



<b>Materia:</b>	Programación Lógica y Funcional		
<b>Nombre</b>	Sistema Experto (El Yerberito ilustrado)		
<b>Informe:</b>			
<b>Alumno(s):</b>	Chelsea Andrea Garcia Telles	<b>Fecha:</b>	18/05/
	America Citlalli Lopez Lemus		2025
	Paola Montserrat Ruiz Carmen		

## Objetivo

El objetivo de esta práctica es desarrollar e implementar un sistema experto basado en Prolog con una interfaz gráfica de usuario (GUI) utilizando XPCE, diseñado para proporcionar información sobre plantas medicinales y sus tratamientos para diversas enfermedades. Se busca integrar una base de datos de plantas y enfermedades, permitir consultas interactivas mediante botones y campos de texto, y mostrar resultados en un entorno visual, fomentando el aprendizaje sobre el uso tradicional de hierbas y la programación lógica.

## Temas del plan de estudios

### TEMA 4: Programación Lógica y Sistemas Expertos

#### 4.1 Desarrollo de Interfaces Gráficas con Prolog

- Introducción a la programación lógica con Prolog.
- Implementación de sistemas expertos para resolver problemas específicos (por ejemplo, consulta de plantas medicinales).
- Diseño y creación de interfaces gráficas usando XPCE para interacción usuario-sistema.
- Manejo de bases de datos dinámicas y consultas interactivas.

## Material

- **Hardware:** Computadora personal con sistema operativo Windows.
- **Software:**
  - SWI-Prolog con módulo XPCE para interfaces gráficas.
  - Editor de texto para escribir y editar el código Prolog.

## Marco Teórico



Un sistema experto es un programa de inteligencia artificial que emula el pensamiento de un humano en un dominio específico, utilizando una base de conocimientos y un motor de inferencia. En este caso, el sistema "El Yerberito Ilustrado" emplea Prolog, un lenguaje de programación lógica basado en predicados y reglas, para almacenar y consultar información sobre plantas medicinales y sus aplicaciones terapéuticas. La interfaz gráfica, desarrollada con XPCE, permite una interacción amigable, mostrando datos como tratamientos para enfermedades mediante botones y diálogos. Las bases de datos se estructuran con hechos y se consultan dinámicamente. Además, el uso de plantas medicinales se basa en conocimientos tradicionales, integrando herbolaria con tecnología moderna.

## Desarrollo

El desarrollo del sistema experto "El Yerberito Ilustrado" se llevó a cabo en varias etapas, a incluir desde la planificación inicial hasta la implementación final y las pruebas del sistema. A continuación, se detalla cada paso del proceso:

### 1. Planificación y Diseño Inicial:

El proyecto comenzó con la definición de los requisitos del sistema. Se identificó la necesidad de crear un sistema experto que permitiera consultar información sobre plantas medicinales y tratamientos para enfermedades, utilizando una interfaz gráfica amigable. Se decidió emplear Prolog como lenguaje principal debido a su capacidad para manejar bases de conocimientos lógicos, y XPCE para la interfaz gráfica, ya que es una librería compatible con SWI-Prolog para desarrollar GUIs. Se diseñó una estructura inicial del sistema que incluía:

- Una base de datos de plantas medicinales y enfermedades
- Una interfaz gráfica con botones, menús y campos de texto para consultas interactivas.
- Funcionalidades como búsqueda rápida de plantas, consulta de tratamientos y listados específicos.

### 2. Configuración del entorno de desarrollo:

Se creó un archivo inicial ( yerberito\_expert\_system.pl ) en el directorio , donde se desarrollaría el código. También se preparó una carpeta ( C:/yerberito\_images/ ) para almacenar imágenes, como yerberito.jpg , que se usarían en la interfaz.

### 3. Creación de la Base de Conocimientos:

La base de datos se implementó mediante hechos en Prolog. Inicialmente, se definieron hechos básicos como:

- Para listar plantas medicinales (p. ej., manzanilla, eucalipto, jengibre).
- Para asociar enfermedades con tratamientos, comenzando con un conjunto pequeño (eg, enfermedad('gripe', ['Sauco (infusion)', 'Paracetamol'])).



