



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE QUÍMICA

**SISTEMA EXPERTO HÍBRIDO BASADO EN
BALANCES MACROSCÓPICOS DE MATERIA Y
ENERGÍA PARA PROCESOS METALÚRGICOS**

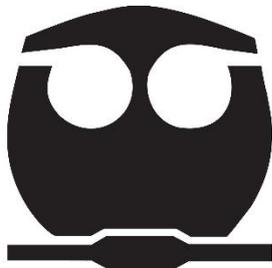
TESIS

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO QUÍMICO METALÚRGICO

PRESENTA:

JOSÉ EDUARDO CALVA DIAZ



CIUDAD DE MÉXICO 2023



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.

Presidente: Prof. JOSÉ BERNARDO HERNÁNDEZ MORALES

Vocal: Prof. GERARDO SANJUAN SANJUAN

Secretario: Prof. BENJAMIN BELTRAN FRAGOSO

1er Suplente: Prof. JUAN CARLOS CONTRERAS GUZMAN

2do Suplente: Prof. FRANCISCO LOPEZ SOSA

Sitio donde se desarrolló el tema:

Departamento de Ingeniería Metalúrgica
Facultad de Química, U.N.A.M., Edificio D.

Asesor del tema:

Dr. José Bernardo Hernández Morales

Sustentante:

José Eduardo Calva Diaz

AGRADECIMIENTOS

A mi asesor el Dr. José Bernardo Hernández Morales, gracias por todo su apoyo en este mi último trabajo como estudiante, gracias por tomarse el tiempo de revisar y estar pendiente no solo de los avances de la tesis, sino también de los trámites de titulación, gracias por guiarme durante todo el desarrollo de este trabajo y por todo lo que me enseñó como profesor durante la carrera.

Al profesor Luis Miguel Villegas Silva, gracias por ayudarme a comprender y abordar un tema totalmente nuevo para mí, su apoyo fue indispensable para el desarrollo de este trabajo.

A cada uno de los profesores que me dio clases en la UNAM, gracias por todo lo que me han enseñado desde hace más de ocho años que entré por primera vez a esta, la máxima casa de estudios, mi segunda casa.

A CCH Sur, gracias por permitirme ser parte de él, gracias porque aquí pasé una de las mejores etapas de mi vida y conocí grandes amigos.

Al subprograma 127 de la Facultad de Química, gracias por el apoyo económico que recibí durante el desarrollo de este trabajo, fue de gran ayuda.

A todos mis profesores, gracias porqué través de su enseñanza en el salón de clase fueron alimentando mis ganas de seguir aprendiendo.

A mis hermanas, gracias por brindarme su apoyo siempre que se los pedía.

A mis padres, muchas gracias porque me dieron el lujo de que mi mayor preocupación fuese entregar mis tareas a tiempo, pasar mis exámenes finales, aprender y conseguir mi título universitario, gracias a ellos no tuve que preocuparme por conseguir un techo donde dormir, ropa que vestir, un plato de comida caliente ni pagar las cuentas de luz, agua, internet, etc., cosas tan básicas y al mismo tiempo tan importantes y valiosas; gracias a ustedes tampoco me hicieron falta palabras y muestras de apoyo, amor, comprensión y motivación.

A mis amigos, gracias por hacer este camino tan divertido, gracias por todos esos momentos que pasamos dentro y fuera de la escuela.

DEDICATORIA

Este trabajo se lo dedico a mi familia, todo su apoyo, confianza, amor y sacrificio me ayudó inmensamente a concluir mi carrera universitaria. Este trabajo es de nosotros porque somos un gran equipo.

Índice

Resumen.....	1
Capítulo 1. Introducción	3
1.1 Marco de trabajo.....	3
1.2 Justificación	6
1.3 Objetivo.....	7
1.4 Metas	7
Capítulo 2. Antecedentes	8
2.1 Sistemas expertos	8
2.2 Construcción de un sistema experto.....	9
2.3 Sistema experto hibrido	12
2.4 Arquitectura de un sistema experto	13
2.5 Programación lógica	15
2.6 Ventajas y limitaciones de los sistemas expertos	16
2.7 <i>Prolog</i>	16
2.8 ¿Qué es un alto horno?	17
2.9 Materias primas del alto horno.....	18
2.10 Productos del alto horno	20
2.11 Reacciones químicas involucradas.....	23
Capítulo 3. Metodología	24
3.1 Etapa 1. Incorporación del balance macroscópico de materia.....	24
3.1.1 Subetapa 1. Desarrollo y programación del balance macroscópico de materia.....	25
3.1.2 Subetapa 2. Construcción de la base de conocimiento del SE, en <i>Prolog</i>	32
3.1.3 Subetapa 3. Integración de los dos sistemas informáticos	35
3.1.4 Subetapa 4. Complementación del modelo matemático.....	38
3.2 Etapa 2. Incorporación del balance macroscópico de materia y energía	42
3.2.1 Subetapa 1. Desarrollo y programación del balance macroscópico simultaneo de materia y energía.....	42
3.2.2 Subetapa 2. Ampliación de la base de conocimiento del SE, en <i>Prolog</i>	48

3.2.3 Subetapa 3. Integración de los dos sistemas informáticos	58
3.3 Etapa 3. Construcción de la interfaz gráfica de usuario.....	62
Capítulo 4. Resultados y análisis	65
Capítulo 5. Conclusiones	73
Bibliografía	74

Índice de figuras

Figura 2.1 Disposición general del alto horno y sus equipos de apoyo (inyección de aire caliente y equipo de manejo y carga de material).....	18
Figura 2.2 a) Pellets de hematita (Fe_2O_3) con un diámetro de 8-16 mm que contienen $\approx 64\%$ en masa de Fe y b) Coque metalúrgico de 70-100 mm de longitud que contiene un 87-91% en masa de carbono y un 9-13% de cenizas de óxido.....	20
Figura 3.1 Esquema del alto horno, indicando las corrientes de entrada y salida.....	26
Figura 3.2 Captura de pantalla de la interfaz del primer código, entrada de datos iniciales en C++.....	31
Figura 3.3 Captura de pantalla del código escrito en <i>Prolog</i> , definición del límite inferior permisible para el CO_2	32
Figura 3.4 Captura de pantalla del código escrito en <i>Prolog</i> , definición de una pregunta sobre el estado de las especies.....	33
Figura 3.5 Captura de pantalla del código escrito en <i>Prolog</i> , Actualización de la pregunta sobre el estado de las especies.....	33
Figura 3.6 Captura de pantalla de la interfaz de interacción con <i>Prolog</i> , consulta sobre estado del CO_2 cuando se encuentra fuera del rango permisible.....	34
Figura 3.7 Captura de pantalla de la interfaz de interacción con <i>Prolog</i> , consulta sobre estado del CO_2 cuando se encuentra dentro del rango permisible.....	35
Figura 3.8 Captura de pantalla de la interfaz de interacción con <i>Prolog</i> , consulta sobre el contenido de CO_2	36
Figura 3.9 Captura de pantalla de la impresión de resultados de la corrección realizada en C++.....	37
Figura 3.10 Diagrama de flujo grueso de la secuencia de ejecución de los códigos de la primera etapa, para una sola iteración.....	38
Figura 3.11 Esquema del alto horno que muestra las entradas y salidas de materia con sus respectivas temperaturas típicas.....	43
Figura 3.12 Captura de pantalla del código escrito en <i>Prolog</i> , definición de los límites permisibles.....	49
Figura 3.13 Captura de pantalla del código escrito en <i>Prolog</i> , regla sobre el estado de las especies.....	50
Figura 3.14 Captura de pantalla de la interfaz de <i>SWI Prolog</i> , consulta sobre el estado de una de las variables.....	51

Figura 3.15 Captura de pantalla del código escrito en <i>Prolog</i> , definición de la propuesta de solución.....	52
Figura 3.16 Captura de pantalla del código escrito en <i>Prolog</i> , nueva regla sobre el estado de las especies.....	52
Figura 3.17 Captura de pantalla del código escrito en <i>Prolog</i> , consulta sobre el estado del CO ₂	52
Figura 3.18 Gráfica del comportamiento del CO, CO ₂ , N y O ₂ en función de la temperatura del aire inyectado.....	54
Figura 3.19 Gráfica del comportamiento del CO, CO ₂ , N y O ₂ en función de la cantidad de carbono en el coque.....	54
Figura 3.20 Captura de pantalla del costo unitario programado en <i>Prolog</i>	55
Figura 3.21 Captura de pantalla, definición de las reglas de selección.....	55
Figura 3.22 Captura de pantalla de la nueva regla para realizar consultas sobre el estado de las variables.....	56
Figura 3.23 Captura de pantalla donde se muestra como señalar que una variable no es sugerencia de solución de otra.....	56
Figura 3.24 Captura de pantalla, consulta sobre el estado del CO ₂ cuando existen dos soluciones diferentes.....	57
Figura 3.25 Diagrama de flujo grueso del código elaborado en la Subetapa 3.....	59
Figura 3.26 Diagrama de flujo grueso del código final.....	60
Figura 3.27 Entorno de desarrollo integrado de <i>Python</i>	62
Figura 3.28 Captura de pantalla de la interfaz gráfica del SE híbrido, después de introducir los datos iniciales y oprimir los botones “%” y “1° Consulta”.....	63
Figura 3.29 Captura de pantalla de la interfaz gráfica del SE híbrido, después de introducir los datos iniciales y oprimir los botones “%” y “Sugerencia”.....	64
Figura 4.1 Captura de pantalla de los datos iniciales solicitados por el SE.....	65
Figura 4.2 Captura de pantalla de los resultados obtenidos por el SE. En este ejemplo, el contenido de CO ₂ y CO está fuera del rango admisible.....	66
Figura 4.3 Captura de pantalla de los datos iniciales solicitados por el SE.....	67
Figura 4.4 Captura de pantalla de los resultados obtenidos por el SE. En este ejemplo, el contenido de CO ₂ y CO está dentro del rango admisible.....	68
Figura 4.5 Captura de pantalla de los datos iniciales solicitados por el SE.....	69
Figura 4.6 Captura de pantalla de los datos obtenidos por el SE, en este ejemplo solo una variable se encuentra fuera de rango.....	70

Figura 4.7 Captura de pantalla de los datos solicitados por el SE híbrido.....71

Figura 4.8 Captura de pantalla de los resultados obtenidos por el SE híbrido, en este ejemplo se aplicó el criterio de selección económico cuando el CO y CO₂ se encuentran fuera de rango.....72

Índice de tablas

Tabla 2.1 Composición química típica del hierro producido por un alto horno.....	21
Tabla 2.2 Composición química típica de la escoria.....	21
Tabla 2.3 Composición típica de los gases de salida.....	22
Tabla 3.1 Arreglo matricial del sistema de ecuaciones algebraicas simultáneas de la Etapa 1.....	30
Tabla 3.2 Arreglo matricial del sistema de ecuaciones algebraicas simultáneas que resultaron de la Etapa 1.....	41
Tabla 3.3 Arreglo matricial del sistema de ecuaciones algebraicas simultáneas que resultaron de los balances de materia y energía.....	47
Tabla 3.4 Nombre y descripción de los elementos que componen la regla escrita en <i>Prolog</i> para consultar el estado de las variables del sistema.....	56
Tabla 3.4 (Cont.) Nombre y descripción de los elementos que componen la regla escrita en <i>Prolog</i> para consultar el estado de las variables del sistema.....	57

Resumen

En este trabajo se desarrolló un sistema experto híbrido (es decir, que une la programación lógica con la programación algorítmica) aplicado a procesos metalúrgicos que requieran modelarse mediante balances macroscópicos de materia y energía. Documentando paso a paso cómo fue que se elaboró, principalmente la conexión entre los lenguajes de programación seleccionados y la construcción del conocimiento lógico acerca del sistema metalúrgico, desde una versión simplificada que se fue puliendo hasta una versión lo suficientemente completa como para que el añadir conocimiento acerca del sistema metalúrgico (lógico y matemático) sea cuestión de agregar unas pocas líneas de código. El sistema metalúrgico seleccionado fue un alto horno para fabricación de arrabio.

Se requirió programar el modelo matemático en un lenguaje de programación de alto nivel (*Python*) mientras que las reglas/conocimiento lógico del sistema se programaron dentro de *Prolog*; este último es un lenguaje de programación declarativo, esto quiere decir que se construye mediante la descripción de funciones o expresiones lógicas del tipo A implica B. Posteriormente se estableció comunicación entre estos lenguajes de programación para unirlos en un solo código; esto es lo que le da el carácter de un sistema experto híbrido.

Al final del proyecto se obtuvo una interfaz de usuario donde se puede interactuar con el sistema experto híbrido. El usuario puede introducir los valores iniciales de las variables del sistema que componen al modelo matemático; a partir de esta información, el sistema experto (SE) analiza al sistema y en base a las reglas previamente establecidas proporciona un diagnóstico indicando la presencia o no de un problema, así como una propuesta de solución al mismo. Posteriormente, el SE aplica la solución propuesta y analiza los valores de las variables resultantes de aplicar la corrección; este proceso de análisis de variables y aplicación de correcciones se realiza hasta que el problema se haya solucionado o de lo contrario el SE avisa que no se llegó a una solución.

Si se logró resolver el problema, el SE reporta cuál fue la solución que llevó a cabo para resolverlo y los nuevos valores de las variables que resultan al aplicar esta solución final.

La versión final del código desarrollado es una versión simple pero funcional de un sistema experto híbrido aplicado a sistemas metalúrgicos. La actualización de esta versión continuaría con la ampliación de la base de conocimiento tanto en *Prolog* como en el modelo matemático y el acoplamiento de esta nueva información a la interfaz gráfica.

Capítulo 1. Introducción

Dentro de las universidades e institutos, así como en la industria, existen profesionistas expertos que poseen una gran cantidad de conocimientos, debido no únicamente a los estudios universitarios que hayan realizado, sino a la experiencia acumulada durante años de trabajo, misma que les permite tomar decisiones y resolver problemas complejos. Desafortunadamente, la toma de decisiones puede requerir mucho tiempo y esfuerzo, lo que a su vez puede reflejarse en pérdidas monetarias o incluso poner en riesgo la vida de personas. Además, la capacidad de análisis de un experto puede disminuir conforme la cantidad de información involucrada en un problema aumenta. Para ahorrar tiempo y esfuerzo, un experto, institución o empresa puede optar por recurrir a la ayuda de sistemas expertos.

1.1 Marco de trabajo

Un sistema experto (SE) es un sistema informático capaz de procesar conocimiento y ayudar en la toma de decisiones durante la resolución de problemas complejos, además de que tienen la capacidad de explicar su propio razonamiento; esta serie de tareas generalmente son realizadas por un experto humano. Los sistemas expertos suelen tener una cantidad considerable de conocimiento sobre el área de interés, el cual debe estar representado de tal manera que el programa sea capaz de inferir conclusiones lógicas [1].

El desarrollo de este tipo de programas siempre se ha enfocado en la resolución de problemas; los primeros sistemas expertos, desarrollados a mediados de los años setenta, no podían resolver problemas prácticos ya que se enfocaron en computarizar la inteligencia general. Fue en 1965 cuando se creó DENTRAL el primer SE enfocado a resolver problemas de la vida real permitiendo identificar estructuras químicas. A partir de este punto el desarrollo de estos programas se enfocó a resolver problemas sobre un área específica del conocimiento intentando simular el razonamiento humano [1].

Existen diferentes criterios para clasificar a los sistemas expertos. Estas clasificaciones aportan una visión más amplia de su alcance. Uno de estos criterios los clasifica por la forma de representar el conocimiento: por lo general

se representa en forma de reglas; reglas del tipo “Si... entonces...”, también se tiene el razonamiento basado en casos, el cual soluciona nuevos problemas basándose en las soluciones de problemas anteriores (casos). Otro tipo se basa en la probabilidad [2].

Otra clasificación común se basa en la configuración que pueden presentar los sistemas expertos [3]:

- Independientes: funcionan de forma autónoma y no comparten recursos informáticos con ningún otro sistema y no se encuentran intermediados por ningún otro programa o componente informático
- Híbridos: este tipo presenta dos configuraciones diferentes: 1) sistemas expertos que tienen incorporados algunos componentes que ejecutan algoritmos convencionales y 2) el sistema experto se encuentra integrado dentro de un programa algorítmico convencional.
- Dedicados: se emplean en un único ordenador, sin interactuar con otros sistemas informáticos.
- Encadenados: donde dos o más sistemas expertos interactúan entre sí
- Embebidos: el SE está integrado dentro de un programa central, de tal manera que su ejecución es imperceptible; representan una pequeña parte del programa principal.

Los sistemas expertos existentes realizan una amplia variedad de actividades. Esto permite agruparlos en función del tipo de tarea que ejecutan [3]:

Control: se encarga de identificar posibles alteraciones de un proceso a partir de diversos parámetros. Posteriormente, si está diseñado para ello, sugerir las modificaciones necesarias para corregir dichas alteraciones; estos sistemas expertos son útiles en procesos que requieren de respuestas inmediatas para resolver los problemas que se puedan presentar. Actualmente se emplean en centrales eléctricas, centrales nucleares, plantas de producción, plantas de ensamblaje, etc.

Formación: este tipo de SE es capaz de evaluar el nivel de conocimiento y capacidad intelectual del alumno permitiéndole diseñar sesiones de enseñanza

acorde a sus necesidades de aprendizaje, este tipo de aprendizaje ha dado origen al término “enseñanza inteligente asistida por ordenador”.

Diagnóstico y reparación: el sistema recibe los datos correspondientes al estado actual de un determinado equipo, mecanismo o individuo y determina si se encuentra en mal funcionamiento. De ser así, prescribe una serie de acciones con el fin de corregir los problemas detectados.

Supervisión: se encargan de realizar un seguimiento y estudio continuo del comportamiento de un sistema con el objetivo de detectar comportamientos inadecuados que provoquen algún accidente.

Planificación: A partir de una serie de objetivos diseñan planes de acción para llevar a cabo proyectos de desarrollo a gran escala, que generalmente implican numerosas secuencias de acciones con distintos grados de complejidad.

Diseño: a partir de una serie de especificaciones, requisitos y restricciones es capaz de diseñar la composición de cualquier objeto, estructura o proceso.

Pronóstico y predicción: a partir del estado inicial de un modelo y de un conjunto de posibles escenarios es capaz de predecir comportamientos futuros.

Interpretación: son programas capaces de hacer un análisis detallado de todo tipo de información ya sea cuantitativa, cualitativa, visual, etc., con el objetivo de obtener una visión más clara de la información recibida.

Aunque no todos los campos favorecen el desarrollo e implementación de un sistema experto, tienen una amplia gama de aplicaciones, algunas de ellas son: facilitar el entrenamiento del personal, control aéreo, diagnóstico médico, para determinar perfiles criminológicos, evaluación de riesgos crediticios, supervisión y control de procesos, control de inventario, supervisión de producción, etc. [2].

Hacer uso de sistemas expertos en la industria metalúrgica podría significar una gran ventaja en diferentes aspectos, por ejemplo: optimizar la toma de decisiones, acelerar la resolución de problemas específicos, permitir que una persona no especializada pueda resolver problemas complejos, etc.; además los SE no poseen las deficiencias fisiológicas de un ser humano. El alto horno es un reactor metalúrgico que procesa mineral de hierro, coque metalúrgico y piedra

caliza para producir arrabio. Tan solo en noviembre del 2022, la producción mundial de acero fue de 139.1 millones de toneladas métricas [4], el cual se fabricó principalmente por la ruta del alto horno. Es un reactor que opera a contracorriente y se caracteriza por su gran tamaño y porque las reacciones químicas que transforman a las materias primas en los productos ocurren en diferentes secciones del alto horno. Para la correcta operación es importante conocer la respuesta del sistema a cambios en la composición de la carga, la composición del aire/hidrocarburos insuflados en el horno, la composición del arrabio, etc. La descripción del sistema es muy compleja, por lo que modelos matemáticos determinísticos podrían no ser eficientes para la toma de decisiones.

1.2 Justificación

El conocimiento acerca de un sistema metalúrgico se suele representar por modelos matemáticos descritos por una serie de ecuaciones. Una práctica común en el estudio de estos sistemas es programar las ecuaciones que componen al modelo matemático en algún lenguaje de programación convencional para realizar rápidamente una serie de cálculos puntuales; por el contrario, una práctica poco utilizada es representar ese conocimiento de la forma como se hace en un sistema experto. En este trabajo se pretende comprobar que el conocimiento sobre un sistema metalúrgico puede ser representado de ambas formas y además que es posible unir estos dos sistemas informáticos en uno solo, formando así un sistema experto híbrido.

Conseguir integrar en este trabajo parte del conocimiento que se tiene en torno a un sistema metalúrgico dentro de un sistema experto híbrido podría marcar una pauta sobre qué dirección tomar en la construcción de sistemas informáticos de este tipo.

1.3 Objetivo

Elaborar un sistema experto híbrido, basado en balances macroscópicos de materia y energía, que sea capaz de analizar un conjunto de condiciones de operación y, si se determina que algunas variables están fuera de rango, proponer cambios para que todas las variables estén en rango. Como sistema de estudio se consideró al alto horno para la producción de hierro.

1.4 Metas

- Desarrollar un modelo matemático determinístico basado en balances macroscópicos de materia
- Implementar el modelo matemático en un lenguaje de programación compatible con *Prolog*
- Desarrollar un SE híbrido, incorporando al modelo matemático de balances macroscópicos de materia
- Desarrollar un modelo matemático determinístico basado en balances macroscópicos de materia y energía térmica
- Implementar el modelo matemático en un lenguaje de programación compatible con *Prolog*
- Desarrollar un SE híbrido, incorporando al modelo matemático balances macroscópicos de energía térmica
- Desarrollar una interfaz de usuario que sea amigable

Capítulo 2. Antecedentes

2.1 Sistemas expertos

Un sistema experto (SE) es un producto de la investigación en inteligencia artificial que emula el comportamiento de un experto humano en la solución de problemas [5].

De una manera más completa, un sistema experto se define como un sistema informático capaz de almacenar la experiencia, conocimiento y habilidades propios de una o más personas especializadas en un área particular del conocimiento humano y resolver problemas específicos de ese área mediante la deducción lógica de conclusiones; los sistemas expertos no son sistemas informáticos convencionales ya que no están conformados por dos partes claramente diferenciadas (datos e instrucciones) [1-3, 6].

Para interactuar con un sistema experto a menudo se requiere de un usuario humano, pero en algunas ocasiones los sistemas expertos funcionan sin la interacción de un usuario. Casi todos los sistemas expertos se desarrollan en laboratorios de inteligencia artificial para resolver problemas prácticos; los más conocidos son los de diagnóstico médico [5]; entre ellos se encuentran los sistemas expertos encargados de diagnosticar infecciones sanguíneas, enfermedades pulmonares, trastornos de los electrolitos ácidos y básicos, entre otros. Existen sistemas expertos para muchas otras áreas, aparte de la medicina, como por ejemplo la minería, en la que se utilizan para evaluar sitios para posibles depósitos minerales, además de sistemas para la educación, ingeniería, derecho, gestión, fabricación y aplicaciones militares [2].

