

**Universidad de Sancti Spíritus
“José Martí Pérez”
Facultad de Informática
Carrera de Ingeniería Informática**



*Trabajo de Diploma para optar por el título de
Ingeniería Informática*

Título: Sistema Experto para Diagnosticar
Enfermedades Fúngicas en los cultivos de interés
agrícola en la provincia de Sancti Spíritus (SEDEF).

Autor: Danay Quintana Sánchez

Tutor: Lic. Luis Alberto Quintero Domínguez

junio de 2014

Curso 2013-2014

Dedicatoria

A las personas que más amo en esta vida, a mi madre María Luisa Sánchez y a mi padre Olber Eduardo Quintana porque gracias a su amor, apoyo y comprensión llegué a ser la persona que hoy soy, ojalá y estuvieran tan orgullosos de mi como yo lo estoy de ellos.

Agradecimientos

- ✓ *A mis padres quienes lo son todo para mí.*
- ✓ *A mi novio Yosmany por su infinito amor y por estar a mi lado durante estos cinco años de mi carrera.*
- ✓ *A mis abuelas por ser las mejores y a mi abuelo Máximo que aunque esté lejos también lo quiero mucho.*
- ✓ *A todos mis tíos, tías y primos en general a los cuales adoro.*
- ✓ *A mi querida suegra que ha sido como una madre para mí y a mis cuñados que son los hermanos que nunca tuve.*
- ✓ *A mi tutor por guiarme y darme lo mejor de sí.*
- ✓ *A todos mis compañeros de grupo.*
- ✓ *A todos los profesores que me impartieron clases.*
- ✓ *A todas aquellas personas que de cierta manera siempre se interesaron por mí:*

Muchas Gracias

Resumen

Desde la antigüedad el hombre siempre soñó con imitar a la naturaleza para reproducir lo alcanzado por ella. El mismo sintió un profundo interés por crear máquinas inteligentes que lo ayudaran en su propósito, es de esta manera que surge la Inteligencia Artificial como una nueva ciencia dentro de la cual los sistemas expertos han constituido un fuerte pilar con sus aplicaciones.

En esta investigación se diseña e implementa un sistema experto capaz de ayudar a diagnosticar enfermedades fúngicas en los cultivos de interés agrícola en la provincia de Sancti Spíritus. Para ello se utilizó el lenguaje de programación lógica Prolog, para la interfaz de usuario se utilizó Java y las reglas de producción como forma de representar el conocimiento. La metodología empleada es la propuesta por el Dr.C Mateo Lezcano en su libro: Prolog y los Sistemas Expertos.

Abstract

Since ancient times man has always dreamed of imitating nature to reproduce what was achieved by it. Himself felt a deep interest in creating intelligent machines to help him on his purpose, it is in this way that the Artificial Intelligence emerges as a new science in which expert systems have been a strong pillar with their applications.

In this research is designed and implemented an expert system that can help diagnose fungal diseases in agricultural interest crops in the province of Sancti Spiritus. For that was used the logic programming language Prolog, for user interface was used Java and the production rules as a way of representing knowledge was used. The methodology employed is the given by the Phd. Mateo Lezcano in his book: Prolog and Expert Systems.

Índice

Introducción	9
Capítulo1: Los sistemas expertos y las enfermedades fúngicas.....	13
1.1 Proceso de identificación de enfermedades en las plantas .	13
1.1.1 Virus	15
1.1.2 Bacterias	16
1.1.3 Hongos.....	16
1.2 Inteligencia Artificial (IA).....	17
1.2.1 Historia de la Inteligencia Artificial (IA).....	18
1.2.2 Campos de la Inteligencia Artificial (IA).....	20
1.3 Sistemas Expertos (SE)	21
1.3.1 Desarrollo Histórico de los sistemas expertos (SE)	22
1.3.2 Ventajas de los sistemas expertos (SE).....	24
1.4 Los sistemas expertos en el diagnóstico de enfermedades.	26
1.5 Metodologías para los sistemas expertos (SE).....	27
1.5.1 Metodología descrita por José Domingo Carrillo Verdun. 28	
1.5.2 Metodología descrita por Marlene Carlos Soto.	28
1.5.3 Metodología Descrita por Mateo Lezcano Brito.	31
1.6 Tecnologías y Herramientas	34
1.6.1 Prolog.....	34
1.6.2 SWI Prolog	35
1.6.3 Java	36

1.6.4	NetBeans.....	37
1.7	Conclusiones Parciales	38
Capítulo 2: Análisis y diseño del SEDEFP.....		39
2.1	Identificación.....	39
2.2	Establecer Conceptos	41
2.3	Formalización.....	42
2.3.1	Clasificación del conocimiento.....	42
2.4	Implementación	43
2.4.1	Formalización de los casos.....	43
2.4.2	Diseño de los módulos del sistema	48
2.4.3	Diálogos del sistema	49
2.4.4	Diagrama de casos de uso del sistema.....	50
2.4.5	Descripción de los Casos de uso del Sistema	51
2.5	Conclusiones Parciales	53
Capítulo 3: Implementación y descripción del SEDEFP.....		54
3.1	Implementación de la interfaz del usuario.....	54
3.1.1	Los JComboBox.....	55
3.1.2	Los JList.	56
3.1.3	El uso de Iconos, botones y etiquetas	57
3.1.4	Otros Contenedores	58
3.1.5	Barra de menú para un JFrame	59
3.1.6	Evento actions.....	60
3.2	Aspectos a tener en cuenta para la interacción de SWI-Prolog y java	60
3.3	Requerimientos del Sistema.....	62

3.3.1	Facilidades que brinda al usuario	63
3.3.2	Menús.....	64
3.4	Estándares en la interfaz de la aplicación	65
3.4.1	Concepción general de la ayuda	66
3.4.2	Tratamiento de excepciones	67
3.5	Conclusiones Parciales	68
	Conclusiones	69
	Recomendaciones	70
	Bibliografía	71

Introducción

Introducción

La producción de cultivos agroindustriales se ha visto diezmada por el ataque de plagas o el padecimiento de enfermedades, lo cual conlleva a pérdidas económicas que pueden ser evitadas si son detectadas con anterioridad. El diagnóstico de enfermedades o plagas se lleva a cabo a través del monitoreo de las plantaciones de cultivos, donde el objetivo es detectar anomalías presentes sobre las plantas, como por ejemplo tallos secos, presencia de pústulas en alguna parte de la planta, presencia de insectos nocivos, entre otros. Así, teniendo un conjunto de síntomas visibles dentro del campo de cultivo se puede realizar un diagnóstico que permita tomar las medidas para minimizar el daño sobre los cultivos.

Las plantas pueden ser atacadas por muchas enfermedades que son causadas por hongos, bacterias o virus, siendo el noventa y cinco por ciento causadas por hongos. Las enfermedades fúngicas que son las que generan los hongos son combatidas por fungicidas, sin embargo una vez que las plantas son atacadas por bacterias o virus no existen productos eficaces y lo único que resta es prevenir estas patologías.

En la Empresa de Sanidad Vegetal de la provincia de Sancti Spíritus se hace muy difícil la tarea de diagnosticar enfermedades en las plantas ya que no existen los suficientes materiales de apoyo, existe lentitud a la hora de darle solución a los problemas que presenta el cultivo y en ocasiones ha habido fallas en algunos diagnósticos.

Desde finales de los años cincuenta, el diagnóstico de enfermedades ha venido siendo un tema de interés en el área computacional de la Inteligencia Artificial (IA). La IA es una de las áreas con más retos de las ciencias de la computación ya que ha tomado a la inteligencia como la característica universalmente aceptada para diferenciar a los humanos de otras criaturas ya sean vivas o inanimadas, para construir programas o

computadoras inteligentes (Ramos, Sistemas Expertos, 2000); uno de los principales objetivos de los investigadores en IA es la reproducción automática del razonamiento humano, donde juegan un papel importante los Sistemas Expertos (SE).

Los SE son programas informáticos que intentan superar la lógica y habilidad de un profesional experto en la resolución de problemas, que requieren experiencia, conocimiento, juicio e interacciones complejas. La fiabilidad de dichos sistemas es proporcional al conocimiento de los expertos que lo produjeron. Estos sistemas pueden utilizar datos en casi cualquier formato y sugerir una solución al problema, incluso utilizar datos incompletos o incorrectos, siempre y cuando el grado de certeza de los mismos sean cuantificados por el experto y se incluyan en la base de conocimientos.

En fitopatología, que es la ciencia que se encarga del estudio de las enfermedades de las plantas, se utilizan con frecuencia estos sistemas con fines de diagnóstico, por ejemplo, identificar la causa de una enfermedad por los síntomas y las observaciones relacionadas. Mediante la incorporación de modelos de infección de enfermedades de cultivos en la base de conocimientos de la computadora, el sistema experto puede asesorar a los productores sobre la probabilidad de ocurrencia real de la enfermedad, períodos de infección y recomendaciones sobre tipo, cantidad y momento de aplicación de pesticidas (Agrios, 2005).

Como **problema de investigación** se plantea:

- Limitaciones en el proceso de identificación de las enfermedades fúngicas que afectan a los cultivos de interés agrícola en la provincia de Sancti Spíritus.

Tomando en cuenta lo expuesto anteriormente, y como solución al problema se define como **objetivo general** de este trabajo:

- Desarrollar un sistema experto para diagnosticar enfermedades fúngicas en los cultivos de interés agrícola en la provincia de Sancti Spíritus.

Las preguntas de investigación que se plantean para darle solución al problema son:

- ¿Cuáles son los fundamentos teóricos y metodológicos que permiten desarrollar una herramienta informática para diagnosticar enfermedades fúngicas en los cultivos?
- ¿Cómo diseñar un sistema experto para diagnosticar enfermedades fúngicas en los cultivos de interés agrícola en la provincia de Sancti Spíritus?
- ¿Cómo implementar un sistema experto para diagnosticar enfermedades fúngicas en los cultivos de interés agrícola en la provincia de Sancti Spíritus?

A partir de las preguntas de investigación anteriormente expuestas se trazan las siguientes tareas de investigación:

- Determinar los fundamentos teóricos y metodológicos que permiten desarrollar una herramienta informática para diagnosticar enfermedades fúngicas en los cultivos.
- Diseñar un sistema experto para diagnosticar enfermedades fúngicas en los cultivos de interés agrícola en la provincia de Sancti Spíritus.
- Implementar un sistema experto para diagnosticar enfermedades fúngicas en los cultivos de interés agrícola en la provincia de Sancti Spíritus.

