuercas sin recubrimiento en almacén: 0.

Como no hay suficientes tuercas para cumplir con la requisición, una o varias partidas no pueden ser procesadas y empacadas. Para decidir cual o cuales partidas desechar, necesitamos información de la Tabla 4.1:

PARTIDA	DIÁMETRO	LARGO	REQUISIÓN	EXISTENCIA	FALTANTE	BARRAS
2	1/2	2	30	0	30	1
3	1/2	2 1/2	262	179	83	2
4	1/2	3	3600	450	3150	69
5	1/2	3 1/2	780	108	672	18
6	1/2	4	156	185	0	0
7	1/2	4 1/2	36	0	36	2
8	1/2	6	24	0	24	2

Tabla 4.8 Cálculo de barras para requisiciones con diámetro de 1/2 pulgada.

Si comparamos la cantidad de espárragos de la partida 4 con la cantidad de tuercas en almacén, vemos que no es posible procesar y empacar esta partida porque la cantidad de espárragos es mucho mayor que la cantidad de tuercas. Si calculamos la cantidad de tuercas C necesarias para el conjunto de requisiciones {2, 3, 5, 6, 7, 8} tenemos

$$C=2 \times (30+262+780+156+36+24) = 2576(\text{tuercas}),$$

donde C es menor que la cantidad de tuercas en almacén (2870).

De esta información y de que no necesitamos procesar ningún espárrago de la partida 6 podemos concluir que el conjunto de partidas con diámetro de 1/2 pulgada {2, 3, 5, 7, 8} integra a las partidas de espárragos que deben ser procesadas y empacadas, y la partida 6 solo debe ser empacada. La cantidad de tuercas a procesar es cero porque no existen tuercas sin recubrimiento de esta medida y porque la cantidad de tuercas con recubrimiento es suficiente para empacar las partidas de juegos de espárragos del conjunto {2, 3, 5, 6, 7, 8}.

Caso 3. Diámetro: 1 1/2 pulgadas.

De las Tablas 4.3 y 4.4 podemos obtener la siguiente información:

- Diámetro: 1 1/2 pulgada.
- Requisición de barras: 239.
- Barras en almacén: 65.
- Requisición de tuercas: 3816.
- Tuercas con recubrimiento en almacén: 141.
- Tuercas sin recubrimiento en almacén: 6900.

Como no hay suficientes en barras para cumplir con la requisición, una o varias partidas no pueden ser procesadas y empacadas. Para decidir cual o cuales partidas desechar, necesitamos información de la Tabla 4.1:

PARTIDA	DIÁMETRO	LARGO	REQUISIÓN	EXISTENCIA	FALTANTE	BARRAS
98	1 1/2	9	160	8	152	11
99	1 1/2	9 1/2	480	33	447	32
100	1 1/2	10	712	52	660	48
101	1 1/2	10 1/2	172	41	131	11
102	1 1/2	11	96	22	74	7
103	1 1/2	11 1/2	20	15	5	1
104	1 1/2	12	152	5	147	14
105	1 1/2	12 1/2	36	0	36	4
106	1 1/2	18 1/2	20	18	2	1
107	1 1/2	23	60	1	59	10

Tabla 4.9 Cálculo de barras para requisiciones con diámetro de 1 1/2 pulgadas.

Para seleccionar las partidas que deben ser procesadas y empacadas se utilizaron los siguientes pasos³³:

1. Sean $D=1\frac{1}{2}$ el diámetro, $C=0$ la cantidad de barras acumuladas, $C_{al}=65$ la cantidad de barras en almacén, $PP=\{\}$ el conjunto de partidas que pueden ser procesadas, $PD=\{\}$ el conjunto de partidas que no pueden ser procesadas, $PT=PP\cap PD$, y PU el conjunto de partidas {98, 99, 100, 101, 102, 103, 104, 105, 106, 107}.
2. Sea P la partida con diámetro D que tiene la mayor longitud de las partidas pertenecientes PU y que no pertenece a PT . De la Tabla 4.12 se concluye que $P=107$.
3. Si $C_p=10$ es la cantidad de barras necesarias para procesar la partida P y $C_p+C\leq 65$ entonces $C_1=C_p+C$, $PP_1=\{107\}$ es el conjunto de partidas que pueden ser procesadas, $PD_1=\{\}$ es el conjunto de partidas que no pueden ser procesadas y $PT_1=PP_1\cap PD_1$.
4. Sea P_1 la partida con diámetro D que tiene la mayor longitud de las partidas pertenecientes al conjunto PU y que no pertenece a PT_1 . De la Tabla 4.12 se concluye que $P_1=106$.
5. Si $C_{p1}=1$ es la cantidad de barras necesarias para procesar la partida P_1 y $C_{p1}+C_1\leq 65$ y entonces $C_2=C_{p1}+C_1$, $PP_2=\{106,107\}$, $PD_2=\{\}$ y $PT_2=PP_2\cap PD_2$.
6. Sea P_2 la partida con diámetro D que tiene la mayor longitud de las partidas pertenecientes al conjunto PU y que no pertenece a PT_2 . De la Tabla 4.12 se concluye que $P_2=105$.
7. Si $C_{p2}=4$ es la cantidad de barras necesarias para procesar la partida P_2 y $C_{p2}+C_2\leq 65$ entonces $C_3=C_{p2}+C_2$, $PP_3=\{105,106,107\}$, $PD_3=\{\}$ y $PT_3=PP_3\cap PD_3$.
8. Sea $P_3\dots$
9. Si $C_{p3}=14\dots$
10. Sea $P_4\dots$
11. Si $C_{p4}=14\dots$
12. Sea $P_5\dots$
13. Si $C_{p5}=1\dots$
14. Sea $P_6\dots$
15. Si $C_{p6}=7\dots$
16. Sea $P_7\dots$
17. Si $C_{p7}=11$ es la cantidad de barras necesarias para procesar la partida P_2 y $C_{p7}+C_7\leq 65$ entonces $C_8=C_{p7}+C_7$, $PP_8=\{101,102,103,104,105,106,107\}$, $PD_8=\{\}$ y $PT_8=PP_8\cap PD_8$.
18. Sea P_8 la partida con diámetro D que tiene la mayor longitud de las partidas pertenecientes al conjunto PU y que no pertenece a PT_8 . De la Tabla 4.12 se concluye que $P_8=100$.
19. Si $C_{p8}=48$ es la cantidad de barras necesarias para procesar la partida P_2 y $C_{p8}+C_8\leq 65$ entonces $C_9=C_{p8}+C_8$, $PP_9=\{100,101,102,103,104,105,106,107\}$, $PD_9=\{\}$ y $PT_9=PP_9\cap PD_9$; como no es el caso, entonces $C_9=C_9$, $PP_9=PP_8$, $PD_9=\{100\}$ y $PT_9=PP_9\cap PD_9$.
20. Sea P_9 la partida con diámetro D que tiene la mayor longitud de las partidas pertenecientes al conjunto PU y que no pertenece a PT_9 . De la Tabla 4.12 se concluye que $P_9=99$.

³³El procedimiento seleccionar, véase Apéndice C, utiliza un algoritmo muy parecido al descrito en esta secuencia de pasos.

21. Si $C_{p9}=32$ es la cantidad de barras necesarias para procesar la partida P2 y $C_{p9}+C9 \leq 65$ entonces $C10=C_{p9}+C9$, $PP10=\{99,101,102,103,104,105,106,107\}$, $PD10=\{100\}$ y $PT10=PP10 \cap PD10$; como no es el caso, entonces $C10=C9$, $PD10=\{99,100\}$ y $PT10=PP10 \cap PD10$.
22. Sea P10 la partida con diámetro D que tiene la mayor longitud de las partidas pertenecientes al conjunto PU y que no pertenece a PT10. De la Tabla 4.12 se concluye que $P10=98$.
23. Si $C_{p9}=11$ es la cantidad de barras necesarias para procesar la partida P2 y $C_{p9}+C9 \leq 65$ entonces $C10=C_{p9}+C9$, $PP11=\{98,101,102,103,104,105,106,107\}$, $PD11=\{99,100\}$ y $PT11=PP11 \cap PD11$.
24. Sea P11 la partida con diámetro D que tiene la mayor longitud de las partidas pertenecientes al conjunto PU y que no pertenece a PT11. Como $PU=PT11$ entonces no se puede deducir P11.
25. El conjunto de partidas que pueden ser procesadas y empacadas es $PP11=\{98,101,102,103,104,105,106,107\}$.

Si calculamos la cantidad de barras B necesarias para el conjunto de requisiciones $\{98,101,102,103,104,105,106,107\}$ tenemos

$$B=2 \times (11+11+7+1+14+4+1+10) = 59(\text{barras}),$$

donde B es menor que la cantidad de barras en almacén (65). Si calculamos la cantidad de tuercas C necesarias para el mismo conjunto de requisiciones tenemos

$$C=2 \times (160+172+96+20+152+36+20+60) = 1432(\text{tuercas}),$$

si restamos a C la cantidad de tuercas con recubrimiento que hay en almacén (141) tenemos que la cantidad de tuercas T que se deben procesar para poder empacar estas requisiciones es

$$T=1432-141=1291(\text{tuercas}).$$

Caso 4. Diámetro: 2½ pulgadas.

De las Tablas 4.3 y 4.4 podemos obtener la siguiente información:

- Diámetro: 2½ pulgada.
- Requisición de barras: 122.
- Barras en almacén: 1.
- Requisición de tuercas: 1904.
- Tuercas con recubrimiento en almacén: 103.
- Tuercas sin recubrimiento en almacén: 2736.

Como no hay suficientes barras para cumplir con la requisición, una o varias partidas no pueden ser procesadas y empacadas. Para decidir cual o cuales partidas desechar, necesitamos información de la Tabla 4.1:

PARTIDA	DIÁMETRO	LARGO	REQUISIÓN	EXISTENCIA	FALTANTE	BARRAS
140	2 1/2	16	112	37	75	10
141	2 1/2	16 1/2	672	0	672	85
142	2 1/2	17	28	0	28	4
143	2 1/2	18	28	0	28	5
144	2 1/2	18 1/2	56	0	56	9
145	2 1/2	19	56	0	56	9

Tabla 4.10 Cálculo de barras para requisiciones con diámetro de 1½ pulgadas.

Como todas las partidas necesitan más de una barra para poder ser procesadas y empacadas, se concluye que no se puede procesar ninguna partida con este diámetro.

La respuesta del intérprete

Si le preguntamos al intérprete por el conjunto de partidas de juegos de espárragos que pueden ser procesados

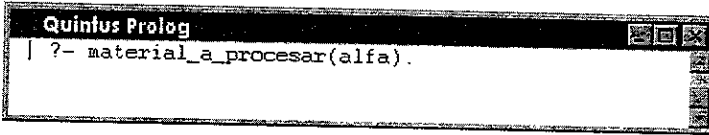


Figura 4.5 Preguntando al intérprete por el material a procesar. su respuesta es:

```

Quintus Prolog
| ?- material_a_procesar(alfa).

      Material a procesar

Diametro   Tuercas   Partidas
0.5        0           8,7,5,3,2
0.625     0           22,21,20,19,18,17,16
0.75      0           42,41,40,39,38,37
0.875     1320        58,57,56,55,54,53,52,51,50,49,48,43
1.125     2204        78,76,75,73,72,71,70,69
1.25      2909        94,92,91,90,89,88,87,86,84,83
1.375     524          97,96,95
1.5       1291        107,106,105,104,103,102,101,98
1.625     0            116,115,113
1.875     0            133,132,131,130,129,128,127

      Partidas para empaque

Diametro   Partidas
0.5        6
0.625     33,28
0.75      80,79,77,74
1.125     93
1.25      117,109
1.375
1.5
1.625
1.875

yes _

```

Figura 4.6 Respuesta del intérprete.

Si comparamos los resultados de los casos particulares con los propuestos por el intérprete para el mismo diámetro tenemos que:

CASO	DIAMETRO	CALCULO	INTERPRETE
1	1 1/8	2204	2204
2	1/2	0	0
3	1 1/2	1291	1291
4	2 1/2	0	0

Tabla 4.11 Número de tuercas a procesar.

CASO	DIAMETRO	CALCULO	INTERPRETE
1	1 1/8	69, 70, 71, 72, 73, 75, 76, 78	78, 76, 75, 73, 72, 71, 70, 69
2	1/2	2, 3, 5, 7, 8	8, 7, 5, 3, 2
3	1 1/2	98, 101, 102, 103, 104, 105, 106, 107	107, 106, 105, 104, 103, 102, 101, 98
4	2 1/2		

Tabla 4.12 Partidas de espárragos a procesar.

CASO	DIAMETRO	CALCULO	INTERPRETE
1	1 1/8	74, 77, 79, 80	80, 79, 77, 74
2	1/2	6	6
3	1 1/2		
4	2 1/2		

Tabla 4.13 Partidas de espárragos para empacar

De las Tablas 4.11, 4.12 y 4.13 se puede concluir que el procedimiento `material_a_procesar` es correcto para los casos estudiados.