La mayoría de los sistemas expertos son complejos y requieren de años de desarrollo y de personal altamente calificado, pero existen algunos a pequeña escala, que pueden ser construidos rápidamente por personal que tenga conocimientos de algún área en especial y de principios básicos de construcción de este tipo de programas. Así, es posible construir un SE para solucionar problemas de algún aparato eléctrico o para saber si alguien tiene derecho a las

prestaciones de la seguridad social. Un SE puede sustituir de manera inteligente a un libro o un manual [5].

Si bien el objetivo de un SE es simular el comportamiento de un experto en la toma de decisiones, no se busca que la máquina le enseñe a una persona el camino a seguir, si no que la apoye en el proceso creativo de la toma de decisiones [6].

2.2 Construcción de un sistema experto

Para construir un sistema experto, es necesario pensar en cómo funciona un experto y en las diferentes tareas que debe realizar. Podría decirse que un experto en ocasiones sirve como un consultor al cual acuden clientes con algún problema. A través de una consulta, el experto le proporciona al cliente un diagnóstico y una prescripción o sugerencia; entonces, el cliente toma la decisión de aceptar o no el diagnóstico y llevar a cabo la prescripción. Para deducir un diagnóstico y una sugerencia, el experto debe recoger información sobre el problema y en base en ello, sus conocimientos especializados y habilidades para resolver problemas da sus conclusiones. Es posible que después de que el experto de sus conclusiones al cliente este pida una explicación, con lo cual se daría por concluida la consulta con el experto; entonces, el SE debe incluir una explicación de la forma en la que llegó a la respuesta, como parte de la salida del programa [5].

Un ejemplo de cómo se lleva a cabo una consulta con un experto puede ser ir a una tienda de jardinería en busca de una manera de eliminar la mala hierba del jardín [5]. Lo primero que hará el experto en jardinería es tratar de identificar la mala hierba; esto se podría considerar como un diagnóstico. Una vez que el experto hace la identificación de la mala hierba es posible que no se necesiten más consejos para resolver el problema. Si esta información no es suficiente para eliminar la mala hierba, el experto podría sugerir formas de hacerlo; su recomendación puede depender del tipo de hierba, de las otras plantas que hay en el jardín, el clima, etcétera. Después de hacer la recomendación, el experto debería ser capaz de explicar porque hizo esa recomendación. Esta consulta se puede dividir en dos partes: 1) la interacción entre el cliente y el experto y 2) la

capacidad del experto para hacer uso de sus conocimientos para resolver problemas.

Conociendo este ejemplo práctico se puede formalizar el proceso de creación de un SE en las siguientes etapas [2-3]:

Etapa 1. Identificación y definición del problema

Esta etapa consiste en la comprensión del problema para verificar si el desarrollo de un SE realmente facilitará la resolución de este, permitiendo alcanzar todos los objetivos (que deben ser establecidos en esta etapa). Por otra parte, el problema debe tener un grado de dificultad apropiado de modo que valga la pena el desarrollo del programa.

Etapa 2. Búsqueda del experto o de la fuente de conocimiento

Es de gran ayuda contar con un experto en el tema que se pretende abordar. Pero, si no se cuenta con él, el conocimiento se puede extraer de otras fuentes, siempre y cuando ofrezcan el conocimiento, en cantidad y calidad, necesario.

El experto deberá conocer todo lo concerniente a la problemática; además, sería ideal que tome conciencia cuanto antes de las dificultades que presenta la adquisición, representación y procesamiento del conocimiento.

Etapa 3. Identificación de los conceptos y datos clave

En la siguiente fase se deberá profundizar en el conocimiento del tema en cuestión, identificando hechos y conceptos que el experto utiliza en su razonamiento durante la solución de un problema. Estos hechos y conceptos se deben escribir buscando entre ellos asociaciones y relaciones.

Etapa 4. Selección del *hardware* y *software*

En este punto se debe tener un panorama general del tipo de *software* y *hardware* que se puede utilizar en el desarrollo del SE; para la selección del *software* y *hardware* se deben tener en cuenta siempre los objetivos fundamentales, el tipo de conocimiento, la forma en que se pretende representar dicho conocimiento y las estrategias de razonamiento para obtener conclusiones lógicas.

Una vez identificada la estructura general del conocimiento a utilizar y las estrategias de solución necesarias, se debe seleccionar las herramientas y lenguajes de programación que se utilizarán.

Etapa 5. Adquisición del conocimiento

Una vez seleccionadas las herramientas de programación se debe nutrir el programa con información. Es suficiente con integrar parte del conocimiento a la base de conocimiento para ejecutar todo el conjunto de pruebas y ensayos necesarios para verificar que es factible el desarrollo del SE. Se harán las pruebas que sean necesarias para familiarizarse con el entorno de desarrollo.

Una vez que se tenga un primer prototipo para realizar suficientes pruebas para comprobar el funcionamiento del programa y, que este haya quedado validado mediante la resolución de problemas sencillos, podrá comenzar el desarrollo de una versión completa.

Etapa 6. Representación del conocimiento y formalización del razonamiento

Esta es la etapa que más tiempo y esfuerzo demandan en el diseño de sistemas expertos; se debe a que, por lo general, no se conocen con precisión cuáles son los procesos que cualquier ser humano lleva a cabo en su mente cuando analiza los datos iniciales de un problema en busca de una solución para el mismo. Afortunadamente el conocimiento puede ser representado en términos de hechos y fenómenos que integran la realidad; de procedimientos o reglas que relacionan tales hechos e información. Esta etapa consiste en la formalización de los hechos y conceptos con sus correspondientes relaciones por medio de alguna de las formas de representación del conocimiento.

Etapa 7. Desarrollo de un prototipo.

Es deseable poder trabajar con una versión reducida del programa a un nivel elemental; esta primera versión podría estar constituida sólo por el motor de inferencia (parte de un SE encargado de procesar el conocimiento) y una pequeña porción de la base de conocimiento.

Este prototipo servirá para comprobar que el diseño es adecuado y si el tratamiento y representación del conocimiento es suficiente, además de que permitirá mostrar un adelanto de lo que el sistema podría ser capaz de hacer.

Etapa 8. Adquisición de conocimiento

El siguiente paso es enriquecer al prototipo hasta alcanzar el nivel previsto. De ser necesario, se cambiará el *hardware* y/o el *software* para asegurar que la versión inicial pueda evolucionar hasta convertirse en un sistema experto, capaz de proporcionar respuestas, consejos y diagnósticos para una amplia variedad de casos, casos que un especialista esté en condiciones de resolver. El desarrollo estará ya concluido cuando opere y actúe como lo haría un especialista.

Etapa 9. Mantenimiento y actualización

Un buen SE debería ser capaz de integrar nuevo conocimiento cuando sea necesario; esta característica le permitirá ser actualizado, así como un especialista se mantiene al día sobre los temas que domina.

2.3 Sistema experto híbrido

Un sistema experto híbrido (SEH) está formado por la combinación de dos o más subsistemas inteligentes. Cada uno mantiene su propio lenguaje de programación y forma para resolver un problema [7]; cada subsistema tiene una sola forma de representación y se combinan entre sí mediante variables de entrada y salida.

El objetivo de los sistemas expertos híbridos es mejorar la forma de desarrollar sistemas expertos y construir mecanismos más potentes de una manera más eficiente que el desarrollo de un sistema basado en un solo sistema inteligente.

Los sistemas expertos híbridos aprovechan las virtudes de cada uno de los lenguajes de programación que los componen, teniendo el potencial para resolver problemas que serían muy difíciles de resolver utilizando solamente un método de razonamiento. Un SEH combina, por ejemplo, a un sistema experto

tradicional con un código que resuelve las ecuaciones que definen a un modelo matemático.

Esta clasificación presenta diferentes configuraciones: la primera de ellas son los sistemas expertos a los que se les añaden algunos componentes, a modo de subrutinas, basados en algoritmos convencionales, de modo que si al resolver un problema el sistema experto necesita ejecutar algunos cálculos u otra operación en segundo plano podría hacer uso de dicha subrutina convencional, que muy posiblemente las realizará con más eficacia que el sistema experto.

En el otro tipo de configuración el sistema experto se incorpora a un programa algorítmico convencional, el cual será el encargado de ejecutar las instrucciones principales y, si se requiere, hará uso del sistema experto para realizar las tareas específicas para las que fue creado [3].

2.4 Arquitectura de un sistema experto

Los sistemas expertos se dividen en dos entornos principales: el ambiente de desarrollo y el ambiente de consulta. El ambiente de desarrollo se utiliza por la persona que está construyendo el SE, para crear sus componentes y aumentar la base de conocimiento. El ambiente de consulta es utilizado por los no-expertos para obtener conocimiento o consejo sobre un tema en específico [2].

Hablando de las partes que componen a un SE, se pueden mencionar 4 principales, las cuales son [1-3,6]:

- Base de conocimiento
- Motor de inferencia
- Subsistema de justificación
- Interfaz

Estos elementos deben estar contruidos de tal forma que puedan desarrollarse de forma independiente entre sí. De los elementos que componen a los sistemas expertos existen dos de gran importancia: la base de conocimiento y el motor de inferencia.

Base de conocimiento

Engloba todo el conocimiento sobre el tema en específico sobre el cual se diseña el SE. El conocimiento se escribe en función de una notación específica, en forma de hechos, reglas asociadas a esos hechos, afirmaciones y preguntas.

Unas de las formas que más se emplean para relacionar conocimiento en los SE es mediante el uso de reglas lógicas y conocimiento probabilístico. Las reglas lógicas, también llamadas reglas de producción, se tratan de enunciados condicionales que se pueden estructurar de diferentes maneras, por ejemplo:

- Si la condición A es verdadera entonces la conclusión B es verdadera.
- Si se da la situación F entonces se da la acción E.
- Si las condiciones A y B son verdaderas entonces la condición C es falsa.

Las relaciones lógicas son absolutas: o son verdaderas o son falsas; con las relaciones basadas en conocimiento probabilístico, las relaciones no siempre son absolutas ya que se asocian a una probabilidad empírica.

Otra forma de representar el conocimiento en un SE es mediante la lógica de predicados o proposicional en la cual se emplean sentencias/predicados que expresan relaciones entre objetos, así como sus cualidades y atributos, estos objetos pueden ser personas, objetos o conceptos. Las proposiciones son equivalentes a frases u oraciones del lenguaje natural.

Motor de inferencia

Es el cerebro del SE: también se le conoce como estructura de control o intérprete de reglas. Es prácticamente un programa de computadora (usualmente no se crea de cero, sino que se utiliza software ya existente) que provee metodologías para el razonamiento de la información que se encuentra en la base de conocimiento, es decir, se encarga de ejecutar las inferencias (deducciones). Al realizar una consulta, el motor de inferencia se encarga de dar dirección sobre cómo utilizar la información que se encuentra en la base de conocimiento hasta llegar a una conclusión lógica.

Algo importante a tomar en cuenta en la creación de un SE es que la base de conocimiento se debe construir con base en la nomenclatura o simbología que el motor de inferencia puede manejar.

Subsistema de justificación

Reporta el camino tomado en el razonamiento. Permite al usuario y al experto conocer las líneas de razonamiento que se siguieron para encontrar una solución.

Interfaz

En la interfaz de usuario o sistema de consulta se establece la comunicación entre el usuario y el SE; su principal objetivo es mantener un dialogo en un lenguaje casi natural con el SE.

Las consultas del usuario se comunican a través de la interfaz hacia el motor de inferencia y los resultados de dicha consulta se anuncian al usuario por medio de la interfaz

2.5 Programación lógica

Una de las formas más eficaces para representar y razonar el conocimiento en las matemáticas es mediante el uso de la lógica de primer orden, también llamada lógica proposicional. En la lógica de primer orden el conocimiento se representa en forma de axiomas, a los cuales se le pueden aplicar reglas formales para deducir cosas verdaderas [8]. El trabajo de los últimos años sobre la manera de implementar computacionalmente el razonamiento lógico ha dado origen a un nuevo tipo de lenguajes de programación, conocidos como lenguajes lógicos o lenguajes declarativos, en los cuales para resolver un problema todo lo que se debe hacer es describirlo vía axiomas y reglas de deducción. La programación lógica surge de esta búsqueda de la automatización de deducciones lógicas.

Esta forma de describir fragmentos del conocimiento mediante axiomas/predicados permite originar nuevo conocimiento al aplicarles un conjunto de reglas de inferencia.

El tipo de proposiciones que se utilizan en este lenguaje de programación equivalen a frases del lenguaje natural y los conectores que se utilizan para relacionar estas proposiciones entre sí son la conjunción “y”, la disyunción “o” y la implicación “sí”. Representar el conocimiento de esta manera es parte indispensable en el desarrollo de un sistema experto; por lo que se recomienda ojear algún texto sobre lógica matemática de primer orden (por ejemplo [9]).

2.6 Ventajas y limitaciones de los sistemas expertos

Ventajas

A diferencia de un experto humano los sistemas expertos no se pueden enfermar, envejecer o migrar a otra empresa, convirtiéndose en una herramienta estable y confiable, además de ser más rápido en el procesamiento de la información; puede ser usado por personas no expertas en la solución de problemas y al usarlo con frecuencia podrían aprender de él. Sus áreas de aplicación son muy amplias [2, 3].

Limitaciones

Las principales limitaciones que presentan los SE son: para ser actualizados se necesitan reprogramar, su elevado costo en dinero y tiempo en las inversiones iniciales, son poco flexibles a cambios, presentan una alta dificultad para acceder a información no estructurada, carecen de sentido común, es complejo que aprendan automáticamente de sus errores, no son capaces de dar soluciones creativas y que su mantenimiento puede ser costoso. También presentan limitaciones a la hora de encontrar posibles aplicaciones y tratar de integrarlo dentro de una organización ya establecida [2, 3].

2.7 Prolog

Prolog [9] es un lenguaje de programación lógica, esto es, un lenguaje declarativo basado en reglas. Fue creado por Alain Colmerauer y sus colegas en la universidad de Aix-Marseille, en Marsella Francia en 1972; su nombre proviene de PROgramation en LOGique (programación en lógica) [8]. En la actualidad, *Prolog* se utiliza principalmente en el desarrollo de aplicaciones de inteligencia artificial, especialmente para sistemas de razonamiento automatizado, por

ejemplo, en sistemas expertos, servicios de asistencia automatizados, bases de datos inteligentes y programas de procesamiento del lenguaje natural.

Los programas en *Prolog* no son como los programas típicos donde se da una lista de instrucciones, paso a paso, para resolver un problema, ya que se trata de un lenguaje no-determinista, es decir, es capaz de explorar muchas posibilidades y resolver miles y miles de pequeñas ecuaciones. En estas ecuaciones se introducen incógnitas/variables del programa, que no se refieren a las variables típicas que ocupan espacio de memoria. Estas incógnitas pueden representar aquellos valores que se buscan, y al establecer ciertas relaciones entre esas incógnitas el programa es capaz de considerar todos los casos posibles y aportar todas las posibles soluciones.

La sintaxis de un programa escrito en *Prolog* consiste en lo siguiente [2, 5]:

- Declarar hechos sobre objetos y sus relaciones
- Hacer preguntas sobre objetos y sus relaciones
- Definir reglas sobre objetos y sus relaciones

2.8 ¿Qué es un alto horno?

El alto horno es un reactor pirometalúrgico vertical (Figura 2.1) el cual se utiliza para fabricar hierro fundido a partir de sus minerales; a este hierro se le conoce comúnmente como “arrabio” o “*pig iron*” [12].

El objetivo principal del alto horno es producir una aleación de hierro fundido de composición constante ($\approx 94.3\%$ en peso de hierro, $\approx 4.5\%$ de carbono, $\approx 0.6\%$ de silicio, $\approx 0.4\%$ de manganeso y cantidades mínimas de azufre y fósforo) con una productividad elevada. Uno de los parámetros críticos del funcionamiento del alto horno es la temperatura del hierro y de la escoria, que debe ser superior a los 1500°C para que estos productos estén en estado líquido.

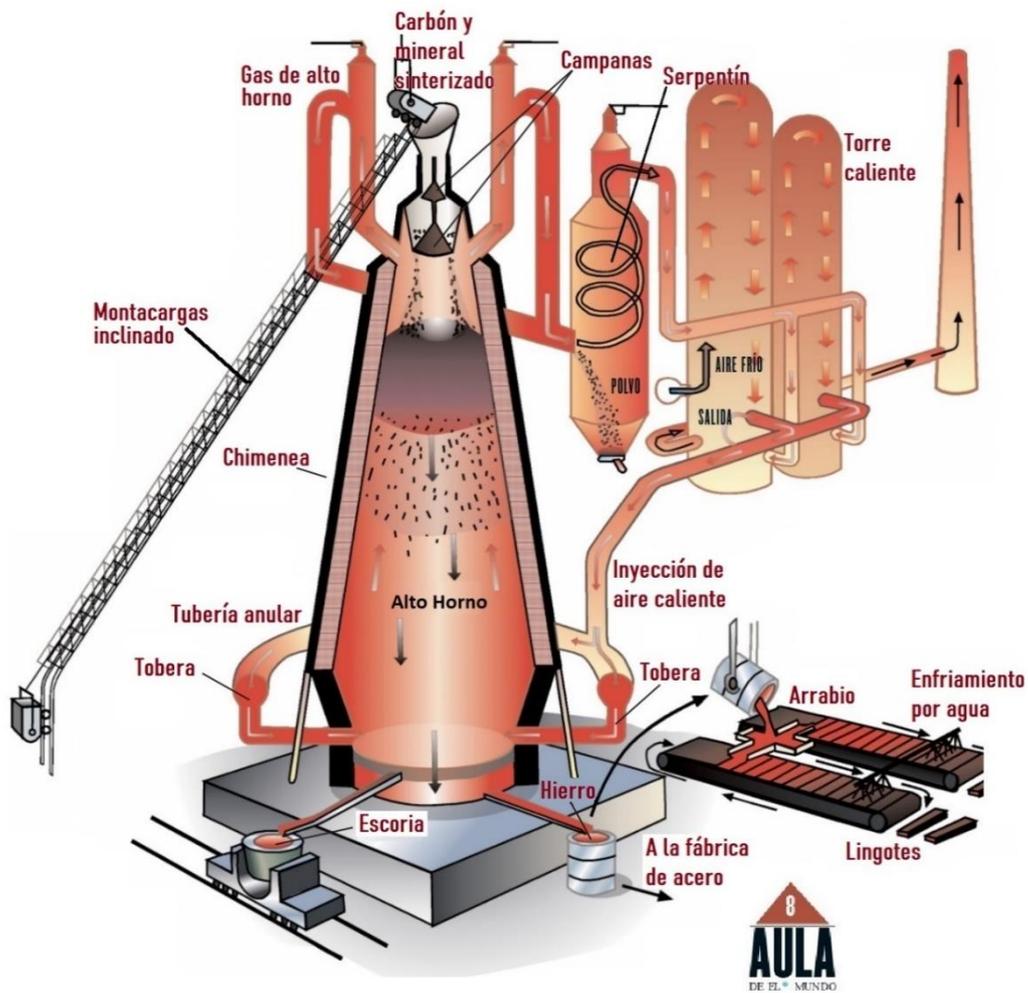


Figura 2.1 Disposición general del alto horno y sus equipos de apoyo (inyección de aire caliente y equipo de manejo y carga de material) [11].

2.9 Materias primas del alto horno

Las principales materias primas del alto horno son [12-13]:

- Sólidos cargados en la parte superior del horno: mineral de hierro (óxidos de hierro), coque y fundentes.
- Aire caliente $\approx 1200^{\circ}\text{C}$, a menudo enriquecido con oxígeno puro, que se introduce con fuerza en el horno a través de las toberas ubicadas en el fondo del horno.
- Hidrocarburos, a través de las toberas también se introducen aditivos como hidrocarburos (gas, líquido o sólidos).

Las materias primas sólidas que se cargan en la parte superior consisten en:

- a) Óxidos de hierro: normalmente hematita (Fe_2O_3), ocasionalmente magnetita (Fe_3O_4). En la actualidad estos óxidos de hierro se añaden en forma de:
- (I) Gránulos/pellets de 8-16 mm de diámetro ($\approx 64\%$ en masa de Fe y el resto de $\text{SiO}_2 + \text{Al}_2\text{O}_3$) producidos a partir de mineral finamente molido y beneficiado (\approx concentrado) (Figura 2.2 a).
 - (II) Piezas de sinterizado de 10-45 mm ($\approx 57\%$ en masa de Fe) producidas a partir de finos de mineral no beneficiado
 - (III) Mineral natural, triturado en piezas de aproximadamente 50 mm ($62\text{-}67\%$ en masa de Fe).

La carga de un alto horno puede contener una, dos o las tres formas de óxido de hierro.

- b) Coque metalúrgico: $87\text{-}91\%$ en masa de carbono, $9\text{-}13\%$ de ceniza y $1\text{-}5\%$ en masa de H_2O , agregado en trozos de 50-60 mm de diámetro (Figura 2.2 b), proporciona la mayor parte del gas reductor y del calor para la reducción de hierro y la fundición del mineral.

La ceniza de coque se compone de alúmina (Al_2O_3), sílice (SiO_2) e impurezas alcalinas (K_2O y Na_2O).

- c) Fundentes: principalmente CaO y MgO que ayudan a fundir las impurezas de sílice y alúmina del mineral y el coque para formar una escoria fluida con un bajo punto de fusión ($\approx 1400\text{ }^\circ\text{C}$), los fundentes se añaden en forma de trozos de piedra caliza (CaCO_3) y dolomita ($\text{CaCO}_3\text{:MgCO}_3$) de 50 mm de diámetro o en forma de CaO y MgO contenidos en pellets y sinterizados. Estos fundentes hacen que el azufre y las impurezas alcalinas, contenidas principalmente en el coque, se absorban en la escoria fundida y no en el hierro fundido.

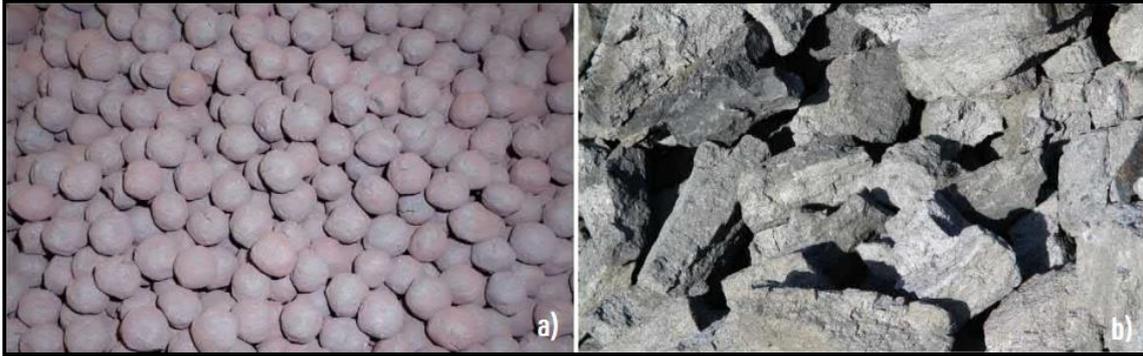


Figura 2.2 a) Pellets de hematita (Fe_2O_3) con un diámetro de 8-16 mm que contienen $\approx 64\%$ en masa de Fe y b) coque metalúrgico de 70-100 mm de longitud que contiene un 87-91% en masa de carbono y un 9-13% de cenizas de óxido [13].