El presente trabajo se estructura de la siguiente forma: introducción, tres capítulos, conclusiones, recomendaciones y bibliografía.

En el **Capítulo 1** se describe el proceso de identificación de enfermedades en las plantas. También se abordan temas relacionados con la IA y los SE, así como su utilización en el diagnóstico de

enfermedades humanas y de las plantas. Además se realiza un bosquejo de las metodologías para la construcción de SE y se determinan las herramientas y tecnologías más apropiadas para su implementación.

En el **Capítulo 2** se detallan las etapas seguidas para la construcción del SE que están descritas en la metodología del DrC. Mateo Lezcano. También se muestra el diagrama de casos de uso del sistema y describen los casos de uso del SEDEFP.

El **Capítulo 3** explica la implementación de la interfaz de usuario y los requerimientos del hardware y software para lograr un rendimiento óptimo del SE. Además se detallan las facilidades desde el punto de vista del usuario que brinda el mismo, terminando este capítulo con un manual de usuario para la correcta utilización del software.

Capítulo 1

Capítulo 1: Los sistemas expertos y las enfermedades fúngicas.

En este capítulo se describe el proceso de identificación de enfermedades en las plantas. También se abordan temas relacionados con la IA y los SE, así como su utilización en el diagnóstico de enfermedades humanas y de las plantas. Además se realiza un bosquejo de las metodologías para la construcción de SE y se determinan las herramientas y tecnologías más apropiadas para su implementación.

1.1 Proceso de identificación de enfermedades en las plantas

El diagnóstico de enfermedades de las plantas es uno de los aspectos más importantes en el proceso de la protección vegetal. En él se integran conocimientos de Fitopatología, Genética, Fisiología Vegetal, Edafología, Climatología, Manejo Agronómico y Estadística, con la finalidad de tomar decisiones acertadas en el momento en que se requieran (Roque, 2014).

Hay muchos factores que influyen el desarrollo de enfermedades en las plantas. Sin embargo, las enfermedades no pueden ocurrir a menos que estén presentes los siguientes elementos: una planta susceptible, un patógeno y un ambiente favorable. Algunas enfermedades no ocurren si no hay un vector que las transmita, lo que es común en las enfermedades causadas por virus, las cuales pueden transmitirse por insectos o nematodos¹. Los agentes patógenos que infectan y enferman a las

¹ Los **nematodos**, **nematodes** o **nemátodos** son gusanos nematelmintos del superfilo **Ecdysozoa**. Estos animales disponen de aparato digestivo con forma de conducto recto, que ocupa toda la extensión del **cuerpo**.

plantas son de diversa naturaleza, entre ellos cabe mencionar a los hongos, bacterias y virus. El diagnóstico de enfermedades no puede hacerse basándose solamente en los síntomas, aunque pueden hacerse algunas generalizaciones. Los síntomas causados por agentes infecciosos (hongos, bacterias, virus) y no infecciosos (deficiencias nutricionales, toxicidades, exceso o escasez de agua, contaminantes ambientales, acidez o alcalinidad del suelo) son similares. Un diagnóstico preciso solo puede hacerse luego de evaluar la planta afectada por observación directa o cultivar los patógenos en medios específicos (Agrios, 2005).

La determinación de la etiología² de una enfermedad en un cultivo es habitualmente un proceso complejo que requiere de la información más completa posible, sobre el cultivo en estudio, las condiciones ambientales donde se desarrolla, prácticas culturales utilizadas y agentes bióticos o abióticos que pudiesen haber interferido el desarrollo normal de las plantas. Los procesos de identificación del agente causal parten de la observación de síntomas y la toma de muestras para el examen de laboratorio, en el cual se buscará pistas del posible agente causal. La calidad de la muestra será determinante para este proceso, la muestra debe contener tejido afectado y sano; cuidando de no llegar a los extremos de enviar muestras totalmente deterioradas (Bustamante, 2007).

Para proceder a realizar un adecuado diagnóstico de un problema, es necesario reconocer perfectamente las condiciones y la apariencia que posee una planta sana para, de esta forma, poder reconocer también cuando sufre alguna afección. También para realizar un diagnóstico certero, es necesario analizar a fondo, además de los síntomas que

² La etiología es la ciencia que estudia las causas de las cosas. En medicina (patogénesis) se refiere al origen de la enfermedad

ocasiona el problema, el comportamiento de este en el campo. Lógicamente, un diagnóstico exacto solo se logra mediante el empleo de métodos rutinarios de laboratorio que permiten, generalmente, la captura del patógeno (Durán Quirós , Mora Acedo, & Ramí, 2009).

1.1.1 Virus

Los virus son patógenos infecciosos demasiado pequeños para ser vistos en el microscopio de luz, pero que a pesar de su tamaño son capaces de causar caos. Las formas más simples de virus están compuestas por una pequeña porción de ácido nucleico rodeado de una cubierta proteica o envoltura proteica o cápside. Como en el caso de otros organismos, los virus portan información genética en sus ácidos nucleicos, los cuales típicamente codifican tres o más proteínas. Todos los virus son parásitos que dependen de la maquinaria celular de sus hospedantes para reproducirse. Ellos no son activos fuera de sus hospederos, lo cual ha llevado a que muchos sugieran que no son organismos vivos (Colectivo de autores, 2011). Todos los tipos de organismos vivos incluyendo animales, plantas, hongos y bacterias son hospedantes de virus, pero la mayoría de los virus infecta solo un tipo de hospedante. Estos causan muchas e importantes enfermedades vegetales y son responsables de pérdidas en el rendimiento y la calidad de los cultivos en todas partes del mundo (Gonsalves, 2004).

1.1.2 Bacterias

Bacteria es una palabra que deriva del griego *bacterion*, y significa bastón pequeño. Estos microorganismos son unicelulares, cuyo material genético carece de una membrana celular y por ello no está organizado en un núcleo. Sus células están compuestas por citoplasmas que contienen ADN y ribosomas pequeños. El citoplasma está rodeado por una membrana y una pared celular. La mayoría de las bacterias fitopatógenas tienen forma de bastón o bacilo y poseen flagelos, usualmente en los polos, y fimbrias o pili (Lambrecht A. K., 2004).

Las bacterias asociadas a las plantas pueden ser benéficas o dañinas. Todas las superficies vegetales tienen microbios sobre ellas (epífitos), y algunos microbios viven dentro de las plantas (endófitos). Algunos son residentes y otros transitorios. Las bacterias se encuentran entre los microorganismos que colonizan a las plantas en forma sucesiva a medida que éstas maduran. Las células bacterianas individuales no se pueden observar sin un microscopio, sin embargo, poblaciones grandes de bacterias se vuelven visibles en forma de agregados en medio líquido, como *biofilms* en plantas, suspensiones viscosas taponando los vasos de las plantas, o como colonias en placas de Petri en el laboratorio (Lambrecht V. &, 2004).

1.1.3 Hongos

Los hongos constituyen un grupo muy numeroso de organismos (se han descrito aproximadamente 500 000, pero se estima que pueden existir entre 1 y 1,5 millones de especies) que presentan una amplia distribución en la naturaleza, contribuyendo a la descomposición de la materia orgánica y participando en los ciclos biológicos (Balvantín C., 2009).

Inicialmente, los hongos fueron clasificados dentro del reino planta ya que fueron considerados organismos inmóviles presentando estructuras que se asientan firmemente en el sustrato sobre el que crecían. Sin embargo, cuando se ha aplicado la biología molecular en los estudios taxonómicos se ha observado que los hongos están más próximos al reino animal que al de las plantas (Echemendía, 2007).

Esta palabra proviene del griego *sphongos*, que significa esponja. La mayoría de las especies están compuestas por filamentos tubulares que pueden ser cenocíticos o septados, cuyo conjunto se denomina micelio. Los componentes principales de las paredes celulares de los hongos son celulosa, pectona, y quitina, y se distinguen por la carencia de clorofila, posesión de núcleos y nucléolos con membranas bien definidas (Harper G, 2010).

En el campo de la agricultura existen reportes de que algunas especies de hongos pueden emplearse para el biocontrol de insectos. Sin embargo, existen otros que son responsables de muchas enfermedades en plantas. Se han reportado más de 8000 especies de hongos que causan enfermedades en las plantas, siendo las especies de las divisiones *ascomycetes*, *basidiomycetes* y *oomycota* las más comunes, que ocasionan pérdidas económicas importantes a nivel nacional (Evleva E.V., 2006). Dichas micosis generan una amplia variedad de síntomas en sus hospederos: manchas cloróticas y necróticas, cribados, canchales, tizones, podredumbres húmedas o secas, agallas, abolladuras, costras, ahogamientos, marchitamientos y pústulas.

1.2 Inteligencia Artificial (IA)

Se sabe que una de las características de los seres vivos es la inteligencia, que todos tienen una capacidad o grado distinto de inteligencia, que depende de factores biológicos, psicológicos y de desarrollo mental. Se puede decir que el concepto de inteligencia es variable dependiendo del tipo de investigador; es decir los conceptos de inteligencia de un psicólogo, un antropólogo, un abogado u otro profesional pueden ser distintos y válidos para sus respectivas áreas.

Existen varias definiciones de IA, pero la mayoría de los autores coinciden en que es, en esencia, lograr que una máquina tenga inteligencia propia. La IA es una de las áreas más fascinantes y con más retos de la ciencia de la computación ya que ha tomado a la inteligencia como la característica universalmente aceptada para diferenciar a los humanos de otras criaturas ya sean vivas o inanimadas, para construir programas o computadoras inteligentes (Ramos, Sistemas Expertos, 2000).

Esta es una definición muy completa e interesante, pero no es la única, para algunos otros autores, la IA es el estudio de cómo hacer que las computadoras hagan cosas que, en estos momentos, hace mejor el hombre (Briz, 2007).

La inteligencia artificial, por lo tanto, consiste en el diseño de procesos que, al ejecutarse sobre una arquitectura física, producen resultados que maximizan una cierta medida de rendimiento. Estos procesos se basan en secuencias de entradas que son percibidas y almacenadas por la mencionada arquitectura (Definición.De, 2008).