Del Apéndice C tenemos que el procedimiento `material_a_procesar(N)` es:

```
material_a_procesar(N):-
    nl,
    write('          Material a procesar'),nl,nl,
    write('Diametro      Tuercas          Partidas'),nl,
    material_a_procesar(N,PL,TcaL),
    presentar_partidas_procesar(N,PL,TcaL),nl,nl,
    write('          Partidas para empaque'),nl,nl,
    write('Diametro      Partidas'),nl,
    presentar_partidas_armar(N,PL,TcaL).
```

Para que la conclusión (`material_a_procesar(N)`) sea verdadera, el argumento tiene que ser verdadero. Para el caso en que $N=\alpha$, concluimos que son verdaderos los procedimientos:

- `material_a_procesar(N,PL,TcaL)`.
- `presentar_partidas_procesar(N,PL,TcaL)`.
- `presentar_partidas_armar(N,PL,TcaL)`.

Suponemos que estos procedimientos son correctos para cualquier caso en donde el material a procesar sean juegos de espárragos, salvo que alguien presente un contraejemplo que demuestre lo contrario. De la misma manera podemos concluir que son válidos los procedimientos:

- `partidas_armar(N,D,PL)`.
- `material_a_procesar(N,D,P,T,PL,TcaL)`.
- `tuercas_a_procesar(A,B,C)`.
- `seleccionar(N,Ps,B_acum1,T_acum1,T,D,PL)`.
- `armar(N,P)`.

Es importante recordar que el algoritmo debe presentar soluciones correctas para todos los casos donde las requisiciones sean de juegos de espárragos. Sin embargo, el hecho de que la solución sea correcta no implica que sea la mejor solución para todos los casos; por ejemplo, en el caso 3 hay mas de una solución.

4.3 Cálculo de rondas.

Para estimar el número de rondas para el lote completo, el coordinador de producción normalmente utiliza una hoja de cálculo para ordenar las partidas a su conveniencia. Primero organiza las partidas en orden descendente utilizando la longitud del material como criterio de selección. Posteriormente define las piezas máximas por ronda para cada partida y utiliza este dato como criterio para ordenar nuevamente la hoja cálculo.

PARTIDA	CANTIDAD	MATERIAL	DIÁMETRO	LARGO	PIEZAS POR RONDA
118	48	jes	1 5/8	26 1/2	6
132	24	jes	1 7/8	30 1/2	6
133	24	jes	1 7/8	31	6
146	16	jes	3 3/4	32	6
58	48	jes	7/8	20	8
80	32	jes	1 1/8	21	8
94	20	jes	1 1/4	21	8
107	60	jes	1 1/2	23	8
117	20	jes	1 5/8	20 1/2	8
122	16	jes	1 3/4	21	8
123	32	jes	1 3/4	22	8
131	16	jes	1 7/8	21	8
136	40	jes	2	20	8
137	80	jes	2	20 1/2	8
138	60	jes	2	21	8
139	20	jes	2	21 1/2	8
135	20	jes	2	19 1/2	16
140	112	jes	2 1/2	16	16
141	672	jes	2 1/2	16 1/2	16
142	28	jes	2 1/2	17	16
143	28	jes	2 1/2	18	16
144	56	jes	2 1/2	18 1/2	16
145	56	jes	2 1/2	19	16
108	20	jes	1 5/8	9 1/2	24
111	480	jes	1 5/8	12	24
112	248	jes	1 5/8	12 1/2	24
113	40	jes	1 5/8	13	24
114	120	jes	1 5/8	13 1/2	24
115	24	jes	1 5/8	14 1/2	24
116	16	jes	1 5/8	18 1/2	24
119	28	jes	1 3/4	12	24
120	32	jes	1 3/4	13	24
121	16	jes	1 3/4	18 1/2	24
124	504	jes	1 7/8	13 1/2	24
125	168	jes	1 7/8	14	24
126	48	jes	1 7/8	14 1/2	24
127	72	jes	1 7/8	15	24
128	24	jes	1 7/8	15 1/2	24
129	24	jes	1 7/8	16	24
130	24	jes	1 7/8	17	24

PARTIDA	CANTIDAD	MATERIAL	DIAMETRO	LARGO	PIEZAS POR RONDA
134	32	jes	2	13 1/2	24
40	8	jes	3/4	10	40
41	32	jes	3/4	10 1/2	40
42	8	jes	3/4	12	40
53	16	jes	7/8	10	40
54	12	jes	7/8	11	40
55	16	jes	7/8	12	40
56	12	jes	7/8	12 1/2	40
57	24	jes	7/8	13	40
66	64	jes	1	10	40
67	36	jes	1	12 1/2	40
68	12	jes	1	13	40
78	12	jes	1 1/8	11	40
79	16	jes	1 1/8	14	40
87	80	jes	1 1/4	10	40
88	80	jes	1 1/4	11	40
89	120	jes	1 1/4	12 1/2	40
90	60	jes	1 1/4	15	40
91	64	jes	1 1/4	18	40
92	20	jes	1 1/4	18 1/2	40
93	20	jes	1 1/4	19 1/2	40
96	80	jes	1 3/8	10	40
97	160	jes	1 3/8	11	40
100	712	jes	1 1/2	10	40
101	172	jes	1 1/2	10 1/2	40
102	96	jes	1 1/2	11	40
103	20	jes	1 1/2	11 1/2	40
104	152	jes	1 1/2	12	40
105	36	jes	1 1/2	12 1/2	40
106	20	jes	1 1/2	18 1/2	40
109	20	jes	1 5/8	11	40
110	428	jes	1 5/8	11 1/2	40
1	192	jes	3/8	9 1/2	80
19	112	jes	5/8	7	80
20	48	jes	5/8	7 1/2	80
21	8	jes	5/8	8 1/2	80
22	128	jes	5/8	9 1/2	80
34	100	jes	3/4	7	80
35	140	jes	3/4	7 1/2	80
36	24	jes	3/4	8	80
37	32	jes	3/4	8 1/2	80
38	8	jes	3/4	9	80
39	36	jes	3/4	9 1/2	80
48	72	jes	7/8	7	80
49	120	jes	7/8	7 1/2	80
50	48	jes	7/8	8	80
51	96	jes	7/8	8 1/2	80
52	84	jes	7/8	9	80
63	3520	jes	1	7	80
64	304	jes	1	7 1/2	80
65	16	jes	1	9 1/2	80

PARTIDA	CANTIDAD	MATERIAL	DIÁMETRO	LARGO	PIEZAS POR RONDA
73	384	jes	1 1/8	7	80
74	148	jes	1 1/8	7 1/2	80
75	468	jes	1 1/8	8	80
76	48	jes	1 1/8	8 1/2	80
77	28	jes	1 1/8	9 1/2	80
81	180	jes	1 1/4	7	80
82	1520	jes	1 1/4	7 1/2	80
83	1020	jes	1 1/4	8	80
84	368	jes	1 1/4	8 1/2	80
85	1736	jes	1 1/4	9	80
86	260	jes	1 1/4	9 1/2	80
95	84	jes	1 3/8	9 1/2	80
98	160	jes	1 1/2	9	80
99	480	jes	1 1/2	9 1/2	80
8	24	jes	1/2	6	120
15	620	jes	5/8	5	120
16	264	jes	5/8	5 1/2	120
17	32	jes	5/8	6	120
18	212	jes	5/8	6 1/2	120
29	3426	jes	3/4	5	120
30	144	jes	3/4	5 1/4	120
31	3278	jes	3/4	5 1/2	120
32	256	jes	3/4	6	120
33	56	jes	3/4	6 1/2	120
44	1672	jes	7/8	5	120
45	964	jes	7/8	5 1/2	120
46	1732	jes	7/8	6	120
47	1464	jes	7/8	6 1/2	120
59	64	jes	1	5	120
60	100	jes	1	5 1/2	120
61	480	jes	1	6	120
62	28	jes	1	6 1/2	120
69	20	jes	1 1/8	5	120
70	48	jes	1 1/8	5 1/2	120
71	144	jes	1 1/8	6	120
72	432	jes	1 1/8	6 1/2	120
6	156	jes	1/2	4	160
7	36	jes	1/2	4 1/2	160
13	10274	jes	5/8	4	160
14	4108	jes	5/8	4 1/2	160
26	2156	jes	3/4	4	160
27	5066	jes	3/4	4 1/2	160
28	32	jes	3/4	4 3/4	160
43	112	jes	7/8	4 1/2	160
2	30	jes	1/2	2	240
3	262	jes	1/2	2 1/2	240
4	3600	jes	1/2	3	240
5	780	jes	1/2	3 1/2	240
9	56	jes	5/8	2 1/2	240
10	546	jes	5/8	3	240

PARTIDA	CANTIDAD	MATERIAL	DIAMETRO	LARGO	PIEZAS POR RONDA
11	5156	jes	5/8	3 1/2	240
12	24	jes	5/8	3 3/4	240
23	56	jes	3/4	2 1/2	240
24	28	jes	3/4	3	240
25	88	jes	3/4	3 1/2	240

Tabla 4.14 Cálculo de rondas

Finalmente se suman todas las partidas que corresponden a la misma cantidad de piezas por ronda y se dividen entre el número máximo de piezas por ronda correspondiente. El resultado se redondea y se calcula el número total de rondas.

Piezas por ronda	Número de rondas
6	19
8	56
16	61
24	110
40	54
80	150
120	129
160	138
240	44
Total:	743

Tabla 4.15 Cálculo de rondas

La respuesta del intérprete

Si le pedimos al intérprete lógico que nos haga el mismo cálculo su resultado será:

```

Quintus Prolog
| ?- [requisicion,almacen,experto]
% compiling file C:/tesis/quintus/requisicion.pl
% requisicion.pl compiled in module user, 0.160 sec 0 bytes
% compiling file C:/tesis/quintus/almacen.pl
% almacen.pl compiled in module user, 0.170 sec -16,104 bytes
% compiling file C:/tesis/quintus/experto.pl
% experto.pl compiled in module user, 0.380 sec -40 bytes

yes
| ?- programar_rondas_de_esparragos(alfa,todas)
Lote:alfa
Partidas:todas
No. de rondas:665
Tiempo estimado:15 semanas

yes
| ?- █

```

Figura 4.7 Cálculo de rondas con el intérprete.

Era de esperarse este resultado debido a que el programa utiliza reglas de optimización de espacio. Si eliminamos estas reglas del programa, véase Apéndice C, el resultado será:

```

Quintus Prolog
yes
| ?- [requisicion,almacen,experto].
% compiling file C:/tesis/quintus/requisicion.pl
% requisicion.pl compiled in module user, 0.880 sec 0 bytes
% compiling file C:/tesis/quintus/almacen.pl
% almacen.pl compiled in module user, 0.220 sec -16.104 bytes
% compiling file C:/tesis/quintus/experto.pl
% experto.pl compiled in module user, 0.490 sec -3.440 bytes

yes
| ?- programar_rondas_de_esparragos(alfa,todas)
Lote:alfa
Partidas:todas
No. de rondas:744
Tiempo estimado:17 semanas

yes
| ?- █

```

Figura 4.8 Cálculo de ronda sin reglas de optimización.

Con este resultado podemos suponer que el procedimiento `programar_rondas_de_esparragos(N,PL)` es correcto y que también son correctos los procedimientos:

- `programar_rondas_de_esparragos(N,PL,ronda(N,No))`.
- `simular_existencia(N,[cantidad(C1)|Cs],M,[Dim|Dims])`.
- `descripcion_de_ronda(N,PL,A,B,C)`.
- `miembro(P,PL)`.
- `piezas_maximas_por_ronda(T,medidas(D,L),C_max)`.
- `ronda(N,PL,[cantidad(C)],material(T,E),[medidas(D,L)],C_max,C,X,Z)`.

4.4 Cálculo de variables de operación

Si queremos estimar la densidad de corriente y la resistencia de la celda para una ronda de ochos espárragos de 1¼x20 pulgadas, y tenemos la siguiente información:

	PROCESO	CONTROL DE CALIDAD
Número de ronda	1	1
Número de piezas	8	8
Material	espárrago	espárrago
Medidas	1¼x20	1¼x20
Espesor	21 micras	
Resistencia	0.01754Ω	
Densidad de corriente	0.028 ($\frac{A}{cm^2}$)	
Fecha de salida	97/6/25	97/6/25
Hora de salida	7:25'30''	
Observaciones		Con defectos superficiales

Tabla 4.16 Datos para cálculos de variables de operación.