Las materias primas inyectadas a través de las toberas son:

- a) Aire caliente: calentado a $\approx 1200^\circ\text{C}$, a menudo enriquecido con oxígeno puro. El aire caliente quema el coque incandescente ($\approx 1500^\circ\text{C}$) delante de las toberas para proporcionar calor suficiente para:
 - (I) Calentar y reducir los óxidos de hierro en todo el alto horno
 - (II) Fundir el hierro producido y la escoria.
- b) Hidrocarburos gaseosos, líquidos o sólidos: La mayoría de las veces se inyecta carbón pulverizado, pero también otros hidrocarburos como, por ejemplo, gas natural, "fuel oil" o el alquitrán, que se queman delante de las toberas para proporcionar calor y gases reductores (CO y H_2) extra. El carbón pulverizado es más barato que el coque por kilogramo de carbono contenido. La inyección de carbón pulverizado reduce la cantidad de coque del alto horno y el coste total de funcionamiento.

2.10 Productos del alto horno

Hierro fundido

El principal producto del alto horno es el arrabio/*pig iron* (hierro fundido) el cual se extrae del alto horno a $\approx 1500^\circ\text{C}$; se vierte a intervalos regulares o continuamente a través de uno o varios orificios (piqueras) situados cerca del fondo de la solera, la composición típica el hierro fundido se puede ver en la Tabla 2.1.

Tabla 2.1 Composición química típica del hierro producido por un alto horno [13].

Elemento	% en peso
Fe	94.4
C	4.5
Si	0.6
Mn	0.4
P	0.06
S	0.03
Ti	0.01

La composición del arrabio de un alto horno se elige en función de los requisitos de la planta siderúrgica a la que se envía el hierro. Se controla ajustando la composición de la escoria y la temperatura del horno (especialmente en la mitad inferior del horno). El arrabio se transporta inmediatamente, aun en estado líquido, a la planta siderúrgica, donde se ajusta a la composición química requerida.

Escoria

La escoria fundida se extrae del alto horno junto con el hierro fundido, la escoria se separa del hierro por gravedad, posteriormente se retira, solidifica y se vende. La escoria es una solución de óxido fundido a $\approx 1500^{\circ}\text{C}$ compuesta por [13]:

Tabla 2.2 Composición química típica de la escoria [13].

Compuesto	% en peso
CaO	40
SiO ₂	38
Al ₂ O ₃	10
MgO	10
MnO	0.4
TiO ₂	0.5
P ₂ O ₅	<0.1
S	0.8

La composición de la escoria se elige para eliminar el SiO₂ y el Al₂O₃ en una escoria fluida, absorber el K₂O₃ y el Na₂O que tienden a acumularse en el horno,

absorber el azufre para evitar que se disuelva en el hierro y controlar el contenido de silicio en el metal [12].

Gas

Los productos gaseosos del proceso salen por la parte superior del horno, donde en seguida son desempolvados. Posteriormente, este gas se quema en estufas auxiliares para calentar el aire que entra al horno y otros hornos de la planta; el polvo se aglomera mediante sinterización y se carga al horno. El producto gaseoso se compone generalmente de:

Tabla 2.3 Composición típica de los gases de salida [13].

Compuesto	% en volumen
CO	23
CO ₂	22
H ₂	3
H ₂ O	3
N ₂	49

Un ciclo completo en un alto horno consiste en introducir, de manera continua, los óxidos de hierro, el coque y los fundentes por la parte superior del alto horno a la presión del horno y en capas separadas, el coque se quema en el fondo del horno con el aire caliente ($\approx 1200^{\circ}\text{C}$) inyectado a través de las toberas que son entre 15 y 45 toberas de cobre refrigerado por agua las cuales se encuentran alrededor del horno; al quemar el coque se obtiene la energía necesaria para para que se lleve a cabo el proceso de reducción del mineral.

El hierro fundido se extrae de la parte inferior del horno y se vierte en contenedores conocidos como carros torpedo. Inmediatamente se transporta el hierro fundido a la acería.

2.11 Reacciones químicas involucradas

Las principales reacciones químicas involucradas en el proceso de reducción en el alto horno son:

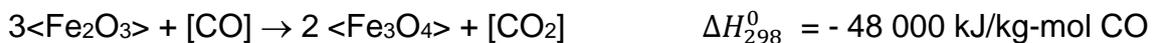
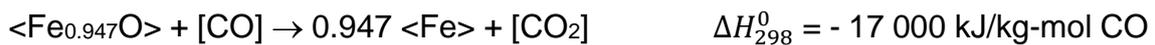
1) La oxidación del carbono por la presencia de aire inyectado a través de las toberas generando CO₂ (gaseoso) y calor (reacción exotérmica):



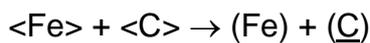
2) Reacción entre el CO₂ con el carbono para producir CO (reacción endotérmica):



3) Reducción de los óxidos de hierro (Fe) sólido:



4) Formación de hierro fundido a partir de especies sólidas (Fe y C):



Donde:

<i>: especie i en estado sólido

(i): especie i en estado líquido

(i): especie i disuelta

[i]: especie i en estado gaseoso

Capítulo 3. Metodología

El SE incorporó un modelo matemático resultante de un balance (macroscópico) simultáneo de materia y energía, por lo que puede clasificarse como un sistema experto híbrido (SEH). Una buena práctica de programación es desarrollar un programa de cómputo en etapas en lugar de intentar construir la versión final “de una sola vez”. Es así, que en esta sección se detallan las etapas que se siguieron para desarrollar al SEH, mostrando resultados intermedios.

Su desarrollo se realizó en dos etapas: 1) incorporando un balance macroscópico de materia y 2) incorporando un balance macroscópico de materia y energía; en ambos casos, se utilizó al alto horno para la fabricación de arrabio como ejemplo. Cada una de estas etapas tuvo, a su vez, subetapas, que también se describirán en esta sección. Tómese en cuenta que esta es la primera ocasión en la que se desarrolla un SEH en una tesis de licenciatura de la Licenciatura en Ingeniería Química Metalúrgica, por lo que se ha puesto énfasis en detallar la metodología lo más posible.

3.1 Etapa 1. Incorporación del balance macroscópico de materia

Parte fundamental fue producir un modelo matemático determinístico, basado (inicialmente) en balances de materia, en un lenguaje de cómputo compatible con *Prolog*, que es el lenguaje que se empleó para elaborar el SE. El balance de materia es de naturaleza macroscópica y se escribió en base atómica. Aunque el proceso involucra intercambio de energía térmica entre las diversas zonas del alto horno, en esta primera etapa no se incluyeron balances de energía térmica.

Esta primera etapa en el desarrollo del sistema experto híbrido se realizó en cuatro subetapas, las cuales se indican a continuación.

Subetapa 1: Desarrollo del modelo matemático y programación del balance macroscópico de materia, utilizando un lenguaje de programación convencional (C++).

Subetapa 2: Construcción de la base de conocimiento del SE, programado en *Prolog*

Subetapa 3: Integración de los dos sistemas informáticos.

Subetapa 4: Complementación del modelo matemático.

3.1.1 Subetapa 1. Desarrollo y programación del balance macroscópico de materia

La primera etapa en la elaboración del sistema experto para el alto horno consistió en resolver un problema relativamente simple utilizando dos lenguajes de programación diferentes, uno convencional (C++) y otro basado en reglas lógicas (*Prolog*), esto con la finalidad de asegurar que es posible intercambiar información entre ambos lenguajes de programación. En esta primera versión, la comunicación se da mediante la lectura y escritura de archivos de texto.

Para realizar la primera parte del modelado matemático del alto horno se hizo una investigación en la literatura para identificar las expresiones algebraicas que describen el comportamiento del reactor. Una vez que se realizó la investigación, el problema se planteó como se describe enseguida.

En primer lugar, se debe tomar en cuenta que la fabricación de hierro en el alto horno implica:

- 1) La combustión del coque con el aire caliente que se inyecta al horno, produciendo monóxido de carbono (CO) el cual asciende a través del alto horno.
- 2) La reacción química entre el CO y el oxígeno en el mineral (óxidos de hierro) que desciende, produciendo CO₂ y hierro fundido que desciende.

Sabiendo esto, el sistema que se pretende modelar se puede describir de la siguiente manera: Un alto horno el cual tiene como entradas de material hematita (Fe₂O₃), coque y aire (O₂ + N₂) y como salidas hierro (líquido), CO, CO₂ y N₂ (gases) (ver Figura 3.1).

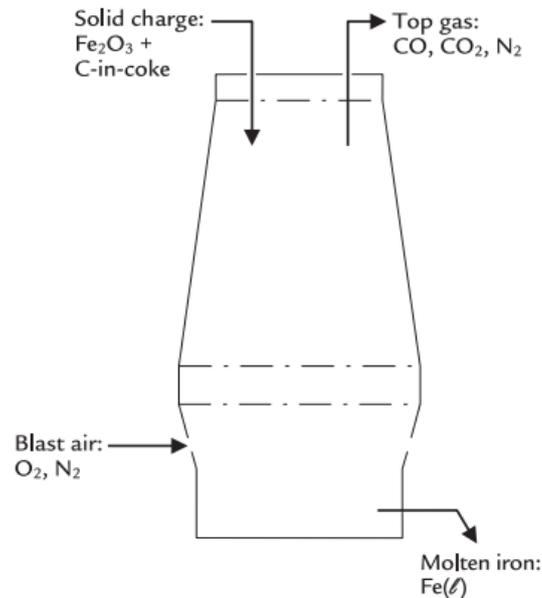


Figura 3.1 Esquema del alto horno, indicando las corrientes de entrada y salida [13].

El siguiente paso es el desarrollo del modelo matemático del alto horno, el cual inicia con el planteamiento de ecuaciones de balance de materia en estado estacionario para un sistema simplificado.

Se establecieron balances de materia en base atómica de tal forma que las expresiones queden en función de la masa de cada una de las especies involucradas. Se consideraron ocho variables:

- Masa de hematita (Fe_2O_3) en el mineral cargado
- Masa de carbono (C) en el coque cargado
- Masa de oxígeno (O_2) en el aire inyectado
- Masa de nitrógeno (N_2) en el aire inyectado
- Masa de hierro (Fe) en el arrabio producido
- Masa de CO en los gases de salida
- Masas de CO_2 en los gases de salida
- Masa de N_2 en los gases de salida

Especificaciones adicionales

Se considera la composición del aire inyectado (76.7% N₂ y 23.3% O₂) y la cantidad de hierro producido (1000 kg), estas consideraciones se describen mediante las ecuaciones siguientes:

Para la composición del aire:

$$\frac{\left(\begin{array}{l} \text{Masa de N}_2 \text{ en el} \\ \text{aire inyectado} \end{array} \right)}{\left(\begin{array}{l} \text{Masa de O}_2 \text{ en el} \\ \text{aire inyectado} \end{array} \right)} = \frac{76.7\% \text{ de N}_2}{23.3\% \text{ de O}_2} = 3.3$$

$$\left(\begin{array}{l} \text{Masa de N}_2 \text{ en el} \\ \text{aire inyectado} \end{array} \right) * 1 = \left(\begin{array}{l} \text{Masa de O}_2 \text{ en el} \\ \text{aire inyectado} \end{array} \right) * 3.3$$

$$0 = - \left(\begin{array}{l} \text{Masa de N}_2 \text{ en el} \\ \text{aire inyectado} \end{array} \right) * 1 + \left(\begin{array}{l} \text{Masa de O}_2 \text{ en el} \\ \text{aire inyectado} \end{array} \right) * 3.3 \quad (3.1)$$

Para los 1000 kg de hierro producido

$$\left(\begin{array}{l} \text{Masa de Fe en} \\ \text{el hierro fundido} \end{array} \right) * 1 = 1000$$

$$1000 = \left(\begin{array}{l} \text{Masa de Fe en} \\ \text{el hierro fundido} \end{array} \right) * 1 \quad (3.2)$$

Balance para el hierro

El hierro (Fe) entra al horno como hematita (Fe₂O₃) y sale del horno como hierro líquido.

Masa que entra = Masa que sale

$$\left(\begin{array}{l} \text{Masa de Fe}_2\text{O}_3 \\ \text{en el mineral} \end{array} \right) * \frac{69.9\% \text{ Fe en Fe}_2\text{O}_3}{100\%} = \left(\begin{array}{l} \text{Masa de Fe en} \\ \text{el hierro fundido} \end{array} \right) * \frac{100\% \text{ Fe}}{100\%}$$

$$0 = - \left(\begin{array}{l} \text{Masa de Fe}_2\text{O}_3 \\ \text{en el mineral} \end{array} \right) * 0.699 + \left(\begin{array}{l} \text{Masa de Fe en} \\ \text{el hierro fundido} \end{array} \right) * 1 \quad (3.3)$$

Balance para el oxígeno

El oxígeno entra al alto horno como Fe_2O_3 y en el aire inyectado, todo el oxígeno sale en forma de CO y CO_2 (gaseosos).

$$\begin{aligned} & \left(\text{Masa de } Fe_2O_3 \text{ en el mineral} \right) * \frac{30.1\% \text{ O en } Fe_2O_3}{100\%} + \left(\text{Masa de } O_2 \text{ en el aire inyectado} \right) * \frac{100\% \text{ O en } O_2}{100\%} = \\ & \left(\text{Masa de CO en los gases de salida} \right) * \frac{57.1\% \text{ O en CO}}{100\%} + \left(\text{Masa de } CO_2 \text{ en los gases de salida} \right) * \frac{72.2\% \text{ O en } CO_2}{100\%} \\ 0 = & - \left(\text{Masa de } Fe_2O_3 \text{ en el mineral} \right) * 0.301 - \left(\text{Masa de } O_2 \text{ en el aire inyectado} \right) * 1 + \left(\text{Masa de CO en los gases de salida} \right) \\ & * 0.571 + \left(\text{Masa de } CO_2 \text{ en los gases de salida} \right) * 0.727 \end{aligned} \quad (3.4)$$

Balance para el carbono

El carbono entra al alto horno en el coque cargado y sale en los gases como CO y CO_2 .

$$\begin{aligned} & \left(\text{Masa de C en el coque} \right) * \frac{100\% \text{ C}}{100\%} = \left(\text{Masa de CO en los gases de salida} \right) * \frac{42.9\% \text{ C en CO}}{100\%} + \\ & \left(\text{Masa de } CO_2 \text{ en los gases de salida} \right) * \frac{27.3\% \text{ C en } CO_2}{100\%} \\ 0 = & - \left(\text{Masa de C en el coque} \right) * 1 + \left(\text{Masa de CO en los gases de salida} \right) * 0.429 + \\ & \left(\text{Masa de } CO_2 \text{ en los gases de salida} \right) * 0.273 \end{aligned} \quad (3.5)$$

Balance para el nitrógeno

El nitrógeno entra al horno en el aire inyectado y sale en los gases.

$$\begin{aligned} & \left(\text{Masa de } N_2 \text{ en el aire inyectado} \right) * \frac{100\% \text{ N en } N_2}{100\%} = \left(\text{Masa de } N_2 \text{ en los gases de salida} \right) * \frac{100\% \text{ N en } N_2}{100\%} \\ 0 = & - \left(\text{Masa de } N_2 \text{ en el aire inyectado} \right) * 1 + \left(\text{Masa de } N_2 \text{ en los gases de salida} \right) * 1 \end{aligned} \quad (3.6)$$

Se tienen 8 variables y solamente 6 ecuaciones, por lo que el sistema no está determinado. Para dar solución a las variables se necesitan 2 ecuaciones más; para ello se proponen 2 especificaciones adicionales, 370 kg de O₂ en el aire inyectado y 400 kg de carbono en el coque por 1000 kg de hierro producido, estos valores corresponden a especificaciones típicas en la operación del alto horno [13]. Descritas como ecuaciones quedarían de la siguiente forma:

$$400 = \left(\begin{array}{c} \text{Masa de C en} \\ \text{el coque} \end{array} \right) * 1 \quad (3.7)$$

$$370 = \left(\begin{array}{c} \text{Masa de O}_2 \text{ en el} \\ \text{aire inyectado} \end{array} \right) * 1 \quad (3.8)$$

Con estas dos nuevas expresiones estaría completo el sistema de ecuaciones y se podría resolver para cualquier combinación de valores de masa de carbono en el coque y oxígeno en el aire inyectado.

El sistema de ecuaciones algebraicas simultáneas se encuentra escrita en forma matricial en la Tabla 3.1.

Tabla 3.1 Arreglo matricial del sistema de ecuaciones algebraicas simultáneas de la Etapa 1.

	Masa de Fe ₃ O ₄ en el mineral	Masa de C en el coque	Masa de O ₂ en el aire inyectado	Masa de N ₂ en el aire inyectado	Masa de Fe en el hierro fundido	Masa de CO en los gases de salida	Masa de CO ₂ en los gases de salida	Masa de N ₂ en los gases de salida
Producción de hierro	1000	0	0	0	1	0	0	0
Balance para el hierro	0	-0.699	0	0	1	0	0	0
Balance para el oxígeno	0	-0.301	0	-1	0	0.571	0.727	0
Balance para el carbono	0	0	-1	0	0	0.429	0.273	0
Balance para el Nitrógeno	0	0	0	0	-1	0	0	1
Composición del aire	0	0	0	3.3	-1	0	0	0
Carbono en el coque	400	0	1	0	0	0	0	0
O ₂ en el aire inyectado	370	0	0	1	0	0	0	0

Hasta este punto solamente se ha planteado el sistema de ecuaciones algebraicas simultáneas (SEAS) que describen el comportamiento del sistema, específicamente de la composición de los gases de salida, en función de la cantidad de hierro producido, de carbono que entra en el coque y la cantidad de oxígeno que entra en el aire inyectado, si bien este desarrollo no constituye un sistema experto sí es parte fundamental del sistema experto híbrido que se desarrolló.

Para finalizar esta subetapa se escribió un código utilizando el lenguaje de programación C++ [8]. Este primer código está encargado de recabar la información actual del alto horno tal y como la conocería un operador. Esto incluye información de cada una de las corrientes de entradas y salidas de material en el alto horno, por ejemplo, entrada de mineral, aire, coque, salida de hierro y gases (CO, CO₂ y N₂). En la Figura 3.2 se muestra la interfaz con la que el usuario interactúa con este código. También se definió una función que construye y resuelve cualquier SEAS que pueda ser representado en una matriz cuadrada, como por ejemplo la matriz de la Tabla 3.1.

```
Introducir valores en kg
Cantidad de Fe2O3 en la carga: 1431
Cantidad de C en coque: 400
Cantidad de O2 en la entrada de gases: 370
Cantidad de N2 en la entrada de gases: 1221
Cantidad de Fe producido: 1000
Cantidad de CO en los gases de salida: 463
Cantidad de CO2 en los gases de salida: 738
Cantidad de N2 en los gases de salida: 1221

Composicion de los gases de salida:
%CO = 19.1164
%CO2 = 30.4707
%N2 = 50.4129
Presione una tecla para continuar . . .

-----
Process exited after 46.98 seconds with return value 0
Presione una tecla para continuar . . .
```

Figura 3.2 Captura de pantalla de la interfaz del primer código, entrada de datos iniciales en C++.

3.1.2 Subetapa 2. Construcción de la base de conocimiento del SE, en *Prolog*

En esta subsección se estableció una base de conocimiento (hechos, reglas y preguntas) en *Prolog*, que es equivalente al criterio de comparación, para dar un diagnóstico de las condiciones del alto horno, que se menciona en el problema planteado a continuación.

El siguiente paso consistió en plantear un problema que el SEH debería ser capaz de resolver. El problema puede definirse como: a partir de los datos (valores de las corrientes de salida y entrada del alto horno) que introduzca el usuario, el SE comparará cada uno de las variables con un criterio determinado y dará un diagnóstico; en el caso de que alguna de las variables no cumpla ese criterio el SE propone e implementa una solución, posteriormente le hace saber al usuario qué correcciones realizó y, si resolvió el problema (este problema se irá resolviendo conforme se avance con las etapas del desarrollo del SE).

La manera de programar en *Prolog* es mediante un conjunto de relaciones lógicas, representadas como hechos y reglas, del tipo A implica B o si se cumple A y B entonces se cumple C, y la interacción o cálculos se lleva a cabo mediante una consulta sobre esas relaciones; es por esto que se utilizó a *Prolog* para comparar los datos iniciales con determinados parámetros y proponer una solución (en caso de ser necesaria) [14]. Todo el conocimiento que tendría un experto en el alto horno debe estar almacenado/recopilado en *Prolog*.

Para comenzar a construir la base de conocimiento se partió desde el caso más simple, definiendo el límite inferior de una de las especies (CO₂) y desarrollando una pregunta sobre si esta especie se encuentra fuera de ese límite inferior; de ser así, propone disminuir en un 10% el contenido de oxígeno. El límite inferior se escribió en forma de hechos en el lenguaje de *Prolog* y quedó expresado como se muestra en la Figura 3.3.

```
1
2  limite_inferior(695,co2).
3
```

Figura 3.3 Captura de pantalla del código escrito en *Prolog*, definición del límite inferior permisible para el CO₂

Que en lenguaje común significa que 695 es el límite inferior de CO₂, las unidades están en kg por 1000 kg de hierro producido. Este límite se estableció de manera arbitraria ya que el principal objetivo de esta etapa es comprobar que es posible construir un sistema experto híbrido utilizando *Prolog* y C++.

La sintaxis básica para construir una regla en *Prolog* es la siguiente:

Cabeza :- Cuerpo

Y se lee de la siguiente manera: “La cabeza es verdadera si el cuerpo es verdad”; el cuerpo podría estar conformado por más de una condición.

En la Figura 3.4 quedó expresada la pregunta sobre el estado del CO₂ en el lenguaje de *Prolog*.

```
4
5 fuera_de_rango(Especiex,ValorEspeciex):-
6 limite_inferior(L,Especiex), ValorEspeciex<L.
7
```

Figura 3.4 Captura de pantalla del código escrito en *Prolog*, definición de una pregunta sobre el estado de las especies.

Que en lenguaje común quiere decir que la especie (*Especiex*) se encuentra fuera de rango si el valor de la especie (*ValorEspeciex*) se encuentra por debajo de su límite inferior permisible (*L*); para que se cumpla la condición “*ValorEspeciex<L*” también se debe cumplir que exista un límite inferior para la especie “*Especiex*” (*limite_inferior(L,Especiex)*).

Los elementos que se encuentran entre paréntesis y comienzan por una letra mayúscula se consideran como variables, también podrían comenzar con un guion bajo “_”;

Para añadir la sugerencia de solución se agregó la variable “*Correccion*” a la misma pregunta como se muestra en la Figura 3.5.