1.2.1 Historia de la Inteligencia Artificial (IA)

Desde tiempos muy remotos en la historia el hombre ha soñado con crear máquinas con inteligencia propia, muestra de esto pueden ser: el mito del coloso de Rodas entre los griegos, las estatuas parlantes del medioevo, el androide de Von Kempel en que jugó al ajedrez con Napoleón, y el motor analítico de Charles Babbage que calculaba logaritmos (Ávila, 2012). Sin embargo estos intentos resultaron infructuosos, hasta 1943 cuando la base de la IA fue asentada, gracias a Warren Mc Culloch y Walter Pitts, quienes propusieron un modelo de neurona de cerebro humano y animal. Estas neuronas nerviosas informáticas proporcionaron una representación simbólica de la actividad cerebral. Un tiempo después, Nobeit Wiener tomó estas y otras ideas y las elaboró dentro de un mismo campo que se llamó Cibernética, a partir del cual nacería la IA (Vega, 2009). En la década de los 50, hubo dos corrientes de investigaciones tras la propuesta de Alan Turing sobre la inteligencia de las máquinas en su artículo Maquinaria Computacional e inteligencia. La primera corriente fue fundada por John Von Neuman, quien trató de hacer analogías del cerebro humano y construyó a partir de esto las computadoras. Así, construyó varias máquinas que tomaban como base los conocimientos que hasta ese entonces existían sobre el cerebro humano, y diseñó los primeros programas que se almacenaban en la memoria de una computadora. La otra corriente fue la que inició Mc Culloch, quien decía que las leyes que gobiernan al pensamiento deben buscarse entre las reglas que gobiernan a la información y no entre las que gobiernan a la materia. Es decir, que a diferencia de Von Neuman, quien trataba de imitar las condiciones físico-químicas del cerebro, Mc Culloch se centró más en la forma en que el cerebro procesa la información (Norving, 1996).

A partir de la década de los 90 y con el desarrollo de la informática, se produce un amplio desarrollo en el campo de la IA y los SE, pudiéndose afirmar que estos se han convertido en una herramienta habitual en determinadas empresas en la actualidad. La evolución histórica de los métodos utilizados en el desarrollo de los SE también se ha producido a medida que se ha ido desarrollando la IA y los diferentes métodos que se han empleado para su resolución. El desarrollo de lenguajes como LISP y PROLOG condicionó esa evolución, así como investigaciones en diversos campos relacionados (Castillo Ron, 1999).

Múltiples son las ramas en las cuales las técnicas de la IA tienen aplicación. Entre ellos se destacan: los juegos, el procesamiento de lenguaje natural, la robótica, la astronáutica, la medicina (donde los computadores pueden ayudar a los médicos a hacer diagnósticos, supervisar la condición de los pacientes, alcanzando notable relevancia en áreas como la neurofisiología, la urología y las enfermedades infecciosas), en la programación automática, en el reconocimiento de modelos, en las telecomunicaciones, la industria, la aeronáutica, la electrónica, en la educación (en la cual los computadores pueden entender por qué los estudiantes cometen errores) y el transporte (Martín, 2010).

1.2.2 Campos de la Inteligencia Artificial (IA)

La IA es una disciplina muy amplia en la cual confluyen un gran número de campos entre los cuales se encuentran:

- Aprendizaje Automático (*Machine Learning*)
- Ingeniería del conocimiento (*Knowledge Engineering*)
- Lógica difusa (*Fuzzy Logic*)
- Redes neuronales artificiales (*Artificial Neural Networks*)
- Sistemas reactivos (*Reactive Systems*)
- Sistemas multi-agente (*Multi-Agent Systems*)
- Sistemas basados en reglas (*Rule-Based Systems*)

- Razonamiento basado en casos (*Case-Based Reasoning*)
- Sistemas expertos (Expert Systems)
- Redes Bayesianas (*Bayesian Networks*)
- Computación evolutiva (*Evolutionary Computation*)
- Estrategias evolutivas
- Algoritmos genéticos (*Genetic Algorithms*)
- Técnicas de Representación de Conocimiento
 - Redes semánticas (*Semantic Networks*)
 - Frames
- Visión artificial
- Audición artificial
- Lingüística computacional
- Procesamiento del lenguaje natural (*Natural Language Processing*)
- Minería de datos (*Data Mining*)

1.3 Sistemas Expertos (SE)

Antes de la aparición del ordenador, el hombre ya se preguntaba si se le arrebataría el privilegio de razonar y pensar. En la actualidad existe un campo dentro de la IA al que se le atribuye esa facultad: el de los SE. Estos sistemas permiten la creación de máquinas que razonan como el hombre, restringiéndose a un espacio de conocimientos limitado. En teoría pueden razonar siguiendo los pasos que seguiría un experto humano (médico, analista, empresario, etc.) para resolver un problema concreto. Este tipo de modelos de conocimiento por ordenador ofrece un extenso campo de posibilidades en resolución de problemas y en aprendizaje. Su uso se extiende ampliamente, debido a su importante impacto sobre los negocios y la industria (Ramos, Trabajos de administración de centros de cómputos, 2000).

No resulta fácil dar una definición de SE, entre otras cosas, porque el concepto de SE va evolucionando, ya que, a medida que se va progresando, sus funciones se van ampliando y resulta un concepto

cambiante. Edward Feigenbaum, de la Universidad de Stanford definió, en un Congreso Mundial de IA, un SE como: un programa de computador inteligente que usa el conocimiento y procedimientos de inferencia para resolver problemas que son lo suficientemente difíciles como para requerir la intervención de un experto humano para su resolución (García Valdivia, 2000). Hoy, con los avances conseguidos, resultaría más correcto definir un SE como: un sistema informático que simula el proceso de aprendizaje, de memorización, de razonamiento, de comunicación y de acción de un experto humano en una determinada rama de la ciencia, suministrando, de esta forma, un consultor que puede sustituirle con unas ciertas garantías de éxito (Quintanar, 2007).

1.3.1 Desarrollo Histórico de los sistemas expertos (SE)

Los SE proceden inicialmente de la IA a mediados de los años sesenta. En esos tiempos, se creía que bastaban unas pocas leyes de razonamiento con potentes computadoras para producir resultados brillantes (Samper, 2002).

En el año 1950 el campo de la automática recibe un gran impulso cuando Wiener desarrolla el principio de la retroalimentación. La teoría de la retroalimentación es base fundamental de los sistemas de control. En 1955 Newell y Simon desarrollan la Teoría de la lógica. Este desarrollo permitió desarrollar un programa que exploraba la solución a un problema utilizando ramas y nudos, seleccionando únicamente las ramas que más parecían acercarse a la solución correcta del problema. En 1956, se celebra una conferencia en Vermont de gran trascendencia en el desarrollo de la IA, John McCarthy propone por primera vez el uso del término IA para denominar el estudio del tema (Bonafío, 2008).

En 1957, aparece la primera versión de "*The General Problem Solver*" (GPS, Solucionador general de problemas), un programa capaz de solucionar problemas de sentido común pero no problemas del mundo real como diagnósticos médicos. El GPS utilizaba la teoría de la

retroalimentación de Wiener. En 1958 McCarthy anuncia su nuevo desarrollo el lenguaje LISP (*List Processing*), el lenguaje de elección para todos aquellos desarrolladores inmersos en el estudio de la IA. En 1963, el Instituto Tecnológico de Massachussets (MIT) recibe una subvención de 2,2 millones de dólares del gobierno de los Estados Unidos en concepto de investigación en el campo de la IA. De esa forma, se comprueba la importancia que el Gobierno concede a la investigación dentro de ese campo (Rendon, 2012).

En 1965 aparece DENDRAL, el primer SE. Es en ese año cuando Feigenbaum entra a formar parte del departamento de informática de Stanford. Allí conoció a Joshua Lederberg, el cual quería averiguar cuál era la estructura de las moléculas orgánicas completas. El objetivo de DENDRAL fue estudiar un compuesto químico. El descubrimiento de la estructura global de un compuesto exigía buscar en un árbol las posibilidades, y por esta razón su nombre es DENDRAL que significa en griego árbol. Antes de DENDRAL los químicos solo tenían una forma de resolver el problema, esta era tomar unas hipótesis relevantes como soluciones posibles, y someterlas a prueba comparándolas con los datos. La realización de DENDRAL duró más de diez años (1965-1975).

En 1972, en la Universidad de Standford se desarrolla MYCIN, SE dentro del campo de la medicina para diagnóstico de enfermedades infecciosas en la sangre. MYCIN se trataba de un SE para el diagnóstico de enfermedades infecciosas. Desde los resultados de análisis de sangre, cultivos bacterianos y demás datos, el programa era capaz de determinar, o en lo menos, sugerir el microorganismo que estaba causando la infección. Después de llegar a una conclusión, MYCIN prescribía una medicación que se adaptaba perfectamente a las características de la persona, tales como el peso corporal de este. Al mismo tiempo, Davir Marr propone nuevas teorías sobre la capacidad de reconocimiento visual de las diferentes máquinas (Rendon, 2012).

En 1973 se desarrolla el SE llamado TIERESIAS. El cometido de este SE era el de servir de intérprete entre MYCIN y los especialistas que lo manejaban, a la hora introducir nuevos conocimientos en su base de datos. El especialista debía utilizar MYCIN de una forma normal, y cuando este cometiera un error en un diagnóstico (hecho producido por la falta o fallo de información en el árbol de desarrollo de teorías) TEIRESIAS corregiría dicho fallo destruyendo la regla si es falsa o ampliándola si es eso lo que se necesita (Palacios, 2008).

Entre los años 80 a 85 se produce la revolución de los SE. En estos 5 años se crearon diversos SE como el DELTA, de General Electric Company, para la reparación de locomotoras diesel y eléctricas. Aldo en Disco para la reparación de calderas hidrostáticas giratorias usadas para la eliminación de bacterias. Se crearon multitud de empresas dedicadas a los SE como Teknowledge Inc., Carnegie Group, Symbolics, Lisp Machines Inc., Thinking Machines Corporation, Cognitive Systems Inc. Formando una inversión total de 300 millones de dólares. Los productos más importantes que creaban estas nuevas compañías eran las máquinas Lisp, que se trataba de unos ordenadores que ejecutaban programas LISP con la misma rapidez que en un ordenador central, y el otro producto fueron las herramientas de desarrollo de SE. En 1987 XCON empieza a no ser rentable. Los técnicos de DEC tuvieron que actualizar XCOM rápidamente llegándose a gastar más de dos millones de dólares al año para mantenimiento y algo parecido ocurrió con el DELTA (Quintanar, 2007).