A lo largo del desarrollo, se expandió esta base para incluir más enfermedades, como "acidez estomacal", "cistitis" y "eczema", cada una con tratamientos que combinaban plantas (por ejemplo, "Aloe vera (gel)" para quemaduras), medicamentos (por ejemplo, "Ibuprofeno" para el dolor), y consejos (por ejemplo, "Descanso" para fiebre).

#### 4. Desarrollo de la Interfaz Gráfica (GUI):

La interfaz se implementó en el predicado `iniciar_sistema`. Se creó un diálogo principal ( `MainDialog` ) de 800x600 píxeles. La GUI se estructuró en secciones:

- **Búsqueda Rápida:** Un campo de texto y un botón ("Buscar Planta") para buscar plantas por nombre.
- **Menú de Plantas:** Un menú desplegable que lista todas las plantas disponibles, ordenadas alfabéticamente con `sort/2`.
- **Botones de Consulta:** Dos grupos de botones ( `ButtonGroup1` y `ButtonGroup2` ) para funciones como "Información de la Planta", "Tratamientos" y "Plantas Venenosas". Inicialmente, se intentó usar una imagen de fondo con el método `stretch` en `XPCE`, pero esto generó errores de compatibilidad. Como solución, se optó por colocar una imagen debajo de los botones, utilizando una imagen de objeto y ajustando su posición con `point(250, 10)` para centrarla horizontalmente.

#### 5. Implementación de Funcionalidades:

Se programaron varios predicados claves para las funcionalidades del sistema:

- **Búsqueda Rápida:** Permite buscar una planta por nombre y muestra su información detallada en un nuevo diálogo. Si la planta no existe, se muestra un mensaje de error.
- **Consulta de Tratamientos:** Un botón "Tratamientos" abre un diálogo donde el usuario ingresa una enfermedad (por ejemplo, "tos"). El sistema consulta la base y muestra los tratamientos ("Eucalipto (vapor)", "Jarabe").
- **Listados Genéricos:** Utilizado para botones como "Plantas Analgésicas" o "Enfermedades Tratadas", mostrando listas formateadas con viñetas. Se implementó manejo de errores con `catch/3` para capturar excepciones (archivo de imagen no encontrado) y mostrar mensajes amigables.

#### 6. Iteraciones y mejoras:

- **Corrección de errores:** Se resolvió el problema con el método `stretch` de `XPCE`, reemplazándolo por una imagen centrada. También se ajustó el manejo de imágenes para verificar su existencia ( `exist_file/1` ) y mostrar un texto alternativo ("Imagen no disponible") si faltaban.
- **Normalización de Entradas:** Se utilizamos `downcase_atom/2` en las consultas de enfermedades para hacerlas sin distinción entre mayúsculas y minúsculas, mejorando la experiencia del usuario.