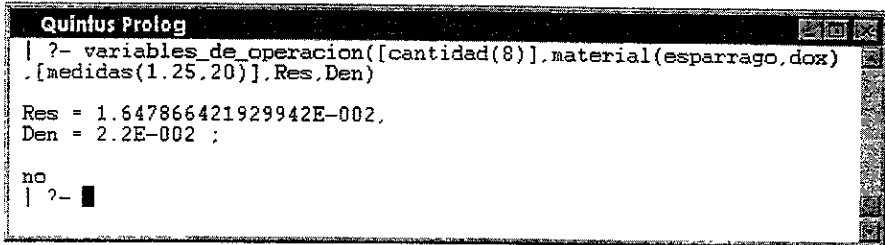
Se utiliza la regla "Si se ha producido un defecto superficial se debe reducir la densidad de corriente según la siguiente ecuación: $j=j_0-0.001 \left(\frac{A}{cm^2}\right)$ " y se concluye que:

$$j = 0.028 - 0.001 = 0.027 \left(\frac{A}{cm^2} \right)$$

$$R \approx 0.01754 \Omega$$

La respuesta del intérprete

Si le pedimos al intérprete que estime las mismas variables de operación, éste contesta:



```

Quintus Prolog
| ?- variables_de_operacion([cantidad(8)],material(esparrago,dox)
.[medidas(1.25,20)].Res,Den)

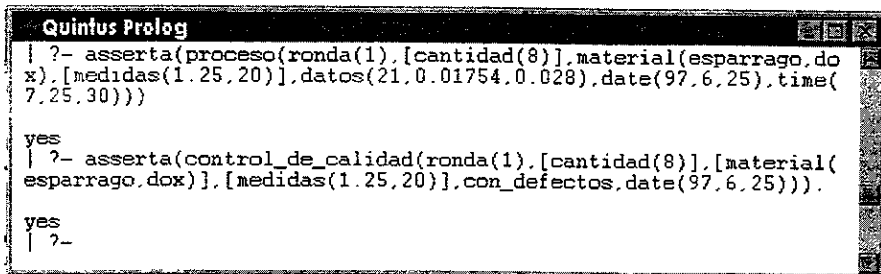
Res = 1.647866421929942E-002,
Den = 2.2E-002 ;

no
| ?- █

```

Figura 4.9 Respuesta del sistema experto.

Si agregamos al sistema experto la información proporcionada en la Tabla 4.16



```

Quintus Prolog
| ?- asserta(proceso(ronda(1).[cantidad(8)].material(esparrago,dox)
.[medidas(1.25,20)].datos(21,0.01754,0.028),date(97.6.25).time(
7.25.30)))

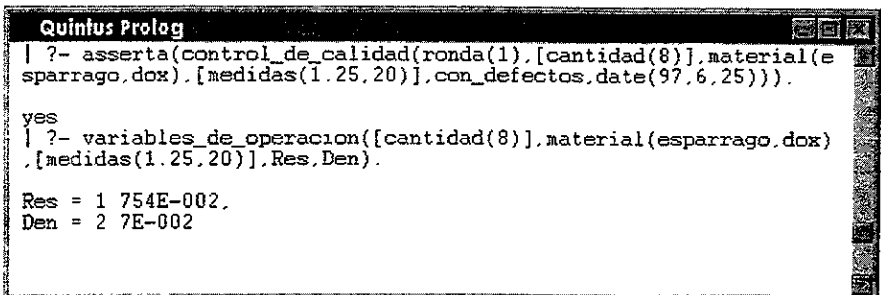
yes
| ?- asserta(control_de_calidad(ronda(1).[cantidad(8)].[material(
esparrago,dox)].[medidas(1.25,20)].con_defectos,date(97.6.25))).

yes
| ?-

```

Figura 4.10 Agregando nuevos hechos al sistema experto.

y volvemos a efectuar la misma pregunta, el intérprete contesta:



```

Quintus Prolog
| ?- asserta(control_de_calidad(ronda(1).[cantidad(8)].material(esparrago,dox)
.[medidas(1.25,20)].con_defectos,date(97.6.25))).

yes
| ?- variables_de_operacion([cantidad(8)],material(esparrago,dox)
.[medidas(1.25,20)].Res,Den).

Res = 1.754E-002,
Den = 2.7E-002

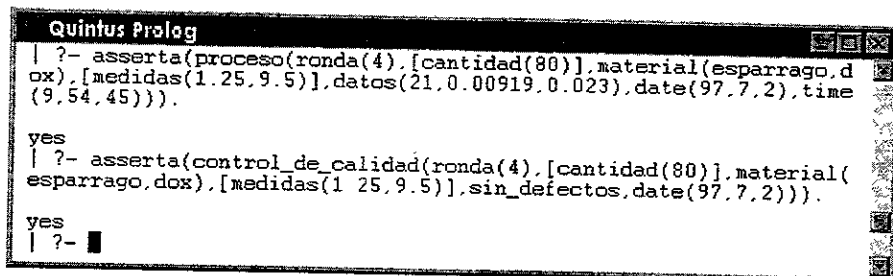
```

Figura 4.11 La respuesta del intérprete usando nuevos conocimientos.

De las dos respuestas del intérprete, Figuras 4.10 y 4.11, se concluye que el procedimiento *variables_de_operacion(A,B,C,D,E)* es correcto porque puede

estimar las variables de operación aunque no tenga información de la ronda y, cuando tiene información, hace la misma deducción que haría el experto humano.

Si agregamos más información al intérprete:



```

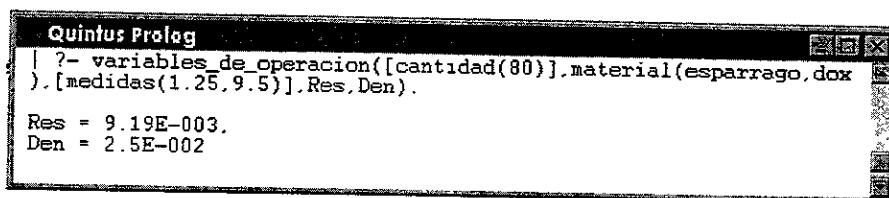
Quintus Prolog
| ?- asserta(proceso(ronda(4).[cantidad(80)].material(esparrago,d
ox).[medidas(1.25,9.5)],datos(21,0.00919,0.023),date(97,7,2),time
(9,54,45))).

yes
| ?- asserta(control_de_calidad(ronda(4).[cantidad(80)].material(
esparrago,dox).[medidas(1.25,9.5)].sin_defectos,date(97,7,2))).

yes
| ?- █
  
```

Figura 4.12 Agregando información al intérprete.

y preguntamos al intérprete:



```

Quintus Prolog
| ?- variables_de_operacion([cantidad(80)].material(esparrago,dox
).[medidas(1.25,9.5)].Res,Den).

Res = 9.19E-003.
Den = 2.5E-002
  
```

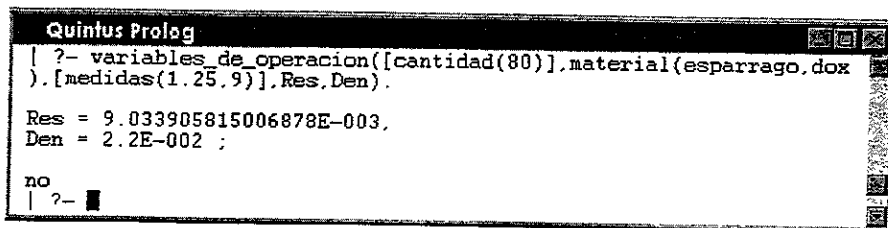
Figura 4.13 Estimación de variables para espárragos de 1¼x9½.

Para deducir estos valores, el intérprete utiliza los siguientes hechos:

proceso(ronda(4).[cantidad(80)].material(esparrago).[medidas(1.25,9.5)],datos(21,0.00919,0.023),date(97,7,2),time(9,54,45)).
 control_de_calidad(ronda(4).[cantidad(80)].material(esparrago).[medidas(1.25,9.5)].sin_defectos,date(97,7,2)).

Al comparar estos hechos, se deduce que la densidad de corriente utilizada en esta ronda puede incrementarse porque no se presentaron defectos y, consecuentemente, la densidad de corriente se obtiene según la ecuación $j = j_0 + 0.002 \left(\frac{A}{cm^2} \right)$.

Si preguntamos al intérprete:



```

Quintus Prolog
| ?- variables_de_operacion([cantidad(80)].material(esparrago,dox
).[medidas(1.25,9)].Res,Den).

Res = 9.033905815006878E-003.
Den = 2.2E-002 ;

no
| ?- █
  
```

Figura 4.14 Estimación de variables para espárragos de 1¼x9.

Como no existen datos para este tipo de ronda se toma como primera aproximación $j=0.022 \frac{A}{cm^2}$. Para estimar la resistencia de la celda utilizamos la ecuación:

$$r=0.4265 \times s^{-0.3902},$$

donde s es la superficie total de los espárragos.

$$s \approx 2.54 \times 2.54 \times 80 \times \pi \times ((1.25^2)/2 + (9 \times 1.25)) = 19508 \text{ cm}^2$$

$$r=0.009034 \Omega$$

4.5 Control de proceso

El programa fue validado en planta y demostró ser capaz de controlar automáticamente la intensidad de corriente. Además, en los casos donde se presentaron resistencias con desviaciones superiores al 5% el programa tomó las mismas decisiones que hubiera tomado el ingeniero de proceso. Por ejemplo, en la Figura 4.15 tenemos los siguientes datos:

	Punto de control	Medición
Intensidad de corriente	392 A	390 A
Voltaje	3.92 V	4.23 V
Resistencia	0.01 Ω	0.01 Ω

Tabla 4.16 Datos para control de proceso.

Donde la desviación de la resistencia fue:

$$\text{desviación} = \frac{0.0108 - 0.01}{0.01} \times 100 = 8\%.$$

Como la desviación fue superior al 5%, véase Tabla 3.6, entonces el programa de control presentó una ventana de diálogo para informar del problema al operador y sugerirle una acción correctiva.

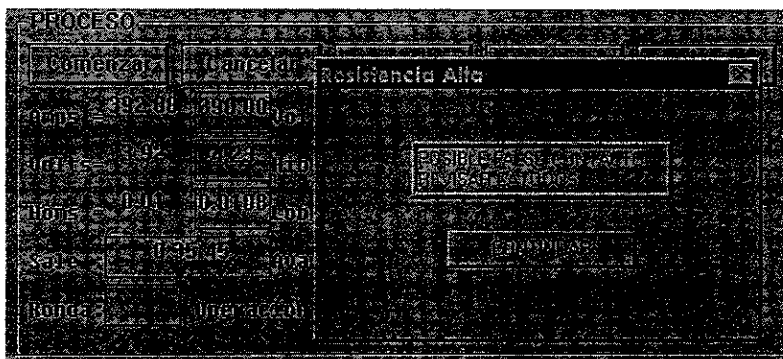


Figura 4.15 Control de proceso DOX.

CAPÍTULO 5

ANÁLISIS DE RESULTADOS

5.1 Cálculos

Generalmente el experto humano no utiliza las reglas de optimización de espacio para hacer una primera evaluación del número de rondas debido a que continuamente tiene que modificar este cálculo. Lo mismo sucede con el cálculo de variables de operación; es difícil que el experto recuerde exactamente los datos de una ronda que fue procesada tres meses antes. Aunque se podría construir una base de datos para consultar esta información, la base de datos tiene la desventaja de que no puede procesar esta información para generar nuevos conocimientos. Por ejemplo, si quisiéramos calcular el tiempo que se necesita para procesar todo este material podríamos utilizar una solución más eficiente que la propuesta por la ecuación:

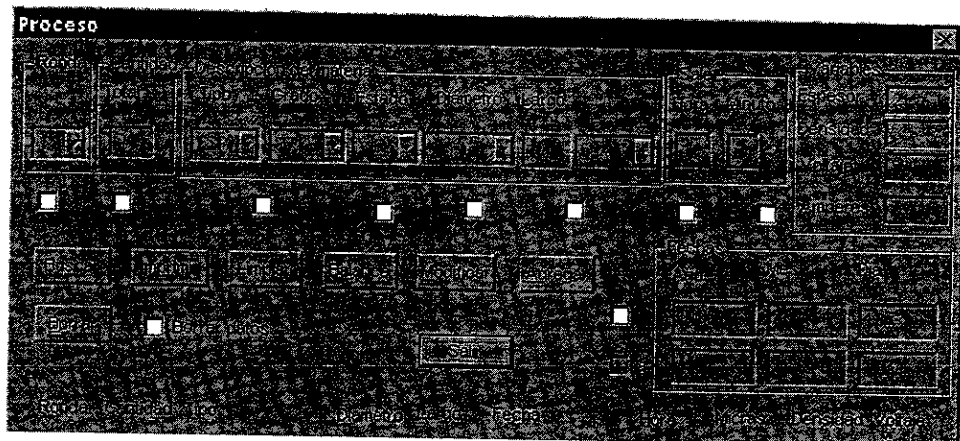
$$\text{Tiempo de producción} = N \text{ rondas} \times \frac{\text{turno}}{9 \text{ rondas}} \times \frac{\text{semanas}}{5 \text{ turno}}$$

Podríamos utilizar el cálculo de tiempos de proceso, Apéndice B, y el tiempo necesario para introducir y sacar una ronda (13'45'') para estimar el número de rondas por día. Esta sería una mejor aproximación, ya que una ronda tuercas de 2 ½ pulgadas puede durar en proceso 64 minutos, mientras que una ronda de espárragos de 2 ½ puede utilizar 28 minutos. La automatización del conocimiento del experto no solo liberaría al experto de una carga tediosa de trabajo sino que también le permitiría explorar nuevas alternativas para desarrollar sus programas de producción, estimar variables de operación y controlar el proceso.

