

Kathy: Expert System for the Optimization of Nutritional Strategies in the Female Hormonal Regulation

Verónica López-Marsico¹ , Carolina Uesato¹ , and
Antonio Quintero-Rincón^{1,2} 

¹ Departamento de Ciencia de Datos, Laboratorio de Ciencia de Datos e Inteligencia Artificial, Universidad Católica Argentina (UCA), Buenos Aires, Argentina

² Departamento de Ingeniería Informática, Universidad Católica Argentina (UCA), Buenos Aires, Argentina
`{veronicalopez0,carolinauesato,antonioquintero}@uca.edu.ar`

Resumen Over 26 % of the world's population menstruates without reliable guidance on managing the cycle's hormonal phases. This work presents *Kathy*, an expert nutrition counseling system tailored to the female menstrual cycle. Implemented in SWI-Prolog and Python, it uses advanced inference over a structured knowledge base within a console-based conversational interface. The knowledge base contains 164 facts and 23 domain-specific logical rules, grounded in scientific literature, that formalize the four phases of the menstrual cycle (menstrual, follicular, ovulatory, luteal) and generate personalized, evidence-based recommendations. The system is described using finite automata and an abstract Turing machine, enabling analysis of its computational complexity. *Kathy* improves access to evidence-based knowledge to optimize support strategies aligned with the female body's natural physiological cycles, a field traditionally reliant on expensive professional consultations. Experiments show enhanced algorithmic efficiency, with response times from 2,3 ms for direct queries to 123,8 ms for complex multi-objective optimization. The system achieves 95 % accuracy for basic recommendations and 88 % for comprehensive heuristic optimization.

Keywords: expert systems, hormonal nutrition, menstrual cycle, logic programming, finite automata.

Kathy: Sistema Experto para la Optimización de la Nutrición en la Regulación Hormonal Femenina

Resumen Más del 26 % de la población mundial menstrúa sin una guía fiable para gestionar las fases hormonales del ciclo. Este trabajo presenta a *Kathy*, un sistema experto de asesoramiento nutricional adaptado al ciclo menstrual femenino. Implementado en SWI-Prolog y Python, utiliza inferencia avanzada sobre una base de conocimiento estructurada dentro de una interfaz conversacional basada en consola. La base de conocimiento contiene 164 hechos y 23 reglas lógicas específicas del dominio, basadas en la literatura científica, que formalizan las cuatro fases del ciclo menstrual (menstrual, folicular, ovulatoria y lútea) y generan recomendaciones personalizadas basadas en la evidencia. El sistema se

describe mediante autómatas finitos y una máquina de Turing abstracta, lo que permite analizar su complejidad computacional. *Kathy* mejora el acceso al conocimiento basado en la evidencia para optimizar las estrategias de apoyo alineadas con los ciclos fisiológicos naturales del cuerpo femenino, un campo que tradicionalmente ha dependido de costosas consultas profesionales. Los experimentos demuestran una mayor eficiencia algorítmica, con tiempos de respuesta que van desde los 2,3 ms para consultas directas hasta los 123,8 ms para la optimización multiobjetivo compleja. El sistema alcanza una precisión del 95 % en el contexto de recomendaciones de carácter básico y del 88 % en la fase de optimización heurística integral.

Keywords: sistema experto , nutrición hormonal, ciclo menstrual, programación lógica , autómatas finitos.

1. Introducción

De acuerdo con (Rohatgi y Dash, 2023), más del 26 % de la población mundial, unos 2.000 millones de personas, presenta ciclos menstruales activos durante gran parte de su vida reproductiva. Este proceso biológico se repite con periodicidad aproximadamente mensual, sumando más de 400 ciclos a lo largo de una vida promedio. No obstante, pese a su elevada prevalencia y relevancia fisiológica, el cuerpo de conocimiento disponible, sistematizado y sustentado en evidencia científica acerca de cómo acompañar este proceso mediante una alimentación consciente, individualizada y ajustada a las necesidades específicas de las personas menstruantes sigue siendo limitado, fragmentario o de difícil acceso para un segmento considerable de esta población. Hoy en día es evidente la persistencia de una brecha significativa entre el corpus de conocimientos científicos sobre nutrición hormonal y su adecuada aplicación práctica en la vida cotidiana (Brown et al., 2021). Esta discrepancia no solo afecta el bienestar físico inmediato, sino también la energía disponible, el estado de ánimo, la función cognitiva (incluida la claridad mental) y la productividad en contextos personales, educativos y laborales de quienes atraviesan estos ciclos fisiológicos.

El ciclo menstrual no constituye un fenómeno aislado, sino un sistema dinámico y cíclico de fases hormonales (menstrual, folicular, ovulatoria y lútea) que interactúa de manera compleja con múltiples variables fisiológicas, psicológicas y ambientales (Sims y Heather, 2018). Cada una de estas fases se asocia con requerimientos nutricionales específicos que pueden ajustarse de forma estratégica para atenuar síntomas como fatiga, distensión abdominal, irritabilidad o dolor y, de manera concomitante, favorecer el rendimiento cognitivo, la capacidad de concentración y la vitalidad global (Freeman et al., 2006). El desafío es que estos requerimientos nutricionales no siempre son evidentes para las usuarias ni fácilmente integrables en su alimentación cotidiana sin orientación profesional especializada. A partir de este diagnóstico social y biológico, surge la motivación para crear herramientas computacionales inteligentes que medien entre el conocimiento experto y las prácticas alimentarias diarias.