1.3.2 Ventajas de los sistemas expertos (SE)

Los SE tienen innumerables ventajas, entre ellas se encuentran:

- Están disponibles ininterrumpidamente de día y noche, ofreciendo siempre su máximo desempeño.
- Pueden duplicarse ilimitadamente, es decir, tener tantos de ellos como se requieran.

- Pueden trabajar en entornos hostiles y peligrosos.
- Siempre se ajustan a las normas establecidas y son consistentes en su desempeño, no desarrollan apreciaciones subjetivas, tendenciosas, irracionales o emocionales. No padece de olvido, fatiga, dolor o comete errores de cálculo.
- No requiere un sueldo, promociones, seguros médicos, incapacidades.
- Siempre están dispuestos a dar explicaciones, asistir o enseñar a la gente, así como a aprender.
- Pueden tener una vida de servicio ilimitada.

Un SE es muy eficaz cuando tiene que analizar una gran cantidad de información, interpretándola y proporcionando una recomendación a partir de la misma. Un ejemplo es el análisis financiero, donde se estudian las oportunidades de inversión, dependiendo de los datos financieros de un cliente y de sus propósitos. Para detectar y reparar fallos en equipos electrónicos, se utilizan los SE de diagnóstico y depuración, que formulan listas de preguntas con las que obtienen los datos necesarios para llegar a una conclusión. Entonces recomiendan las acciones adecuadas para corregir los problemas descubiertos (Pacheco, 2009).

Este tipo de sistemas se utilizan también en medicina (como MYCIN y PUFF), y para localizar problemas en sistemas informáticos grandes y complejos. Los SE son buenos para predecir resultados futuros a partir del conocimiento que tienen. Los sistemas meteorológicos y de inversión en bolsa son ejemplos de utilización en este sentido. El sistema PROSPECTOR es de este tipo.

La planificación es la secuencia de acciones necesaria para lograr una meta. Conseguir una buena planificación a largo plazo es muy difícil. Por ello, se usan SE para gestionar proyectos de desarrollo, planes de producción de fábricas, estrategia militar y configuración de complejos sistemas informáticos, entre otros. Cuando se necesita controlar un

proceso tomando decisiones como respuesta a su estado y no existe una solución algorítmica adecuada, es necesario usar un SE. Este campo comprende el supervisar fábricas automatizadas, factorías químicas o centrales nucleares. Estos sistemas son extraordinariamente críticos porque normalmente tienen que trabajar a tiempo real (Quintanar, 2007).

El diseño requiere una enorme cantidad de conocimientos debido a que hay que tener en cuenta muchas especificaciones y restricciones. En este caso, el sistema experto ayuda al diseñador a completar el diseño de forma competente y dentro de los límites de costes y de tiempo. Se diseñan circuitos electrónicos, circuitos integrados, tarjetas de circuito impreso, estructuras arquitectónicas, coches, piezas mecánicas, etc. Por último, un sistema experto puede evaluar el nivel de conocimientos y comprensión de un estudiante, y ajustar el proceso de aprendizaje de acuerdo con sus necesidades (Ramos, Sistemas Expertos, 2000).

1.4 Los sistemas expertos en el diagnóstico de enfermedades

Las tecnologías de la información y comunicación auxilian en el desarrollo de equipamiento y sensores adecuados para recolectar datos en relación al manejo de enfermedades. En este sentido, existen diversos tipos de sistemas de pronóstico que se utilizan para monitorear condiciones propicias para la infección, en las principales enfermedades de numerosos cultivos. Así, diversas empresas, desarrollan en distintos países, productos informatizados para la fitopatología (Bombelli, 2011).

Con la rápida evolución de la tecnología, se han realizado avances notables para lograr mejorar la calidad de la salud de los seres humanos. Como era de esperar, en la salud también se han desarrollado SE. Existen SE que permiten detectar infecciones a partir de exámenes de sangre y cultivos de bacterias. Otros permiten hacer el diagnóstico médico y formulación de medicamentos, a partir de los síntomas que reporta el paciente. En el mercado se encuentran varios SE para el diagnóstico médico. Muchos médicos practicantes o estudiantes de los

últimos semestres de la carrera de medicina utilizan este tipo de aplicaciones, debido a que les permite tener una ayuda al momento de la consulta médica. La fiabilidad de estos sistemas es un factor importante, por lo que la decisión final beneficiará al paciente luego del diagnóstico, jamás se desea que el paciente salga perjudicado por un mal diagnóstico del SE (Alfonso, 2011).

En los últimos años, se ha desarrollado una serie de programas de computación denominados SE, relacionados con las enfermedades en las plantas. En estos programas, lo que hay que hacer es recolectar la información de campo y de laboratorio, introducirla a la computadora y con esa información, esta brindará un diagnóstico tentativo del agente causal. Lógicamente, estos programas de cómputo facilitarán en gran manera la labor del diagnóstico de los problemas fitopatológicos. (Durán Quirós , Mora Acedo, & Ramí, 2009).

1.5 Metodologías para los sistemas expertos (SE)

La creación de las bases de conocimiento es un complejo y largo período de adquisición de información, llevado a cabo por el ingeniero del conocimiento, quien es el encargado de codificar la información obtenida utilizando un lenguaje formal de representación del conocimiento partiendo de los juicios y opiniones de diferentes expertos. La culminación exitosa de este trabajo requiere un consenso en cuanto al criterio de cada uno de los expertos, eliminando cualquier discrepancia que pudiera haber surgido en el proceso de refinamiento con respecto a un tema determinado. Si esto se logra, la base de conocimiento resultante contará con una biblioteca de información mayor que cualquier caudal de información individual.

Para desarrollar un SE, debemos identificar y analizar el problema a partir de una serie de reglas y experiencias para después idear algún modo de adquirir y modelar el conocimiento y por último reducirlo a nivel simbólico y educar así a la máquina.

1.5.1 Metodología descrita por José Domingo Carrillo Verdun.

La metodología ideal desarrollada en la tesis de José Domingo Carrillo intenta una primera aproximación a la sistematización del desarrollo de SE, estructurándolo en una serie de fases, etapas y actividades que intentan facilitar la labor del ingeniero del conocimiento en la construcción del SE.

En esta metodología, se hace especial referencia a los problemas más importantes que debe resolver el ingeniero del conocimiento en cuanto a:

- Selección del dominio de Aplicación del SE y del experto en ese dominio
- Su relación (Jacobson, 2006) con el experto, y extracción del conocimiento del mismo, donde tienen especial importancia técnicas psicológicas tales como el análisis del protocolo o técnicas de entrevistas.
- Adquisición del conocimiento por el ingeniero y selección de la técnica de representación adecuada.
- Selección de la herramienta más adecuada para el desarrollo.
- Construcción de prototipos incrementales.

Se establecen unas pautas para la posible aplicación de esta Metodología al sector financiero, señalándose las principales posibles áreas de aplicación, así como la problemática que para este sector tienen los SE (Carrillo Verdun, 1987).

1.5.2 Metodología descrita por Marlene Carlos Soto.

Para el desarrollo de un SE se cuenta con un equipo de desarrollo en el cual las personas que participan son:

- El experto: pone su conocimiento a disposición del SE.
- El ingeniero de conocimientos: plantea las preguntas al experto, estructura sus conocimientos y los implementa en la base de conocimiento.

- El usuario: aporta sus deseos e ideas, determinando especialmente el escenario en el que debe aplicarse el SE.

En la fase de desarrollo el peso del trabajo recae en el experto y el ingeniero del conocimiento. En el desarrollo del SE estas tres personas trabajan muy unidas, ya que el experto y el ingeniero deben tener en cuenta los deseos y las ideas del usuario. Primeramente elaboran los problemas que deben ser resueltos por el sistema. Una vez delimitado el dominio hay que alimentar poco a poco al sistema con los conocimientos del experto, este último debe constantemente chequear que sus conocimientos hayan sido transmitidos de la forma más conveniente. El ingeniero del conocimiento es responsable de una implementación correcta pero no de la exactitud del conocimiento, esta exactitud recae en el experto antes mencionado. Además de esto no debe ignorarse nunca al usuario durante el desarrollo, para que al final se disponga de un sistema que le sea de máxima utilidad (Carlos Soto).

Los pasos a seguir en el desarrollo de un SE son:

Paso 1 Elección de la aplicación: En esta fase lo primero que debemos saber es si cumple con las siguientes condiciones:

- La información declarativa: es necesario considerar el conocimiento como modular y de este modo si ocurre algún error poder actualizarlos sin perderse en largas líneas de código.
- Las ventajas de la interfaz: como las inferencias hechas por los SE son similares por los mismos expertos humanos, el comportamiento del SE es amigable y los usuarios pueden mantener el sistema. Otra ventaja es que el conocimiento faltante en la base de conocimientos puede ser fácilmente obtenido de un modo natural.
- El SE debe ser capaz de explicar sus conclusiones: estos se debe a que se puede defender las conclusiones en términos de las oraciones de la base de datos que fueron usados para llegar a esa

conclusión. De esta forma no solo se puede verificar razonamientos incorrectos o la falta de alguno de ellos sino que también la máquina será capaz de enseñar a los humanos a partir de sus experiencias.

Paso 2 Elección de la herramienta apropiada: Para la elección de la herramienta de desarrollo del SE se han establecido algunos principios, ejemplo: la herramienta debía poseer solamente el grado de generalidad necesario para resolver el problema dado y probar el software de partida construyendo un pequeño prototipo antes de su realización.

La herramienta debe poseer las siguientes características:

- El lenguaje de realización deberá ser lo más simple y universal posible.
- Un medio de acceso a los mecanismos de control si la generalidad es más importante que la eficacia, o a la inversa, un sistema de control muy limitado si se busca un aprendizaje, una automodificación o explicaciones elaboradas.
- Capacidades de diálogos elaboradas (lenguaje casi natural, diccionario, etc) si el tiempo de desarrollo es un factor crítico.

Por último es necesario utilizar una herramienta que ya haya servido para una aplicación comparable.

Para la realización del SE es necesario respetar el formalismo de representar el conocimiento a usar, así como las técnicas de solución del problema adecuado a este formalismo y al problema que se está considerando, pero la determinación de estos se obtiene luego de haber efectuado la adquisición y el modelado de la información para posteriormente reducirlo a un nivel simbólico y así poderlo implementar como un Sistema Informático.