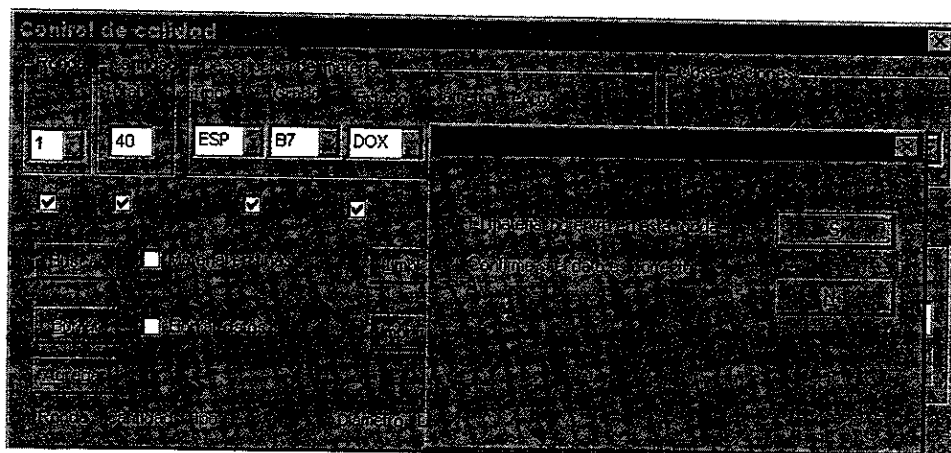
5.2 Hojas de control

En el desarrollo del modelo se propuso que los formatos de hojas de control pueden ser traducidos a estructuras proposicionales. La utilización de estas estructuras permite la construcción de programas de inventario³⁴ que pueden ser utilizados por técnicos para capturar su informe de departamento en computadora sin la supervisión del experto humano.

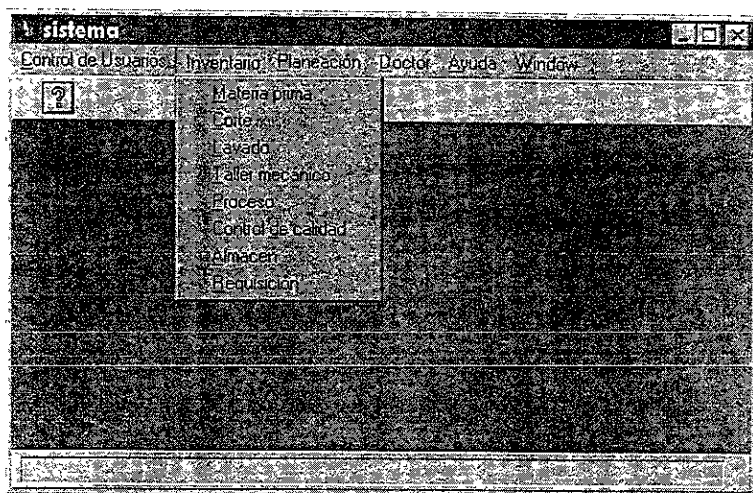
³⁴De hecho se desarrolló un programa de inventarios, pero no se consideró conveniente incluirlo en la explicación.



Utilizando reglas y balances de materia el sistema experto puede revisar la información proporcionada por el técnico y concluir si su información es correcta.



Esto permitiría construir un sistema experto con un servidor y computadoras colocadas en cada departamento conectadas al servidor que le permitiría al sistema recibir información del medio ambiente y permitirle tomar decisiones para controlar y optimizar el proceso.



5.3 Cálculo de variables de operación

Si comparamos los resultados con las reglas presentadas en el Capítulo 3 para cálculo de variables de operación vemos que los resultados concuerdan con lo esperado. Además en la última pregunta que se le hizo al intérprete lógico respecto a los espárragos de 1¼X9 el sistema respondió adecuadamente, la resistencia propuesta es similar a la de los espárragos de 9¼, aunque la medida no se encontraba en la base de conocimiento (proceso y control de calidad). Sin embargo, esta predicción tiene un pequeño error ya que los espárragos de 9 tienen una superficie total menor a los de 9¼ y consecuentemente su resistencia debería ser mayor a este valor de referencia. Es de esperarse que conforme se incremente la base de conocimientos la respuesta del sistema al medio ambiente sea mejor.

El ejemplo presentado sólo es una muestra de cómo se debe desarrollar el programa para cálculo de variables. Por ejemplo, el procedimiento no incluye la posibilidad de buscar una ronda parecida en caso de que la ronda deseada no esté en la base de conocimientos. Es evidente que si se hubiera utilizado la densidad de la ronda de espárragos de 9¼ para la ronda de espárragos de 9 la aproximación hubiera sido mejor. Otra situación que no se incorporó en este programa fue desarrollar ecuaciones o modelos de extrapolación estadístico específicas para cada tipo de arreglo (6, 8, 16, 24, ..., 400 piezas por ronda)

y tipo de material y pedir al procedimiento que identificara de qué tipo de arreglo se trata. De hecho una estrategia que se puede utilizar para reducir el tamaño de la base de conocimientos, y consecuentemente reducir el tiempo de búsqueda, es analizar la ronda antes de iniciar la búsqueda y definir un caso más general; por ejemplo, si tenemos una ronda con 38 piezas de un material A y dos piezas de un material B podríamos sugerirle al intérprete que busque la información para una ronda con 40 piezas del material A.

Un problema que se puede presentar con este algoritmo es el crecimiento de la base de conocimiento. Si se procesan mas de 2000 rondas al año, tomando en cuenta solo un turno, el programa puede perder mucho tiempo buscando una ronda que nunca se haya procesado. Aunque existen estrategias para enfrentar este problema no se incluyó en este trabajo porque dificultaría la comprensión del algoritmo para los lectores potenciales³⁵.

³⁵Con poco conocimiento de programación lógica.

CAPÍTULO 6

CONCLUSIONES

1. La construcción de un sistema experto para este proceso se justifica porque:
 - Por ser un proceso único en el mundo, la desaparición de un experto resultaría desastrosa para la empresa.
 - Debido a las políticas de la empresa el desarrollo de programas de producción solo puede ser resuelto utilizando modelos heurísticos. Esto no implica que la programación lineal y otras técnicas de modelos cuantitativos para administración no pueden ser empleadas, debido a que son conocimientos y como tales pueden ser incluidos en los sistemas expertos. Siempre será mejor utilizar un modelo cuantitativo cuando la situación lo permita.
 - Los cálculos manuales tienen un elevado costo debido a que implican una gran cantidad de horas-hombre para que, finalmente, se tengan que modificar por situaciones imprevistas.
 - La gran cantidad de información que se tiene que manejar puede generar errores, descuidos, olvidos u omisiones, normales en un experto humano.
 - Aunque la programación de la producción no es propiamente un problema metalúrgico, sólo el ingeniero de proceso puede definir el tipo de arreglos y las variables de operación. Al incorporar el conocimiento del coordinador de producción y del ingeniero de proceso se pueden desarrollar planes de producción que contemplen el aspecto metalúrgico.
 - La automatización del conocimiento permitirá la investigación de nuevas estrategias para desarrollar planes de producción. Actualmente la planta trabaja con una sola línea de producción; al incrementar el número de líneas³⁶ será necesario efectuar estudios de tiempos y movimientos que eviten los tiempos muertos; este problema sólo podrá ser resuelto utilizando técnicas de planeación propias de la inteligencia artificial.
 - La implementación de un sistema experto permitirá la utilización del material humano en la generación de nuevos conocimientos.
2. El sistema experto está basado fundamentalmente en reglas producto de la experiencia adquirida en planta y conocimientos básicos de electroquímica. El desarrollo de un modelo fenomenológico para predecir las variables de

³⁶ De hecho, la planta está instalando una nueva línea de producción.

operación puede ser muy interesante desde el punto de vista científico; sin embargo el costo y el tiempo de desarrollo de éste podría inhibir el interés de los principales interesados.

- 3 El modelo para cálculo de variables de operación puede responder satisfactoriamente al ambiente. Sin embargo, conforme aumente su conocimiento respecto al proceso mejorará sus predicciones. Al mejorar la estimación de la densidad de corriente será posible reducir los tiempos de producción sin riesgo de afectar la calidad del material.
4. Respecto al desarrollo del modelo, considero importante los ejemplos presentados para la traducción del conocimiento, fundamentalmente:
 - La traducción de las hojas de control a expresiones proposicionales.
 - La construcción de las reglas para calcular las piezas máximas por ronda a partir de la Tabla 3.2.
 - La traducción de las reglas de cálculo de variables de operación a procedimientos.
 - La utilización de reglas de inferencia para la construcción de un modelo de control inteligente.
5. La utilización de programación lógica para resolver este problema se justifica porque:
 - Evita la necesidad de diseñar algoritmos de búsqueda.
 - La programación lógica ha sido diseñada específicamente para trabajar con la lógica de predicados. A diferencia de otros lenguajes de programación no hay que darle una receta de cocina, un algoritmo que haga la tarea encomendada. En lugar de eso lo que se le da es la descripción del problema y Prolog entonces busca la solución. Esto reduce significativamente el número de líneas por programa. Es capaz de deducir soluciones y adquirir conocimientos. La descripción de nuevas situaciones no requiere la modificación de la estructura del programa; simplemente se agregan nuevos hechos a la base de datos o conocimientos.
 - La programación lógica puede manipular adecuadamente estructuras de datos complejas.
 - Por ser un problema de deducción lógica, la traducción de las reglas a este lenguaje de programación es relativamente sencilla.

RECOMENDACIONES

A continuación, presentamos sugerencias de posibles líneas de investigación para trabajos futuros:

- *Programación de producción.* El estudio de tiempos y movimientos para dos o más líneas de producción y su automatización utilizando programación lógica.
- *Proceso DOX.* El estudio de la termodinámica, cinética y formación de la microestructura podría permitir la optimización y simulación del proceso y además daría herramientas para desarrollar otras aleaciones utilizando electrólisis.
- *Sistemas expertos híbridos.* El programa desarrollado en este trabajo es un sistema basado en el conocimiento empírico, la combinación de este tipo de conocimiento con modelos fenomenológicos y estadísticos puede mejorar la calidad del sistema experto. La incorporación de un modelo fenomenológico para el cálculo de variables de operación y el diseño de ánodos permitiría mejores predicciones. En general, la construcción de sistemas expertos debe incluir modelos matemáticos cuando sea posible.
- Este trabajo se enfocó al estudio de la representación del conocimiento; aunque se utilizan, las otras ramas de la inteligencia artificial (estrategias de búsqueda, heurística, deducción, estrategias de razonamiento y planeación) no se estudian en detalle.

APÉNDICE A

Prolog

El lenguaje de programación más usado en los últimos años en la Inteligencia Artificial es Prolog. Aunque Lisp tiene sus adeptos, Prolog ha resultado muy poderoso en algunas ramas de la lógica, en el procesamiento del lenguaje natural y en los sistemas expertos (o basados en conocimientos). De esta manera, poco a poco en los cursos que se imparten de esta asignatura, Prolog se ha colocado cada vez más como el favorito natural.

Prolog (PROgramming in LOGic), trabaja con la que se denomina lógica de predicados. Esto se acerca de manera muy familiar a la lógica formal y es por eso que programar en este último lenguaje resulta parecido a simplemente especificar el problema. Esto quiere decir que, a diferencia de otros lenguajes de programación a Prolog no hay que darle una receta de cocina, un algoritmo que haga la tarea encomendada. En lugar de eso lo que se le da es la descripción del problema y Prolog entonces busca la solución. Para quienes no hayan programado jamás en un lenguaje como Prolog, suena absurdo que éste resuelva el problema sin haber descrito las rutinas que lo resuelven. En realidad Prolog ya tiene un mecanismo genérico de búsqueda de respuestas, llamado algoritmo de Robinson, el cual le da la potencia necesaria para que el sistema pueda funcionar en cierto tipo de problemas específicos. Así entonces, en Prolog uno plantea la descripción del problema y entonces le preguntamos al sistema sobre resultados específicos, los cuales el algoritmo mencionado encuentra haciendo inferencias lógicas.

En Prolog, podemos obtener soluciones por la deducción lógica de algo ya conocido con anterioridad. Comúnmente, un programa en Prolog no es una secuencia de acciones, es una colección de hechos y reglas para obtener conclusiones de estos hechos. Por esto, Prolog es conocido como un lenguaje declarativo. Prolog usa variaciones simplificadas de la sintaxis de la lógica de predicados porque esta da una sintaxis fácil de entender muy similar al lenguaje natural. Prolog incluye una máquina de inferencia, el cual es un proceso para razonar lógicamente la información. Prolog trata de inferir si una hipótesis es verdad (en otras palabras, responder una pregunta) por el cuestionamiento del conjunto de información alrededor del conocimiento a ser provado. El mundo de conocimiento de Prolog es el conjunto finito de hechos y reglas que son dados en un programa.

Una característica importante de Prolog es que, además de que encuentra lógicamente respuestas a preguntas, éste puede tratar con diferentes alternativas y encontrar todas las soluciones posibles en vez de una sola. En vez de simplemente proceder del principio del programa al final, Prolog puede regresar y mirar más de un camino para resolver cada parte del problema.