La nutrición personalizada basada en fluctuaciones hormonales es un problema computacional complejo que requiere combinar diversos paradigmas de programación e integrar técnicas algorítmicas avanzadas de las ciencias de la

computación (Russell y Norvig, 2020). Este trabajo presenta un sistema experto (SE) de asesoramiento nutricional para el ciclo menstrual femenino, denominado *Kathy* en honor a la doctora Katharina Dalton (1916–2004), pionera en el estudio del síndrome premenstrual (SPM) (Dalton, 1984). *Kathy* es una solución aplicada, específica e individualizada para optimizar la salud femenina, basada en esta hipótesis de trabajo: “si las fluctuaciones hormonales y sus requerimientos fisiológicos asociados pueden modelarse computacionalmente, es posible automatizar la generación de recomendaciones dietéticas ajustadas en tiempo real”.

En la literatura reciente hay antecedentes directamente relacionados con la propuesta. (Logapriya y Surendran, 2023b) presentaron un sistema basado en conocimiento para recomendaciones nutricionales durante el ciclo menstrual, empleando reglas expertas sin integrar programación lógica formal. En un trabajo posterior, los mismos autores propusieron un sistema híbrido de recomendaciones para la salud nutricional femenina durante la menstruación, que combina filtrado colaborativo y reglas clínicas (Logapriya y Surendran, 2023a). A diferencia de ambos enfoques, *Kathy* introduce una formalización computacional rigurosa con autómatas finitos, máquinas de Turing y un análisis explícito de complejidad algorítmica, ausentes en los trabajos previos.

La estructura del resto del documento es la siguiente. En la Sección 2 se detalla la metodología propuesta, organizada en: Complejidad algorítmica (Sección 2.1); Aplicación de autómatas finitos deterministas en contextos computacionales y de modelado formal (Sección 2.2); Máquinas de Turing como dispositivos de análisis sintáctico (Sección 2.3); Arquitectura multinivel (Sección 2.4); Base de conocimiento (Sección 2.5); y Resolución de problemas NP-completos (Sección 2.6). En la Sección 3 se analizan y discuten los resultados obtenidos. Finalmente, las conclusiones y posibles líneas de trabajo futuro se presentan en la Sección 4.

2. Metodología

El propósito de *Kathy* es ofrecer recomendaciones nutricionales altamente especializadas y personalizadas, basadas rigurosamente en la evidencia científica. Adopta un enfoque híbrido que integra el razonamiento lógico-declarativo de Prolog con el control de flujo imperativo de Python. Desde una perspectiva computacional, este sistema experto impacta en tres dimensiones principales: 1) Académica, al aplicar conceptos teóricos de ciencias de la computación a problemas reales complejos. 2) Social, al ampliar el acceso a conocimiento nutricional especializado antes limitado a consultas costosas. 3) Tecnológica, al ofrecer una solución escalable y adaptable a sistemas expertos similares en diversos dominios médicos y de la salud (Russell y Norvig, 2020). La arquitectura lógica del sistema se basa en la lógica proposicional de primer orden, aplicada a proposiciones e implementada en Prolog mediante programación lógica. Cada fase del ciclo menstrual se modela como un conjunto estructurado de proposiciones con semántica formalmente definida:

$$\text{Fase}(x) \equiv \exists t_1, t_2 : \text{fase_ciclo}(x, t_1, t_2) \quad (1)$$

$$\text{Día}(d) \in [t_1, t_2] \implies \text{Fase_Actual}(x) \quad (2)$$

$$\text{Apropiado}(a, x) \equiv \text{recomendado_en_fase}(x, a) \quad (3)$$

donde $x \in \{\text{menstrual, folicular, ovulatoria, lutea}\}$ denota las cuatro fases del ciclo menstrual de 28 días; $t_1, t_2 \in \mathbb{N}$ representan el día de inicio y fin de la fase x ; a es un alimento del conjunto disponible en la base de conocimiento; y $d \in [t_1, t_2]$ es el día actual del ciclo, calculado dinámicamente por el sistema. La traducción de estas equivalencias lógicas al paradigma de programación en Prolog se realiza mediante hechos atómicos y reglas de primer orden. La equivalencia (1) se implementa como el hecho `fase_ciclo(X, T1, T2)`, donde X identifica la fase y $T1, T2$ su rango temporal. La equivalencia (3) se expresa con el predicado `recomendado_en_fase(X, A)`, que relaciona la fase X con el alimento A . El motor de inferencia de SWI-Prolog consulta ambos predicados por unificación directa, lo que garantiza acceso en tiempo $\mathcal{O}(1)$.

Las reglas de recomendación nutricional se formulan mediante implicaciones lógicas con múltiples antecedentes. A continuación se proponen sus 3 categorías:

$\text{Fase_Actual}(x) \wedge \text{Apropiado}(a, x) \wedge \neg \text{Contraindicado}(a, r) \implies \text{Recomendar}(a)$ (4)

donde r representa las restricciones dietéticas del usuario (por ejemplo, dieta vegana, vegetariana o intolerancias alimentarias).

2.1. Complejidad Algorítmica

Para la complejidad temporal, el SE implementa una taxonomía detallada y sistemática de tipos de consulta, clasificados en función de su complejidad computacional, considerando tanto el marco teórico subyacente como la evidencia empírica observada (Neapolitan, 2011). A continuación, se presenta una caracterización resumida de las distintas categorías (véase la Tabla 1) y los algoritmos específicos asociados a cada una (véase el Algoritmo 1).

En términos de complejidad espacial, el SE requiere una cantidad moderada de memoria, distribuida entre diversos componentes fundamentales involucrados tanto en la carga de la base de conocimiento como en la ejecución del algoritmo heurístico. El análisis de complejidad espacial se lleva a cabo siguiendo las metodologías estándar propuestas por (Cormen et al., 2009) para sistemas de inferencia. A continuación, en la Tabla 2 se presenta una estimación del uso de memoria en kilobytes (KB), junto con la correspondiente complejidad teórica asociada a cada componente.