Paso 3 Transferencia de Experiencia: La transferencia de experiencia se efectúa del experto al SE, ayudado por el ingeniero del conocimiento.

La disciplina que interviene en la transferencia de experiencia se denomina ingeniería del conocimiento o cognimática.

El proceso se descompone en cuatro etapas:

1. Análisis del problema: Consiste en elaborar los problemas que deben ser resueltos por el sistema. Es importante tener en cuenta los deseos y sugerencias del usuario. El resultado de esta etapa es el proyecto de construcción del SE.
2. Adquisición del conocimiento y conceptualización: Consiste en identificar todos los elementos que intervienen en la solución del problema a través de estudios y seleccionar los conceptos básicos que harán posible el funcionamiento del sistema.
3. Formalización y representación del conocimiento: La actividad del ingeniero del conocimiento en esta fase se concreta en las siguientes tareas:
 - Elección de un formalismo de representación del conocimiento
 - Elección de una arquitectura que permita coordinar y manejar los distintos tipos de elementos que intervienen en la solución del problema.
 - Creación de la base de conocimientos utilizando el formalismo y la arquitectura elegida.
 - Diseño de la interfaz del SE con el usuario y con el resto del entorno lógico.
4. Validación: El desarrollo de un SE no se considera que está acabado una vez que funciona este sino que se continúa desarrollando y actualizando tanto el conocimiento del sistema como los métodos del procesamiento, quedando reflejados los progresos o modificaciones en el campo, área o sistema.

1.5.3 Metodología Descrita por Mateo Lezcano Brito.

Cuando se diseña una base de conocimientos, se está construyendo algo como una teoría sobre un determinado dominio o que al menos aspira a

llegar a serlo, en este sentido deben prevalecer en su diseño los aspectos conceptuales. La información debe estar organizada en forma armónica y debe ser, ante todo, lo más exacta posible. Sin embargo, el conocimiento de los expertos es a menudo incompleto e inconsistente, por lo que el proceso puede tornarse bastante engorroso y en algunos casos, el resultado pudiera no ser lo suficientemente exacto (Lezcano Brito, 1995).

Los métodos de construcción de SE parten, en primer lugar, de la identificación de los elementos del dominio y los conceptos descriptivos de sus propiedades, definiendo además las relaciones que se establecen entre ellos, en base a las cuales se organiza el conocimiento y posteriormente se pasa a su formalización. Desde el punto de vista cognoscitivo, la formalización del conocimiento va de la contemplación viva al pensamiento abstracto y de ahí nuevamente a la realidad. En este camino se parte del plano real (observación de la realidad), se pasa por el plano conceptual (creación del modelo para partes de esa realidad) y se llega al plano formal (elaboración de la teoría).

Existe un método clásico general, que permite la separación de este proceso en varias etapas. El primer paso a seguir cuando se habla de la construcción de un SE es el estudio del dominio, con vista a determinar si es realmente propicia la construcción del SE o no (Bello Pérez, 2000).

Este análisis se puede hacer en base a los siguientes criterios:

- Inexistencia de una solución algorítmica para el problema.
- Se trata de problemas relativamente estáticos, comparados con el tiempo requerido para analizarlos.
- Las tareas no son muy fáciles de resolver (se requieren años para formar un perito y además son muy escasos).
- La tarea debe tener interés práctico, produciendo altos beneficios.
- El conocimiento humano pudiera perderse.

Después de hecho este análisis, si se determina que es posible la construcción del SE, se pasa a las diferentes etapas como son:

Identificación del problema: En esta etapa se determina, básicamente, cuál es el problema que se quiere resolver y sus características, así como quienes van a participar en el levantamiento de la base de conocimiento y el papel de cada cual en ese proceso, entre otras cosas. Es más bien un periodo de familiarización, donde tiene lugar la relación inicial entre el experto y el ingeniero del conocimiento. En una entrevista inicial (y otras sucesivas si es necesario) el experto debe lograr que el ingeniero de conocimiento se lleve una idea general del dominio que se quiere modelar, haciendo una caracterización informal del problema y mostrando algunas descripciones de problemas típicos y los posibles pasos para su solución.

Establecer conceptos: Se definen los conceptos para la representación del conocimiento. El experto y el ingeniero del conocimiento determinan los aspectos claves del problema, las relaciones entre ellos y sus características con el objetivo de llevar a cabo la descripción del proceso de solución del problema.

Formalización: En esta etapa se formalizan los conceptos claves y subproblemas que estaban aislados durante la conceptualización. Se diseñan las estructuras para organizar el conocimiento. Después de un análisis intensivo, por parte del ingeniero de conocimiento, de los diferentes medios de representación con que se cuenta, se determina cuál se adapta mejor a las condiciones del problema, estableciendo un lenguaje formal que incorpore los conceptos formalizados del tema objeto de representación y describa a la vez el mecanismo de solución.

La definición de los conceptos y relaciones derivadas (conocimiento inducido) y la creación de reglas que los relacionan con el conocimiento explícito, marca el paso del plano conceptual al formal.

En esta etapa el ingeniero del conocimiento juega un papel más activo, determinando: la existencia de datos redundantes, si hay incertidumbre asociada a los datos, si los datos son consistentes y completos para resolver el problema y además si la interpretación lógica de los datos depende de su orden de ocurrencia en el tiempo.

Cumpliendo correctamente todas las etapas se formalizan los conceptos claves y subproblemas y se pasa a la implementación.

1.6 Tecnologías y Herramientas

En el proceso de selección de las herramientas a utilizar para la implementación del software que respalda esta investigación, se realizó un levantamiento del estado actual de las tecnologías que actualmente se usan en el desarrollo de sistemas similares. Haciendo énfasis en las conocidas como código abierto.

1.6.1 Prolog

Prolog es el resultado de muchos años de trabajo. La versión oficial de Prolog se desarrolló en la Universidad de Marsella, Francia por Alain Colmerauer y Robert Kowalski en los tempranos 1970, desarrollan un probador de teoremas especializado, escrito en FORTRAN, el cual utilizaron para implementar sistemas de procesamiento del lenguaje natural, como una herramienta para Programación en la Lógica (*Programming in Logic*), expresión que da origen a su nombre (Lezcano Brito, 1995).

Perspectiva del lenguaje

- El programa Prolog se compone de unos hechos (datos) y un conjunto de reglas, es decir, relaciones entre objetos de la base de datos.
- La ejecución del programa cargado en memoria consiste en realizar una pregunta de forma interactiva: el intérprete generará por inferencia los resultados que se deducen a partir del contenido de la base de datos.

- PROLOG tiene una sintaxis y semántica simples. Sólo busca relaciones entre los objetos creados, las variables y las listas, que son sus estructuras básicas.

El Prolog se ha ido en general de los laboratorios de las inteligencias artificiales. Es lo que se conoce como un idioma declaratorio. Esto significa que dado los hechos necesarios y reglas, Prolog usará el razonamiento deductivo para resolver sus problemas de la programación. Esto está en contraste con los lenguajes de la computadora tradicionales, como el C, Básic y Pascal que son los lenguajes procedurales (Contreras, 2004).

Entre sus beneficios podemos destacar:

1. Permite crear programas de la IA mucho más rápido y más fácil.
2. Es ideal para implementar SE y procesamiento del lenguaje natural.
3. Los mecanismos de inferencias y los procedimientos son partes de él (*built-in*).

Entre los objetivos de la creación de Prolog está la evolución de los lenguajes, como también el bajo costo y facilidad de programación que este puede ofrecer, ya que como su nombre lo indica PROLOG (*Programming in Logic*), es un programa que usa la lógica para resolver los problemas. Por otra parte, el lenguaje está orientado para trabajar en aplicaciones de inteligencia artificial (Montecino, 2003).

Prolog posee una máquina de inferencia, esta máquina puede usarse parcialmente para implementar algunos SE. Las reglas de Prolog son usadas para representar conocimiento y su máquina de inferencia es usada para derivar conclusiones. Otras partes del sistema, como la interfaz con el usuario deben escribirse usando Prolog.

1.6.2 SWI Prolog

SWI-Prolog trabaja en un entorno de Software Libre, licenciado bajo la *Lesser Gnu Public License* y posee un conjunto de herramientas de gráficos XPCE³, su desarrollo se inició en 1987 y ha sido impulsada por las necesidades de aplicaciones en el mundo real. Actualmente SWI-Prolog se utiliza en la investigación y la educación, así como para aplicaciones comerciales (Martín, 2010).

SWI-Prolog es una herramienta basada en una máquina virtual muy sencilla de Prolog llamada ZIP que define solamente 7 instrucciones. Además de simplificar el diseño del propio sistema Prolog, este enfoque tiene ventajas para el desarrollo del programa: el compilador es simple y rápido, el usuario no tiene que decidir de antemano si la depuración es necesaria y el sistema sólo se ejecuta ligeramente a una velocidad más baja cuando está en modo de depuración (Wielemaker, 2012).

1.6.3 Java

Java surgió en 1991 cuando un grupo de ingenieros de Sun Microsystems trataron de diseñar un nuevo lenguaje de programación destinada a electrodomésticos que llamaron Oak. La reducida potencia de cálculo y memoria de los electrodomésticos llevó a desarrollar un lenguaje sencillo capaz de generar código de tamaño muy reducido.

Como lenguaje de programación para computadoras, Java se introdujo a finales de 1995, fecha en la que se da a conocer al público, alcanzando notoriedad rápidamente, y Sun Microsystems logró su propósito de crear un lenguaje totalmente independiente de plataforma y a su vez potente y orientado a objetos. La clave fue la incorporación de un intérprete Java en la versión 2.0 del programa Netscape Navigator, produciendo una verdadera revolución en Internet. Java 1.1 apareció a principios de 1997, mejorando sustancialmente esta primera versión de Java 1.2, más tarde rebautizado como Java 2, nació a finales de 1998. Esa filosofía y su

³ Biblioteca de interfaz gráfica de usuario nativa SWI-Prolog

facilidad para crear aplicaciones para redes TCP/IP ha hecho que sea uno de los lenguajes más utilizados en la actualidad (Sánchez, 2003).

Los programas en Java cuando se compilan no generan código binario como los programas en C o C++. En su lugar al compilar se genera un código intermedio (que no es directamente ejecutable), conocido como *bytecode* o *J-code*. Ese código, generado normalmente en archivos con extensión class es el que es ejecutado por la máquina virtual de Java (JVM) que interpreta las instrucciones generando el código ejecutable de la aplicación.