Sintaxis

El lenguaje de programación prolog se compone de:

- **Constantes.** Hay dos clases de constantes: *átomos* y *números*. Los *átomos* son secuencias de caracteres con las siguientes restricciones:
 1. Si el átomo se pone entre comillas simples "'", la secuencia puede incluir cualquier carácter.
 2. Si el átomo no se pone entre comillas, la secuencia debe iniciar con una letra minúscula y puede incluir letras no acentuadas, números y el carácter "_".

`a`, `ana`, `'Ana Hernández'`, `manuel_hernandez`, `x1` son ejemplos de átomos. `Ana`, `manuel hernandez`, `1x` no son nombres válidos de átomos. Los números se utilizan para realizar operaciones aritméticas y pueden ser enteros, negativos y de coma flotante.

- **Variables.** Es una secuencia de letras no acentuadas y números; la secuencia debe iniciar con una letra mayúscula o con el carácter `"_"`. `Ana`, `_ana` son ejemplos de variables.
- **Símbolo de predicado.** Se utiliza para construir predicados de la forma `predicado(t1,t2,...,tn)`, donde `ti` son términos. Tiene la misma notación que los átomos.
- **Operadores.** Se utilizan para representar funciones y permitimos utilizar una sintaxis que facilite la lectura de algunas estructuras. Por ejemplo, las operaciones aritméticas suelen escribirse comúnmente como operadores. Cuando escribimos la expresión aritmética `"x+y*z"` representamos la estructura `+(x,*(y,x))`. Algunos operadores importantes son los operadores lógicos y aritméticos.

Operador	Tipo de operador	Traducción ³⁷
+	aritmético	Suma (+)
-	aritmético	Resta (-)
*	aritmético	Multiplicación (')
/	aritmético	División (—)
=	aritmético	Igualdad (=)
\=	aritmético	desigualdad (≠)
<	aritmético	menor que (<)
>	aritmético	mayor que (>)
=<	aritmético	menor o igual (≤)
>=	aritmético	mayor o igual (≥)
,	lógico	conjunción (∧)
;	lógico	disyunción (∨)
:-	lógico	implicación inversa (←)
->	lógico	implicación (→)

Hechos

Son relaciones o predicados cuyos términos son constantes. Esta formado por un símbolo de predicado, que en este caso es el símbolo del predicado, seguido por una eneada de términos constantes separados por `","` y agrupados entre paréntesis. Al final del hecho se debe colocar un carácter `"."`. Según la estructura:

$$\text{predicado}(t_1, t_2, \dots, t_n) .$$

Reglas

Es un enunciado de la forma:

$$A :- B_1, B_2, \dots, B_n .$$

Donde `:-` representa al conectivo lógico \leftarrow y `","` a la conjunción (\wedge). Su traducción al lenguaje de cálculo de predicados sería:

$$A \leftarrow B_1 \wedge B_2 \wedge \dots \wedge B_n .$$

Donde `"."` indica la conclusión de la regla.

³⁷Algunos de los operadores utilizados por el lenguaje de programación no representan exactamente el significado de su equivalente matemático; se sugiere consultar la bibliografía correspondiente [15,16,21] para ampliar el conocimiento y uso de los operadores.

Predicados predefinidos

Igual que cualquier lenguaje común de programación, Prolog tiene un conjunto de instrucciones o predicados predefinidos que el usuario puede utilizar. Por ejemplo:

- **consult(~)**. Se utiliza para suministrar nuevos enunciados, los cuales están almacenados en archivos de texto; por ejemplo, el programa del Apéndice C.
- **write(~)**. Salida a pantalla. Despliega átomos; cuando escribimos una variable, write despliega en la pantalla el valor de la variable.
- **nl**. Línea nueva (salto de línea).
- **fail**. Es un predicado que siempre fracasa.

Preguntas

Prolog utiliza el operador `?-` para indicarle al usuario que está listo para recibir instrucciones. Por ejemplo, si queremos suministrarle al intérprete la siguiente base de conocimientos:

```
es_padre_de(teraj,abraham).
es_padre_de(teraj,najor).
```

Que está almacenada en el archivo `padre.pl`, podemos utilizar el predicado `consult(~)`.

```
?-consult(padre.pl).
```

El intérprete intenta cumplir el objetivo, si lo satisface contesta afirmativamente (`yes`) y si no lo puede satisfacer contesta negativamente (`no`). Debemos incluir un carácter `."` para indicarle al intérprete el final de las instrucciones o del predicado.

Una vez que hemos suministrado una base de conocimientos al intérprete, podemos hacer preguntas acerca de ellos. En Prolog, una pregunta se representa igual que un hecho puesto delante del operador `?-`. Por ejemplo, si preguntamos:

```
?-es_padre_de(teraj,abraham).
```

El intérprete contesta:

```
yes.
```

```
?-
```

Cuando el intérprete nos presenta el operador `?-` nos indica que nuevamente está listo para recibir instrucciones.

Aprendizaje

Una de las herramientas más poderosas de Prolog es su capacidad para modificar su base de conocimiento. Prolog agrega o borra hechos y reglas utilizando los predicados `asserta`, `assertz` y `retract`.

- `asserta(~)`. Se utiliza para agregar cláusulas a la base de conocimiento; al utilizarla, `asserta` agrega una cláusula al principio de la base de datos.
- `asserta(~)`. Tiene la misma aplicación que `asserta(~)` pero agrega la cláusula al final de la base de datos. Por ejemplo, si queremos agregar el hecho `es_padre_de(teraj,haran)` a la base de conocimientos del programa "padre.pl", le diremos al intérprete:

```
?-asserta(es_padre_de(teraj,haran)).
```

El hecho `es_padre_de(teraj,haran)` se coloca en la base de datos de la siguiente manera:

```
es_padre_de(teraj,haran).
```

```
es_padre_de(teraj,abraham).
```

```
es_padre_de(teraj,najor).
```

Si utilizamos la instrucción

```
?-assertz(es_padre_de(abraham,isaac)).
```

La cláusula se agrega de la siguiente manera:

```
es_padre_de(teraj,abraham).
```

```
es_padre_de(teraj,najor).
```

```
es_padre_de(teraj,haran).
```

```
es_padre_de(abraham,isaac).
```

- `retract(~)`. Permite a un programa quitar cláusulas de la base de datos.

Reevaluación

Un grupo de hechos puede proporcionar varias soluciones a una pregunta si utilizamos una o varias variables en la pregunta. Por ejemplo, si le decimos al intérprete:

```
?
```

El intérprete trata de cumplir el objetivo unificando el predicado `es_padre_de(teraj,X)` con alguno de los hechos que tiene en su base de conocimientos. Particularmente, Prolog inicia la búsqueda de soluciones en la cima de la base de conocimientos de manera tal el intérprete contesta:

```
X=abraham
```

El intérprete no contesta afirmativa o negativamente, ni presenta el operador `?-` porque todavía existen hechos que pueden satisfacer la pregunta. Si se

teclea carácter ";" después de cada respuesta Prolog nos dará la siguiente información:

```
X=abraham;
```

```
X=najor;
```

```
X=haran;
```

```
no
```

```
?-
```

Prolog contesta en este orden porque busca nuevas soluciones en orden descendente, de arriba hacia abajo.

Si queremos conocer todas las soluciones posibles a esta pregunta, podemos utilizar el siguiente predicado:

```
?-es_padre_de(teraj,X),write('X='),write(X),nl,fail.
```

La respuesta del intérprete será:

```
X=abraham
```

```
X=najor
```

```
X=haran
```

```
no
```

```
?-
```

Prolog contesta así porque todos los predicados atómicos previos a fail (`es_padre_de(teraj,X),write('X='),write(X),nl`) pero debido a que fail es un predicado que siempre fracasa entonces hace que el predicado `es_padre_de(teraj,X),write('X='),write(X),nl,fail` fracase y provoca que prolog reinicie la búsqueda de una solución con una nueva X. Como este predicado, conjunción, tiene un predicado atómico que siempre es falso se provoca un proceso de reevaluación que se detendrá solo hasta que se hayan revisado todas las alternativas posibles.

Una instrucción útil para el control de la reevaluación es el operador ! (Corte). Este operador le indica al intérprete cuales son las opciones previas que no hace falta que vuelva a considerar cuando pasa a reevaluar la cadena de objetivos. Un ejemplo de su aplicación sería el siguiente predicado.

```
?-!,es_padre_de(teraj,X),write('X='),write(X),nl,fail.
```

Cuya respuesta sería:

```
X=abraham
```

```
no
```

```
?-
```

El intérprete sólo muestra una solución porque el corte le indica que no hace falta que vuelva a reevaluar.

APÉNDICE B

Cálculo de variables de operación

Notación

- A es el área de la superficie de las piezas (cm^2)
- N es el número de piezas y a procesar
- M es el número de piezas diferentes
- j es la densidad de corriente ($\frac{A}{\text{cm}^2}$)
- i es el flujo de corriente (A)
- ρ es la densidad del recubrimiento ($\frac{g}{\text{cm}^3}$)
- t es el tiempo de proceso (s)
- \overline{PM} es el peso molecular promedio ($\frac{g}{\text{mol}}$)
- z es el equivalente mol
- V es el voltaje (volts)
- R es la resistencia del baño (ohm)

Datos

- Número y tipo de piezas a procesar.
- Espesor del recubrimiento (cm)
- j
- R

Cálculos

$$\text{Superficie}_{total} = \sum_{i=1}^m A_i \times N_i \quad (\text{cm}^2)$$

$$\text{Volumen}_{total} = \text{Superficie}_{total} \times \text{Espesor} \quad (\text{cm}^3)$$

$$i = j \times \text{Superficie}_{total} \quad (\text{A})$$

$$g_{Dox} = \text{Volumen}_{total} \times \rho \quad (\text{g})$$

$$t = \frac{g_{Dox} \times z \times F \times l}{PM} \quad (\text{s})$$

$$V = I \times R \quad (\text{V})$$

APÉNDICE C

Reglas para planeación de producción

```

requisicion_de_tuercas(N):-
    requisicion(lote(N,total),_,material(tuerca,dox),_,_),
    write('Error, ya se calculó la requisición de tuercas. '),nl,
    !.

requisicion_de_tuercas(N):-
    requisicion(lote(N,P),C,material(tuerca,E),M,F),
    sumatoria(lote(N,P),C,material(tuerca,E),M,F),
    fail.

requisicion_de_tuercas(_):-!.

sumatoria(lote(N,_),cantidad(C),material(T,E),M,F):-
    retract(requisicion(lote(N,total),cantidad(C_vieja),
    material(T,E),M,F)),
    C_nueva is C_vieja+C,
    asserta(requisicion(lote(N,total),cantidad(C_nueva),
    material(T,E),M,F)),
    !.

sumatoria(lote(N,_),cantidad(C),material(T,E),M,F):-
    C_nueva is C,
    asserta(requisicion(lote(N,total),cantidad(C_nueva),
    material(T,E),M,F)),
    !.

modificar_existencia(N):-
    requisicion(lote(N,P),cantidad(C_req),material(esparrago,dox),Dim,_),
    retract(existencia(almacen,lote(reserva,total),cantidad(C_inv),
    material(esparrago,dox),Dim)),
    modificacion(C_req,C_inv,lote(N,P),material(esparrago,dox),Dim,almacen),
    fail.

modificar_existencia(N):-
    requisicion_de_barras(N).

modificar_existencia(N):-
    requisicion(lote(N,total),cantidad(C_req),material(tuerca,dox),Dim,_),
    retract(existencia(almacen,lote(reserva,total),cantidad(C_inv),
    material(tuerca,dox),Dim)),
    modificacion(C_req,C_inv,lote(N,total),material(tuerca,dox),Dim,almacen),
    fail.

modificar_existencia(_):-!.

modificar_barras(N):-
    requisicion(lote(N,total),cantidad(C_req),material(barra,virgen),Dim,_),
    retract(existencia(materia_prima,lote(reserva,total),
    cantidad(C_inv),material(barra,virgen),Dim)),
    modificacion(C_req,C_inv,lote(N,total),material(barra,virgen),Dim,materia_prima),
    fail.

modificar_barras(_):-!.

modificar_tuercas(N):-
    requisicion(lote(N,total),cantidad(C_1),material(tuerca,dox),Dim,_),
    revisar_existencia(almacen,lote(N,total),cantidad(C_2),material(tuerca,dox),Dim),
    C_req is C_1-C_2,
    retract(existencia(materia_prima,lote(reserva,total),
    cantidad(C_inv),material(tuerca,virgen),Dim)),
    modificacion(C_req,C_inv,lote(N,total),material(tuerca,virgen),Dim,materia_prima),
    fail.

modificar_tuercas(_):-!.

modificacion(C_req,_,_,_,_):-