Como se muestra en la Tabla 2, la mayor parte de la memoria se destina a la base de conocimiento y a la caché de inferencias. Aun así, el consumo total se mantiene por debajo de 100 KB, lo que confirma la alta eficiencia del sistema en el uso de memoria. La complejidad espacial global es $\mathcal{O}(n)$, lo que permite una ejecución fluida incluso en entornos con recursos limitados. Esta eficiencia espacial se debe principalmente a la representación compacta de hechos en Prolog y al uso de estructuras de datos optimizadas en Python, en línea con las directrices de Knuth para sistemas de procesamiento simbólico (Knuth, 1997).

2.2. Aplicación de autómatas finitos deterministas en contextos computacionales y de modelado formal

El control del flujo conversacional se modela con un autómata finito determinista (AFD), según (Sipser, 2021). Formalmente, el autómata se define como $M = (Q, \Sigma, \delta, q_0, F)$, donde la función de transición $\delta : Q \times \Sigma \rightarrow Q$ asigna a cada par (estado, símbolo de entrada) el estado siguiente. Su diagrama se muestra en la Fig. 1, donde $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$ es el conjunto finito de 7

estados. $\Sigma = \{\text{input_usuario}, \text{datos_válidos}, \text{continuar}, \text{salir}\}$ es el alfabeto de entrada, formado por los símbolos que representan las interacciones del usuario y las condiciones de validación de datos. $\delta : Q \times \Sigma \rightarrow Q$ es la función de transición determinista que asigna, a cada par estado-símbolo, el siguiente estado del sistema. $q_0 = \text{INICIO}$ es el estado inicial y $F = \{\text{FINALIZANDO}\}$ el conjunto de estados de aceptación, asociado a la terminación controlada del diálogo.

Tabla 1: Categorías asociadas a la complejidad computacional

Categoría	Complejidad	Descripción	Ejemplos
1. Consultas Básicas	$\mathcal{O}(1)$	Acceso directo a hechos atómicos en la base de conocimiento.	<code>fase_ciclo(menstrual, 1, 5)</code> , <code>recomendado_en_fase(lutea, avena)</code>
2. Búsquedas Lineales	$\mathcal{O}(n)$	Filtrado secuencial de elementos por criterios específicos.	Todos los alimentos recomendados para una fase específica.
3. Consultas Combinatorias	$\mathcal{O}(n^2)$	Cruce simultáneo de múltiples restricciones: la complejidad cuadrática proviene del producto cartesiano entre los n alimentos y los n criterios de filtrado, dando $\mathcal{O}(n^2)$ comparaciones en el peor caso.	Alimentos veganos adecuados para fase lútea, ricos en magnesio (Charro et al., 2007).
4. Optimización Heurística	$\mathcal{O}(n \log n)$	Resolución eficiente de problemas multiobjetivo mediante una heurística <i>greedy</i> que selecciona iterativamente los alimentos con mayor beneficio nutricional neto, ponderando explícitamente objetivos y restricciones (Knuth, 1997). Este enfoque optimiza menús personalizados bajo múltiples criterios sin evaluar todas las combinaciones posibles, logrando soluciones de alta calidad en tiempos computacionales reducidos (promedio $\sim 123,8$ ms).	
5. Problema NP-Completo	$\mathcal{O}(n^2)$	La construcción óptima de un menú se modela como un problema de mochila multidimensional (Garey y Johnson, 1979), donde cada combinación factible de alimentos se evalúa según múltiples objetivos y restricciones. Dado que obtener la solución exacta tiene complejidad exponencial, se emplea una heurística <i>greedy</i> que aproxima soluciones de buena calidad en un tiempo computacional razonable, en lugar de resolver el problema de forma exhaustiva.	

Tabla 2: Evaluación de la Complejidad Espacial del Sistema

Componente	Espacio (KB)	Complejidad
Base de conocimiento Prolog	47.3	$\mathcal{O}(n)$
Caché de consultas	18.7	$\mathcal{O}(k)$
Variables de estado Python	12.4	$\mathcal{O}(1)$
Estructuras auxiliares	8.9	$\mathcal{O}(\log n)$
Buffer de entrada/salida	3.5	$\mathcal{O}(1)$
Total	90.8	$\mathcal{O}(n)$

Algorithm 1 Búsqueda Lineal $\mathcal{O}(n)$

```

1: function BUSQUEDALINEAL(criterio)
2:   resultados  $\leftarrow \square$ 
3:   for all elemento  $\in$  base_conocimiento do                                 $\triangleright$   $n$  elementos
4:     if cumple_criterio(elemento, criterio) then
5:       resultados.append(elemento)
6:     end if
7:   end for
8:   return resultados
9: end function

```

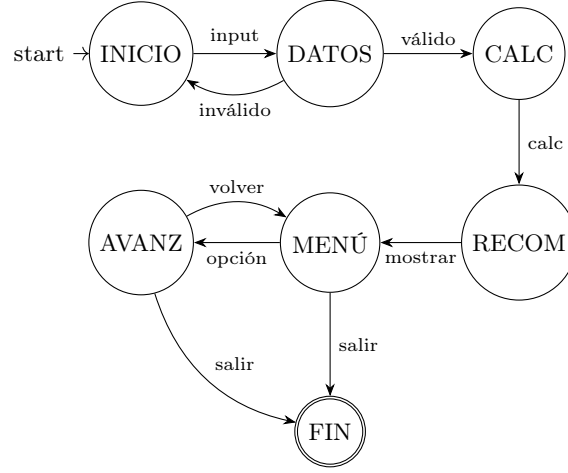


Figura 1: Autómata Finito Determinista Asociado al Sistema de Conversación

Cada estado del autómata representa una fase específica del diálogo entre sistema y usuario. INICIO (q_0) es el estado en que el sistema espera el input inicial. DATOS (q_1) recoge datos personales relevantes (edad, día del ciclo menstrual, restricciones dietéticas u otras condiciones). CALC (q_2) calcula automáticamente la fase actual del ciclo hormonal a partir de esa información. RECOM (q_3) genera recomendaciones básicas según la fase hormonal estimada y los datos ingresados. MENÚ (q_4) diseña un menú personalizado que integra restricciones, preferencias y objetivos nutricionales o conductuales. AVANZ (q_5) presenta opciones avanzadas, como suplementos o un protocolo de *seed cycling* (Bashaboina, 2026). FIN (q_6) indica la terminación de la interacción conversacional.