La máquina virtual de Java es un programa pequeño que se distribuye gratuitamente para la mayoría de los sistemas operativos (Colectivo de autores, Aprende Java como si estuvieras en primero, 2000). Esto es lo que permite que los programas creados en Java sean altamente portables, ya que solo se necesita tener la JVM para un sistema con la que podemos ejecutar cualquier programa escrito en java pues no contiene código binario específico para una plataforma.

1.6.4 NetBeans

El IDE NetBeans es un producto libre y gratuito sin restricciones de uso y de amplias comodidades para el programador, de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, fue fundada en junio de 2000 por Sun Microsystems. La Plataforma NetBeans es una base modular y extensible usada como una estructura de integración para crear aplicaciones de escritorio grandes (Carpio, 2010).

Para la realización de este proyecto se utilizó el IDE 6.9 de NetBeans, una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación.

1.7 Conclusiones Parciales

En este capítulo se determinó el marco teórico metodológico que permite desarrollar una herramienta informática para diagnosticar enfermedades fúngicas en los cultivos. Luego de este análisis se llega a la conclusión de que es factible la realización de un SE para diagnosticar las enfermedades fúngicas en los cultivos, utilizando la metodología del DrC. Mateo Lezcano. Además se decidió la utilización del Prolog para confeccionar la base de conocimientos y el lenguaje Java para crear la interfaz visual, utilizando los IDE SWI-Prolog y Netbeans.

Capítulo 2: Análisis y diseño del SEDEFP

En este capítulo se detallan las etapas seguidas para la construcción del SE que están descritas en la metodología del DrC. Mateo Lezcano. También se muestra el diagrama de casos de uso del sistema y describen los casos de uso del SEDEFP.

2.1 Identificación

Los métodos de construcción de un SE parten, en primer lugar, de la identificación de los elementos del dominio y los conceptos descriptivos de sus propiedades, definiendo además las relaciones que se establecen entre ellos, en base a las cuales se organiza el conocimiento (Martín, 2010). Este es más bien un período de familiarización, donde tiene lugar la relación inicial entre los experto y el ingeniero del conocimiento. Es un encuentro donde los experto hacen que el ingeniero de conocimiento se lleve una idea general del dominio que se quiere modelar, haciendo una caracterización informal del problema y mostrando algunas descripciones de problemas típicos y los posibles pasos para su solución.

En esta etapa se determinó, básicamente, que el problema que se quiere resolver está basado en como contribuir con el diagnóstico de enfermedades fúngicas llevado a cabo en La Empresa de Sanidad Vegetal, así como quienes van a participar en el levantamiento de la base de conocimiento y el papel de cada cual en ese proceso.

Los participantes en el levantamiento de la base de conocimiento fueron:

- Ing. Yamile especialista en enfermedades fúngicas en La Empresa de Sanidad vegetal.
- Lic. Luis Quintero Domínguez
- DrC. Miguel Salvat Quesada profesor de agronomía de la UNISS

- MSc. Gorge Félix Melendrez Rodríguez profesor de agronomía de la UNISS

De forma conjunta, se determinaron los elementos del dominio con sus elementos descriptivos, los que se relacionan seguidamente:

Elementos del dominio:

- Plantas.
 - **Conceptos descriptivos:** hojas, tallos, raíz, frutos
- Enfermedades.
 - **Conceptos descriptivos:** cambios de color, necrosis, desarrollo anormal, marchitamiento

El **objetivo relevante** de la investigación es suministrar una herramienta para el diagnosticar enfermedades fúngicas en las plantas de interés agrícola.

Los recursos con que se cuenta para la obtención de información son los siguientes:

- Expertos en agronomía de la UNISS y de la Empresa de Sanidad Vegetal.
- Libros
 - Prolog y los Sistemas Expertos por el Dr. Mateo Lezcano Brito de la Universidad Central “Martha Abreu” de las Villas, 1995.
 - Manejo Integrado de Plagas por un colectivo de autores en el Centro Nacional de Sanidad Vegetal (CNSV), Cuba, 2006.
- Tesis:
 - Sistema Experto para la clasificación temprana de Dengue Severo en el Instituto Pedro Kourí (IPK) de la Ciudad de la Habana.

- Sistema Experto para contribuir a desarrollar habilidades para Aprender a Aprender (SEPAA).
- Sistema Experto para contribuir a un mejor aprendizaje de la Ortografía (SEPAO).
- Sistema Experto para el Diagnóstico Fitosanitario del Espárrago usando Redes Bayesianas por Pedro Nelson Shiguihara Juárez y Jorge Carlos Valverde Rebaza.

El alcance que vaya a tener, en un inicio, el problema y las diferentes submetas trazadas son:

Alcance:

- El diagnóstico de los cultivos afectados que tienen mayor importancia económica para la provincia de Sancti Spíritus.

Submetas trazadas:

- Estudiar toda la bibliografía disponible.
- Confeccionar la base de conocimiento.
- Confeccionar la interfaz visual.

2.2 Establecer Conceptos

En este período se definieron los conceptos para la representación del conocimiento, se determinaron los aspectos claves del problema, las relaciones entre ellos y sus características con el objetivo de llevar a cabo la descripción del proceso de solución del problema.

Se establecen una serie de elementos esenciales, tales como:

- Los tipos de datos disponibles son exactos e incompletos (hasta el momento).
- No se trabaja con factor de certidumbre.
- Los datos de salida son los nombres de los hongos que se identifican.

El conocimiento se organizó atendiendo a los diferentes elementos de dominio planteados con anterioridad y los conceptos descriptivos de estos, y en base al elemento del dominio que el usuario desee consultar es que se realiza el proceso de razonamiento.

2.3 Formalización

En esta etapa se diseñaron las estructuras básicas para organizar el conocimiento. Se formalizaron los conceptos claves y los subproblemas que estaban aislados durante la conceptualización. Después de la realización de un análisis intensivo, se determinó usar las reglas de producción en lenguaje Prolog, por adaptarse mejor a las condiciones del problema.

2.3.1 Clasificación del conocimiento

La organización del conocimiento quedó dividida en once módulos con el objetivo de organizar el conocimiento desde el punto de vista lógico.

Estos son:

Módulo 1: Buscar conceptos

Módulo 2: Enferm ajo y cebolla

Módulo 3: Enferm arroz

Módulo 4: Enferm plátano

Módulo 5: Enferm cacao

Módulo 6: Enferm cafeto

Módulo 7: Enferm garbanzo

Módulo 8: Enferm maíz

Módulo 9: Enferm tabaco

Módulo10: Enferm frijol

Módulo11: Enferm berenjena

El predicado principal para los 10 módulos correspondientes a las enfermedades de los cultivos es: **consulta**.

El predicado principal para el módulo de Buscar Conceptos es: **conceptos**.

2.4 Implementación

El ingeniero de conocimiento combina y reorganiza el conocimiento formalizado para hacerlo compatible con las características del flujo de información del problema.

Se lleva a cabo la formalización de las reglas que abarcan todo el conocimiento, obteniéndose un primer prototipo, que muestra sobre un esquema elemental la forma en que opera el sistema. El objetivo fundamental de este primer prototipo es obtener una solución inmediata del problema. La realización de pruebas al primer prototipo y el análisis de las críticas que se hicieron a su funcionamiento sirvieron como punto de partida para la formalización de una versión más avanzada de la base de conocimiento.

2.4.1 Formalización de los casos

Los casos que a continuación son descritos fueron implementados en SWI-Prolog, en una versión inicial, luego se hizo una segunda versión con la intención de lograr una interfaz visual más amigable mediante la interacción con NetBeans.

En la segunda versión de la base de conocimiento los módulos correspondientes a las enfermedades de las plantas quedaron divididos en dos partes, un bloque de preguntas y un bloque de respuestas, tal como se muestra en el ejemplo siguiente:

Enferm frijol

/***PREGUNTAS*****/**

pregunta1('¿Tiene manchas pequeñas? ').

pregunta2('¿Las manchas se tornan en un color blanquecino a amarillo? ').

pregunta3('¿Las uredósporas del hongo tienen una coloración pardo-rojiza? ').

pregunta4('¿Tiene manchas blancas?').

pregunta5('¿El aspecto de las manchas es polvoriento? ').

pregunta6('¿Aparecen primero en hojas más viejas y después en vainas y tallos? ').

pregunta7('¿Debajo de las hojas se necrosa el tejido? ').

pregunta8('¿Se observan chancros negros debajo del nudo de los cotiledones? ').

pregunta9('¿Presenta lesiones en forma de anillos concéntricos que debilitan el tallo? ').

pregunta10('¿Se observan manchas verde claro en la base del tallo? ').

pregunta11('¿En el tallo y raíces se observa un micelio blanco afieltrado? ').

pregunta12('¿Se observan lesiones chancrosas de color pardo claro a oscuras? ').

pregunta13('¿La lesiones son secas y duras en la base de los tallos? ').

pregunta14('¿Se observa pudrición seca de la corona? ').

pregunta15('¿Se observa pudrición en la base del tallo? ').

pregunta16('¿Los tejidos del tallo se tornan rojizos y se oscurecen hasta necrosar? ').

pregunta17('¿El follaje muestra una clorosis lenta y marchitez? ').

Enferm frijol

/***RESPUESTAS*****/**

resp_roya_frijol('El cultivo es afectado por Uromyces appendiculatus, conocida comúnmente como Roya del frijol.').

resp_mildiu_polvoriento('El cultivo es afectado por Botrytis aclada, conocida comúnmente como Mildio Polvoriento.').

resp_tizon_ceniciento('El cultivo es afectado por Macrophomina phaseolina, conocida comúnmente como Tizón ceniciento.').

resp_tizon_sureño('El cultivo es afectado por Corticium rolfsii Curzi, conocida comúnmente como Tizón sureño.').

resp_rhizoctoniosis('El cultivo es afectado por Phyrenochaeta, conocida comúnmente como Rhizoctoniosis.').