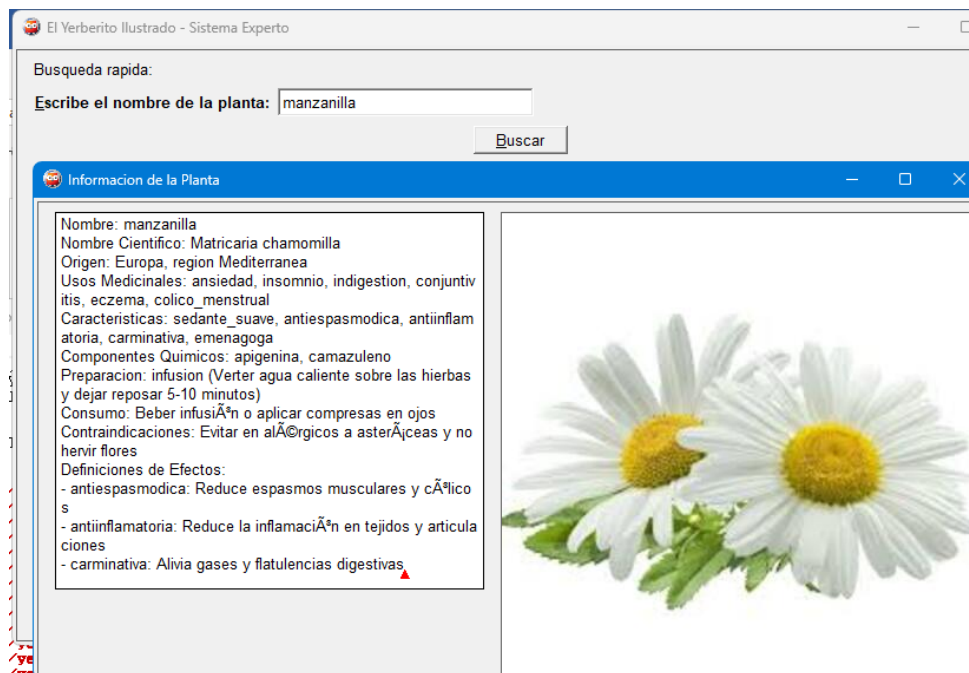
## 7. Pruebas

y

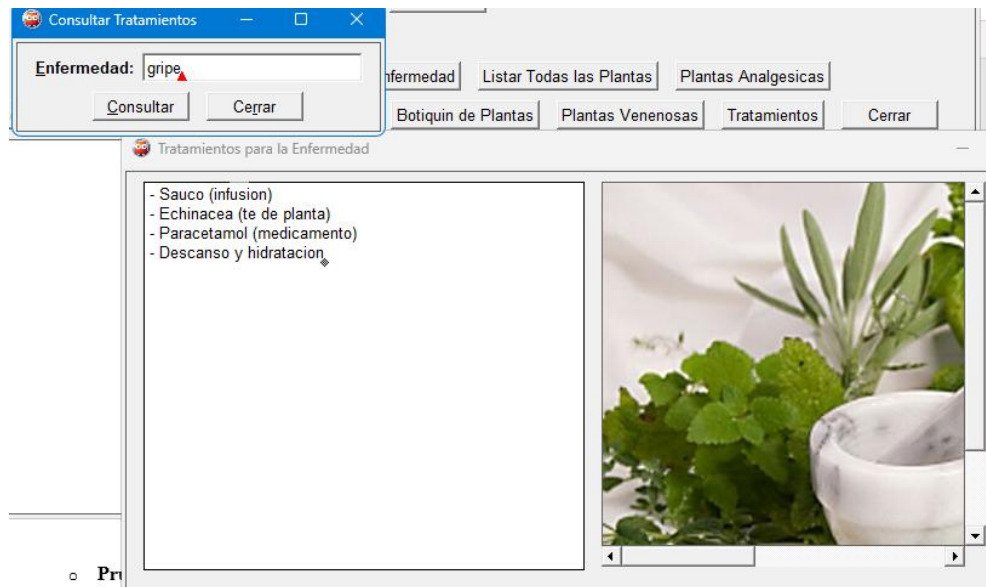
Depuración:

Se realizaron pruebas iterativas para validar cada funcionalidad:

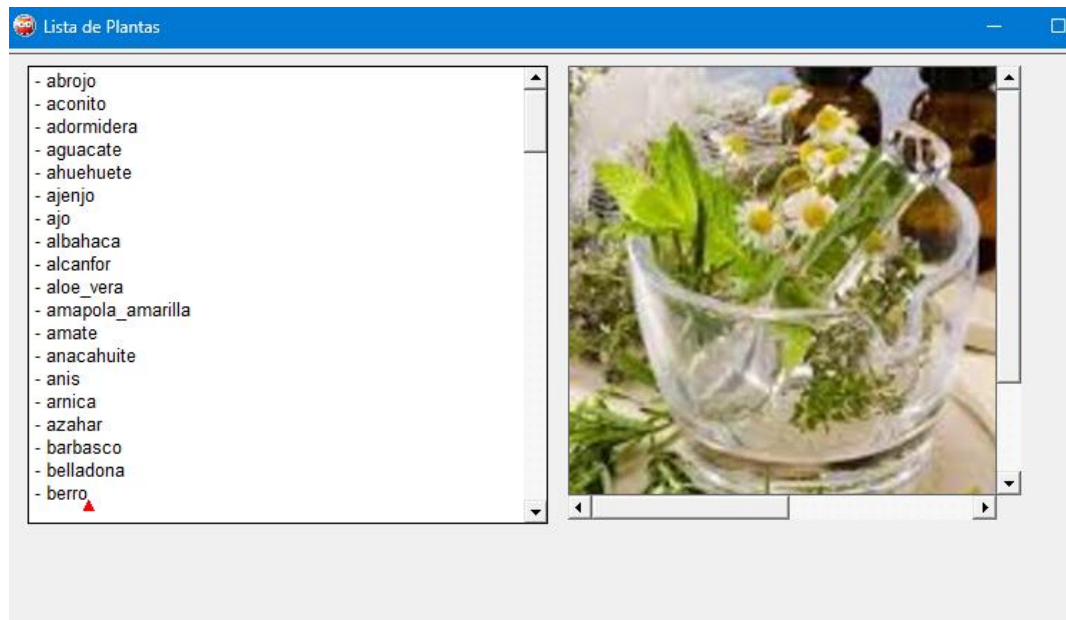
- **Prueba de Búsqueda:** Consultar "manzanilla" mostró su información correctamente, incluyendo una imagen específica si estaba disponible.