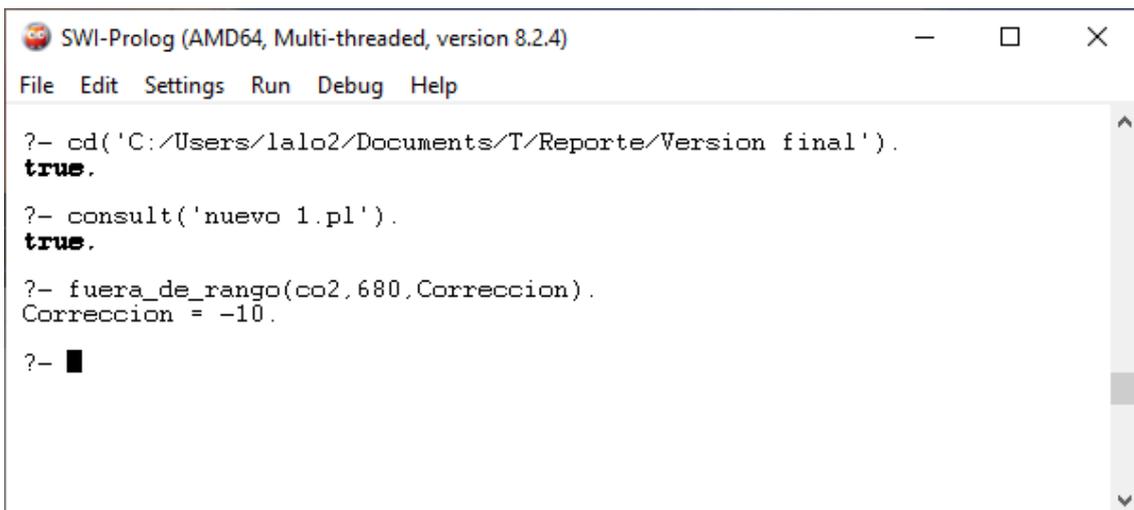
```
4
5 fuera_de_rango(Especiex,ValorEspeciex,Correccion):-
6 limite_inferior(L,Especiex), ValorEspeciex<L, Correccion is -10.
7
```

Figura 3.5 Captura de pantalla del código escrito en *Prolog*, Actualización de la pregunta sobre el estado de las especies.

Esto no afecta el resultado previo, simplemente la variable “Correccion” tomara el valor de “-10” si las condiciones “ValorEspeciex<L” y “limite_inferior(L,Especiex)” se cumplen.

Cabe señalar que todos los códigos escritos en el lenguaje de *Prolog* se modifican en un editor de texto y se guardan con la extensión “.pl”. Para hacer consultas sobre el archivo que se acaba de crear primero se deben introducir un par de comandos donde se especifica la ubicación y el nombre del archivo; la manera de realizar la consulta es sustituyendo en la pregunta anterior (ver Figura 3.5) las variables “Especiex” y “ValorEspeciex” por el nombre y valor correspondientes de la variable CO₂; esta serie de instrucciones se ejecutan en la interfaz de SWI *Prolog* tal como se muestra en la Figura 3.6.

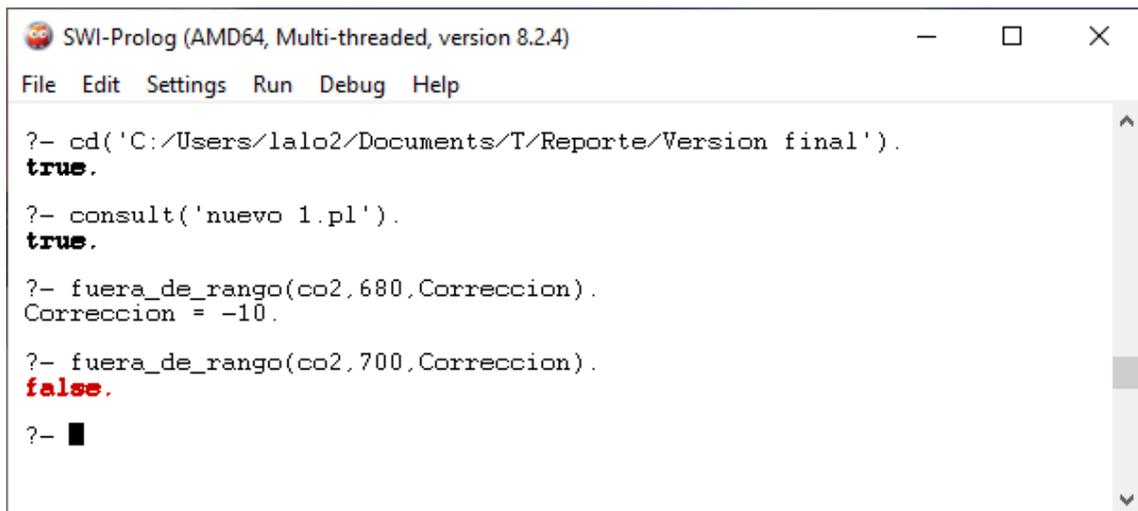
Esta consulta dará como resultado “Correccion = -10”, esto quiere decir que es verdadero que la especie “co2” se encuentra fuera de rango y propone disminuir en un 10% el contenido de oxígeno que entra al horno.



```
SWI-Prolog (AMD64, Multi-threaded, version 8.2.4)
File Edit Settings Run Debug Help
?- cd('C:/Users/lalo2/Documents/T/Reporte/Version final').
true.
?- consult('nuevo 1.pl').
true.
?- fuera_de_rango(co2,680,Correccion).
Correccion = -10.
?- █
```

Figura 3.6 Captura de pantalla de la interfaz de interacción con *Prolog*, consulta sobre estado del CO₂ cuando se encuentra fuera del rango permisible.

Si en lugar de escribir 680 se introduce un valor por encima de 695 el resultado de la consulta será “false”, como se puede observar en la Figura 3.7, lo que significa que la cantidad de la especie “co2” no se encuentra fuera del rango permisible.



```
?- cd('C:/Users/lalo2/Documents/T/Reporte/Version final').
true.

?- consult('nuevo 1.pl').
true.

?- fuera_de_rango(co2, 680, Correccion).
Correccion = -10.

?- fuera_de_rango(co2, 700, Correccion).
false.

?- █
```

Figura 3.7 Captura de pantalla de la interfaz de interacción con *Prolog*, consulta sobre estado del CO₂ cuando se encuentra dentro del rango permisible.

3.1.3 Subetapa 3. Integración de los dos sistemas informáticos

En esta subetapa se buscó comunicar ambos lenguajes de programación (C++ y *Prolog*) para resolver el problema planteado de una manera más eficiente, aprovechando las características de cada uno de ellos.

A simple vista parece sencillo el problema que se está resolviendo; esto se debe a que el principal objetivo de esta primera etapa es confirmar que es posible construir un SEH de un sistema metalúrgico a partir de balances macroscópicos haciendo uso de un lenguaje de programación convencional (C++) y otro basado en reglas (*Prolog*).

Para resumir, el lenguaje de programación C++ se utilizó para capturar los datos iniciales y dar solución al sistema de ecuaciones algebraicas simultáneas (SEAS); en el lenguaje de programación *Prolog* se programó una pequeña porción de conocimiento, el cual permite hacer consultas sobre si el valor de la especie CO₂ se encuentra fuera del rango inferior permisible.

Ya que ambos lenguajes de programación son capaces de escribir y leer archivos de texto, la comunicación entre ellos se dio de esa manera, por lo que se modificaron ligeramente los programas desarrollados en las subetapas

anteriores añadiendo instrucciones encargadas de escribir y leer archivos de texto.

A continuación, se describen, en orden de ejecución, las tareas que realizó cada lenguaje de programación para establecer comunicación entre sí y resolver una versión simple del problema planteado en la Subetapa 2; estas tareas se llevan a cabo mediante la ejecución de tres diferentes códigos, que se describen a continuación.

Programa 1 (escrito en C++): Solicita al usuario introducir los valores de cada una de las variables del sistema; el programa escribe estos valores en un archivo de texto en un orden específico.

Programa 2 (escrito en *Prolog*): Este código lee el archivo de texto que se creó con el código anterior y permite hacer una consulta/pregunta sobre si la cantidad de CO₂ producido se encuentra fuera del límite inferior permisible. En el caso de que el CO₂ se encuentre fuera del límite permisible, *Prolog* arroja una respuesta positiva y una sugerencia de solución. Al finalizar la consulta se crea un segundo archivo de texto con la información que recibió *Prolog* y una sugerencia de corrección de parámetros (en caso de ser necesario). La Figura 3.8 muestra la interfaz de *SWI Prolog* en la que el usuario interactúa para realizar las consultas.

```
SWI-Prolog (AMD64, Multi-threaded, version 8.2.4)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- cd('C:/Users/lalo2/Documents/Asignatura Proyecto/Semana 10').
true.

?- ls.
% corregir variable.cpp          gastopinic.txt
% corregir variable.exe        pruebas 1.pl
% corregir.txt                 pruebas 2.pl
% Entrada de datos iniciales 2.cpp Sist. de ec. 2.cpp
% Entrada de datos iniciales 2.exe texto.txt
% gastop2.txt
true.

?- consult('pruebas 2.pl').
true.

?- fuera_de_rango('gastopinic.txt',L).
L = [1431, 400, 370, 1221, 1000, 19.1164, 30.4707, 50.4129, 2422].

?- █
```

Figura 3.8 Captura de pantalla de la interfaz de interacción con *Prolog*, consulta sobre el contenido de CO₂.

Programa 3: (escrito en C++): Con este código se implementa la sugerencia a la que llegó *Prolog* mediante la lectura del archivo de texto creado con el código anterior. Después de aplicar la corrección, que en este caso fue la reducción de la cantidad de oxígeno que entra en el aire inyectado (dado que el experto sabe que esto es cierto), el programa resuelve el nuevo SEAS. La nueva información se escribe en un archivo de texto el cual se podría consultar nuevamente con el segundo código y al realizar la misma consulta/pregunta averiguar si la sugerencia de *Prolog* resolvió el problema; de no ser así, se podrían volver a ejecutar los programas 3 y 2 hasta que la cantidad de CO₂ producido se encuentre por encima del límite inferior permisible. La Figura 3.9 muestra la ventana que arroja C++ al ejecutar el tercer código.

```

Daton de entrada:
1431
400
370
1221
1000
19.1164
30.4707
50.4129
-10
2422

Sistema de ecuaciones a resolver:
 0.00  0.00  0.00  0.00  1.00  0.00  0.00  0.00  | 1000.00
-0.70  0.00  0.00  0.00  1.00  0.00  0.00  0.00  |  0.00
-0.30  0.00 -1.00  0.00  0.00  0.57  0.73  0.00  |  0.00
 0.00 -1.00  0.00  0.00  0.00  0.43  0.27  0.00  |  0.00
 0.00  0.00  0.00 -1.00  0.00  0.00  0.00  1.00  |  0.00
 0.00  0.00  3.30 -1.00  0.00  0.00  0.00  0.00  |  0.00
 0.00  1.00  0.00  0.00  0.00  0.00  0.00  0.00  |  400.00
 0.00  0.00  1.00  0.00  0.00  0.00  0.00  0.00  |  333.00

Solucion:

X1=1430.615234
X2=399.999939
X3=332.999969
X4=1098.899902
X5=1000.000000
X6=527.775879
X7=635.839233
X8=1098.899902

Composicion de los gases de salida:
%CO = 23.327
%CO2 = 28.1032
%N2 = 48.5698
Presione una tecla para continuar . . .

```

Figura 3.9 Captura de pantalla de la impresión de resultados de la corrección realizada en C++.

Se observa, que estos programas son válidos sólo para un caso particular y deben ser modificados para cubrir toda una gama de casos. Además, esta manera de comunicar ambos lenguajes de programación no es la más cómoda y eficiente; este inconveniente se resolvió en la Etapa 2.

En la Figura 3.10 se muestra un diagrama de la secuencia de ejecución; esta secuencia de pasos solamente representa una iteración del circuito final desarrollado en el SE híbrido.

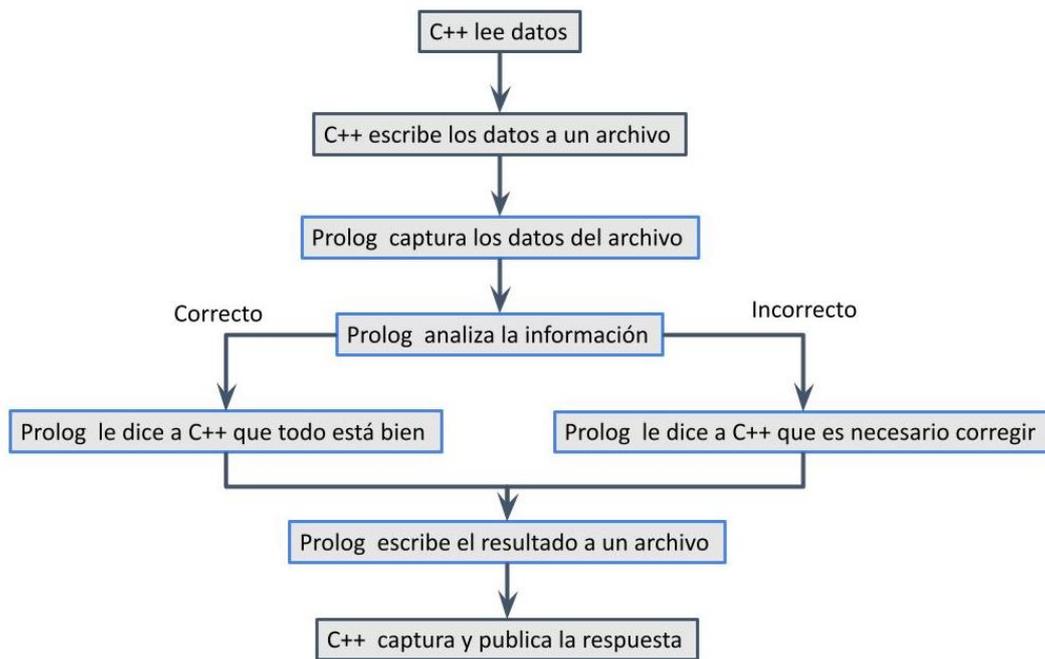


Figura 3.10 Diagrama de flujo grueso de la secuencia de ejecución de los códigos de la primera etapa, para una sola iteración.

3.1.4 Subetapa 4. Complementación del modelo matemático.

Esta última subetapa consistió en complementar el sistema de ecuaciones que describen el comportamiento del alto horno, es decir, agregar otras posibles condiciones de operación; esto se logró modificando algunos de los balances de materia que se describieron en etapas anteriores y agregando nuevas variables.

Se sabe que el alto horno no solo funciona con hematita, sino que también se suelen cargar otro tipo de minerales como la magnetita (Fe_3O_4) o estos mismos

minerales, pero en diferentes presentaciones, como en pellets, sinter y trozos de mineral. Para completar el SEAS, tal que se tenga en cuenta que en el alto horno se puede cargar mineral de hematita y/o magnetita, es necesario modificar dos de las ecuaciones de balance de materia.

Suponiendo que todo el hierro que entra al horno lo hace en forma de mineral de magnetita, y dado que la magnetita contiene 72.45 % en masa de hierro, la Ec. 3.3 se tendría que modificar; de la Ec. 3.3 se conoce que la cantidad de hierro que entra al horno se calcula multiplicando la cantidad de mineral que entra por la fracción peso de hierro que contiene el mineral:

$$(Masa\ de\ mineral) * (Fracción\ peso\ de\ Fe\ en\ el\ mineral)$$

Para calcular la cantidad de hierro que entra al horno en un mineral que contiene hematita y/o magnetita se debe considerar que la cantidad de hierro que puede entrar es $0.724 * mineral$ (100% magnetita) y la mínima $0.699 * mineral$ (100% hematita). Entonces para considerar una mezcla de minerales, la ecuación del balance para el hierro quedaría expresada de la siguiente forma:

$$0 = - \left(\begin{matrix} Masa\ de \\ mineral \end{matrix} \right) * \left[0.724 * \left(\begin{matrix} \% magnetita \\ en\ el\ mineral \end{matrix} \right) + 0.699 * \left(\begin{matrix} \% hematita \\ en\ el\ mineral \end{matrix} \right) \right] + \left(\begin{matrix} Masa\ de\ Fe\ en \\ hierro\ fundido \end{matrix} \right) * 1 \quad (3.9)$$

Al tener una mezcla de óxidos de hierro (hematita y magnetita) también se debe modificar el balance de materia para el oxígeno. Aplicando el razonamiento anterior y sabiendo que la cantidad de oxígeno que entra en el mineral es $0.301 * mineral$ (100% hematita) y la mínima $0.276 * mineral$ (100% magnetita), el balance de materia para el oxígeno quedaría expresado de la siguiente forma:

$$0 = - \left(\begin{matrix} Masa\ de \\ mineral \end{matrix} \right) * \left[0.276 * \left(\begin{matrix} \% magnetita \\ en\ el\ mineral \end{matrix} \right) + 0.301 * \left(\begin{matrix} \% hematita \\ en\ el\ mineral \end{matrix} \right) \right] - \left(\begin{matrix} Masa\ de\ O_2\ en\ el \\ aire\ inyectado \end{matrix} \right) * 1 + \left(\begin{matrix} Masa\ de\ CO\ en\ los \\ gases\ de\ salida \end{matrix} \right) * 0.571 + \left(\begin{matrix} Masa\ de\ CO_2\ en\ los \\ gases\ de\ salida \end{matrix} \right) * 0.727 \quad (3.10)$$

Otra de las variables que se tomó en cuenta fue el porcentaje de carbono en el arrabio producido. En el diagrama de la Figura 3.1 no se consideró al carbono

en el hierro fundido, aunque el hierro producido en el alto horno contiene alrededor del 4.5% de carbono. Agregar otra variable al sistema implica agregar otra ecuación; sin tomar en cuenta las impurezas, el arrabio se puede considerar como una mezcla: $\approx 4.5\%$ de carbono y el resto de hierro, esta composición se puede expresar de la siguiente manera:

$$\frac{\left(\begin{array}{c} \text{Masa de C en el} \\ \text{hierro fundido} \end{array}\right)}{\left(\begin{array}{c} \text{Masa de Fe en el} \\ \text{hierro fundido} \end{array}\right)} = \frac{4.5\% \text{ de C}}{95.5\% \text{ de Fe}} = 0.047$$

$$0 = - \left(\begin{array}{c} \text{Masa de C en el} \\ \text{hierro fundido} \end{array}\right) * 1 + \left(\begin{array}{c} \text{Masa de Fe en el} \\ \text{hierro fundido} \end{array}\right) * \left(\frac{\% \text{ de C}}{\% \text{ de Fe}}\right) \quad (3.11)$$

También se modificó la Ec. 3.5, que describe el balance para el carbono, ahora se debe añadir que el carbono también sale en el hierro.

$$0 = - \left(\begin{array}{c} \text{Masa de C} \\ \text{en el coque} \end{array}\right) * 1 + \left(\begin{array}{c} \text{Masa de CO en los} \\ \text{gases de salida} \end{array}\right) * 0.429 + \left(\begin{array}{c} \text{Masa de CO}_2 \text{ en los} \\ \text{gases de salida} \end{array}\right) * 0.273 + \left(\begin{array}{c} \text{Masa de C en el} \\ \text{hierro fundido} \end{array}\right) * 1 \quad (3.12)$$

La última consideración que se añadió fue el enriquecimiento con oxígeno del aire inyectado; esta es una práctica común en el alto horno. Para considerar esto en el SEAS se modificó la Ec. 3.1 que representa la composición del aire, de esta manera se puede modelar el sistema para cualquier composición de aire inyectado.

$$0 = - \left(\begin{array}{c} \text{Masa de N}_2 \text{ en el} \\ \text{aire inyectado} \end{array}\right) * 1 + \left(\begin{array}{c} \text{Masa de O}_2 \text{ en el} \\ \text{aire inyectado} \end{array}\right) * \left(\frac{\% \text{ de N}_2}{\% \text{ de O}_2}\right) \quad (3.13)$$

Para aplicar estas consideraciones se modificó ligeramente el SEAS; este cambio se puede ver en la Tabla 3.2. En los programas desarrollados en las subetapas anteriores es posible aplicar estos complementos modificando ligeramente el código que prepara el SEAS y lo resuelve; esto da muestra de lo sencillo que es ampliar las capacidades del SEH

Tabla 3.2 Arreglo matricial del sistema de ecuaciones algebraicas simultáneas que resultaron de la Etapa 1.

	Mineral	C en el coque	O ₂ en el aire inyectado	N ₂ en el aire inyectado	Fe en el hierro fundido	C en el hierro fundido	CO en los gases de salida	CO ₂ en los gases de salida	N ₂ en los gases de salida
Producción de hierro	1000	0	0	0	1	0	0	0	0
Balance para el hierro	0	- G *	0	0	1	0	0	0	0
Balance para el oxígeno	0	- F **	0	-1	0	0	0.571	0.727	0
Balance para el carbono	0	0	-1	0	0	1	0.429	0.273	0
Balance para el Nitrógeno	0	0	0	0	1	0	0	0	1
Composición del aire	0	0	0	%N ₂ /O ₂	-1	0	0	0	0
Carbono en el coque	400	0	1	0	0	0	0	0	0
O ₂ en el aire inyectado	370	0	0	1	0	0	0	0	0
Composición del hierro fundido	0	0	0	0	0	%C/%Fe	-1	0	0

$$* G = 0.724 * \left(\begin{matrix} \% \text{ magnetita} \\ \text{en el mineral} \end{matrix} \right) + 0.699 * \left(\begin{matrix} \% \text{ hematita} \\ \text{en el mineral} \end{matrix} \right)$$

$$** F = 0.276 * \left(\begin{matrix} \% \text{ magnetita} \\ \text{en el mineral} \end{matrix} \right) + 0.301 * \left(\begin{matrix} \% \text{ hematita} \\ \text{en el mineral} \end{matrix} \right)$$

3.2 Etapa 2. Incorporación del balance macroscópico de materia y energía

En la segunda etapa se buscó una manera más eficiente de conectar los lenguajes de programación seleccionados. Se encontró que es posible ejecutar el programa *SWI Prolog* dentro un código de programación escrito en *Python*.

La ejecución de los códigos elaborados en la Etapa 1 requiere que el usuario esté ejecutando alternativamente de *C++* a *Prolog*. Esto no es cómodo o intuitivo para el usuario; por esto se buscó una forma de unir en un solo código los dos lenguajes de programación. Se encontró que el lenguaje de programación *Python* se puede conectar con *Prolog* de una manera relativamente sencilla: *Python* puede ejecutar un programa de *SWI Prolog* para realizar consultas. Por lo que se decidió utilizar estos dos lenguajes para desarrollar el sistema experto híbrido en esta etapa; en *Python* se continuó con el desarrollo del modelo matemático y en *Prolog* se continuó construyendo la base de conocimiento.

Esta segunda etapa también se dividió en subetapas que se describen a continuación:

Subetapa 1: Desarrollo del balance macroscópico simultáneo de materia y energía, programado en *Python*.

Subetapa 2: Ampliación de la base de conocimiento del SE, programada en *Prolog*.

Subetapa 3: Integración de los dos sistemas informáticos.

3.2.1 Subetapa 1. Desarrollo y programación del balance macroscópico simultáneo de materia y energía

El desarrollo del balance de energía comenzó con un balance relativamente simple. Para ello se consideró lo siguiente:

- La energía que ingresa al sistema en forma calorífica lo hace a través de las materias primas que entran al alto horno (mineral, coque y aire inyectado).
- La energía calorífica sale en los productos gaseosos, el hierro líquido y una fracción se pierde a los alrededores.
- La mayor parte de las pérdidas de calor de un alto horno van a parar al agua de refrigeración.
- La temperatura de los gases de salida será una variable independiente.

En la Figura 3.11 se muestra un esquema del cual se desprende el balance de energía.