La base de conocimiento de buscar conceptos quedó formalizada de la siguiente forma:

/***CONCEPTOS BASICOS*****/**

concepto1('humus','Sustancia compuesta por ciertos productos orgánicos de naturaleza coloidal, que proviene de la descomposición de los restos orgánicos por organismos y microorganismos benéficos (hongos y bacterias). Se caracteriza por su color negruzco debido a la gran cantidad de carbono que contiene. Se encuentra principalmente en las partes altas de los suelos con actividad orgánica.').

concepto1('fotoperiodo','Conjunto de procesos de las especies vegetales mediante los cuales regulan sus funciones biológicas (como por ejemplo su reproducción y crecimiento) usando como parámetros la alternancia de los días y las noches del año y su duración según las estaciones y el ciclo solar.').

concepto1('fotosíntesis','Es la conversión de materia inorgánica en materia orgánica gracias a la energía que aporta la luz. En este proceso la energía luminosa se transforma en energía química

estable, siendo el adenosín trifosfato (ATP) la primera molécula en la que queda almacenada esa energía química.').

concepto1('invernadero','Lugar cerrado, estático y accesible a pie, que se destina a la producción de cultivos, dotado habitualmente de una cubierta exterior translúcida de vidrio o plástico, que permite el control de la temperatura, la humedad y otros factores ambientales para favorecer el desarrollo de las plantas.').

concepto1('fitorregulador','Producto regulador del crecimiento de las plantas; normalmente se trata de hormonas vegetales (fitohormonas), y sus principales funciones son estimular o paralizar el desarrollo de las raíces y de las partes aéreas.').

concepto1('posología','Es la rama de la farmacología que estudia la dosificación de los fármacos').

concepto1('perlita','Es un vidrio volcánico amorfo que tiene un contenido de agua relativamente alto. Es un mineral que aparece en la naturaleza, y tiene la propiedad poco común de expandirse muchísimo cuando se la calienta lo suficiente.').

concepto1('limbo','Parte plana y visible de la hoja; generalmente ancha, recorrida por los nervios y sostenida por el pecíolo.').

concepto1('nervadura','Es la distribución de los nervios que componen el tejido vascular de la hoja de una planta. Se ubican en el estrato esponjoso del mesófilo de la hoja; a través de ellos circula la savia, comunicando los órganos de la hoja con el resto de la planta.').

concepto1('ápice ', 'Expresa el extremo superior o punta de la hoja. En un órgano, por ejemplo una hoja, hay que distinguir entre el ápice orgánico, por donde puede crecer distalmente el órgano, dotado de tejido meristemático, y el ápice geométrico, que es simplemente el punto más distanciado de la base.').

concepto1('foliolo', 'Cada una de las piezas separadas en que a veces se encuentra dividido el limbo de una hoja. Cuando el limbo foliar está formado por un solo foliolo, es decir no está dividido, se dice que la hoja es una hoja simple. Cuando el limbo foliar está dividido en foliolos se dice que la hoja es hoja compuesta.').

concepto1('pecíolo', 'Es el rabillo que une la lámina de una hoja a su base foliar o al tallo. Falta en las hojas sésiles.').

concepto1('haz', 'Cara superior o cara adaxial del limbo de la hoja de una planta. Se diferencia del envés en que no posee estomas, tiene una cutícula algo más gruesa y posee menor abundancia de tricomas. Su color suele ser más oscuro que el del envés (como máximo pueden llegar a ser del mismo color).').

concepto1('envés', 'Cara inferior o cara abaxial de la lámina o limbo de la hoja de una planta. La epidermis del envés suele ser diferente de la del haz o cara superior; casi siempre presenta una cutícula más fina, mayor densidad de estomas y, frecuentemente también, mayor abundancia de tricomas (pelos epidérmicos). Casi siempre es de color más claro que el haz.').

concepto1('tricomas', 'Son excrecencias de origen epidérmico y de formas muy variables, pudiendo ser glandulares o no. Pueden hallarse

vivos o muertos a su madurez y tienen caracteres suficientemente constantes en distintas especies como para llegar a tener mucho valor en la identificación de plantas.').

concepto1('tricomas glandulares ', 'son aquellos que presentan células secretoras. La cutícula es delgada, reviste a todas las células y se puede separar de la pared ya que presenta por debajo una alta concentración en pectina, haciendo que sea la trama de la pared más débil.').

2.4.2 Diseño de los módulos del sistema

Cada módulo posee una base de conocimiento, la cual está dividida en dos partes, un bloque de preguntas y un bloque de respuestas; además cuenta con un motor de inferencia y una interfaz del usuario. En la figura 2.1 se presenta el diseño.

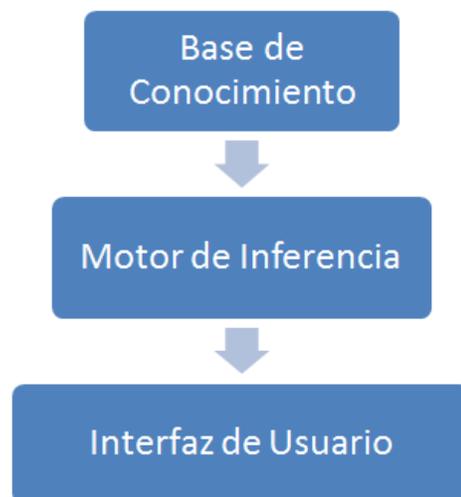


Ilustración 1: Diseño del módulo.

La creación de las bases de conocimiento es un complejo y largo período de adquisición de información, llevado a cabo por el ingeniero del conocimiento; quién, partiendo de los juicios y opiniones de diferentes

expertos, debe codificar la información obtenida utilizando un lenguaje formal de representación del conocimiento.

Al diseñar una base de conocimiento, se está construyendo algo como una teoría sobre un determinado dominio o que al menos aspira a llegar a serlo, en este sentido deben prevalecer en su diseño los aspectos conceptuales. La información debe estar organizada en forma armónica y debe ser, ante todo, lo más exacta posible. Sin embargo, el conocimiento de los expertos es a menudo incompleto e inconsistente, por lo que el proceso puede tornarse bastante engorroso y en algunos casos, el resultado pudiera no ser lo suficientemente exacto (Lezcano Brito, 1995).

2.4.3 Diálogos del sistema

Cada módulo comienza con una pregunta del sistema para que el usuario responda, en dependencia de la respuesta SEDEFP sigue un camino u otro en el árbol de soluciones posibles. El diálogo continúa hasta que el sistema sea capaz de identificar un hongo o indicando que no pudo identificar la enfermedad.

Diálogo del módulo Enferm frijol:

A: ¿Tiene manchas pequeñas?

Respuesta del usuario.

A: ¿Las manchas se tornan en un color blanquecino a amarillo?

Respuesta del usuario.

A: ¿Las uredósporas del hongo tienen una coloración pardo-rojiza?

Respuesta del usuario.

A: ¿Tiene manchas blancas?

Respuesta del usuario.

A: ¿El aspecto de las manchas es polvoriento?

Respuesta del usuario.

A: ¿Aparecen primero en hojas más viejas y después en vainas y tallos?

Respuesta del usuario.

A: ¿Debajo de las hojas se necrosa el tejido?

Respuesta del usuario.

A: ¿Se observan chancros negros debajo del nudo de los cotiledones?

Respuesta del usuario.

A: ¿Presenta lesiones en forma de anillos concéntricos que debilitan el tallo?

Respuesta del usuario.

A: ¿Se observan manchas verde claro en la base del tallo?

Respuesta del usuario.

A: ¿En el tallo y raíces se observa un micelio blanco afieltrado?

Respuesta del usuario.

A: ¿Se observan lesiones chancrosas de color pardo claro a oscuras?

Respuesta del usuario.

A: ¿Las lesiones son secas y duras en la base de los tallos?

Respuesta del usuario.

A: ¿Se observa pudrición seca de la corona?

Respuesta del usuario.

A: ¿Se observa pudrición en la base del tallo?

Respuesta del usuario.

A: ¿Los tejidos del tallo se tornan rojizos y se oscurecen hasta necrosar?

Respuesta del usuario.

A: ¿El follaje muestra una clorosis lenta y marchitez?

Respuesta del usuario.

B: Respuesta del sistema

Seguido de la respuesta que da el sistema se muestra el tratamiento que se debe de seguir para combatir la enfermedad.

2.4.4 Diagrama de casos de uso del sistema

El modelo de casos de uso permitió al desarrollador del sistema y a los clientes que llegaron a un acuerdo sobre las condiciones y posibilidades que debe tener el sistema. Este diagrama se muestra en la figura.

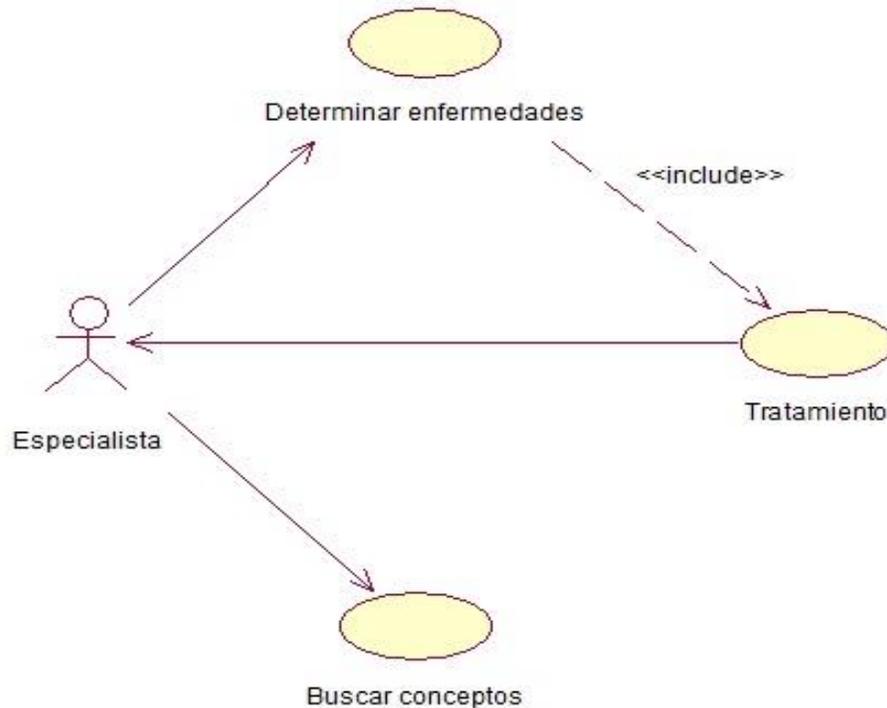


Ilustración 2: Diagrama de Casos de Uso del Sistema.