```



```

C_req=<0,!.

modificacion(C_req,C_inv,lote(N,P),M,Dim,Etapa):-
    C_req>C_inv,!,
    asserta(existencia(Etapa,lote(N,P),cantidad(C_inv),M,Dim)).

modificacion(C_req,C_inv,lote(N,P),M,Dim,Etapa):-
    asserta(existencia(Etapa,lote(N,P),cantidad(C_req),M,Dim)),
    C is C_inv-C_req,
    asserta(existencia(Etapa,lote(reserva,total),cantidad(C),M,Dim)).

orden_de_produccion(lote(N,total),cantidad(C),material(tuerca,dox),Dim,F):-
    requisicion(lote(N,total),cantidad(C_requisicion),material(tuerca,dox),Dim,F),
    revisar_existencia(almacen,lote(N,total),cantidad(C_almacen),material(tuerca,dox),Dim),
    C_requisicion>C_almacen,
    C is C_requisicion-C_almacen.

orden_de_produccion(lote(N,P),cantidad(C),material(esparrago,dox),Dim,F):-
    requisicion(lote(N,P),cantidad(C_requisicion),material(esparrago,dox),Dim,F),
    revisar_existencia(almacen,lote(N,P),cantidad(C_almacen),material(esparrago,dox),Dim),
    C_requisicion>C_almacen,
    C is C_requisicion-C_almacen.

revisar_existencia(Etapa,lote(N,P),C,M,Dim):-
    existencia(Etapa,lote(N,P),C,M,Dim),!.

revisar_existencia(_,_,cantidad(0),_,_).

requisicion_de_barras(N):-
    requisicion(lote(N,total),_,material(barra,virgen),_,_),
    write('Error, ya se calculó la requisición de barras. '),nl,!.

requisicion_de_barras(N):-
    orden_de_produccion(lote(N,P),cantidad(C),material(esparrago,dox),medidas(D,L),F),
    Esparragos_por_barra is integer(144 / (L+0.125)),
    Barras_por_partida is C//Esparragos_por_barra+1,
    sumatoria(lote(N,P),cantidad(Barras_por_partida),
    material(barra,virgen),medidas(D,144),F),
    fail.

solicitud_de_material(N,tuerca):-
    requisicion(lote(N,comprar),_,material(tuerca,virgen),_,_),
    write('Error, ya se calculo la solicitud de compra de Tuercas. '),nl,!.

solicitud_de_material(N,tuerca):-
    date(F_solicitud),
    orden_de_produccion(lote(N,total),cantidad(C_produccion),material(tuerca,dox),Dim,_),
    revisar_existencia(materia_prima,lote(N,total),
    cantidad(C_existencia),material(tuerca,virgen),Dim),
    C_produccion>C_existencia,
    C_solicitud is C_produccion-C_existencia,
    asserta(requisicion(lote(N,comprar),cantidad(C_solicitud),
    material(tuerca,virgen),Dim,F_solicitud)),
    fail.

solicitud_de_material(_,tuerca):-!.

solicitud_de_material(N,barra):-
    requisicion(lote(N,comprar),_,material(barra,virgen),_,_),
    write('Error, ya se calculó la solicitud de compra de barras. '),nl,
    !.

solicitud_de_material(N,barra):-
    date(F_solicitud),
    requisicion(lote(N,total),cantidad(C_requisicion),material(barra,virgen),Dim,_),
    revisar_existencia(materia_prima,lote(N,total),
    cantidad(C_existencia),material(barra,virgen),Dim),
    C_requisicion>C_existencia,
    C_solicitud is C_requisicion-C_existencia,
    asserta(requisicion(lote(N,comprar),cantidad(C_solicitud),
    material(barra,virgen),Dim,F_solicitud)),
    fail.

solicitud_de_material(_,barra):-!.

```

```

solicitud_de_material(N):-
    requisicion_de_tuercas(N),
    modificar_existencia(N),
    modificar_barras(N),
    modificar_tuercas(N),
    solicitud_de_material(N,tuerca),
    solicitud_de_material(N,barra),
    write('Lote '),write(N),nl,
    requisicion(lote(N,comprar),_,_,F_solicitud),
    write('Fecha de solicitud: '),write(F_solicitud),nl,
    write('Comprar: '),nl,
    write('Material          Diametro          Cantidad'),nl,
    desplegar(N,comprar).

desplegar(N,P):-
    requisicion(lote(N,P),cantidad(C),material(T,_),medidas(D,_),_),
    write(T),put(9),put(9),write(D),put(9),put(9),write(C),nl,fail.

desplegar(_,_) -.

/*piezas_maximas_por_ronda(Tipo,medidas(D,L),Cantidad)*/
piezas_maximas_por_ronda(esparrago,medidas(_,L),4):-L<=55,L>=35.
piezas_maximas_por_ronda(esparrago,medidas(_,L),6):-L<35,L>=25.
piezas_maximas_por_ronda(esparrago,medidas(_,L),8):-L<25,L>=19.5.
piezas_maximas_por_ronda(esparrago,medidas(D,L),16):-D=2.5,L<=19.5.
piezas_maximas_por_ronda(esparrago,medidas(D,L),16):-D=2.25,L<=19.5,L>=14.
piezas_maximas_por_ronda(esparrago,medidas(D,L),16):-D=2,L<=19.5,L>=16.
piezas_maximas_por_ronda(esparrago,medidas(D,L),24):-D=2,L<16.
piezas_maximas_por_ronda(esparrago,medidas(D,L),24):-D<2,D=1.625,L<=19.5.
piezas_maximas_por_ronda(esparrago,medidas(D,L),40):-D<1.625,L<=19.5,L>=10.
piezas_maximas_por_ronda(esparrago,medidas(D,L),80):-D<1.625,L<10,L>=6.75.
piezas_maximas_por_ronda(esparrago,medidas(D,L),120):-D<1.375,L<6.75,L>=4.75.
piezas_maximas_por_ronda(esparrago,medidas(D,L),160):-D<1.375,L<4.75,L>=3.75.
piezas_maximas_por_ronda(esparrago,medidas(D,L),240):-D<0.875,L<3.75.
piezas_maximas_por_ronda(tuerca,medidas(D,_),64):-D<=2.5,D>=1.75.
piezas_maximas_por_ronda(tuerca,medidas(D,_),144):-D<1.75,D>=1.375.
piezas_maximas_por_ronda(tuerca,medidas(D,_),208):-D<1.375,D>=1.
piezas_maximas_por_ronda(tuerca,medidas(D,_),400):-D<1,D>=0.5.

/*REGLAS PARA OPTIMIZACIÓN DE ESPACIO*/

descripcion_de_ronda(N,PL,[cantidad(6),cantidad(6)],material(esparrago,dox),
[medidas(D1,L1),medidas(D2,L2)]):-
    orden_de_produccion(lote(N,P1),cantidad(C1),material(esparrago,dox),
    medidas(D1,L1),_,miembro(P1,PL),
    piezas_maximas_por_ronda(esparrago,medidas(D1,L1),6),
    C1>=6,D1<=2.5,D1>=1.625,
    orden_de_produccion(lote(N,P2),cantidad(C2),material(esparrago,dox),
    medidas(D2,L2),_,miembro(P2,PL),
    piezas_maximas_por_ronda(esparrago,medidas(D2,L2),16),C2>=6.

descripcion_de_ronda(N,PL,[cantidad(6),cantidad(12)],material(esparrago,dox),
[medidas(D1,L1),medidas(D2,L2)]):-
    orden_de_produccion(lote(N,P1),cantidad(C1),material(esparrago,dox),
    medidas(D1,L1),_,miembro(P1,PL),
    piezas_maximas_por_ronda(esparrago,medidas(D1,L1),6),C1>=6,
    D1<=2.5,D1>=1.625,
    orden_de_produccion(lote(N,P2),cantidad(C2),material(esparrago,dox),
    medidas(D2,L2),_,miembro(P2,PL),
    piezas_maximas_por_ronda(esparrago,medidas(D2,L2),24),C2>=12.

descripcion_de_ronda(N,PL,[cantidad(8),cantidad(8)],material(esparrago,dox),
[medidas(D1,L1),medidas(D2,L2)]):-
    orden_de_produccion(lote(N,P1),cantidad(C1),material(esparrago,dox),
    medidas(D1,L1),_,miembro(P1,PL),
    piezas_maximas_por_ronda(esparrago,medidas(D1,L1),8),
    C1>=8,D1<=2.5,D1>=1.625,
    orden_de_produccion(lote(N,P2),cantidad(C2),material(esparrago,dox),
    medidas(D2,L2),_,miembro(P2,PL),
    piezas_maximas_por_ronda(esparrago,medidas(D2,L2),16),C2>=8.

descripcion_de_ronda(N,PL,[cantidad(8),cantidad(16)],material(esparrago,dox),
[medidas(D1,L1),medidas(D2,L2)]):-
    orden_de_produccion(lote(N,P1),cantidad(C1),material(esparrago,dox),
    medidas(D1,L1),_,miembro(P1,PL),

```

```

piezas_maximas_por_ronda(esparrago,medidas(D1,L1),8),
C1>=8,D1<=2.5,D1>=1.625,
orden_de_produccion(lote(N,P2),cantidad(C2),material(esparrago,dox),
medidas(D2,L2),_),miembro(P2,PL),
piezas_maximas_por_ronda(esparrago,medidas(D2,L2),24),C2>=16.

descripcion_de_ronda(N,PL,[cantidad(8),cantidad(32)],material(esparrago,dox),
[medidas(D1,L1),medidas(D2,L2)]) :-
orden_de_produccion(lote(N,P1),cantidad(C1),material(esparrago,dox),
medidas(D1,L1),_),miembro(P1,PL),
piezas_maximas_por_ronda(esparrago,medidas(D1,L1),8),
C1>=8,D1<=1.625,
orden_de_produccion(lote(N,P2),cantidad(C2),material(esparrago,dox),
medidas(D2,L2),_),miembro(P2,PL),
piezas_maximas_por_ronda(esparrago,medidas(D2,L2),40),C2>=32.

descripcion_de_ronda(N,PL,[cantidad(8),cantidad(72)],material(esparrago,dox),
[medidas(D1,L1),medidas(D2,L2)]) :-
orden_de_produccion(lote(N,P1),cantidad(C1),material(esparrago,dox),
medidas(D1,L1),_),miembro(P1,PL),
piezas_maximas_por_ronda(esparrago,medidas(D1,L1),8),
C1>=8,D1<=1.625,
orden_de_produccion(lote(N,P2),cantidad(C2),material(esparrago,dox),
medidas(D2,L2),_),miembro(P2,PL),
piezas_maximas_por_ronda(esparrago,medidas(D2,L2),80),
C2>=72.

descripcion_de_ronda(N,PL,[cantidad(40),cantidad(40)],material(esparrago,dox),
[medidas(D1,L1),medidas(D2,L2)]) :-
orden_de_produccion(lote(N,P1),cantidad(C1),material(esparrago,dox),
medidas(D1,L1),_),miembro(P1,PL),
piezas_maximas_por_ronda(esparrago,medidas(D1,L1),40),
C1=40,L1<=13.5,
orden_de_produccion(lote(N,P2),cantidad(C2),material(esparrago,dox),
medidas(D2,L2),_),miembro(P2,PL),
piezas_maximas_por_ronda(esparrago,medidas(D2,L2),80),
C2>=40,L2+L1<=19.5.

descripcion_de_ronda(N,PL,[cantidad(40),cantidad(80)],material(esparrago,dox),
[medidas(D1,L1),medidas(D2,L2)]) :-
orden_de_produccion(lote(N,P1),cantidad(C1),material(esparrago,dox),
medidas(D1,L1),_),miembro(P1,PL),
piezas_maximas_por_ronda(esparrago,medidas(D1,L1),80),C1>=40,
orden_de_produccion(lote(N,P2),cantidad(C2),material(esparrago,dox),
medidas(D2,L2),_),miembro(P2,PL),
piezas_maximas_por_ronda(esparrago,medidas(D2,L2),120),
C2>=80,2*L2+L1<=19.

/* TERMINAN REGLAS PARA OPTIMIZACIÓN DE ESPACIO */

descripcion_de_ronda(N,PL,[cantidad(C)],material(T,dox),[medidas(D,L)]) :-
orden_de_produccion(lote(N,P),cantidad(C1),material(T,dox),medidas(D,L),_),
miembro(P,PL),
piezas_maximas_por_ronda(T,medidas(D,L),C),C1>=C.

descripcion_de_ronda(N,PL,X,material(T,E),Z) :-
orden_de_produccion(lote(N,P),cantidad(C),material(T,E),
medidas(D,L),_),miembro(P,PL),
piezas_maximas_por_ronda(T,medidas(D,L),C_max),C<C_max,
ronda(N,PL,[cantidad(C)],material(T,E),[medidas(D,L)],C_max,C,X,Z).

ronda(N,PL,Cs,material(T,E),Ds,C_max,Sum1,X,Z) :-
C_max>Sum1,
orden_de_produccion(lote(N,P),cantidad(C),material(T,E),
medidas(D,L),_),miembro(P,PL),
piezas_maximas_por_ronda(T,medidas(D,L),C_max),C<C_max,
not(member(medidas(D,L),Ds)),
acumulacion(Sum2,Sum1,C_max,C,C_ad),
ronda(N,PL,[cantidad(C_ad)|Cs],material(T,E),[medidas(D,L)|Ds],C_max,Sum2,X,Z).

ronda(_,_,Cs,_,Ds,C_max,C_max,Cs,Ds).

miembro(_,todas):-!.
miembro(P,PL):-!,member(P,PL).