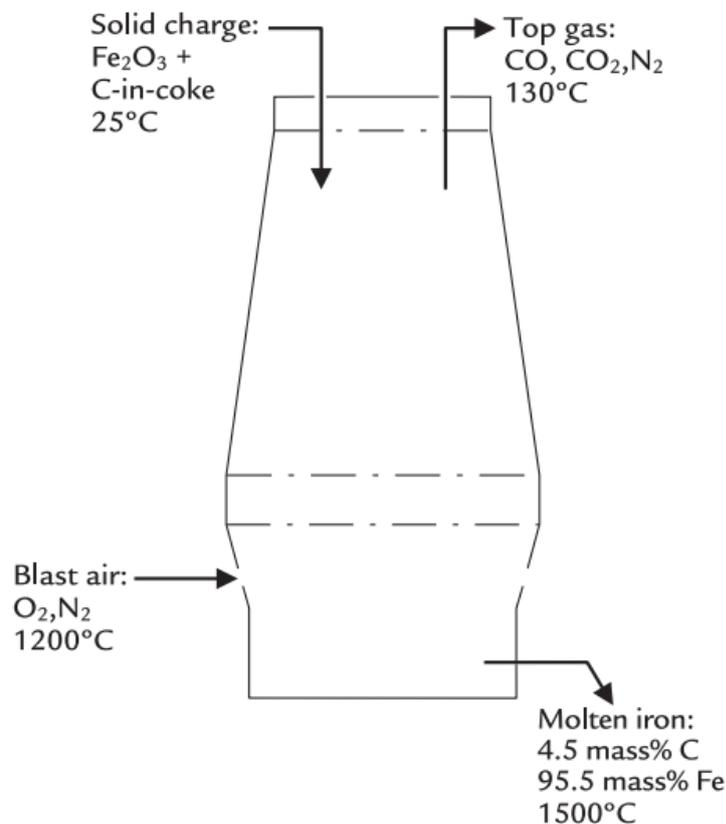


Figura 3.11 Esquema del alto horno que muestra las entradas y salidas de materia con sus respectivas temperaturas típicas [13].

Para incorporar el balance de energía al SEAS ya construido, este debe estar expresado en términos de las masas de cada una de las variables del sistema. Para comenzar, se establece un balance entálpico general para el alto horno, cuya ecuación general es:

Entalpía total de entrada = Entalpía total de salida + Pérdidas de calor por conducción, convección y radiación del horno

Que de manera formal se escribe de acuerdo con la Ec 3.14:

$$\sum_{i=1}^n m_i H_i \text{Entrada} = \sum_{j=1}^n m_j H_j \text{Salida} + \left[\begin{array}{l} \text{pérdidas de calor por conducción,} \\ \text{convección y radiación del horno} \end{array} \right] \quad (3.14)$$

Donde:

m_i y m_j son las masas de las sustancias de entrada y salida del horno, en kg.

H_i y H_j son las entalpías específicas, base masa, de las sustancias de entrada y salida, en MJ por kg de sustancia.

Las pérdidas de energía en forma de calor se encuentran en MJ.

De forma desarrollada la Ec. 3.14 quedaría de la siguiente forma:

Para las entradas:

$$\sum_{i=1}^n m_i H_i \text{Entrada} = \left[\begin{array}{l} \text{masa de } Fe_2O_3 \\ \text{en el mineral} \end{array} \right] * \left[\frac{\Delta H_{<Fe_2O_3>}^{25^\circ C}}{PM_{Fe_2O_3}} \right] + \left[\begin{array}{l} \text{masa de } C \\ \text{en el coque} \end{array} \right] * \left[\frac{\Delta H_{<C>}^{25^\circ C}}{PM_C} \right] + \\ \left[\begin{array}{l} \text{masa de } O_2 \text{ en} \\ \text{el aire inyectado} \end{array} \right] * \left[\frac{\Delta H_{[O_2]}^{1200^\circ C}}{PM_{O_2}} \right] + \left[\begin{array}{l} \text{masa de } N_2 \text{ en} \\ \text{el aire inyectado} \end{array} \right] * \left[\frac{\Delta H_{[N_2]}^{1200^\circ C}}{PM_{N_2}} \right] \quad (3.15)$$

Para las salidas:

$$\sum_{j=1}^n m_j H_j \text{Salida} = \left[\begin{array}{l} \text{masa de } Fe \text{ en el} \\ \text{arrabio producido} \end{array} \right] * \frac{\Delta H_{<Fe>}^{1500^\circ C}}{PM_{Fe}} + \left[\begin{array}{l} \text{masa de } C \text{ en el} \\ \text{arrabio producido} \end{array} \right] * \\ \frac{\Delta H_{(C)}^{1500^\circ C}}{PM_C} + \left[\begin{array}{l} \text{masa de } CO \text{ en los} \\ \text{gases de salida} \end{array} \right] * \frac{\Delta H_{[CO]}^{130^\circ C}}{PM_{CO}} + \left[\begin{array}{l} \text{masa de } CO_2 \text{ en los} \\ \text{gases de salida} \end{array} \right] * \frac{\Delta H_{[CO_2]}^{130^\circ C}}{PM_{CO_2}} + \\ \left[\begin{array}{l} \text{masa de } N_2 \text{ en los} \\ \text{gases de salida} \end{array} \right] * \frac{\Delta H_{[N_2]}^{130^\circ C}}{PM_{N_2}} \quad (3.16)$$

Donde:

ΔH_X^T : Entalpía de la sustancia X a la temperatura T, en MJ por kg-mol; tomando como 0 la entalpía de los elementos puros a 25°C y 1 bar.

PM_X : Peso molecular de la especie X, en kg por kg-mol.

La entalpía de reacción de la mezcla Fe + C a 1600°C con un 95.5 % en masa de Fe y un 4.5 % en masa de C es de 5.4 MJ/kg-mol de aleación, que es aproximadamente la entalpía de la mezcla a 1500°C. Este valor equivale a 2.5 MJ por kg de carbono disuelto, como se describe en el Apéndice A. Con este valor, la ecuación de entalpía del carbono es:

$$\frac{\Delta H_{(C)}^{1500^{\circ}C}}{PM_C} = \frac{\Delta H_{<C>}^{1500^{\circ}C}}{PM_C} + 2.5 \quad (3.17)$$

En la literatura se reporta la tasa de pérdida de calor en el alto horno como 400 MJ por 1000 kg de hierro producido; este valor se obtuvo a partir de mediciones del flujo y temperatura del agua de refrigeración [13].

Sustituyendo los valores de entalpías en la Ec. 3.14 se obtiene la siguiente expresión:

$$\begin{aligned} & \left[\begin{array}{l} \text{masa de } Fe_2O_3 \\ \text{en el mineral} \end{array} \right] * [-5.169] + \left[\begin{array}{l} \text{masa de C} \\ \text{en el coque} \end{array} \right] * [0] + \left[\begin{array}{l} \text{masa de } O_2 \text{ en} \\ \text{el aire inyectado} \end{array} \right] * \\ & [1.239] + \left[\begin{array}{l} \text{masa de } N_2 \text{ en} \\ \text{el aire inyectado} \end{array} \right] * [1.339] = \left[\begin{array}{l} \text{masa de Fe en el} \\ \text{arrabio producido} \end{array} \right] * [1.169] + \\ & \left[\begin{array}{l} \text{masa de C en el} \\ \text{arrabio producido} \end{array} \right] * [5] + \left[\begin{array}{l} \text{masa de CO en los} \\ \text{gases de salida} \end{array} \right] * [-3.836] + \\ & \left[\begin{array}{l} \text{masa de } CO_2 \text{ en los} \\ \text{gases de salida} \end{array} \right] * [-8.844] + \left[\begin{array}{l} \text{masa de } N_2 \text{ en los} \\ \text{gases de salida} \end{array} \right] * [0.1099] \\ & + 400 \text{ MJ} \end{aligned} \quad (3.18)$$

Reordenando esta ecuación se obtiene:

$$\begin{aligned} -400 \text{ MJ} = & - \left[\begin{array}{l} \text{masa de } Fe_2O_3 \\ \text{en el mineral} \end{array} \right] * [-5.169] - \left[\begin{array}{l} \text{masa de C} \\ \text{en el coque} \end{array} \right] * [0] \\ & - \left[\begin{array}{l} \text{masa de } O_2 \text{ en} \\ \text{el aire inyectado} \end{array} \right] * [1.239] - \left[\begin{array}{l} \text{masa de } N_2 \text{ en} \\ \text{el aire inyectado} \end{array} \right] * [1.339] \\ & + \left[\begin{array}{l} \text{masa de Fe en el} \\ \text{arrabio producido} \end{array} \right] * [1.169] + \left[\begin{array}{l} \text{masa de C en el} \\ \text{arrabio producido} \end{array} \right] * [5] \\ & + \left[\begin{array}{l} \text{masa de CO en los} \\ \text{gases de salida} \end{array} \right] * [-3.836] + \left[\begin{array}{l} \text{masa de } CO_2 \text{ en los} \\ \text{gases de salida} \end{array} \right] \\ & * [-8.844] + \left[\begin{array}{l} \text{masa de } N_2 \text{ en los} \\ \text{gases de salida} \end{array} \right] * [0.1099] \end{aligned} \quad (3.19)$$

Los valores de entalpía se calcularon a partir de las expresiones de entalpías en función de la temperatura del Apéndice A.

Ahora solo basta añadir esta última ecuación al SEAS de la Etapa 1; para no tener más ecuaciones que variables se reemplazó la ecuación de la especificación del oxígeno (Ec. 3.8) por la Ec. 3.19 (ver Tabla 3.3); esto permite calcular el requerimiento de oxígeno en el aire inyectado.

Al incorporar este nuevo SEAS al modelo matemático, ya se podría considerar al programa como una primera versión de un sistema experto híbrido de un alto horno basado en balances simultáneos de materia y energía.

Tabla 3.3 Arreglo matricial del sistema de ecuaciones algebraicas simultáneas que resultaron de los balances de materia y energía.

	Mineral	C en el coque	O ₂ en el aire inyectado	N ₂ en el aire inyectado	Fe en el hierro fundido	C en el hierro fundido	CO en los gases de salida	CO ₂ en los gases de salida	N ₂ en los gases de salida	
Producción de hierro	1000	0	0	0	1	0	0	0	0	
Balance para el hierro	0	- G *	0	0	1	0	0	0	0	
Balance para el oxígeno	0	- F **	0	-1	0	0	0.571	0.727	0	
Balance para el carbono	0	0	-1	0	0	1	0.429	0.273	0	
Balance para el Nitrógeno	0	0	0	0	1	0	0	0	1	
Composición del aire	0	0	0	%N ₂ /O ₂	-1	0	0	0	0	
Carbono en el coque	400	0	1	0	0	0	0	0	0	
Composición arrabio	0	0	0	0	0	%C/%Fe	-1	0	0	
Balance de entalpía	-400	5.169	0	-1.239	-1.339	1.269	5	-3.836	-8.844	0.1099

$$* G = 0.724 * \left(\begin{matrix} \% magnetita \\ en\ el\ mineral \end{matrix} \right) + 0.699 * \left(\begin{matrix} \% hematita \\ en\ el\ mineral \end{matrix} \right)$$

$$** F = 0.276 * \left(\begin{matrix} \% magnetita \\ en\ el\ mineral \end{matrix} \right) + 0.301 * \left(\begin{matrix} \% hematita \\ en\ el\ mineral \end{matrix} \right)$$

3.2.2 Subetapa 2. Ampliación de la base de conocimiento del SE, en *Prolog*

Lo que el SE realizaba hasta este punto de su desarrollo era analizar el estado de una de las variables (CO_2) del sistema; si se encuentra fuera del límite permisible propone e implementa una solución y al final da una explicación de lo que hizo, tal como se describió para la Etapa 1.

El único conocimiento que el SE contiene hasta este punto es sobre una de las especies (CO_2). Conoce el límite inferior permisible para esta especie, sabe cómo deducir si la cantidad de CO_2 en los gases de salida sobrepasa dicho límite, y qué hacer si eso pasa.

Por lo tanto, solo se puede preguntar por el estado del CO_2 . Sin embargo, una de las grandes ventajas de *Prolog* es que no hace falta rehacer por completo el código para complementar la base de conocimiento del SE; basta con añadir los hechos y reglas adecuadas para aumentar este conocimiento. En esta etapa se complementó la base de conocimiento del SE gradualmente. Primero se introdujeron los límites inferior y superior de cada una de las variables del sistema en forma de hechos, tal como se hizo con el CO_2 (ver Figura 3.12).

```

1
2 limite_superior(713,co2).
3 limite_superior(1403,fe2o3).
4 limite_superior(1500,fe3o4).
5 limite_superior(400,ccoque).
6 limite_superior(370,o2).
7 limite_superior(1218,n2in).
8 limite_superior(1100,fe).
9 limite_superior(50,cIron).
10 limite_superior(390,co).
11 limite_superior(1218,n2out).
12
13 limite_inferior(680,co2).
14 limite_inferior(1300,fe2o3).
15 limite_inferior(1200,fe3o4).
16 limite_inferior(300,ccoque).
17 limite_inferior(300,o2).
18 limite_inferior(900,n2in).
19 limite_inferior(900,fe).
20 limite_inferior(40,cIron).
21 limite_inferior(350,co).
22 limite_inferior(1000,n2out).
23

```

Figura 3.12 Captura de pantalla del código escrito en *Prolog*, definición de los límites permisibles.

Los elementos que se encuentran dentro de los paréntesis y comienzan con minúsculas se consideran constantes, se suelen ocupar para nombrar objetos, propiedades o relaciones.

Un primer gran avance podría ser generalizar la pregunta que se realiza sobre el estado del CO₂, de tal forma que el SE sea capaz de analizar el estado de cualquiera de las especies. Para esto se parte de una regla que en el lenguaje común quiere decir:

“Una especie X se encuentra fuera de rango cuando su valor sobrepasa su límite inferior o su límite superior, de lo contrario la especie X se encuentra dentro del rango permisible”; este enunciado escrito en el lenguaje de programación de *Prolog* se muestra en la Figura 3.13.

```

31 fuera_de_rango(Especie,Valor,Mensaje):-
32 limite_inferior(Limite,Especie),Valor<Limite,Mensaje='Se encuentra fuera de rango (menor)';
33 limite_superior(Limite,Especie),Valor>Limite,Mensaje='Se encuentra fuera de rango (mayor)';
34 limite_inferior(Limite1,Especie),Valor>Limite1,limite_superior(Limite2,Especie),Valor<Limite2,
35 Mensaje='Se encuentra dentro de rango'.

```

Figura 3.13 Captura de pantalla del código escrito en *Prolog*, regla sobre el estado de las especies.

Como se vio en la primera etapa la estructura básica de una regla en *Prolog* es la siguiente:

Cabeza :- cuerpo

Esta estructura se lee de la siguiente manera: “La cabeza es verdadera si el cuerpo es verdad”. Los elementos del “cuerpo” pueden estar separados por comas y/o punto y coma, donde la coma representa la conjunción “y” y el punto y coma representa la disyunción “o”.

Sabiendo esto, la regla anterior se lee textualmente de la siguiente forma:

El Valor de Especie está fuera de rango si Limite es el límite inferior de Especie y Valor es menor que Limite o si Limite es el límite superior de Especie y Valor es mayor que Limite o si Limite1 es el límite inferior de Especie y Valor es mayor que Limite1 y Limite2 es el límite superior de Especie y Valor es menor que Limite2.

Otra forma de representar esta regla en lenguaje natural es la siguiente: el Valor de Especie se encuentra fuera de rango si se presenta uno de los tres casos:

- 1) Limite es el límite inferior de Especie y Valor es menor que Limite
- 2) Limite es el límite superior de Especie y Valor es mayor que Limite
- 3) Limite1 es el límite inferior de Especie y Limite2 es el límite superior de Especie y Valor es mayor que Limite1 y menor que Limite2

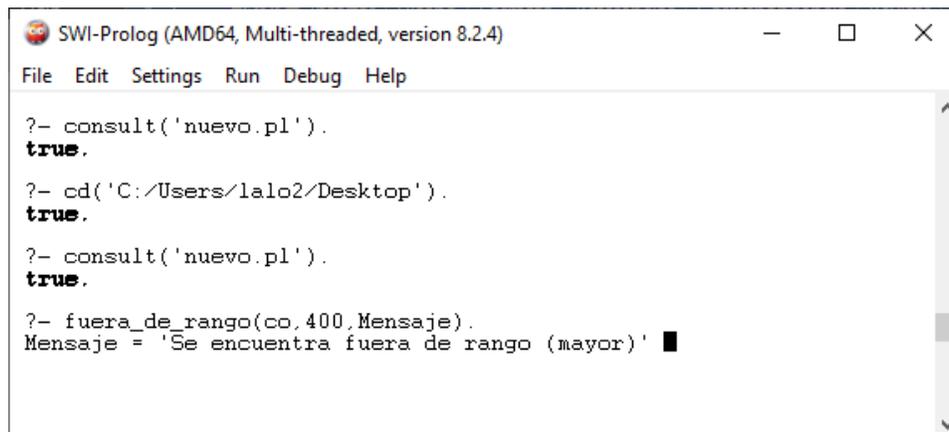
La última parte de esta regla parece contradictoria porque afirma que es verdadero que Especie se encuentra fuera de rango si su Valor se encuentra dentro de los límites permisibles, pero esto se debe a la etiqueta que se le colocó a esta regla

(fuera_de_rango), que bien podría ser “estado_de” en cuyo caso simplemente se tendría un mensaje u otro sobre el estado de la especie en función de su valor.

La variable Mensaje no afecta el resultado, simplemente permitirá más adelante comunicar a través de *Python* la conclusión a la que llegó el SE.

Entonces, cuando se hace una consulta, solo uno de los tres grupos de condiciones se satisface y devuelve el mensaje correspondiente.

Para realizar una consulta se deben sustituir las variables “Especies” y “Valor” que se encuentran dentro de los paréntesis por los datos de alguna de las variables. En la interfaz de *SWI Prolog* se realiza como se muestra en la Figura 3.14.



```
SWI-Prolog (AMD64, Multi-threaded, version 8.2.4)
File Edit Settings Run Debug Help

?- consult('nuevo.pl').
true.

?- cd('C:/Users/lalo2/Desktop').
true.

?- consult('nuevo.pl').
true.

?- fuera_de_rango(co, 400, Mensaje).
Mensaje = 'Se encuentra fuera de rango (mayor)' ■
```

Figura 3.14 Captura de pantalla de la interfaz de *SWI Prolog*, consulta sobre el estado de una de las variables.

En este caso como la cantidad de CO se encuentra por arriba del límite permisible la variable Mensaje se convierte en: Se encuentra fuera de rango (mayor).

Para complementar la base de conocimiento se implementó una nueva manera en la que el SE proponga una solución cuando alguna de las especies se encuentre fuera de rango. Ya que las variables del sistema están relacionadas de diferente manera entre sí, por ejemplo, el modificar la temperatura del aire inyectado, no afecta de igual forma a todas las variables, se deben establecer propuestas de solución específicas para cada especie. Para facilitar la comprensión del progreso de construcción de la base de conocimiento, en la siguiente parte de esta subetapa se utilizó al CO₂ como ejemplo.

Primero se estableció en forma de hechos las propuestas de solución (ver Figura 3.15), que provienen de saber que la temperatura del aire inyectado influye de manera inversamente proporcional sobre la cantidad de CO₂ producido.

```

24  solucion(+0.5,tblast,porArriba,co2).
25  solucion(-0.5,tblast,porAbajo,co2).
26

```

Figura 3.15 Captura de pantalla del código escrito en *Prolog*, definición de la propuesta de solución.

Para utilizar este nuevo conocimiento se modificó ligeramente la regla de la Figura 3.8 (ver Figura 3.16), de esta manera al realizar una consulta, el SE sugerirá la variable que debe ser modificada (aumentarla o disminuirla) en caso de que el CO₂ se encuentre fuera de rango (Ver Figura 3.17).

```

30 fuera_de_rango(Especie,Valor,Mensaje,T,Var):-
31 limite_inferior(Limite,Especie),Valor<Limite,Mensaje='Se encuentra fuera de rango (menor)',
32 solucion(T,Var,porArriba,Especie);
33 limite_superior(Limite,Especie),Valor>Limite,Mensaje='Se encuentra fuera de rango (mayor)',
34 solucion(T,Var,porAbajo,Especie);
35 limite_inferior(Limite1,Especie),Valor>Limite1,limite_superior(Limite2,Especie),Valor<Limite2,
36 Mensaje='Se encuentra dentro de rango',T is 0.
37

```

Figura 3.16 Captura de pantalla del código escrito en *Prolog*, nueva regla sobre el estado de las especies.



```

SWI-Prolog (AMD64, Multi-threaded, version 8.2.4)
File Edit Settings Run Debug Help

?- cd('C:/Users/lalo2/Desktop').
true.

?- consult('nuevo.pl').
true.

?- fuera_de_rango(co2,650,Mensaje,T,VarCorreccion).
Mensaje = 'Se encuentra fuera de rango (menor)',
T = 0.5,
VarCorreccion = tblast .

?- █

```

Figura 3.17 Captura de pantalla del código escrito en *Prolog*, consulta sobre el estado del CO₂.

Como se puede ver, el SE funciona de manera correcta cuando se realizan consultas sobre el CO₂; el siguiente paso es aumentar su capacidad para que sea capaz de realizar consultas sobre otras especies. La forma en que se escribió esta regla (ver Figura 3.16) permite realizar consultas para cada una de las especies, siempre y cuando se encuentren definidos sus límites permisibles y una posible solución cuando sobrepasa dichos límites; estos límites no se pueden establecer de manera arbitraria. Como no todas las variables se relacionan de la misma forma entre sí, se graficó el comportamiento de cuatro variables del alto horno (CO, CO₂, N₂ y O₂) en función de la temperatura del aire inyectado y la cantidad de carbono en el coque cargado, para determinar un rango permisible válido de esas especies. Se deben fijar los rangos permisibles de tal forma que exista una zona de coincidencia entre las variables. Para establecer los rangos permisibles primero se seleccionó un rango de temperaturas para dos variables diferentes (CO₂ y CO) (ver Figura 3.18) después se verificó que se mantuviera una zona de coincidencia al modificar la cantidad de carbono en el coque cerca de los límites de temperatura seleccionados (1100-1450°C). Esto se consiguió graficando el comportamiento de estas variables en función de la cantidad de carbono manteniendo fija la temperatura del aire inyectado (ver Figura 3.19), después de observar con detenimiento las gráficas de las Figuras 3.13 y 3.14 y colocar en ellas diferentes combinaciones de límites se encontró un par de rangos permisibles válidos para ambas variables, los cuales van de 684 a 715 kg para el CO₂ y de 360 a 392 kg para el CO; de esta forma nos aseguramos que siempre se mantenga una zona de coincidencia al modificar ya sea la temperatura del aire inyectado o la cantidad de carbono en el coque. Este paso lleva algo de tiempo, pero de no hacer este análisis al establecer los límites permisibles, el SE no podrá encontrar una solución; además, deben ser valores típicos en la operación de un alto horno.

En las Figuras 3.18 y 3.19 las líneas punteadas de color azul representan los límites permisibles para el CO₂ y las de color verde los límites del CO.