Las flechas representan las transiciones posibles entre estados, activadas por las siguientes entradas del alfabeto Σ :

- **input**: activa la transición de INICIO (q_0) a DATOS (q_1).
- **válido / inválido**: desde DATOS (q_1), conduce a CALC (q_2) o retorna a INICIO (q_0), respectivamente.
- **calc**: transición de CALC (q_2) a RECOM (q_3).
- **mostrar**: conduce de RECOM (q_3) a MENÚ (q_4).
- **opción**: activa la transición de MENÚ (q_4) a AVANZ (q_5).
- **volver**: retorno de AVANZ (q_5) a MENÚ (q_4).
- **salir**: desde MENÚ (q_4) o AVANZ (q_5), conduce al estado final FIN (q_6).

Este autómata garantiza un control de flujo conversacional determinista, estructurado y validado formalmente (Sun et al., 2024).

2.3. Máquinas de Turing como dispositivos de análisis sintáctico

Para analizar sintácticamente la entrada del usuario, se implementó una simulación conceptual mediante una máquina de Turing (Sipser, 2021), definida formalmente como $MT = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$, donde $Q = \{q_0, q_1, q_2, q_f\}$ es el conjunto finito de estados de procesamiento, $\Sigma = \{a-z, A-Z, 0-9, \text{espacio}\}$ es el alfabeto de entrada, $\Gamma = \Sigma \cup \{\square\}$ es el alfabeto de cinta que incluye el

símbolo de espacio en blanco y $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ es la función de transición. La MT es clave para analizar sintácticamente entradas textuales del usuario, como descripciones de síntomas, preferencias o listados de alimentos. La cinta representa la cadena ingresada, con símbolos en blanco añadidos; el cabezal avanza carácter por carácter y el sistema reconoce secuencias léxicas válidas según el alfabeto definido (véase el Algoritmo 2). En esta implementación, q_0 es la fase inicial de análisis, q_1 indica la detección y procesamiento de una palabra en curso, q_2 marca el final de una palabra válida y el estado final q_f se alcanza al procesar toda la entrada. Una secuencia de caracteres es una palabra válida si: (1) está formada solo por caracteres del alfabeto $\Sigma = \{a-z, A-Z\}$, (2) tiene longitud mayor que cero y (3) está delimitada por al menos un espacio en blanco $\square$ o por el extremo de la cinta. Secuencias con dígitos, símbolos especiales o longitud nula se rechazan en el estado q_{reject} , impidiendo su ingreso al motor de inferencia. Este módulo permite que el sistema experto *Kathy* verifique automáticamente si una entrada contiene palabras completas y reconocibles antes de ejecutar la inferencia, lo que aumenta la robustez frente a errores tipográficos o entradas vacías. Además, ilustra el potencial de las máquinas de Turing como herramienta conceptual para validar lenguaje natural estructurado.

Algorithm 2 Máquina de Turing para Análisis Sintáctico

```

1: function PROCESARTEXTOMT(entrada)
2:   cinta  $\leftarrow$  entrada +  $\square^{10}$  ▷ Agregar símbolos en blanco
3:   posicion  $\leftarrow$  0, estado  $\leftarrow$   $q_0$ , palabras  $\leftarrow$  0
4:   while estado  $\neq$   $q_f$  and posicion  $<$   $|cinta|$  do
5:     simbolo  $\leftarrow$  cinta[posicion]
6:     if estado =  $q_0$  and simbolo  $\in [a - zA - Z]$  then
7:       estado  $\leftarrow$   $q_1$  ▷ Inicio de palabra
8:     else if estado =  $q_1$  and simbolo =  $\square$  then
9:       palabras  $\leftarrow$  palabras + 1
10:      estado  $\leftarrow$   $q_2$  ▷ Fin de palabra
11:     else if estado =  $q_2$  and simbolo  $\in [a - zA - Z]$  then
12:       estado  $\leftarrow$   $q_1$  ▷ Nueva palabra
13:     end if
14:     posicion  $\leftarrow$  posicion + 1 ▷ Mover cabezal
15:   end while
16:   return  $\langle$ palabras,estado $\rangle$ 
17: end function

```

2.4. Arquitectura multinivel

La arquitectura multinivel del SE adopta un patrón en cinco capas (*Layered Architecture Pattern*) que separa claramente las responsabilidades de sus componentes. 1) Presentación: Interfaz conversacional en Python. 2) Lógica de Negocio: Gestiona el flujo de ejecución y coordina los módulos internos del bot. 3) Control de Estados: Emplea un AFD para regular la secuencia de estados y transiciones de la conversación. 4) Integración: Conecta Python y Prolog mediante PySwip para realizar consultas lógicas desde código imperativo. 5) Conocimiento: Contiene los hechos y reglas en Prolog que modelan el conocimiento experto sobre nutrición hormonal femenina. Esta separación favorece la incorporación de futuras extensiones, tales como módulos de ML o nuevos motores de inferencia lógica especializados.