- **Prueba de Tratamientos:** Ingresar "gripe" mostró "Sauco (infusión)" y "Paracetamol". Se verificó que enfermedades no registradas devolvieron "No se encontraron tratamientos".



- **Prueba de Listados:** El botón "Enfermedades Tratadas" listó todas las enfermedades de la base, formateadas con viñetas.



Se depuraron errores como problemas de diseño (por ejemplo, botones superpuestos) ajustando



el tamaño y la disposición de los elementos en la GUI. También se verificó que la imagen centrada ( yerberito.jpg ) se mostrará correctamente en C:/yerberito\_images/ .

## Resultado

El resultado fue un sistema experto funcional que permite a los usuarios:

- Buscar información sobre plantas medicinales (p. ej., detalles de "manzanilla").
- Consultar tratamientos para enfermedades (eg, "tos" muestra "Eucalipto" y "Jarabe").
- Listar plantas por categoría (por ejemplo, "Plantas Analgésicas").
- Visualizar una interfaz gráfica con botones, menús y una imagen centralizada.
- La base de datos se expandió con éxito, cubriendo condiciones como "acidez estomacal" y "varices", y la GUI se adaptó para evitar errores de compatibilidad con XPCE. Sin embargo, la precisión depende de la coincidencia exacta de los nombres de enfermedades.

## Evaluación

La práctica cumplió con los objetivos establecidos, demostrando habilidades en programación lógica, diseño de interfaces y manejo de bases de conocimientos. Las fortalezas incluyen la integración de una interfaz interactiva y una base de datos extensa. Debilidades observadas fueron la dependencia de rutas de archivos (C:/yerberito\_images/ ) y la necesidad de normalizar entradas de usuario para evitar fallos

## Conclusiones

En conclusión el desarrollo de "El Yerberito Ilustrado" resultó exitoso al integrar de manera efectiva Prolog y XPCE, dando lugar a un sistema experto intuitivo y accesible que combina conocimientos tradicionales sobre plantas medicinales con tecnología moderna. Esta práctica no solo fortaleció la comprensión de bases de datos lógicas y el diseño de interfaces gráficas de usuario (GUIs), sino que también resaltó la relevancia de las hierbas medicinales como recurso terapéutico en un contexto tecnológico, promoviendo su valoración y aplicación práctica. El proceso permitió identificar áreas de mejora, como la necesidad de optimizar la robustez del sistema frente a entradas no válidas y ampliar la base de datos con más plantas y enfermedades regionales. En conjunto, el proyecto demostró el potencial de la programación lógica para resolver problemas del mundo real, dejando una base sólida para futuras expansiones y aplicaciones en campos como la herbolaria digital o la educación en salud.



## Bibliografía

- *Objetos con PROLOG – acervo para el mejoramiento del aprendizaje de alumnos de ingeniería, en Inteligencia Artificial.* (n.d.). [https://virtual.cuautitlan.unam.mx/intar/?page\\_id=432](https://virtual.cuautitlan.unam.mx/intar/?page_id=432)
- *Graphical applications.* (n.d.). <https://www.swi-prolog.org/Graphics.html>