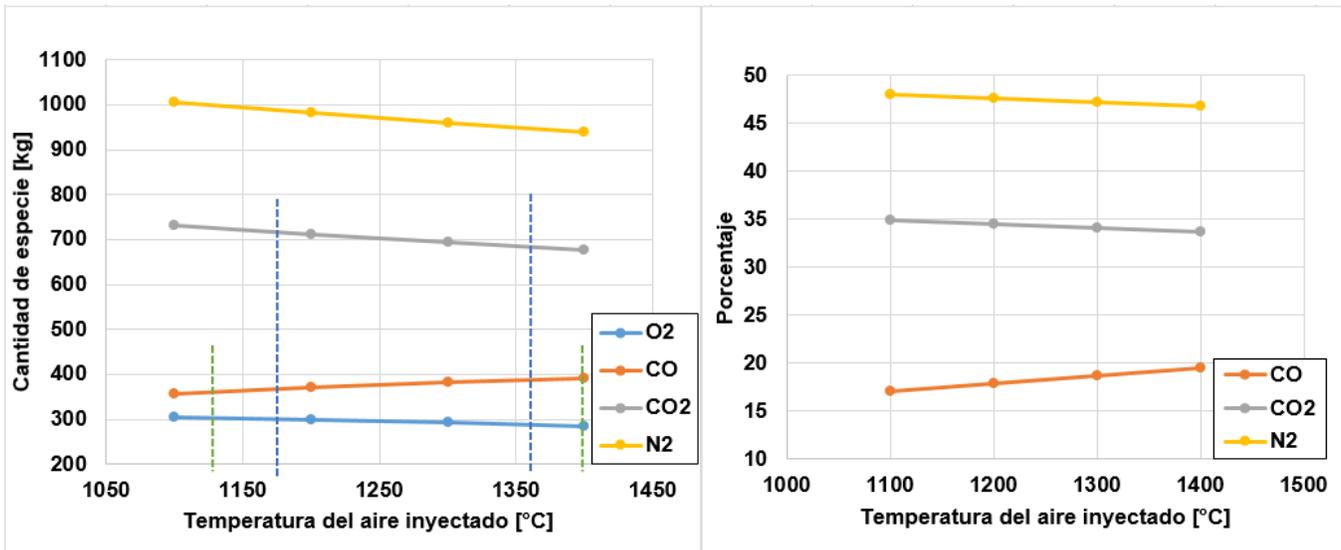


Figura 3.18 Gráfica del comportamiento del CO, CO₂, N₂ y O₂ en función de la temperatura del aire inyectado.

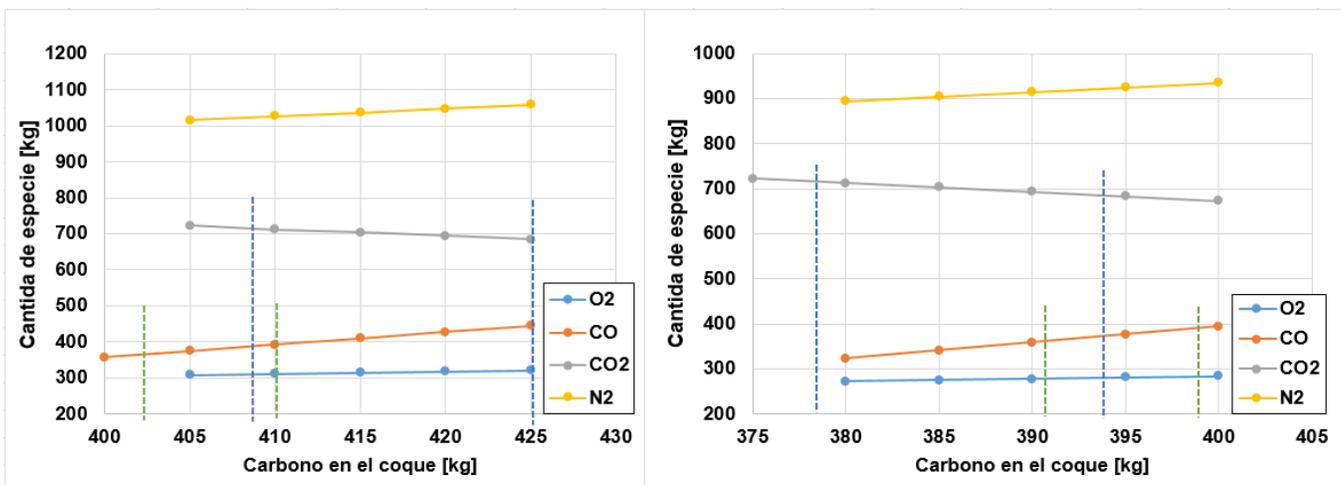


Figura 3.19 Gráfica del comportamiento del CO, CO₂, N₂ y O₂ en función de la cantidad de carbono en el coque. La temperatura del aire inyectado para la gráfica de la izquierda se fijó en 1100°C y para la de la derecha en 1450°C.

Evidentemente existe más de una manera de corregir el valor de una especie que se encuentra fuera de rango; por ello es necesario construir un criterio de elección cuando se presente el caso donde exista más de una solución para el mismo problema.

La última parte de esta subetapa se realizó de forma hipotética con el objetivo de mostrar cómo se construye el conocimiento dentro de *Prolog*. El criterio que se decidió fue que el SE debe elegir la solución que genere el menor gasto o el mayor ahorro económico; se consideró tomar en cuenta sólo dos posibles soluciones para cada especie.

Primero se planteó en forma de hechos un costo unitario hipotético y la definición de una expresión para calcular cuánto se pretende modificar la variable de corrección (ver Figura 3.20).

```

35 costoUnitario(20,tblast).
36 costoUnitario(15,ccoque).
37
38 aumento(0.01*PorcentajeEspecie*ValInicial,PorcentajeEspecie,ValInicial).
39

```

Figura 3.20 Captura de pantalla del costo unitario programado en *Prolog*.

A partir del costo unitario y conociendo cuánto se pretende modificar las variables de corrección propuestas se puede calcular el gasto o ahorro que genera la aplicación de ambas propuestas utilizando una nueva regla (ver Figura 3.21); teniendo esta información se puede plantear otra regla para seleccionar la solución con el menor gasto o el mayor ahorro económico (ver Figura 3.21).

```

43 solucion_final(Correccion1,Correccion2,V1cambio,V2cambio,Varf,Edo,Especie,
44 | Var1,Porcentaje1,ValInic1,Var2,Porcentaje2,ValInic2):-
45 solucion(T1,Var1,Edo,Especie),Tf1 is Porcentaje1+T1,costo(CostoA,Var1,Tf1,ValInic1),
46 solucion(T2,Var2,Edo,Especie),Tf2 is Porcentaje2+T2,costo(CostoB,Var2,Tf2,ValInic2),
47 CostoA<CostoB,Varf=Var1,Correccion1=Tf1,V1cambio=T1,V2cambio=0,Correccion2=0;
48 solucion(T1,Var1,Edo,Especie),Tf1 is Porcentaje1+T1,costo(CostoA,Var1,Tf1,ValInic1),
49 solucion(T2,Var2,Edo,Especie),Tf2 is Porcentaje2+T2,costo(CostoB,Var2,Tf2,ValInic2),
50 CostoA>CostoB,Varf=Var2,Correccion2=Tf2,V2cambio=T2,V1cambio=0,Correccion1=0;
51 solucion(V1cambio,Var1,Edo,Especie),no_solucion(Var2,Especie),Varf=Var1,V2cambio=0,
52 Correccion1 is Porcentaje1+V1cambio,Correccion2=0;
53 solucion(V2cambio,Var2,Edo,Especie),no_solucion(Var1,Especie),Varf=Var2,V1cambio=0,
54 Correccion2 is Porcentaje2+V2cambio,Correccion1=0.
55

```

Figura 3.21 Captura de pantalla de la definición de las reglas de selección.

Ahora solo queda modificar nuevamente la regla “fuera_de_rango” para incluir el criterio de selección económico para el caso en el que se deba elegir entre dos propuestas de solución para un mismo problema (ver Figura 3.22). Como se ha venido haciendo, la forma en que se programó esta regla permite hacer consultas

sobre cualquier variable, siempre y cuando se encuentren definidos sus límites permisibles y al menos una sugerencia de solución, con la opción de añadir una segunda sugerencia de solución. Es importante señalar que si una variable solo cuenta con una sugerencia de solución se debe señalar en el código qué variable no se considera sugerencia de solución para esta variable. En la Figura 3.23 se muestra un ejemplo donde se establece que la variable “ccoque” (carbono en el coque) no es solución de la variable “co” (CO).

```

55 fuera_de_rango(Especie,Valor,Mensaje,Tf1,Tf2,V1,V2,Var,Var1,Porcentaje1,ValInic1,Var2,Porcentaje2,ValInic2):-
56 limite_inferior(Limite,Especie),Valor<Limite,Mensaje='Se encuentra fuera de rango (menor)',
57 solucion_final(Tf1,Tf2,V1,V2,Var,porAbajo,Especie,Var1,Porcentaje1,ValInic1,Var2,Porcentaje2,ValInic2);
58 limite_superior(Limite,Especie),Valor>Limite,Mensaje='Se encuentra fuera de rango (mayor)',
59 solucion_final(Tf1,Tf2,V1,V2,Var,porArriba,Especie,Var1,Porcentaje1,ValInic1,Var2,Porcentaje2,ValInic2);
60 limite_inferior(Limite1,Especie),Valor>Limite1,limite_superior(Limite2,Especie),Valor<Limite2,
61 Mensaje='Se encuentra dentro de rango',V1=0,V2=0,Var='Ok'.
62

```

Figura 3.22 Captura de pantalla de la nueva regla para realizar consultas sobre el estado de las variables.

```

33 no_solucion(ccoque,co).
34

```

Figura 3.23 Captura de pantalla donde se muestra como señalar que una variable no es sugerencia de solución de otra.

Las equivalencias de los elementos que se encuentran dentro de los paréntesis en “fuera_de_rango” se describen en la siguiente tabla:

Tabla 3.4 Nombre y descripción de los elementos que componen la regla escrita en *Prolog* para consultar el estado de las variables del sistema.

Nombre	Descripción
Especie	Nombre de la variable sobre la cual se hace la consulta
Valor	Valor de la variable sobre la cual se hace la consulta
Mensaje	Variable que expresa en palabras el resultado de la consulta
Tf1	Porcentaje en que se debe modificar la variable solución 1
Tf2	Porcentaje en que se debe modificar la variable solución 2
V1	Porcentaje de corrección que debe aplicar el SE sobre la variable solución 1

Tabla 3.4 (Cont.) Nombre y descripción de los elementos que componen la regla escrita en *Prolog* para consultar el estado de las variables del sistema.

V2	Porcentaje de corrección que debe aplicar el SE sobre la variable solución 2
Var	Variable que el SE recomienda modifica
Var1	Nombre de la variable solución 1
Var2	Nombre de la variable solución 2
Porcentaje1	Último porcentaje en que se ha modificado la variable solución 1
Porcentaje2	Último porcentaje en que se ha modificado la variable solución 1
Vallnic1	Valor original de la variable solución 1
Vallnic2	Valor original de la variable solución 2

Las variables “Var1” y “Var2” solo pueden tomar uno de los siguientes valores:

- tblast: temperatura del aire inyectado.
- ccoque: cantidad de carbono en el coque.

Al realizar una consulta solamente los elementos: Especie, Valor, Var1, Var2, Porcentaje1, Porcentaje2, Vallnic1 y Valnic2 se deben sustituir por los datos del alto horno; los demás elementos son variables que tomarán un valor cuando termine la consulta. En la Figura 3.24 se muestra un ejemplo de cómo realizar una consulta.

```

SWI-Prolog (AMD64, Multi-threaded, version 8.2.4)
File Edit Settings Run Debug Help
?- cd('C:/Users/lalo2/Documents/T/Version 3').
true.
?- consult('Pruebas BE 3.pl').
true.
?- fuera_de_rango(co2,690,Estado,P1,P2,Cambio1,Cambio2,Varf,tblast,0,1500,ccoque,0,400).
Estado = 'Se encuentra fuera de rango (menor)',
P1 = Cambio1, Cambio1 = -0.5,
P2 = Cambio2, Cambio2 = 0,
Varf = tblast ,
?- █
  
```

Figura 3.24 Captura de pantalla, consulta sobre el estado del CO₂ cuando existen dos soluciones diferentes.

Este es el ejemplo de una primera consulta, donde no se ha modificado ninguna de las variables solución (temperatura del aire inyectado y cantidad de carbono en el coque). El resultado de la consulta es que el CO₂ se encuentra fuera de rango y la sugerencia de solución es reducir un 0.5% (Cambio1 = - 0.5) la temperatura del aire inyectado (Varf = tblast).

En esta Subetapa 2 solo se planteó una dirección en la construcción de la base de conocimiento, pero no es la única y probablemente aún se podrían pulir algunos aspectos del código, pero no se pretendía cubrir todo el conocimiento en torno al alto horno; el objetivo fue mostrar una manera de construir la base de conocimiento de un SE híbrido aplicado a sistemas metalúrgicos.

3.2.3 Subetapa 3. Integración de los dos sistemas informáticos

El objetivo de esta Subetapa fue desarrollar un código en *Python* que permita unir la base de conocimiento desarrollada en la Subetapa 2 con el modelo matemático basado en balances macroscópicos simultáneos de materia y energía desarrollado en la Subetapa 1 de la Etapa 2.

La diferencia más significativa entre el código escrito en *C++* y *Python* es el método que se utilizó para resolver el SEAS: en *C++* se programó el método Gauss-Jordan “desde cero”; en *Python* se utilizó una biblioteca diseñada para aplicar el método de matriz inversa. La secuencia en la que se utilizan a *Prolog* y *Python* sigue una estrategia similar a la que se describe en la Figura 3.5, solo que ahora no es necesario exportar e importar datos de un archivo de texto, simplemente se deben escribir algunas líneas de código parecidas a las consultas que se hacen a través de la interfaz de *SWI Prolog*. La nueva secuencia de pasos se describe en la Figura 3.25. Cuando se menciona la aplicación de la corrección implica que se está resolviendo el SEAS.

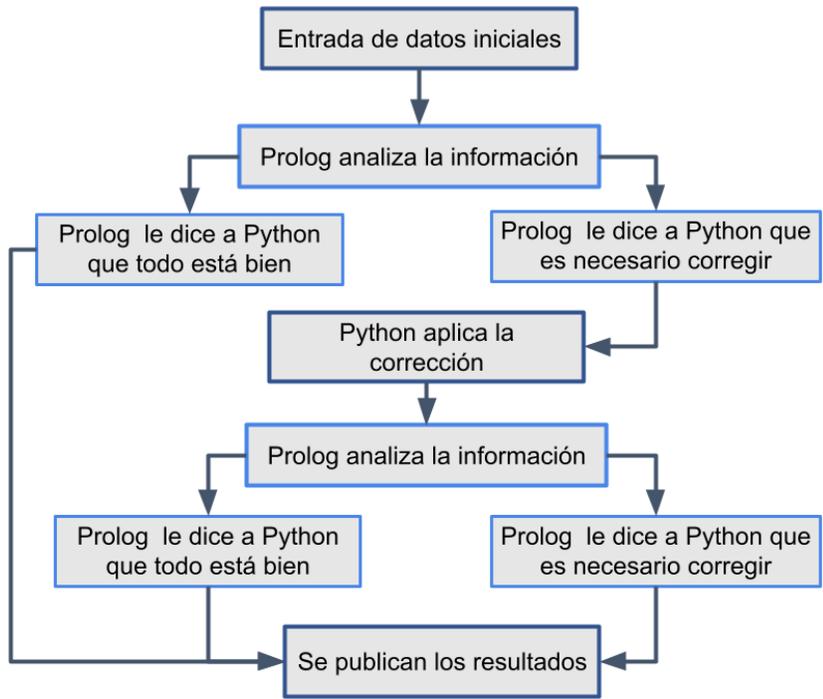


Figura 3.25 Diagrama de flujo grueso del código elaborado en la Subetapa 3.

Al igual que en la Etapa 1, solamente se ha mostrado una cantidad limitada de consultas. Después de recibir los datos iniciales sobre el estado del alto horno, se realiza la consulta en *Prolog* sobre la cantidad de CO₂ en los gases de salida. En caso de ser necesario, se aplica la corrección pertinente; con los nuevos valores se realiza otra consulta y, sea cual sea el resultado, el programa termina, aunque el problema aún no se haya resuelto.

Para dar solución al problema anterior, se escribió un código en *Python* de tal forma que se realizan las consultas y correcciones necesarias. Este código, al igual que los anteriores, comienza con la entrada de los datos iniciales. En seguida se realiza una primera consulta en *Prolog*; si el CO₂ se encuentra dentro del rango permisible la ejecución termina, en el caso contrario se inicia un ciclo en el cual mientras *Prolog* arroje una respuesta incorrecta (CO₂ fuera de rango) se sigue aplicando una corrección. El ciclo termina en cuanto *Prolog* determina que el CO₂ se encuentra dentro de rango. Las correcciones que hace *Prolog* están en función del número de consultas que se hagan; la ejecución del programa concluye con una explicación de qué fue lo que se hizo para solucionar el problema y la presentación de los datos

finales después de la última corrección. Esta secuencia de tareas realizadas en el código de *Python* se muestra en la Figura 3.26.

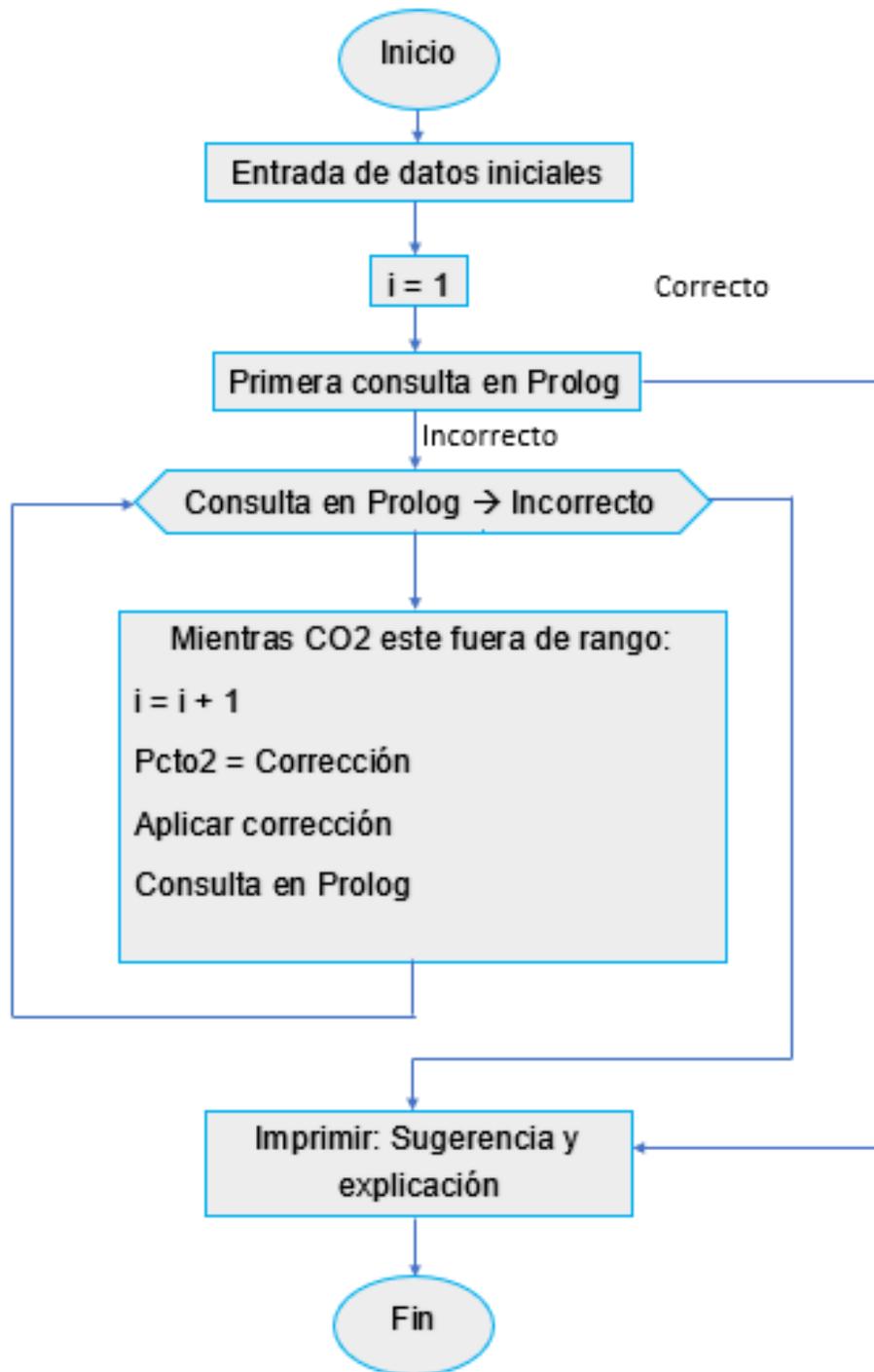


Figura 3.26 Diagrama de flujo grueso del código final.

En esta versión del código solo se puede consultar sobre el estado de una especie a la vez, para abarcar todas las variables del sistema dentro de una sola ejecución del código se debe realizar una consulta individual para cada una de las variables dentro del ciclo de consulta. De esta manera, cada iteración dentro del ciclo involucra una serie de consultas sobre el estado de las variables; esta serie de consultas están limitadas por la extensión de la base de conocimiento ya que para realizar consultas sobre una variable deben estar definidos sus límites permisibles y al menos una propuesta de solución en caso de encontrarse fuera de rango. En este caso solo se pueden realizar consultas para dos variables: CO y CO₂

Este código final permite que el usuario introduzca el valor de catorce variables diferentes y dos parámetros que describen las condiciones del alto horno. El modelo matemático desarrollado permite al SE modificar las siguientes variables: composición del mineral cargado (hematita-magnetita), composición del hierro producido (Fe + C), porcentaje de oxígeno en el aire inyectado, cantidad de carbono en el coque cargado, la temperatura de del aire inyectado, del hierro producido y de los gases de salida.

VARIABLES QUE CAPTURA EL PROGRAMA:

- Masa en kg de:
 - ❖ Hematita (Fe₂O₃) en el mineral cargado
 - ❖ Magnetita (Fe₃O₄) en el mineral cargado
 - ❖ Carbono (C) en el coque cargado
 - ❖ Oxígeno (O₂) en el aire inyectado
 - ❖ Nitrógeno (N₂) en el aire inyectado
 - ❖ Hierro (Fe) en el arrabio producido
 - ❖ CO en los gases de salida
 - ❖ CO₂ en los gases de salida
 - ❖ N₂ en los gases de salida
- Temperatura en °C de:
 - ❖ Aire inyectado
 - ❖ Mineral

- ❖ Coque
- ❖ Hierro fundido
- ❖ Gases de salida (CO, CO₂ y N₂)

Consideraciones adicionales:

- Composición del arrabio producido (C + Fe)
- Enriquecimiento del aire inyectado con oxígeno

3.3 Etapa 3. Construcción de la interfaz gráfica de usuario

La interacción con el sistema experto de la forma que se venía haciendo no era la mejor, ya que al ejecutar el programa se debían introducir los datos iniciales en la terminal del Entorno de Desarrollo Integrado (*IDE*) de *Python* (zona punteada de la Figura 3.27) lo cual no es nada cómodo, principalmente por el espacio tan reducido. Por ello, se desarrolló una interfaz de usuario para interactuar con el sistema experto de una forma más cómoda, interactiva e intuitiva.