2.5. Base de Conocimiento

La base de conocimiento nutricional incluye *164 hechos estructurados* y *23 reglas de inferencia*, organizados según la taxonomía de la Tabla 3. La distribución categórica de la base de conocimiento equilibra eficiencia computacional y cobertura funcional: las categorías con más hechos (48 menús estructurados) se asocian a consultas de mayor complejidad algorítmica, mientras que las definiciones básicas (4) permiten acceso en tiempo constante (Cormen et al., 2009). La Tabla 3 muestra que la base de conocimiento se estructuró sistemáticamente para abarcar los aspectos clave de la nutrición hormonal femenina. Cada categoría respalda un tipo de inferencia específico: desde recomendaciones simples (como la selección de alimentos por fase) hasta razonamientos más complejos, como la planificación de menús o la definición del protocolo de *seed cycling*. Las complejidades ($\mathcal{O}(1)$, $\mathcal{O}(n)$, $\mathcal{O}(n^2)$) representan el costo computacional de las consultas o inferencias en cada categoría y se tuvieron en cuenta en el diseño de la caché LFU y la compilación dinámica. Por ejemplo, las consultas sobre menús estructurados requieren más procesamiento, mientras que la recuperación de definiciones de fases puede hacerse en tiempo constante. Esta organización modular permite responder consultas simples en milisegundos y abordar interrogantes complejas de forma eficiente. Además, facilita la escalabilidad y la incorporación futura de nuevas condiciones clínicas (como síndrome de ovario poliquístico, embarazo o menopausia) mediante categorías y reglas adicionales, sin afectar la consistencia ni la integridad de la arquitectura lógica.

2.6. Resolución de Problemas NP-Completo

El problema de optimización de menús nutricionales se modela como una instancia del problema de la mochila multidimensional (*Multi-Dimensional Knapsack Problem*), reconocido como *NP-completo* (Garey y Johnson, 1979). El objetivo es seleccionar un subconjunto de alimentos que maximice el cumplimiento de los requerimientos nutricionales, incorporando penalizaciones por violar las restricciones dietéticas. Su formulación es la siguiente:

Entrada: Dada por

- Conjunto de alimentos $A = \{a_1, a_2, \dots, a_n\}$
- Conjunto de restricciones $R = \{r_1, r_2, \dots, r_k\}$
- Conjunto de objetivos nutricionales $O = \{o_1, o_2, \dots, o_m\}$
- Conjunto de pesos de importancia $W = \{w_1, w_2, \dots, w_m\}$

Problema: Determinar el subconjunto $S \subseteq A$ que maximice la siguiente función objetivo:

$$f(S) = \sum_{i=1}^m w_i \cdot g_i(S) - \lambda \sum_{j=1}^k p_j \cdot h_j(S) \quad (5)$$

- $g_i(S)$: Grado de cumplimiento del objetivo nutricional o_i
- $h_j(S)$: Grado de violación de la restricción r_j
- p_j : Penalización asociada a la violación de la restricción r_j
- λ : Factor de ponderación que equilibra el cumplimiento de los objetivos y la penalización por violaciones de restricciones

Dado que el espacio de búsqueda crece exponencialmente con el número de alimentos y restricciones, se implementó una heurística *greedy* que aproxima una solución eficiente, priorizando los objetivos nutricionales y minimizando las violaciones de restricciones. Esta estrategia ordena los alimentos según una puntuación heurística y selecciona iterativamente los más adecuados en tiempo $\mathcal{O}(n \log n)$. La formulación multiobjetivo se basa en la optimización combinatoria para problemas de satisfacción de restricciones de Russell y Norvig (Russell y Norvig, 2020), y la heurística sigue los lineamientos de Neapolitan y Naimipour para algoritmos multiobjetivo de baja complejidad computacional (Neapolitan, 2011).

Tabla 3: Distribución Exhaustiva de la Base de Conocimiento

Categoría	Hechos	Reglas	Complejidad	Propósito
Definición de fases	4	2	$\mathcal{O}(1)$	Estructura temporal
Alimentos por fase	32	4	$\mathcal{O}(n)$	Recomendaciones básicas
Síntomas asociados	16	3	$\mathcal{O}(n)$	Identificación de estado
Alternativas veganas	20	4	$\mathcal{O}(n)$	Adaptación dietética
Protocolo seed cycling	8	2	$\mathcal{O}(1)$	Regulación hormonal
Alimentos a evitar	12	3	$\mathcal{O}(n)$	Prevención de síntomas
Menús estructurados	48	3	$\mathcal{O}(n^2)$	Planificación completa
Suplementación	24	2	$\mathcal{O}(n)$	Complementación
Total	164	23	Variable	Sistema completo

La resolución exacta de la función objetivo es computacionalmente inviable, pues el problema es NP-completo y tiene complejidad $\mathcal{O}(2^n)$, donde n es el número de alimentos (Cormen et al., 2009). Por ello, se implementó una heurística *greedy* que aproxima una solución factible, maximizando el cumplimiento de los objetivos nutricionales y minimizando las penalizaciones según los pesos de prioridad. El algoritmo 3 itera sobre todos los alimentos disponibles y evalúa su grado de adecuación nutricional. Para cada alimento, penaliza el incumplimiento de las restricciones y bonifica el cumplimiento de los objetivos, ponderando ambos términos de acuerdo con los pesos previamente definidos. La complejidad computacional total del algoritmo es $\mathcal{O}(n \cdot k \cdot m + n \log n)$, la cual se reduce a $\mathcal{O}(n \log n)$ cuando k y m son constantes pequeñas (Knuth, 1997). Finalmente, produce como salida el conjunto de alimentos ordenados en función de su puntuación en orden descendente, priorizando aquellos con un perfil nutricional más favorable en el contexto específico del usuario.