```

```

acumulacion(C_acum2,C_acum1,C_max,C,N):-
    C_acum2 is C_acum1+C,
    C_acum2=<C_max,!N is C.

acumulacion(C_max,C_acum1,C_max,_,N2):-!,N2 is C_max-C_acum1.

variables_de_operacion(A,B,C,Res,Den):-
    proceso(R,_,_,_,datos(Res,Den1),F,_),
    control_de_calidad(R,_,_,_,con_defectos,F),!,
    Den is Den1-0.001.

variables_de_operacion(A,B,C,Res,Den):-
    proceso(R,_,_,_,datos(Res,Den1),F,_),
    control_de_calidad(R,_,_,_,sin_defectos,F),!,
    Den is Den1+0.002.

variables_de_operacion(A,material(T,_),C,Res,0.022):-
    !,superficie_total(A,material(T,_),C,0,Area),
    log(Area,Ln),
    Prod is -0.3902*Ln,
    exp(Prod,Exp),
    Res is 0.4265*Exp.

superficie_total([],_,[],Sum,Sum).

superficie_total([cantidad(C)|Cs],material(esparrago,_),[medidas(D,L)|Ms],
    Sum,Area):-
    Sum1 is C*(L*D*3.1416*2.54*2.54+2*D*D/4*3.1416*2.54*2.54)+Sum,
    superficie_total(Cs,material(esparrago,_),Ms,Sum1,Area).

programar_rondas_de_esparragos(N,PL):-
    programar_rondas_de_esparragos(N,PL,ronda(N,0)).

programar_rondas_de_esparragos(N,PL,ronda(N,No)):-
    N1 is No+1,
    descripcion_de_ronda(N,PL,A,B,C),
    assertz(ronda_programada(ronda(N,N1),A,B,C)),
    simular_existencia(N,A,B,C),
    programar_rondas_de_esparragos(N,PL,ronda(N,N1)).

programar_rondas_de_esparragos(N,PL,ronda(N,No)):-
    !,write('Lote:'),write(N),nl,
    write('Partidas:'),write(PL),nl,
    write('No. de rondas:'),write(No),nl,
    T is integer(No/45)+1,
    write('Tiempo estimado:'),write(T),write(' semanas'),nl,
    tell(rondas),listing(ronda_programada/4),told.

simular_existencia(_,[],_,[]):-!.

simular_existencia(N,[cantidad(C1)|Cs],M,[Dim|Dims]):-
    revisar_existencia(almacen,lote(N,P),cantidad(C0),M,Dim),
    {(C0>0)->retract(existencia(almacen,lote(N,P),cantidad(C0),M,Dim));true},
    C2 is C1+C0,
    asserta(existencia(almacen,lote(N,P),cantidad(C2),M,Dim)),
    simular_existencia(N,Cs,M,Dims).

material_a_procesar(N):-
    nl,
    write('          Material a procesar'),nl,nl,
    write('Diametro          Tuercas          Partidas'),nl,
    material_a_procesar(N,PL,TcaL),
    presentar_partidas_procesar(N,PL,TcaL),nl,nl,
    write('          Partidas para empaque'),nl,nl,
    write('Diametro          Partidas'),nl,
    presentar_partidas_armar(N,PL,TcaL).

presentar_partidas_procesar(,_,[],):-!.

presentar_partidas_procesar(N,PL,[preparar(tuerca,D,C)|TcaLs]):-
    write(D),put(9),put(9),write(C),
    put(9),put(9),partidas(N,D,PL),put(8),put(32),nl,
    presentar_resultados(N,PL,TcaLs).

partidas(N,D,PL):-
    orden_de_produccion(lote(N,P),_,material(esparrago,dox),medidas(D,_,_),

```

```

member(P, PL), write(P), put(44), fail.

partidas(_,_,_):-!.

presentar_partidas_armar(,_,{):-!.

presentar_partidas_armar(N, PL, [preparar(tuerca, D, _) | TcaLs]):-
    write(D), put(9), put(9),
    partidas_armar(N, D, PL), put(8), put(32), nl,
    presentar_partidas_armar(N, PL, TcaLs).

partidas_armar(N, D, PL):-
    requisicion(lote(N, P), _, material(esparrago, dox), medidas(D, _, _),
    member(P, PL), armar(N, P),
    write(P), put(44), fail.

partidas_armar(,_,_):-!.

material_a_procesar(N, PL, TcaL):-!,
    material_a_procesar(N, 4, [], [], PL, TcaL).

material_a_procesar(N, D, Ps, Ts, PL, TcaL):-
    D1 is D-0.125, D1>=0.375,
    seleccionar(N, [], 0, 0, T_acum, D1, P),
    append(P, Ps, Ps1),
    T_acum>0,

revisar_existencia(almacen, lote(N, total), Cantidad(T_al), material(tuerca, dox), medidas(D1, _)),
    Tuerca_a_procesar(T_acum, T_al, T_prod),
    material_a_procesar(N, D1, Ps1, [preparar(tuerca, D1, T_prod) | Ts], PL, TcaL).

material_a_procesar(N, D, Ps, Ts, PL, TcaL):-
    D1 is D-0.125, D1>=0.375,
    material_a_procesar(N, D1, Ps, Ts, PL, TcaL).

material_a_procesar(,_,_ , Ps, Ts, Ps, Ts):-!.

tuercas_a_procesar(A, B, C):-A>B, !, C is A-B.
tuercas_a_procesar(,_,_ , 0):-!.

seleccionar(N, Ps, B_acum1, T_acum1, T, D, PL):-
    requisicion(lote(N, P), cantidad(C), material(esparrago, dox), medidas(D, _, _),
    nonmember(P, Ps), armar(N, P),
    revisar_existencia(almacen, lote(N, total), cantidad(C_al_tca),
    material(tuerca, dox), medidas(D, _)),
    revisar_existencia(materia_prima, lote(N, total), cantidad(C_mp_tca),
    material(tuerca, virgen), medidas(D, _)),
    C_total_tca is C_al_tca+C_mp_tca,
    T_acum2 is T_acum1+C*2,
    C_total_tca>=T_acum2,
    seleccionar(N, [P|Ps], B_acum1, T_acum2, T, D, PL).

seleccionar(N, Ps, B_acum1, T_acum1, T, D, PL):-
    orden_de_produccion(lote(N, P), cantidad(C), material(esparrago, dox), medidas(D, L, _),
    nonmember(P, Ps),
    Esparragos_por_barra is integer(144 / (L+0.125)),
    Barras_por_partida is C//Esparragos_por_barra+1,
    B_acum2 is B_acum1+Barras_por_partida,
    rRevisar_existencia(materia_prima, lote(N, total),
    cantidad(C_barras_existencia), material(barra, virgen), medidas(D, _)),
    C_barras_existencia>=B_acum2,
    requisicion(lote(N, P), cantidad(C_esp_partida), material(esparrago, dox), _, _),
    revisar_existencia(almacen, lote(N, total),
    cantidad(C_al_tca), material(tuerca, dox), medidas(D, _)),
    revisar_existencia(materia_prima, lote(N, total),
    cantidad(C_mp_tca), material(tuerca, virgen), medidas(D, _)),
    C_total_tca is C_al_tca+C_mp_tca,
    T_acum2 is T_acum1+C_esp_partida*2,
    C_total_tca>=T_acum2,
    seleccionar(N, [P|Ps], B_acum2, T_acum2, T, D, PL).

seleccionar(,_ , Ps, _, T_acum, T_acum, _, Ps):-!.

armar(N, P):-orden_de_produccion(lote(N, P), _, material(esparrago, dox), _, _), !, fail.
armar(,_):-!.

```

APÉNDICE E

Hechos para planeación de producción

*/*Inventario de material, archivo almacen*/*

existencia(almacen,lote(reserva,total),cantidad(5000),material(tuerca,dox),medidas(0 375,0))
 existencia(almacen,lote(reserva,total),cantidad(2870),material(tuerca,dox),medidas(0 5,0)).
 existencia(almacen,lote(reserva,total),cantidad(1640),material(tuerca,dox),medidas(0.625,0))
 existencia(almacen,lote(reserva,total),cantidad(450),material(tuerca,dox),medidas(0 75,0))
 existencia(almacen,lote(reserva,total),cantidad(4847),material(tuerca,dox),medidas(1,0)).
 existencia(almacen,lote(reserva,total),cantidad(1356),material(tuerca,dox),medidas(1.125,0)).
 existencia(almacen,lote(reserva,total),cantidad(1315),material(tuerca,dox),medidas(1.25,0))
 existencia(almacen,lote(reserva,total),cantidad(124),material(tuerca,dox),medidas(1.375,0)).
 existencia(almacen,lote(reserva,total),cantidad(141),material(tuerca,dox),medidas(1.5,0)).
 existencia(almacen,lote(reserva,total),cantidad(1636),material(tuerca,dox),medidas(1 625,0))
 existencia(almacen,lote(reserva,total),cantidad(600),material(tuerca,dox),medidas(1.75,0)).
 existencia(almacen,lote(reserva,total),cantidad(445),material(tuerca,dox),medidas(1.875,0))
 existencia(almacen,lote(reserva,total),cantidad(18),material(tuerca,dox),medidas(2,0)).
 existencia(almacen,lote(reserva,total),cantidad(386),material(tuerca,dox),medidas(2.25,0)).
 existencia(almacen,lote(reserva,total),cantidad(103),material(tuerca,dox),medidas(2.5,0)).
 existencia(almacen,lote(reserva,total),cantidad(96),material(esparrago,dox),medidas(0.375,5)).
 existencia(almacen,lote(reserva,total),cantidad(64),material(esparrago,dox),medidas(0 375,9 5)).
 existencia(almacen,lote(reserva,total),cantidad(179),material(esparrago,dox),medidas(0.5,2 5))
 existencia(almacen,lote(reserva,total),cantidad(450),material(esparrago,dox),medidas(0 5,3)).
 existencia(almacen,lote(reserva,total),cantidad(66),material(esparrago,dox),medidas(0.5,3.25))
 existencia(almacen,lote(reserva,total),cantidad(108),material(esparrago,dox),medidas(0 5,3.5)).
 existencia(almacen,lote(reserva,total),cantidad(185),material(esparrago,dox),medidas(0.5,4)).
 existencia(almacen,lote(reserva,total),cantidad(7),material(esparrago,dox),medidas(0.5,7))
 existencia(almacen,lote(reserva,total),cantidad(1543),material(esparrago,dox),medidas(0.625,3 25))
 existencia(almacen,lote(reserva,total),cantidad(4038),material(esparrago,dox),medidas(0.625,4)).
 existencia(almacen,lote(reserva,total),cantidad(54),material(esparrago,dox),medidas(0 625,4.25))
 existencia(almacen,lote(reserva,total),cantidad(86),material(esparrago,dox),medidas(0 625,4 5)).
 existencia(almacen,lote(reserva,total),cantidad(14),material(esparrago,dox),medidas(0.625,7)).
 existencia(almacen,lote(reserva,total),cantidad(5),material(esparrago,dox),medidas(0.625,9)).
 existencia(almacen,lote(reserva,total),cantidad(45),material(esparrago,dox),medidas(0 625,9 5)).
 existencia(almacen,lote(reserva,total),cantidad(24),material(esparrago,dox),medidas(0 625,15)).
 existencia(almacen,lote(reserva,total),cantidad(14),material(esparrago,dox),medidas(0.75,2 5)).
 existencia(almacen,lote(reserva,total),cantidad(9),material(esparrago,dox),medidas(0.75,3))
 existencia(almacen,lote(reserva,total),cantidad(700),material(esparrago,dox),medidas(0.75,4 5)).
 existencia(almacen,lote(reserva,total),cantidad(200),material(esparrago,dox),medidas(0.75,4 75))
 existencia(almacen,lote(reserva,total),cantidad(99),material(esparrago,dox),medidas(0 75,5 5)).
 existencia(almacen,lote(reserva,total),cantidad(450),material(esparrago,dox),medidas(0.75,6)).
 existencia(almacen,lote(reserva,total),cantidad(130),material(esparrago,dox),medidas(0.75,6 5))
 existencia(almacen,lote(reserva,total),cantidad(2),material(esparrago,dox),medidas(0 75,7 5)).
 existencia(almacen,lote(reserva,total),cantidad(228),material(esparrago,dox),medidas(0 875,5))