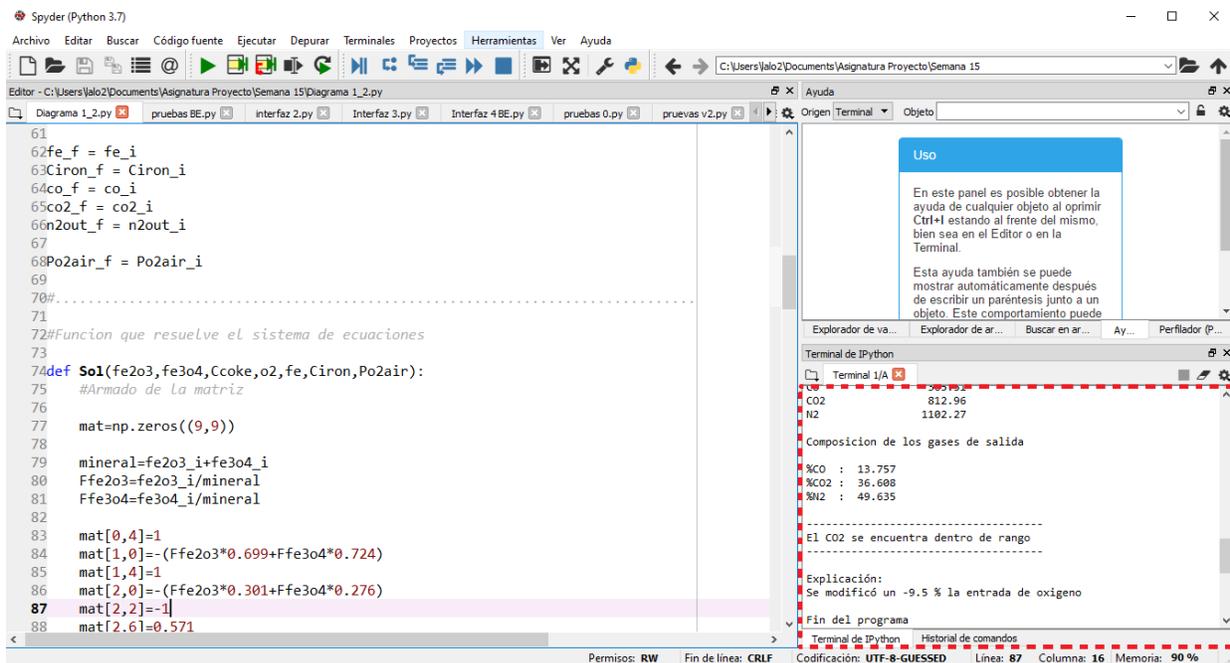


Figura 3.27 Entorno de desarrollo integrado de *Python*

Para la construcción de la interfaz de usuario se utilizó la biblioteca Tkinter para *Python*. Primero se definió una raíz/ventana la cual va a contener todos los *widgets*

/elementos de la interfaz. Enseguida se definieron un par de marcos dentro de la ventana: en uno de ellos se colocaron cuadros de texto con su respectiva etiqueta donde el usuario puede introducir los datos iniciales: el otro marco se destinó para mostrar los resultados de las consultas hechas al SE, lo que incluye mensajes acerca del estado de la variable sobre la cual se realiza la consulta, una explicación de la solución aplicada y los nuevos valores de las variables después de aplicar dicha sugerencia de solución.

En la primera ventana también se colocaron botones que permiten hacer consultas después de haber ejecutada el código y modificar los datos iniciales sin necesidad de cerrar y volver a ejecutar el programa. Para estos botones se definieron funciones de tal forma que al pulsarlos se llama a la función y se ejecuta el ciclo de consultas descrito en la Figura 3.26; también se colocó un botón para calcular la composición de los gases de salida. La interfaz resultante se muestra en las Figuras 3.28 y 3.29.

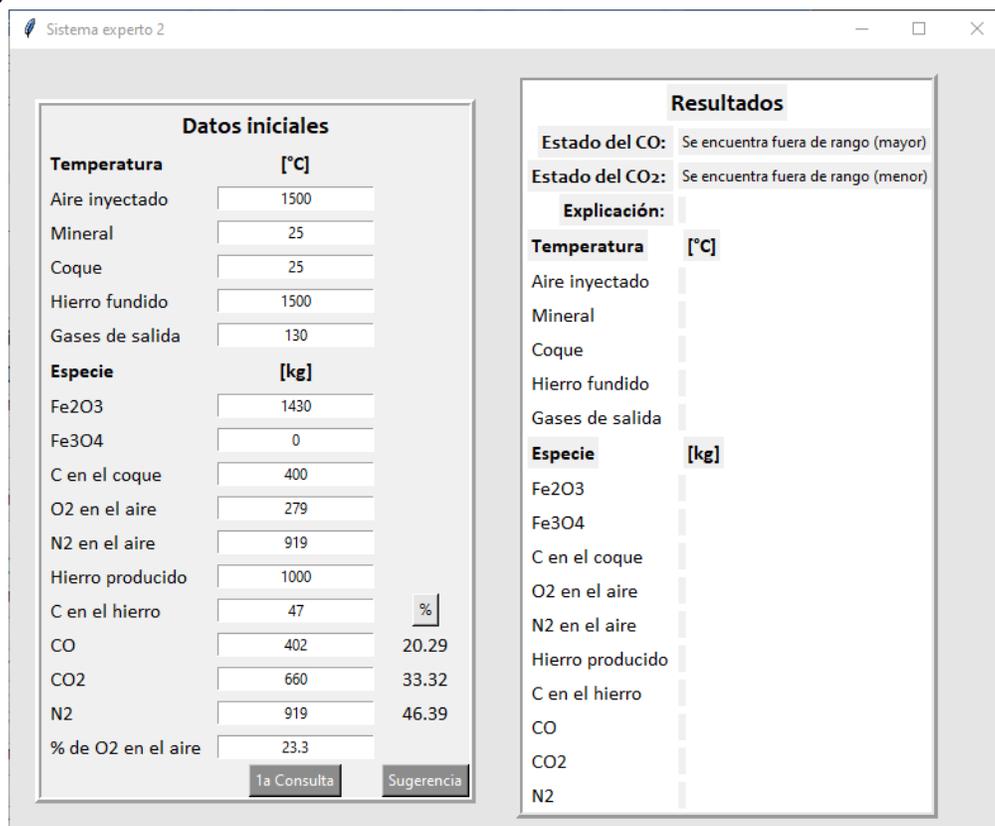


Figura 3.28 Captura de pantalla de la interfaz gráfica del SE híbrido, después de introducir los datos iniciales y oprimir los botones “%” y “1a Consulta”.

Sistema experto 2

—
□
×

Datos iniciales

Temperatura	[°C]
Aire inyectado	1500
Mineral	25
Coque	25
Hierro fundido	1500
Gases de salida	130

Especie	[kg]
Fe2O3	1430
Fe3O4	0
C en el coque	400
O2 en el aire	279
N2 en el aire	919
Hierro producido	1000
C en el hierro	47
CO	402
CO2	660
N2	919
% de O2 en el aire	23.3

%
1a Consulta
Sugerencia

Resultados

Estado del CO: Se encuentra dentro de rango

Estado del CO2: Se encuentra dentro de rango

Explicación: Se modifico un -9.5% la temperatura de los gases

Temperatura	[°C]
Aire inyectado	1357.5
Mineral	25.0
Coque	25.0
Hierro fundido	1500.0
Gases de salida	130.0

Especie	[kg]
Fe2O3	1430.62
Fe3O4	0.0
C en el coque	400.0
O2 en el aire	288.02
N2 en el aire	948.12
Hierro producido	1000.0
C en el hierro	47.0
CO	387.46
CO2	684.18
N2	948.12

Figura 3.29 Captura de pantalla de la interfaz gráfica del SE híbrido, después de introducir los datos iniciales y oprimir los botones “%” y “Sugerencia”.

Capítulo 4. Resultados y análisis

En este capítulo se presentan y discuten los resultados para los diferentes casos que se pueden presentar durante una consulta al SE híbrido desarrollado en el capítulo anterior.

En las Figuras 4.1 y 4.2 se reportan los resultados para uno de los casos que se puede presentar: el caso en el que la cantidad de CO₂ se encuentra fuera de los límites permisibles.

En la Figura 4.1 se presenta el primer paso de la consulta al SE, el cual es recabar el valor de diferentes parámetros que describen el estado actual del alto horno. Después de llenar todos los campos de la sección “Datos iniciales” el usuario puede realizar una primera consulta sobre el estado de las variables presionando el botón “1ra consulta”, en esta versión el SE híbrido solo analiza el estado de dos variables, el CO y CO₂; también se puede calcular la proporción de los gases de salida (CO, CO₂ y N₂) con el botón “%”.

The screenshot shows a software interface titled "Sistema experto 2". It is divided into two main panels: "Datos iniciales" (Initial Data) and "Resultados" (Results).

Datos iniciales: This panel contains two tables for inputting data. The first table is for "Temperatura [°C]" and the second is for "Especie [kg]". The "CO" and "CO2" rows in the second table are highlighted with a red border.

Temperatura [°C]	
Aire inyectado	1450
Mineral	25
Coque	25
Hierro fundido	1500
Gases de salida	130

Especie [kg]	
Fe2O3	1430
Fe3O4	0
C en el coque	400
O2 en el aire	282
N2 en el aire	929
Hierro producido	1000
C en el hierro	47
CO	397
CO2	668
N2	929
% de O2 en el aire	23.3

Buttons: "1a Consulta" and "Sugerencia".

Resultados: This panel displays the system's output. It shows the status of CO and CO2, an explanation, and a table for gas composition.

Estado del CO: Se encuentra fuera de rango (mayor)
Estado del CO2: Se encuentra fuera de rango (menor)
Explicación:

Temperatura [°C]	
Aire inyectado	
Mineral	
Coque	
Hierro fundido	
Gases de salida	

Especie [kg]	
Fe2O3	
Fe3O4	
C en el coque	
O2 en el aire	
N2 en el aire	
Hierro producido	
C en el hierro	
CO	
CO2	
N2	

Figura 4.1 Captura de pantalla de los datos iniciales solicitados por el SE.

Como los rangos permisibles para el CO y CO₂ se establecieron de 360 a 392 kg y de 684 a 715 kg respectivamente es evidente que no se cumple con este criterio. El programa se encarga de identificar este problema y modificar algún parámetro para corregirlo.

En la Figura 4.2 se muestran los resultados obtenidos tras presionar el botón “Sugerencia”. Esto implica que de manera cíclica el SE realiza consultas en *Prolog*, aplica las correcciones sugeridas por las consultas y resuelve el SEAS resultante hasta que las variables se encuentren dentro del rango permissible. Además de reportar el valor de los nuevos parámetros también se da una explicación cuantitativa de cómo se llegó a esos resultados.

En la Figura 4.2 se puede ver que las cantidades de CO y CO₂ ahora se encuentran dentro del rango permissible, lo que se consiguió al disminuir en un 6.5 % (con respecto al valor inicial) la temperatura del aire inyectado.

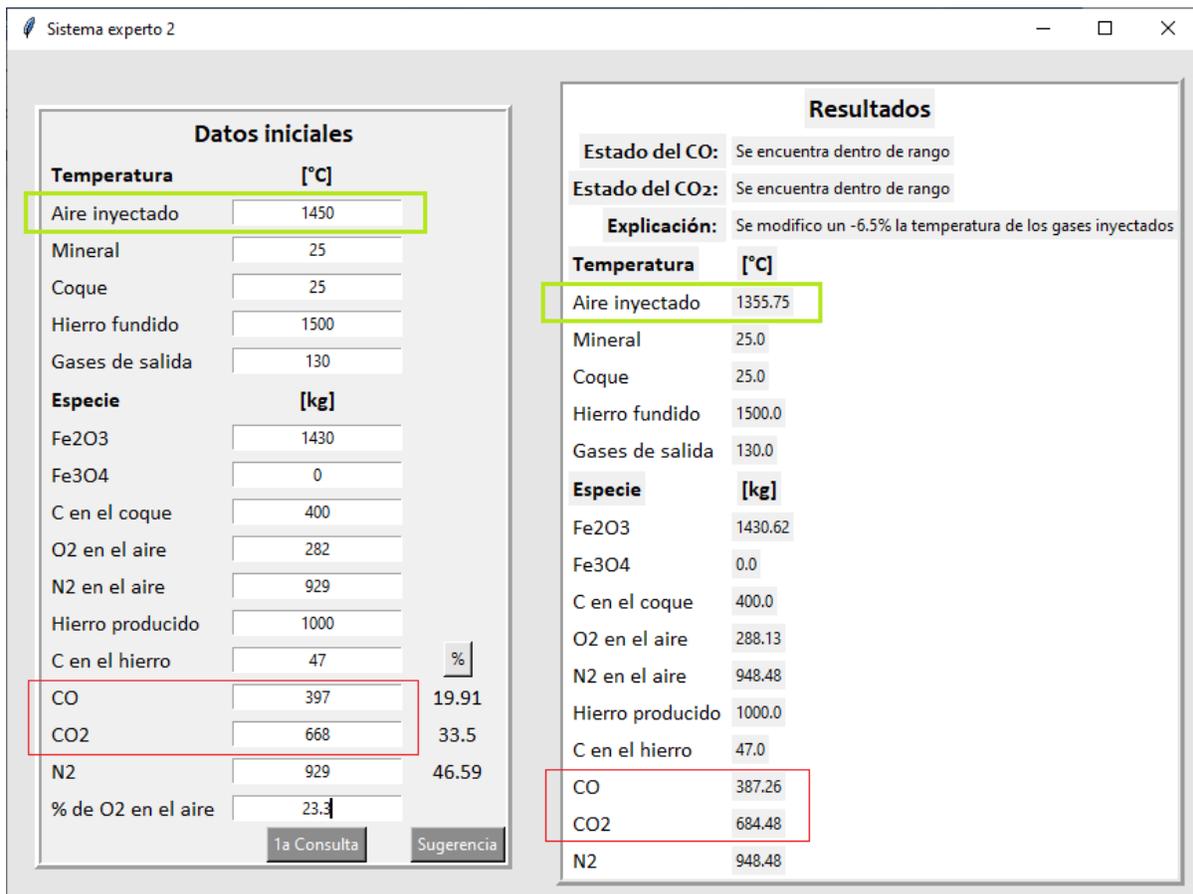


Figura 4.2 Captura de pantalla de los resultados obtenidos por el SE. En este ejemplo, el contenido de CO₂ y CO está fuera del rango admisible.

En la Figura 4.4 se puede ver que después de presionar el botón “Sugerencia” se actualizan los parámetros de la sección “Resultados”. Estos son prácticamente los mismos que introduce el usuario en la sección “Datos iniciales” solo que, con los decimales faltantes, ya que para completar esos campos el SE resuelve el SEAS con los parámetros que introdujo el usuario. Como esta vez no hace falta realizar más consultas o alguna corrección, el SE reporta que se realizó una modificación del 0%.

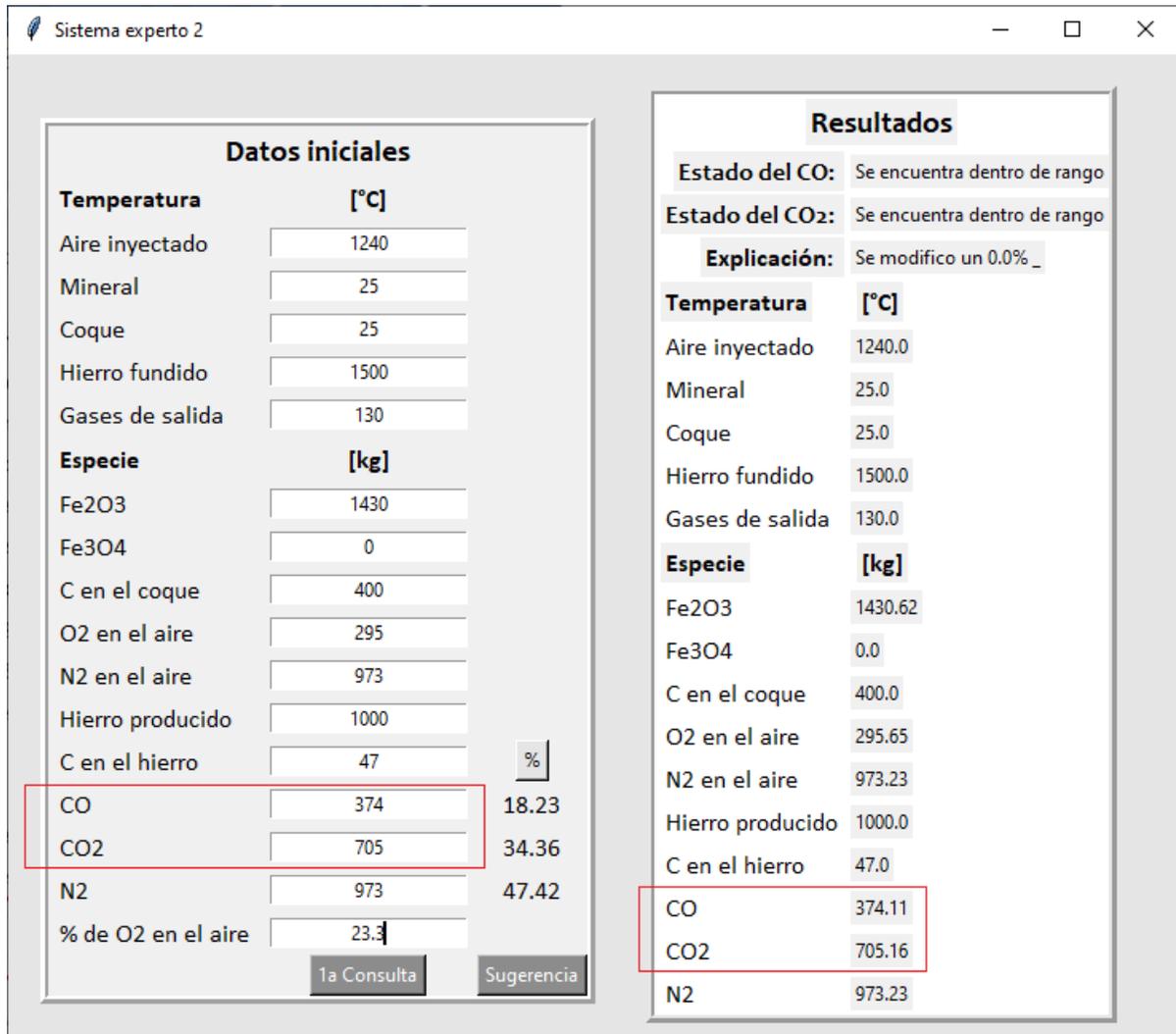


Figura 4.4 Captura de pantalla de los resultados obtenidos por el SE. En este ejemplo, el contenido de CO₂ y CO está dentro del rango admisible.

En los casos anteriores no se tomó en cuenta el criterio de selección económico desarrollado en el capítulo anterior; en los casos siguientes si se tomará en cuenta para comprobar y observar su funcionamiento.

En el caso donde solo una de las variables que se analiza se encuentre fuera de rango, si existen dos propuestas de solución, se aplicará el criterio de selección más económico, de lo contrario el SE sugerirá aplicar la única propuesta de solución disponible. Por ejemplo, en el caso que se presenta en las Figuras 4.5 y 4.6 solo el CO se encuentra fuera del rango permisible; como la base de conocimiento cuenta con una propuesta de solución para cuando el CO esté fuera de rango, solo se sugerirá y aplicará esa sugerencia de solución. En la figura 4.6 se puede ver que hizo falta aumentar un 3% la temperatura del aire inyectado para mantener dentro del rango permisible la cantidad de CO producido.

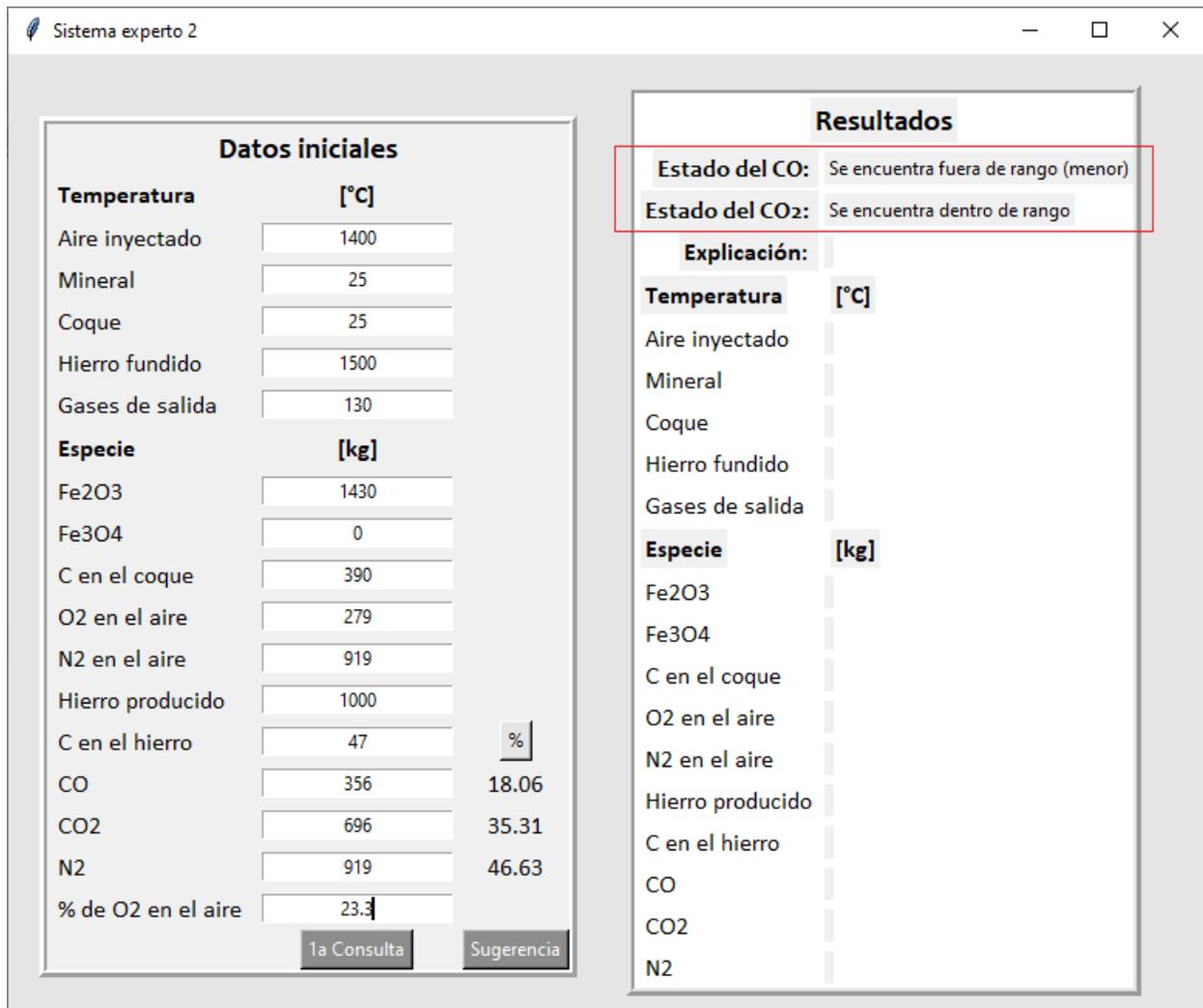


Figura 4.5 Captura de pantalla de los datos iniciales solicitados por el SE.