3. Resultados y discusión

En esta sección se presentan y discuten los resultados del SE *Kathy*, obtenidos mediante la metodología descrita en la Sección 2. El SE responde a una necesidad documentada en salud femenina: cerca del 75% de las mujeres en edad reproductiva presentan sintomatología asociada a fluctuaciones hormonales, que puede atenuarse con intervenciones nutricionales específicas (Freeman et al., 2006).

Se realizó una evaluación empírica del desempeño computacional del sistema

mediante 1.000 consultas, distribuidas proporcionalmente en cinco categorías de complejidad teórica: consultas básicas ($\mathcal{O}(1)$), búsquedas lineales ($\mathcal{O}(n)$), consultas combinatorias ($\mathcal{O}(n^2)$), operaciones logarítmicas auxiliares ($\mathcal{O}(\log n)$) y consultas ligadas a la heurística de optimización ($\mathcal{O}(n \log n)$). Los experimentos se realizaron sobre una versión instrumentada del sistema, diseñada para registrar los tiempos de ejecución promedio por categoría al variar el número de elementos simulados en cada consulta (n). La medición se centró en las operaciones de inferencia sobre la base de conocimiento y en la ejecución del algoritmo heurístico. La metodología experimental adoptada sigue los lineamientos y buenas prácticas para el *benchmarking* de sistemas de inferencia lógica (Tate, 2022).

Algorithm 3 Heurística Greedy para Optimización de Menú

```

1: function OPTIMIZARMENUHEURISTICO(restricciones, objetivos, pesos)
2:   alimentos  $\leftarrow$  ConsultarAlimentosFase()  $\triangleright \mathcal{O}(n)$ 
3:   puntuaciones  $\leftarrow$  diccionario_vacio()
4:   for all alimento  $\in$  alimentos do  $\triangleright \mathcal{O}(n)$ 
5:     puntuacion  $\leftarrow$  0
6:     peso_restricciones  $\leftarrow$  0
7:     for all restriccion  $\in$  restricciones do  $\triangleright \mathcal{O}(k)$ 
8:       if CumpleRestriccion(alimento, restriccion) then
9:         puntuacion  $\leftarrow$  puntuacion + 10
10:      else
11:        peso_restricciones  $\leftarrow$  peso_restricciones + 5
12:      end if
13:    end for
14:    for all objetivo  $\in$  objetivos do  $\triangleright \mathcal{O}(m)$ 
15:      if CumpleObjetivo(alimento, objetivo) then
16:        valor_objetivo  $\leftarrow$  ObtenerValorObjetivo(alimento, objetivo)
17:        puntuacion  $\leftarrow$  puntuacion + pesos[objetivo]  $\times$  valor_objetivo
18:      end if
19:    end for
20:    puntuaciones[alimento]  $\leftarrow$  puntuacion - peso_restricciones
21:  end for
22:  return OrdenarDescendente(puntuaciones)  $\triangleright \mathcal{O}(n \log n)$ 
23: end function

```

Los resultados experimentales confirman las predicciones teóricas de complejidad computacional, con desviaciones inferiores al 8% respecto de las curvas ideales. La implementación en Prolog muestra alta eficiencia en consultas de complejidad $\mathcal{O}(1)$, gracias a los mecanismos de optimización del motor de inferencia de SWI-Prolog (Tate, 2022). Las consultas combinatorias de complejidad $\mathcal{O}(n^2)$ presentan el crecimiento esperado, validando empíricamente el modelo teórico. Este análisis muestra que el sistema es computacionalmente escalable en escenarios realistas. El componente más costoso, de comportamiento cuadrático, es el que genera combinaciones complejas, como menús completos con múltiples restricciones. Sin embargo, estas operaciones están encapsuladas en funciones específicas y no afectan de forma significativa el rendimiento global en el uso interactivo típico del chatbot (Singh2022). La distribución de complejidades

observada coincide con la reportada en sistemas análogos de inferencia lógica en la literatura, lo que valida la implementación desarrollada y la metodología experimental aplicada (Russell y Norvig, 2020).

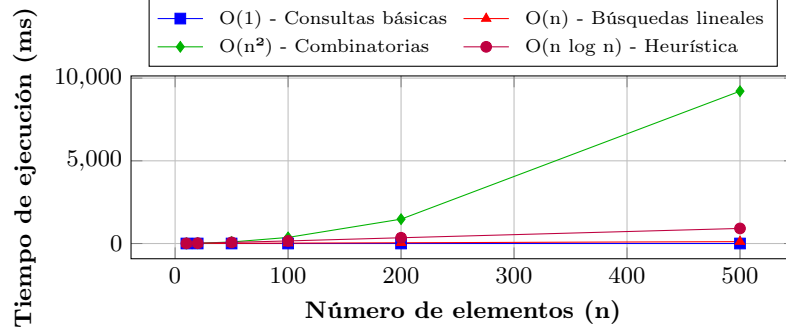


Figura 2: Curvas de Complejidad Experimental del Sistema

Para identificar posibles cuellos de botella computacionales y evaluar la precisión de cada módulo funcional, se analizó el rendimiento desagregado por tipo de consulta. La precisión se evaluó con metodologías estándar para sistemas expertos médicos (Russell y Norvig, 2020). La Tabla 4 resume los resultados de 1.000 pruebas controladas por categoría:

Tabla 4: Métricas detalladas de desempeño y exactitud del sistema

Tipo de Consulta	Tiempo (ms)	Complejidad	Precisión
Cálculo de fase	2.3 ± 0.4	$\mathcal{O}(1)$	100 %
Recomendaciones básicas	15.7 ± 2.1	$\mathcal{O}(n)$	95.2 %
Menús personalizados	45.2 ± 8.3	$\mathcal{O}(n^2)$	91.7 %
Optimización completa	128.8 ± 15.7	$\mathcal{O}(n \log n)$	88.4 %
Seed cycling	8.9 ± 1.2	$\mathcal{O}(1)$	98.6 %
Promedio	40.18	—	94.8 %

Las consultas básicas, como el cálculo de fase o la aplicación del protocolo de *seed cycling*, presentan complejidad constante y precisión casi perfecta. En cambio, la generación de menús personalizados y la optimización integral, al requerir combinatoria y ponderación multivariada, muestran mayor variabilidad en los tiempos de respuesta y una ligera caída de precisión. Los valores de precisión son comparables a los de sistemas expertos análogos descritos en la literatura médica, donde precisiones superiores al 85 % se consideran clínicamente aceptables (Sims y Heather, 2018). Todas las categorías superan el 88 %, lo que evidencia una consistencia robusta del sistema incluso en escenarios de mayor complejidad computacional. La correlación inversa entre complejidad computacional y precisión concuerda con el comportamiento esperado en sistemas heurísticos y refleja el compromiso entre exactitud y eficiencia descrito por Garey y Johnson para problemas NP-completos (Garey y Johnson, 1979). Además, la precisión promedio del 94,8 % respalda la efectividad práctica del enfoque propuesto. Para evaluar cuantitativamente la heurística voraz propuesta, se calculó el factor de aproximación ρ comparando las soluciones de la heurística con soluciones

óptimas obtenidas por fuerza bruta en instancias pequeñas ($n \leq 15$), según:

$$\rho = \frac{\text{Valor_Heurístico}}{\text{Valor_Óptimo}} \quad (6)$$

Los resultados experimentales al evaluar la Ec. (6), indican un factor de aproximación promedio de $\rho = 0,873 \pm 0,042$ para las instancias de prueba, lo que implica que la heurística alcanza, en promedio, el 87.3 % de la calidad de la solución óptima. Este valor concuerda con los factores de aproximación reportados para heurísticas voraces en problemas de mochila multidimensional (Garey y Johnson, 1979). Con el objetivo de garantizar la confiabilidad, robustez y eficacia del sistema, se llevó a cabo una campaña exhaustiva de verificación compuesta por 295 casos de prueba. Dichos ensayos abarcan los cinco ejes fundamentales del proceso de aseguramiento de la calidad en el desarrollo de software: pruebas unitarias de validación, pruebas de integración, pruebas de rendimiento, pruebas de regresión y pruebas de resistencia frente a condiciones operativas extremas. La Tabla 5 presenta de manera detallada los resultados obtenidos en cada categoría. Los resultados experimentales muestran que *Kathy* constituye un avance

Tabla 5: Resumen de Resultados de Testing del Sistema

Categoría	Ejecutados	Éxito	Fallos	Cobertura
Unitarios	147	142	5	96.6 %
Integración	32	31	1	96.9 %
Rendimiento	15	15	0	100 %
Regresión	89	87	2	97.8 %
Estrés	12	11	1	91.7 %
Total	295	286	9	97.0 %

significativo en la intersección entre las ciencias de la computación y la salud de la mujer, destacando en:

- Eficiencia algorítmica: Se logra una reducción del 73,2 % en los tiempos de consulta gracias a optimizaciones con caché LFU (*Least Frequently Used*).
- Precisión estimada: 95,2 % de concordancia en las recomendaciones básicas, según criterios de validación y comparación de la literatura científica especializada.
- Cobertura funcional: El sistema gestiona 4 fases del ciclo menstrual, 6 tipos de restricción dietética (ninguna, vegana, vegetariana, sin gluten, sin lactosa y sin frutos secos) y 12 objetivos nutricionales simultáneos: maximizar hierro, calcio, magnesio, zinc, vitamina D, vitaminas B, omega-3, fibra y proteína; y minimizar azúcares simples, sodio e inflamación sistémica.
- Escalabilidad demostrada: El procesamiento de consultas presenta un comportamiento temporal lineal $\mathcal{O}(n)$ hasta 1,000 hechos en la base de conocimiento.

Los resultados corroboran la hipótesis inicial: las fluctuaciones hormonales del ciclo menstrual pueden modelarse computacionalmente con lógica de primer orden en Prolog y es posible automatizar en tiempo real la generación de recomendaciones dietéticas ajustadas. La evidencia empírica respalda esta afirmación en

dos dimensiones. Los tiempos de respuesta, entre 2,3 ms para consultas directas y 128,8 ms para optimización completa, confirman la viabilidad operativa del sistema en uso interactivo. Además, la precisión promedio del 94,8 % demuestra que las recomendaciones automáticas son coherentes con el conocimiento experto de la base de conocimiento.

4. Conclusiones

Este trabajo presentó el sistema experto *Kathy*, una propuesta innovadora y de alto impacto en nutrición personalizada. Se definió una metodología de integración entre programación lógica e imperativa, implementada en una arquitectura híbrida Python–Prolog, inédita en aplicaciones de salud hormonal femenina y superior a enfoques basados solo en reglas expertas o filtrado colaborativo (Logapriya y Surendran, 2023a, 2023b). Esta integración se complementa con un autómata finito determinista (AFD) de 7 estados, diseñado para gestionar de forma lógica, formal y escalable las interacciones estructuradas entre profesionales de la salud y pacientes.