existencia(almacen,lote(reserva,total),cantidad(127),material(esparrago,dox),medidas(0 875,6))
 existencia(almacen,lote(reserva,total),cantidad(113),material(esparrago,dox),medidas(0.875,6 5)).
 existencia(almacen,lote(reserva,total),cantidad(47),material(esparrago,dox),medidas(0.875,8)).
 existencia(almacen,lote(reserva,total),cantidad(74),material(esparrago,dox),medidas(1,5,5))
 existencia(almacen,lote(reserva,total),cantidad(10),material(esparrago,dox),medidas(1,6,5))
 existencia(almacen,lote(reserva,total),cantidad(157),material(esparrago,dox),medidas(1,7))
 existencia(almacen,lote(reserva,total),cantidad(17),material(esparrago,dox),medidas(1,7 5)).
 existencia(almacen,lote(reserva,total),cantidad(24),material(esparrago,dox),medidas(1,13))
 existencia(almacen,lote(reserva,total),cantidad(5),material(esparrago,dox),medidas(1.125,5)).
 existencia(almacen,lote(reserva,total),cantidad(14),material(esparrago,dox),medidas(1.125,6))
 existencia(almacen,lote(reserva,total),cantidad(15),material(esparrago,dox),medidas(1.125,7))
 existencia(almacen,lote(reserva,total),cantidad(404),material(esparrago,dox),medidas(1 125,7 5)).
 existencia(almacen,lote(reserva,total),cantidad(80),material(esparrago,dox),medidas(1.125,8)).
 existencia(almacen,lote(reserva,total),cantidad(28),material(esparrago,dox),medidas(1.125,9,5)).
 existencia(almacen,lote(reserva,total),cantidad(26),material(esparrago,dox),medidas(1 125,12 5)).
 existencia(almacen,lote(reserva,total),cantidad(30),material(esparrago,dox),medidas(1 125,14))
 existencia(almacen,lote(reserva,total),cantidad(143),material(esparrago,dox),medidas(1 125,21))
 existencia(almacen,lote(reserva,total),cantidad(40),material(esparrago,dox),medidas(1 25,7))
 existencia(almacen,lote(reserva,total),cantidad(127),material(esparrago,dox),medidas(1.25,7,5))
 existencia(almacen,lote(reserva,total),cantidad(29),material(esparrago,dox),medidas(1 25,8)).
 existencia(almacen,lote(reserva,total),cantidad(113),material(esparrago,dox),medidas(1.25,9)).
 existencia(almacen,lote(reserva,total),cantidad(10),material(esparrago,dox),medidas(1 25,9 5)).
 existencia(almacen,lote(reserva,total),cantidad(11),material(esparrago,dox),medidas(1 25,11)).
 existencia(almacen,lote(reserva,total),cantidad(41),material(esparrago,dox),medidas(1 25,14)).
 existencia(almacen,lote(reserva,total),cantidad(18),material(esparrago,dox),medidas(1 25,15)).
 existencia(almacen,lote(reserva,total),cantidad(37),material(esparrago,dox),medidas(1 25,18)).
 existencia(almacen,lote(reserva,52),cantidad(10),material(esparrago,dox),medidas(1 25,18,5)).
 existencia(almacen,lote(reserva,53),cantidad(41),material(esparrago,dox),medidas(1.25,19,5)).
 existencia(almacen,lote(reserva,54),cantidad(285),material(esparrago,dox),medidas(1.25,20)).
 existencia(almacen,lote(reserva,55),cantidad(48),material(esparrago,dox),medidas(1.25,48))
 existencia(almacen,lote(reserva,56),cantidad(34),material(esparrago,dox),medidas(1.375,9,5))
 existencia(almacen,lote(reserva,57),cantidad(24),material(esparrago,dox),medidas(1.375,10)).
 existencia(almacen,lote(reserva,58),cantidad(5),material(esparrago,dox),medidas(1.375,10,5))
 existencia(almacen,lote(reserva,59),cantidad(69),material(esparrago,dox),medidas(1.375,11)).
 existencia(almacen,lote(reserva,60),cantidad(8),material(esparrago,dox),medidas(1,5,9)).
 existencia(almacen,lote(reserva,61),cantidad(33),material(esparrago,dox),medidas(1 5,9,5))
 existencia(almacen,lote(reserva,62),cantidad(52),material(esparrago,dox),medidas(1 5,10))
 existencia(almacen,lote(reserva,63),cantidad(41),material(esparrago,dox),medidas(1 5,10,5)).
 existencia(almacen,lote(reserva,64),cantidad(22),material(esparrago,dox),medidas(1 5,11)).
 existencia(almacen,lote(reserva,65),cantidad(15),material(esparrago,dox),medidas(1 5,11,5)).
 existencia(almacen,lote(reserva,66),cantidad(5),material(esparrago,dox),medidas(1,5,12)).
 existencia(almacen,lote(reserva,67),cantidad(58),material(esparrago,dox),medidas(1 5,16))
 existencia(almacen,lote(reserva,68),cantidad(1),material(esparrago,dox),medidas(1 5,17))
 existencia(almacen,lote(reserva,69),cantidad(18),material(esparrago,dox),medidas(1,5,18 5)).
 existencia(almacen,lote(reserva,70),cantidad(10),material(esparrago,dox),medidas(1,5,22)).
 existencia(almacen,lote(reserva,71),cantidad(1),material(esparrago,dox),medidas(1 5,23)).
 existencia(almacen,lote(reserva,72),cantidad(14),material(esparrago,dox),medidas(1,5,14)).

existencia(almacen,lote(reserva,73),cantidad(9),material(esparrago,dox),medidas(1.625,9.25))
 existencia(almacen,lote(reserva,74),cantidad(90),material(esparrago,dox),medidas(1.625,10.5))
 existencia(almacen,lote(reserva,75),cantidad(46),material(esparrago,dox),medidas(1.625,11)).
 existencia(almacen,lote(reserva,76),cantidad(6),material(esparrago,dox),medidas(1.625,11.5))
 existencia(almacen,lote(reserva,77),cantidad(23),material(esparrago,dox),medidas(1.625,12))
 existencia(almacen,lote(reserva,78),cantidad(48),material(esparrago,dox),medidas(1.625,13.5)).
 existencia(almacen,lote(reserva,79),cantidad(20),material(esparrago,dox),medidas(1.625,14.5)).
 existencia(almacen,lote(reserva,80),cantidad(9),material(esparrago,dox),medidas(1.625,18.5))
 existencia(almacen,lote(reserva,81),cantidad(28),material(esparrago,dox),medidas(1.625,20.5))
 existencia(almacen,lote(reserva,82),cantidad(20),material(esparrago,dox),medidas(1.625,21.5))
 existencia(almacen,lote(reserva,83),cantidad(18),material(esparrago,dox),medidas(1.875,13)).
 existencia(almacen,lote(reserva,84),cantidad(30),material(esparrago,dox),medidas(1.875,13.5)).
 existencia(almacen,lote(reserva,85),cantidad(11),material(esparrago,dox),medidas(1.875,14))
 existencia(almacen,lote(reserva,86),cantidad(47),material(esparrago,dox),medidas(1.875,14.5)).
 existencia(almacen,lote(reserva,87),cantidad(40),material(esparrago,dox),medidas(1.875,15)).
 existencia(almacen,lote(reserva,88),cantidad(5),material(esparrago,dox),medidas(1.875,15.5)).
 existencia(almacen,lote(reserva,89),cantidad(1),material(esparrago,dox),medidas(1.875,16))
 existencia(almacen,lote(reserva,90),cantidad(15),material(esparrago,dox),medidas(1.875,21)).
 existencia(almacen,lote(reserva,91),cantidad(2),material(esparrago,dox),medidas(2,14.5))
 existencia(almacen,lote(reserva,92),cantidad(27),material(esparrago,dox),medidas(2,14.75)).
 existencia(almacen,lote(reserva,93),cantidad(23),material(esparrago,dox),medidas(2,20.5))
 existencia(almacen,lote(reserva,94),cantidad(37),material(esparrago,dox),medidas(2.5,16))
 existencia(almacen,lote(reserva,95),cantidad(25),material(esparrago,dox),medidas(2.5,16.4)).
 existencia(materia_prima,lote(reserva,1),cantidad(6),material(barra,virgen),medidas(0.375,144)).
 existencia(materia_prima,lote(reserva,2),cantidad(160),material(barra,virgen),medidas(0.500,144)).
 existencia(materia_prima,lote(reserva,3),cantidad(334),material(barra,virgen),medidas(0.625,144)).
 existencia(materia_prima,lote(reserva,4),cantidad(314),material(barra,virgen),medidas(0.750,144)).
 existencia(materia_prima,lote(reserva,5),cantidad(112),material(barra,virgen),medidas(0.875,144)).
 existencia(materia_prima,lote(reserva,6),cantidad(101),material(barra,virgen),medidas(1.000,144))
 existencia(materia_prima,lote(reserva,7),cantidad(118),material(barra,virgen),medidas(1.125,144))
 existencia(materia_prima,lote(reserva,8),cantidad(145),material(barra,virgen),medidas(1.250,144)).
 existencia(materia_prima,lote(reserva,9),cantidad(48),material(barra,virgen),medidas(1.375,144)).
 existencia(materia_prima,lote(reserva,10),cantidad(65),material(barra,virgen),medidas(1.500,144)).
 existencia(materia_prima,lote(reserva,11),cantidad(9),material(barra,virgen),medidas(1.625,144)).
 existencia(materia_prima,lote(reserva,13),cantidad(34),material(barra,virgen),medidas(1.875,144)).
 existencia(materia_prima,lote(reserva,14),cantidad(23),material(barra,virgen),medidas(2.000,144)).
 existencia(materia_prima,lote(reserva,15),cantidad(10),material(barra,virgen),medidas(2.250,144)).
 existencia(materia_prima,lote(reserva,16),cantidad(1),material(barra,virgen),medidas(2.500,144)).
 existencia(materia_prima,lote(reserva,17),cantidad(35),material(barra,virgen),medidas(3.000,144)).
 existencia(materia_prima,lote(reserva,19),cantidad(10),material(barra,virgen),medidas(3.500,144)).
 existencia(materia_prima,lote(reserva,5),cantidad(2600),material(tuerca,virgen),medidas(0.875,0)).
 existencia(materia_prima,lote(reserva,7),cantidad(3500),material(tuerca,virgen),medidas(1.125,0)).
 existencia(materia_prima,lote(reserva,8),cantidad(6750),material(tuerca,virgen),medidas(1.25,0)).
 existencia(materia_prima,lote(reserva,9),cantidad(1060),material(tuerca,virgen),medidas(1.375,0)).
 existencia(materia_prima,lote(reserva,10),cantidad(6900),material(tuerca,virgen),medidas(1.5,0)).
 existencia(materia_prima,lote(reserva,11),cantidad(120),material(tuerca,virgen),medidas(1.625,0))
 existencia(materia_prima,lote(reserva,12),cantidad(570),material(tuerca,virgen),medidas(1.75,0))

BIBLIOGRAFÍA

1. S. Russel y P. Norving, *Inteligencia Artificial*, Prentice Hall, México, 1996.
2. S.L Jamsa y A.J. Niemi editores, *Expert Systems in Mineral and Metal Processing*, Pergamon Press, Reino Unido, 1992.
3. K. Otsuka y otros, *A Hybrid Expert System Combined with Mathematical Model for Blast Furnace Operation*, ISIJ International, Vol. 30 (1990), No. 2, pp. 119-127.
4. S.A. Argyropoulos. *Artificial Intelligence in Materials Processing Operations: a Review and Future Directions*, ISIJ International, Vol. 30 (1990), No. 2, pp. 83-89.
5. D. Merrit, *Building Expert Systems in Prolog*; Springer-Verlag, Nueva York, 1989.
6. A.B. Badiru, *Expert Systems Applications in Engineering and Manufacturing*, Prentice Hall, Nueva Jersey, 1992.
7. G.F. Luger y W. A. Stubblefield, *Artificial Intelligence*, Benjamin Cummings Publishing Company, Redwood City, 1993.
8. H.M. Deitel y P.J. Dietel, *Como programar en C/C++*, Prentice Hall, México, 1995.
9. D. J. Kruglinski, *Programación avanzada con Microsoft Visual C++*, McGraw Hill, México, 1997.
10. G. Voss, *Programación orientada a objetos*, McGraw Hill, México, 1994.
11. L. Joyanes Aguilar, *Programación orientada a objetos*, McGraw Hill, México, 1996.
12. C. Petzold, *Programación en Windows 95*, McGraw Hill, México, 1996.
13. J. Richter, *Programación avanzada en Windows*, McGraw Hill, México, 1997.
14. A.V. Aho, J.E. Hopcroft y J.D. Ullman, *Estructura de datos y algoritmos*, SITEA, México, 1988.
15. L. Sterling y E. Shapiro, *The Art of Prolog*, The MIT Press, Estados Unidos de América, 1991.
16. W.F. Clocksin y C.S. Mellish, *Programación en Prolog*, Gustavo Gili, Barcelona, 1987.

17. Jean E. Rubin, *Mathematical Logic: Applications and Theory*, Saunders College Publishing, Estados Unidos de América, 1990.
18. C. J. Hogger, *Essentials of Logic Programming*, Claredon Press, Reino Unido, 1990.
19. M. R. Genesereth y N. J. Nilsson, *Logical Foundations of Artificial Intelligence*, Morgan Kaufmann Publishers, Estados Unidos de América, 1987.
20. K. Doets, *From Logic to Logic Programming*, The MIT Press, Estados Unidos de América, 1994.
21. M.A. Covington, D. Nute y A. Vellino, *Prolog Programming in Depth*, Scoot, Foresman and Company, Estados Unidos de América, 1988.
22. G.W. Castellán, *Fisicoquímica*, SITESA, México, 1987.
23. C.H. Hamann, A. Hamnett y W. Vielstich, *Electrochemistry*, Wiley-VCH, República Federal de Alemania, 1998.
24. A.L. Hines, y R.N. Maddux, *Transferencia de Masa*, Prentice Hall, México, 1987.
25. G. Prentice, *Electrochemical Engineering Principles*, Prentice Hall, Nueva Jersey, 1992.