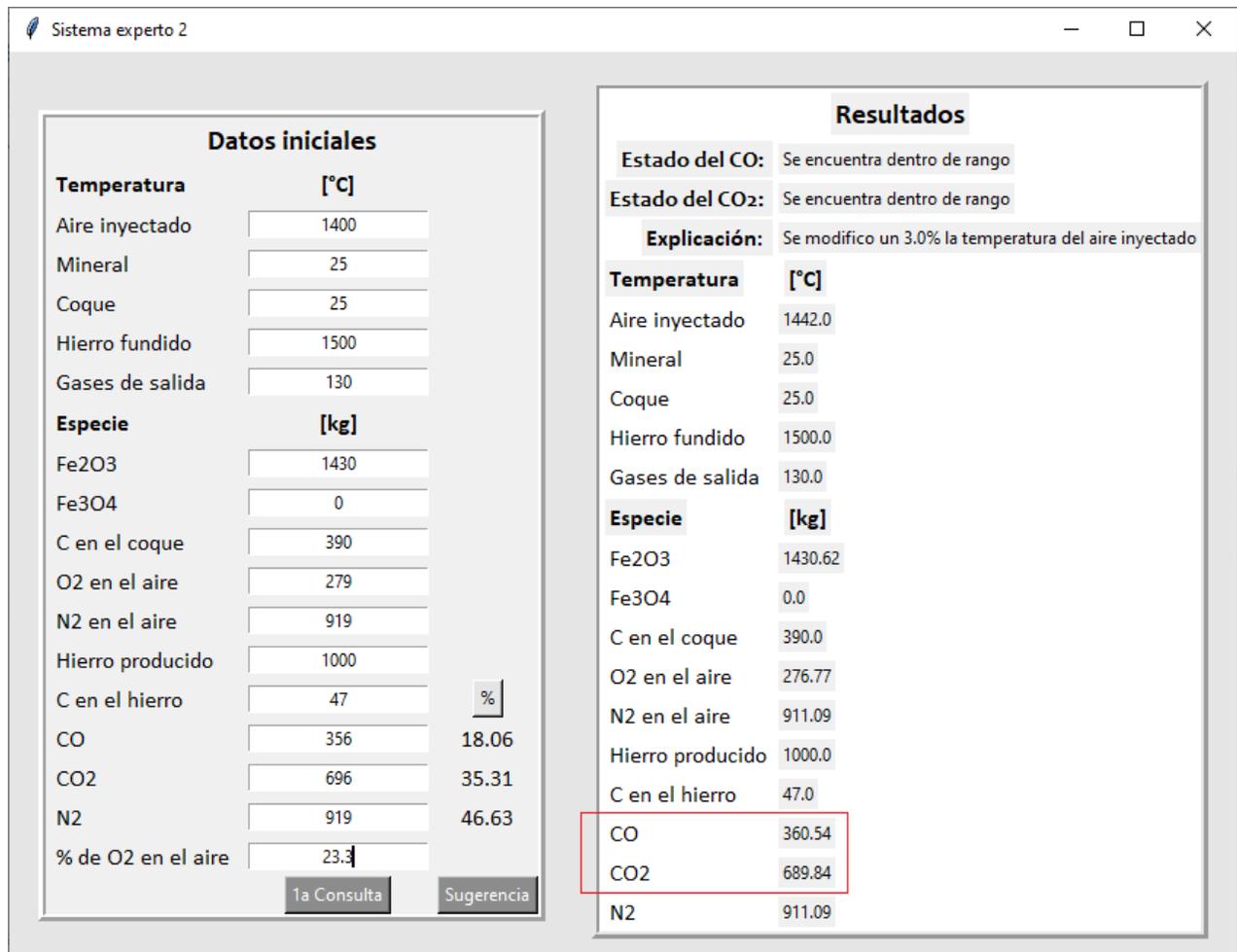


Figura 4.6 Captura de pantalla de los datos obtenidos por el SE (en este ejemplo solo una variable se encuentra fuera de rango).

La forma en que el SE analiza el estado de las variables es realizando una consulta individual para cada una de estas (en este caso solo analiza dos variables). Como para el CO se estableció solo una variable para modificar en caso de que se encuentre fuera de rango y para el CO₂ se establecieron dos, cuando ambas variables se encuentran fuera de rango, las consultas hechas para el estado del CO solo sugerirán modificar la temperatura del aire inyectado y las consultas hechas para el estado del CO₂ podrían sugerir modificar la temperatura del aire inyectado o la cantidad de carbono en el coque. Entonces, el orden de consulta determinará qué corrección final se llevará a cabo, si la propuesta por las consultas al CO₂ o al CO; esto se refiere a que, si durante una consulta al SE el CO₂ ya se encuentra dentro de sus límites permisibles pero el CO aun no, al final de la consulta solo se habrá modificado la temperatura, pero si se da el caso contrario, donde el CO se encuentre dentro de sus límites permisibles pero el CO₂ aún no, el SE podría sugerir modificar

la temperatura del aire inyectado o la cantidad de carbono en el coque, aplicando el criterio de selección económico para elegir entre una sugerencia de solución y otra.

En la Figuras 4.7 y 4.8 se presenta una consulta donde ambas variables se encuentran fuera del rango permisible. A diferencia de los otros ejemplos dados, en este caso el SE no modificó la temperatura del aire inyectado para corregir el error, si no que la solución que implementó fue modificar la cantidad de carbono en el coque. Se puede ver en la Figura 4.8 que el aumentar la cantidad de carbono un 2.25% efectivamente corrigió el error que se reporta en la Figura 4.7 sobre el estado de las variables.

Sistema experto 2

Datos iniciales

Temperatura [°C]

Aire inyectado	1100
Mineral	25
Coque	25
Hierro fundido	1500
Gases de salida	130

Especie [kg]

Fe2O3	1430	
Fe3O4	0	
C en el coque	400	
O2 en el aire	305	
N2 en el aire	1004	
Hierro producido	1000	
C en el hierro	47	
CO	357	17.07
CO2	731	34.94
N2	1004	47.99
% de O2 en el aire	23.3	

1a Consulta Sugerencia

Resultados

Estado del CO: Se encuentra fuera de rango (menor)

Estado del CO2: Se encuentra fuera de rango (mayor)

Explicación:

Temperatura [°C]

Aire inyectado	
Mineral	
Coque	
Hierro fundido	
Gases de salida	

Especie [kg]

Fe2O3	
Fe3O4	
C en el coque	
O2 en el aire	
N2 en el aire	
Hierro producido	
C en el hierro	
CO	
CO2	
N2	

Figura 4.7 Captura de pantalla de los datos solicitados por el SE híbrido.

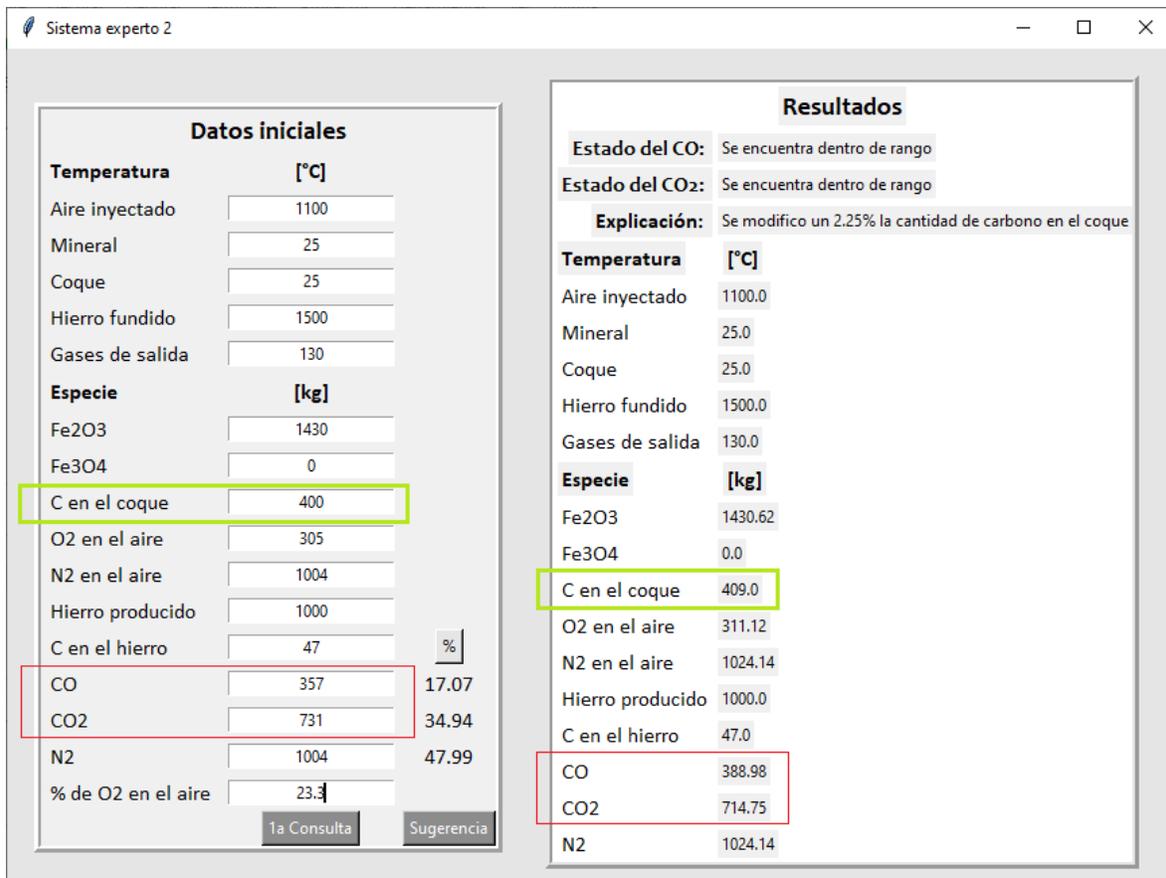


Figura 4.8 Captura de pantalla de los resultados obtenidos por el SE híbrido, en este ejemplo se aplicó el criterio de selección económico cuando el CO y CO₂ se encuentran fuera de rango.

Haciendo un seguimiento de los cálculos y sugerencias de solución que realizó el SE durante la consulta, hacía falta aumentar un 2.5% la temperatura del aire inyectado solo para que la cantidad de CO se mantenga dentro del rango permisible, pero el CO₂ aún se mantendría fuera de rango. Aplicando el criterio de selección económico el SE llegó a la conclusión, a través de las consultas hechas sobre el estado del CO₂, que es más económico aumentar un 2.25% la cantidad de carbono en el coque que seguir modificando la temperatura para mantener ambas variables dentro de sus rangos permisibles. Esto se puede comprobar fácilmente conociendo que los costos unitarios para la temperatura del aire inyectado y la cantidad de carbono en el coque son 20 y 15 respectivamente. Si se tuvieran que establecer unidades se podrían expresar como 20 pesos por grado Celsius y 15 pesos por gramo de carbono en el coque, cabe recalcar que estos costos son hipotéticos; entonces aumentar en un 2.5% la temperatura del aire inyectado tendría un costo de 550 pesos y aumentar 2.25% la cantidad de carbono en el coque tan solo 135 pesos. Con estos resultados se puede comprobar que el criterio de selección económico funciona como se esperaba.

Capítulo 5. Conclusiones

- Se consiguió representar parte del conocimiento acerca de un sistema metalúrgico (alto horno para la producción de hierro) de dos maneras diferentes; la primera con un modelo matemático determinístico basado en balances macroscópicos simultáneos de materia y energía; la segunda representación se construyó en forma de hechos, reglas y preguntas, estructura típica empleada en el desarrollo de sistemas expertos.
- Se logró fusionar ambas formas de representar el conocimiento sobre el proceso de obtención de hierro en un programa informático, conocido como sistema experto híbrido, el cual tiene la capacidad de analizar y corregir algunos parámetros de operación del alto horno.
- Se logró desarrollar un diseño amigable del sistema experto híbrido construyendo una interfaz de interacción que permite al usuario realizar consultas sobre el estado actual de las condiciones de operación del alto horno.
- El sistema experto que se construyó es del tipo monitoreo/control de proceso, aplicado específicamente a la obtención de hierro a través del alto horno, con la capacidad de corregir parámetros de operación.
- A través del desarrollo de diferentes versiones de código se demostró que es posible construir un sistema experto híbrido orientado a sistemas metalúrgicos utilizando los lenguajes de programación *Python* (lenguaje algorítmico convencional) y *Prolog* (lenguaje basado en reglas).
- No se encontró una manera eficiente de construir un experto híbrido utilizando los lenguajes de programación *Prolog* y *C++*.

Bibliografía

1. León, T., (2007), **Sistemas expertos y sus aplicaciones**, Monografía para obtener título de licenciatura, UAEH, México. pp. 3-70
2. Badaro S., Ibáñez J. y Agüero J., (2013), **Sistemas Expertos: Fundamentos, Metodologías y Aplicaciones**, Colección Ciencia y Tecnología #13, Universidad de Palermo, Argentina. pp. 349-364
3. Amador, L., (1996), **Inteligencia Artificial y Sistemas Expertos, Universidad de Córdoba**, Serie Monográfica N° 244, España. pp. 131-241
4. World Steel Association, *November 2022 crude steel production*, Recuperado el 10 de enero del 2023: <https://worldsteel.org/media-centre/press-releases/2022/november-2022-crude-steel-production/>
5. Covington, M.A., Nute, D., Vellino, A. (1997), **Prolog Programming in Depth**, Prentice Hall, EUA.
6. Davara, M.A., (1994), **Introducción a los sistemas expertos**, Revista iberoamericana de derecho informático N°5, ICADE, España. pp. 909-919
7. Fernandez, F., (2000), **Sistemas híbridos neuro-simbólicos: una revisión**, Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial, vol. 4, no. 11.
8. Serra, S., (2011), **Apunte teórico/práctico de la cátedra: Paradigmas de programación-Programación lógica**, Universidad Tecnológica Nacional, Facultad de Córdoba, Departamento de sistemas, Argentina.
9. Mosterín, J., (1976), **Lógica de primer orden**, Colección Convivium, 2ª edición, Ariel, España.
10. *Interfaz para Prolog*, Recuperado el 28 de septiembre del 2022: <https://www.swi-prolog.org>
11. **Altos Hornos**, AULA DE EL MUNDO, España.
12. Geerdes, M., Chaigneau, R., Kurunov, I., Lingiard, O., Ricketts, J., (2015), **Modern Blast Furnace Ironmaking**, 3a edición, IOS Press BV, Países bajos.

13. Cameron, I., Sukhram, M., Lefebvre, K. y Davenport, W., (2020), **Blast Furnace Ironmaking Analysis, Control, and Optimization**, 1ra edición, Elsevier, Amsterdam.
14. Bramer, M., (2013), **Logic Programming with Prolog**, 2a edición, Springer.
15. Sexton, C., (2002), **C++ Programming**, 2a edición, Made Simple, EUA.
16. Peacey, J., Davenport, W., (1979), **The Iron Blast Furnace Theory and Practice**, 1a edición, Pergamon Press, EUA.
17. (2020), **Blast Furnace Process Automation**, Primetals Technologies, Austria.
18. Kuman, S., (2019), **Numerical Methods of Mathematics Implemented in Fortran**, 1a edición, Springer, India.
19. *Guía de Uso Básico de Prolog*, Recuperado el 28 de septiembre del 2021: <http://www.lcc.uma.es/~pacog/apuntes/pd/guia.pdf>

Apéndice A

Entalpía del carbono disuelto en el arrabio

El arrabio que produce el alto horno se representa como una aleación hierro-carbono, por lo que la entalpía se representa de la siguiente manera:

$$\frac{\Delta H_{(C)}^{1500^{\circ}C}}{PM_C} = \frac{\Delta H_{<C>}^{1500^{\circ}C}}{PM_C} + \Delta H_{(Fe)+<C>\rightarrow(Fe-C)aleación}^{1500^{\circ}C}$$

Donde:

$\Delta H_{(Fe)+<C>\rightarrow(Fe-C)aleación}^{1500^{\circ}C}$: entalpía de reacción para producir la aleación Fe-C

fundida a 1500°C, a partir de Fe y C.

El valor para la entalpía de reacción requerido es de 1300 cal/g-mol de aleación que equivale a 5.4 MJ/kg-mol de aleación, este valor se obtuvo aproximándolo al valor de entalpía de reacción a 1600°C, ya que es el único valor reportado en [13].

Para utilizar este valor debe estar expresado en términos de kg-mol de carbono, para ello primero se debe calcular la fracción mol de carbono en la aleación Fe-C con una composición de 4.5% en masa de carbono, considerando 100 kg de aleación. Las masas moleculares del C y Fe son 12.01 y 55.85 kg/kg-mol respectivamente.

$$kg\ mol\ C = \frac{4.5\ kg\ C}{12.01\ kg\ C/kg\ mol\ de\ C} = 0.37$$

$$kg\ mol\ Fe = \frac{95.5\ kg\ Fe}{55.85\ kg\ Fe/kg\ mol\ de\ Fe} = 1.71$$

$$X_C \frac{0.37}{0.37 + 1.71} = 0.18$$

$$X_{Fe} \frac{1.71}{0.37 + 1.71} = 0.82$$

Por lo tanto, un kg de hierro fundido que contiene 0.18 fracción mol de C y 0.82 fracción mol de Fe contiene 0.18 kg de mol de C, dividiendo el valor de entalpía de reacción por este valor se obtiene:

$$\begin{aligned} \Delta H_{Fe(l)+C(s) \rightarrow (Fe-C)aleación}^{1500^{\circ}C} &= \frac{5.4 \text{ MJ/kg} - \text{mol de aleación}}{0.18 \text{ kg} - \text{mol de C/kg} - \text{mol de aleación}} \\ &= 30 \text{ MJ/kg} - \text{mol de C en la aleación} \end{aligned}$$

Por último, se divide el valor anterior por la masa atómica del carbono:

$$\begin{aligned} \Delta H_{Fe(l)+C(s) \rightarrow (Fe-C)aleación}^{1500^{\circ}C} &= \frac{30 \text{ MJ/kg} - \text{mol de C}}{12 \text{ kg/kg} - \text{mol de carbono}} \\ &= \underline{2.5 \text{ MJ/kg de carbono disuelto}} \end{aligned}$$

Con este valor la entalpía para el carbono disuelto queda expresada de la siguiente manera:

$$\frac{\Delta H_{(C)}^{1500^{\circ}C}}{PM_C} = \frac{\Delta H_{<C>}^{1500^{\circ}C}}{PM_C} + 2.5 \text{ MJ/kg}$$

El valor que hace falta ($\frac{\Delta H_{<C>}^{1500^{\circ}C}}{PM_C}$) se obtiene a partir de la Tabla A1.

$$\begin{aligned} \frac{\Delta H_{(C)}^{1500^{\circ}C}}{PM_C} &= 2.488 \frac{\text{MJ}}{\text{kg}} \text{ de C disuelto} + 2.5 \frac{\text{MJ}}{\text{kg}} \text{ de C disuelto} \\ &= 4.988 \text{ MJ/kg de carbono disuelto} \end{aligned}$$

Tabla A1. Valores útiles de entalpía de los compuestos en MJ por kg de compuesto [13].

Temperature	Compound	H°/MW	Temperature	Compound	H°/MW	
1500°C	Al ₂ O ₃ (s)	-14.67	930°C	O ₂ (g)	1.239	
	Al ₂ O ₃ (oxide in molten slag)	-13.58		Al ₂ O ₃ (s)	-15.41	
	C(s)	2.488		C(s)	1.359	
	C (dissolved in molten iron)	5		CO(g)	-2.926	
	CaO(s)	-9.925		CO ₂ (g)	-7.926	
	CaO (oxide in molten slag)	-8.495		CaO(s)	-10.50	
	Fe(<i>l</i>)	1.269		Fe	0.6164	
	MgO(s)	-13.08		Fe _{0.947} O(s)	-3.152	
	MgO (oxide in molten slag)	-11.14		H ₂ (g)	13.35	
	Mn(<i>l</i>)	1.343		H ₂ O(g)	-11.49	
	Mn (dissolved in molten iron)	1.27		MgO(s)	-13.84	
	MnO(s)	-4.300		MnO(s)	-4.770	
	MnO (oxide in molten slag)	-3.530		N ₂ (g)	1.008	
	Si(<i>l</i>)	3.155		SiO ₂ (s)	-14.13	
	Si (dissolved in molten iron)	-2.15		TiO ₂ (s)	-11.03	
	SiO ₂ (s)	-13.44		25°C	Al ₂ O ₃ (s)	-16.43
	SiO ₂ (oxide in molten slag)	-13.28			C(s)	0
TiO ₂ (s)	-10.48	CH ₄ (g)	-4.664			
TiO ₂ (oxide in molten slag)	-9.64	CO(g)	-3.946			
1200°C	CO ₂ (g)	-7.573	CO ₂ (g)		-8.942	
	CO(g)	-2.591	CaO(s)		-11.32	
	H ₂ O(g)	-10.81	CaCO ₃ (s)		-12.06	
	N ₂ (g)	1.339	Fe(s)		0	
			Fe _{0.947} O		-3.865	
		Fe ₂ O ₃ (s)	-5.169			
		Fe ₃ O ₄ (s)	-4.841			
		H ₂ (g)	0			
		H ₂ O(<i>l</i>)	-15.87			
		MgO(s)	-14.92			
		MgCO ₃ (s)	-13.20			
		MnO ₂ (s)	-5.98			
		SiO ₂ (s)	-15.16			
		Coal, 25°C	-1.2			
		Natural gas, 25°C	-4.52			

Para determinar la entalpía de las demás especies se utilizaron ecuaciones de entalpía en función de la temperatura simplificadas, válidas para rangos de temperatura típicas en la operación del alto horno reportadas en las Tablas A2, A3 y A4.

Tabla A2. Ecuaciones simplificadas para calcular la entalpía de diferentes especies en función de la temperatura del aire inyectado (900°C – 1400°C) [13].

$N_2(g)$	$0.001237 * T_{blast}^{\circ C} - 0.1450$
$O_2(g)$	$0.001137 * T_{blast}^{\circ C} - 0.1257$
$H_2O(g)$	$0.002582 * T_{blast}^{\circ C} - 13.91$
$C(s)$	$0.00197 * T_{blast}^{\circ C} - 0.482$

Tabla A3. Ecuaciones simplificadas para calcular la entalpía de diferentes especies en función de la temperatura de los gases de salida (25°C – 225°C) [13].

$CO(g)$	$0.001049 * T_{top\ gas}^{\circ C} - 3.972$
$CO_2(g)$	$0.0009314 * T_{top\ gas}^{\circ C} - 8.966$
$H_2(g)$	$0.01442 * T_{top\ gas}^{\circ C} - 0.3616$
$H_2O(g)$	$0.001902 * T_{top\ gas}^{\circ C} - 13.47$
$N_2(g)$	$0.001044 * T_{top\ gas}^{\circ C} - 0.02624$

Tabla A4. Ecuación simplificada para calcular la entalpía del hierro fundido en función de la temperatura del hierro fundido (1400°C – 1600°C) [13].

$Fe(l)$	$0.0008264 * T_{molten\ iron}^{\circ C} + 0.02863$
---------	--