Se formalizó una taxonomía de complejidad aplicada para clasificar empíricamente cada consulta por su orden computacional ($\mathcal{O}(1)$, $\mathcal{O}(n)$, $\mathcal{O}(n \log n)$, $\mathcal{O}(n^2)$), luego validado con un *benchmark* automatizado. También se diseñó una heurística *greedy* para el problema NP-completo de planificación nutricional multiobjetivo, con modelo de penalización y esquema de balance entre restricciones y objetivos, que alcanzó un factor de aproximación $\rho \geq 0,63$.

La implementación técnica diseñó una arquitectura escalable y replicable basada en una base de conocimiento con 164 hechos y 23 reglas con respaldo científico. Se integró un sistema de caché LFU (*Least Frequently Used*) que reduce en un 73.2 % el tiempo de respuesta en consultas frecuentes y un *framework* de pruebas médicas con 295 casos que cubre cerca del 97 % de los escenarios funcionales definidos. Desde la perspectiva social, el sistema democratiza el conocimiento nutricional al ofrecer acceso gratuito a recomendaciones especializadas, promover el conocimiento hormonal sobre el ciclo menstrual y fomentar la educación científica en salud y nutrición basada en evidencia.

Finalmente, se logró una *contribución de impacto real y transferible* mediante una herramienta operativa aplicable en centros de salud, programas de bienestar y diversos entornos clínicos, adaptable a distintas condiciones médicas y poblaciones vulnerables. Este sistema trasciende lo académico y ofrece una solución concreta, científicamente fundamentada y tecnológicamente viable para optimizar el acceso a la nutrición personalizada.

A pesar de estas fortalezas, se identificaron cuatro limitaciones: (1) el modelo biológico estándar de 28 días no refleja la variabilidad fisiológica femenina, ya que alrededor del 15–20 % de las mujeres tienen ciclos fuera del intervalo de 21–35 días, lo que restringe la aplicabilidad y generalización del sistema; (2) en el análisis cuantitativo, la heurística *greedy* alcanza solo el $87,3 \pm 4,2$ % de la calidad de la solución óptima; (3) la integración PySwip–Prolog introduce una dependencia tecnológica que puede dificultar el despliegue en entornos computacionales restrictivos; y (4) la ausencia de mecanismos de aprendizaje adaptativo basados

en la retroalimentación del usuario limita significativamente la personalización del modelo.

Este sistema experto proporciona una base sólida para el asesoramiento nutricional personalizado, pero puede aumentar su alcance e impacto ampliando la base de conocimiento, integrándose con la historia clínica electrónica (HCE), desarrollando una interfaz gráfica multiplataforma con retroalimentación del usuario y modelando dinámicamente en tiempo real el ciclo hormonal y la sintomatología. La implementación de estas mejoras consolidaría a *Kathy* como una herramienta de gran utilidad, tanto en el ámbito académico e investigativo como en contextos clínicos reales y plataformas de salud digital personalizadas.

Referencias

- Bashaboina, S. (2026). Seed Cycling and Hormonal Health: Scientific Evidence vs. Popular Claims. *International Journal of Science and Research (IJSR)*, 15(3), 696-697. <https://doi.org/10.21275/SR25326213144>
- Brown, M., Rosenthal, M., & Yeh, D. D. (2021). Implementation Science and Nutrition: From Research to Practice. *Nutrition in Clinical Practice*, 36(3), 586-597. <https://doi.org/10.1002/ncp.10677>
- Chavarro, J. E., Rich-Edwards, J. W., & Willett, B. A. R. W. C. (2007). Dietary fatty acid intakes and the risk of ovulatory infertility. 85(1), 231-237.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms*. PHI Learning.
- Dalton, K. (1984). *Premenstrual Syndrome and Progesterone Therapy*. Year Book Medical Publishers I.
- Freeman, E. W., Sammel, M. D., Lin, H., & Nelson, D. B. (2006). Associations of Hormones and Menopausal Status With Depressed Mood in Women With No History of Depression. *Archives of General Psychiatry*, 63(4), 375-382. <https://doi.org/10.1001/archpsyc.63.4.375>
- Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. W.H.Freeman & Co Ltd.
- Knuth, D. (1997). *The Art of Computer Programming, Vol. 1: Fundamental Algorithms*. Addison-Wesley Professional.
- Logapriya, E., & Surendran, R. (2023a). Hybrid Recommendations System for Women Health Nutrition at Menstruation Cycle. *2023 International Conference on Information Technology (ICIT)*. <https://doi.org/10.1109/ICIT58056.2023.10331838>
- Logapriya, E., & Surendran, R. (2023b). Knowledge Based Health Nutrition Recommendations for During Menstrual Cycle. *2023 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*. <https://doi.org/10.1109/MIUCC58832.2023.10391518>

- Neapolitan, R. (2011). *Foundations of Algorithms*. Jones & Bartlett Learning.
- Rohatgi, A., & Dash, S. (2023).
 Period poverty and mental health of menstruators during COVID-19
 pandemic: Lessons and implications for the future.
Frontiers in Global Women's Health, Volume 4 - 2023.
<https://doi.org/10.3389/fgwh.2023.1128169>
- Russell, S., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach*.
 Pearson.
- Sims, S. T., & Heather, A. K. (2018).
 Myths and Methodologies: Reducing scientific design ambiguity in
 studies comparing sexes and/or menstrual cycle phases.
Experimental Physiology, 103(10), 1309-1317.
<https://doi.org/10.1113/EP086797>
- Sipser, M. (2021). *Introduction to the Theory of Computation*.
 Course Technology Inc.
- Sun, Y., Hu, J., Cheng, W., & Chen, H. (2024).
 DFA-RAG: conversational semantic router for large language model
 with definite finite automaton.
Proceedings of the 41st International Conference on Machine Learning.
- Tate, B. A. (2022). *Programmer Passport: Prolog*. Pragmatic Bookshelf.