2.4.5 Descripción de los Casos de uso del Sistema

Caso de uso: Mostrar ayuda.
Actores: Especialista.
Propósito: Solicitar al sistema la ayuda para saber cómo trabajar con el software.
Resumen: El caso de uso se inicia cuando el especialista, desea consultar la ayuda del sistema para obtener ayuda acerca de las opciones a la que tiene acceso. El caso de uso finaliza cuando el sistema le muestra la ayuda que solicita el especialista.

Pre-condiciones: La ayuda debe estar creada en el sistema.

Pos-condiciones: Que el sistema muestre la ayuda.

Tabla 1: Descripción del caso de uso del sistema Mostrar ayuda.

Caso de uso: Buscar conceptos.

Actores: Especialista.

Propósito: Solicitar ayuda al sistema para conocer los conceptos de algunas palabras claves.

Resumen:

El caso de uso se inicia cuando el especialista, desea consultar el sistema para conocer los conceptos relacionados con algunas palabras que son complejas. El sistema muestra la opción para que el especialista escriba la palabra que desea buscar. El caso de uso finaliza cuando el sistema le muestra la información sobre el concepto que se deseaba conocer.

Pre-condiciones: Debe existir en la base de conocimiento el conocimiento sobre los conceptos.

Pos-condiciones: Que el sistema muestre los conceptos.

Tabla 2: Descripción del caso de uso del sistema Buscar conceptos.

Caso de uso: Determinar Enfermedades.

Actores: Especialista.

Propósito: Solicitar ayuda al sistema para determinar una enfermedad.

Resumen:

El caso de uso se inicia cuando el especialista, desea consultar el sistema para determinar una enfermedad en una determinada

<p>planta. El sistema muestra una serie de preguntas que el especialista debe responder para saber si el cultivo es afectado por un virus, una bacteria o un hongo. Si la enfermedad es provocada por un hongo el sistema da la opción de que el especialista seleccione el cultivo que desee diagnosticar, continuando con las demás preguntas. El caso de uso finaliza cuando el sistema llega a la identificación del hongo en caso de que exista.</p>
<p>Pre-condiciones: Debe existir en la base de conocimiento las preguntas y respuestas para identificar la enfermedad.</p>
<p>Pos-condiciones: Que el sistema muestre las preguntas.</p>

Tabla 3: Descripción del caso de uso del sistema Determinar Enfermedades.

2.5 Conclusiones Parciales

En este capítulo se identificaron los elementos del dominio y los conceptos descriptivos de sus propiedades; y las relaciones entre ellos. Se formalizaron e implementaron las reglas y se definieron y describieron los casos de usos del sistema.

Capítulo 3

Capítulo 3: Implementación y descripción del SEDEFP

En este capítulo se explica la implementación de la interfaz de usuario y los requerimientos del hardware y software para lograr un rendimiento óptimo del SE. Además se detallan las facilidades desde el punto de vista del usuario que brinda el mismo, terminando este capítulo con un manual de usuario para la correcta utilización del software.

3.1 Implementación de la interfaz del usuario

Después de la formalización de la base de conocimiento, mediante las reglas de producción y la definición de los requerimientos del sistema se pasó a la implementación de la interfaz del usuario. SWI-Prolog posee una interfaz simple para generar aplicaciones de consola, pero esto no es suficiente cuando necesitamos utilizar Prolog para aplicaciones profesionales que requieren una interfaz más amigable y accesible para todo tipo de usuarios, para solucionar este problema algunas implementaciones incluyen bibliotecas que permiten realizar la interfaz de Prolog con superlenguajes como Java.

Para desarrollar la interface visual se utilizó NetBeans 6.9 por ser un entorno de desarrollo rápido de aplicaciones y poseer facilidades para interactuar con SWI-Prolog. En SWI- Prolog se genera un fichero .pl, pero este no se puede utilizar con esta extensión para consultar la base de conocimiento mediante el lenguaje java, para esta acción se necesita que el fichero posea la extensión .pro, que se obtiene con solo cambiar en el nombre la extensión.

El uso de SWI-Prolog con Java se facilita por una componente (la clase Query) que implementa las llamadas a la biblioteca, así como el acceso a otros tipos de datos necesarios. La biblioteca jpl.jar disponible en SWI-

Prolog brinda una gran variedad de procedimientos y funciones para la manipulación de datos, predicados y términos de Prolog.

Para lograr una mayor eficiencia en el proceso de trabajo, y sobre todo para lograr una coherencia formal entre todos los módulos del sistema, se han pautado una serie de elementos comunes que facilitan su uso y reconocimiento.

- ✓ Se diseñó una pantalla inicial global del SE, desde la cual se accederá a los diferentes módulos de este.
- ✓ El diseño está determinado fundamentalmente por el principio de la usabilidad, donde este tiene como principal propósito facilitar su uso y comprensión manteniendo pautas estéticas y agradables.
- ✓ El logo siempre estará ubicado en la parte superior de cada módulo. La tipografía será siempre Arial, por su amplia legibilidad y por las facilidades conocidas que brinda para la lectura digital.
- ✓ El diseño de la interfaz es muy simple de comprender por el usuario.

3.1.1 Los JComboBox

Se utilizan ComboBox no editable que desactiva el tecleo y permite al usuario seleccionar un valor desde una lista, para evitar errores introducidos por el mismo.

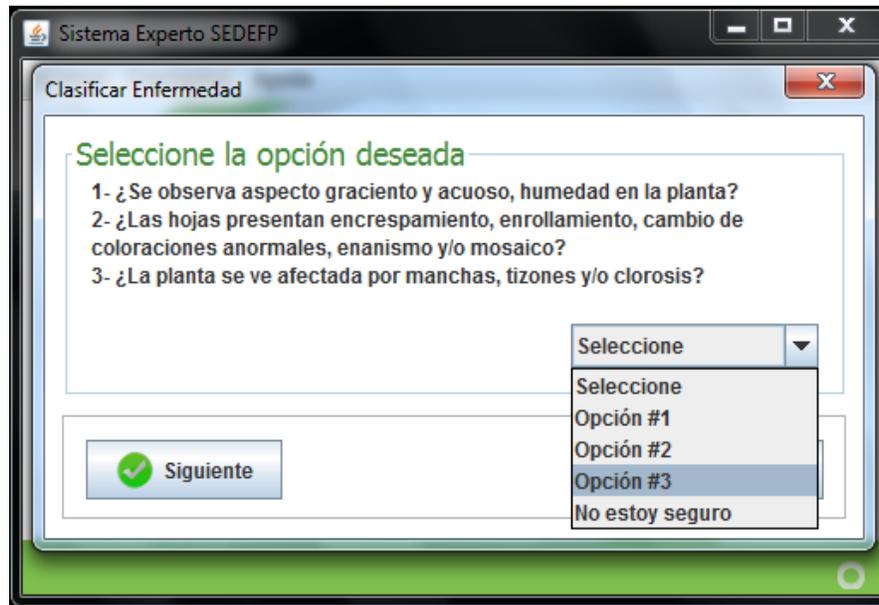


Figura 3: Uso de JComboBox en SEDEFP.

3.1.2 Los JList.

Los JList se utilizan para mostrar textos extraídos de la base de conocimiento.

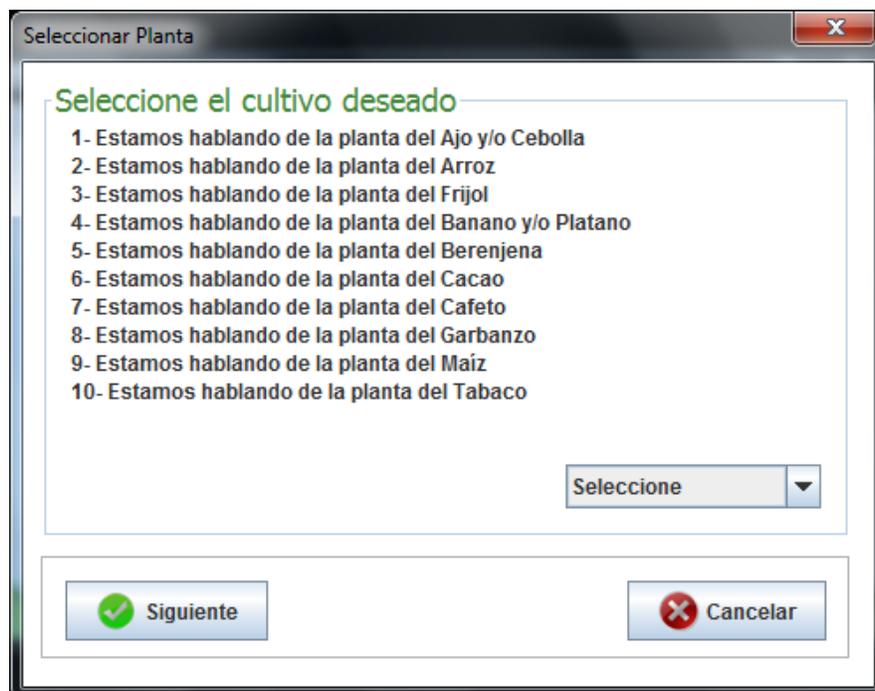


Figura 4: Uso de JList en SEDEFP

3.1.3 El uso de Iconos, botones y etiquetas

Algunos componentes Swing, como JLabel y JButton, pueden ser decorados con un icono, una imagen de tamaño fijo. En Swing, un icono es un objeto que se adhiere al interface Icon.

Un objeto ImageIcon puede cargar una imagen desde un nombre de fichero, una URL u otra fuente. Todas las imágenes que fueron utilizadas se cargaron mediante una URL, resulta muy cómoda y eficiente esta alternativa.

La clase JButton descende de AbstractButton define un pequeño API público adicional, la implementación del manejo de eventos depende del tipo de botón escogido y de cómo se utiliza. Generalmente, se implementa un ActionListener, que es notificado cada vez que el usuario pulsa el botón, en esta aplicación se siguió la misma política para el manejo de eventos con los botones.

El botón y la etiqueta son componentes atómicos, su papel no es contener otros componentes Swing, sino que actúan como entidades auto-suficientes que representan bits de información para el usuario. Frecuentemente, los componentes atómicos también obtienen entrada del usuario, tal es el caso de las cajas de selección (JComboBox), campos de texto (JTextField), y tablas (JTable). SEDEF no contiene tablas pero si utiliza a las cajas de selección en varias de las opciones que propone.

Este es un ejemplo de los botones que utiliza SEDEF:

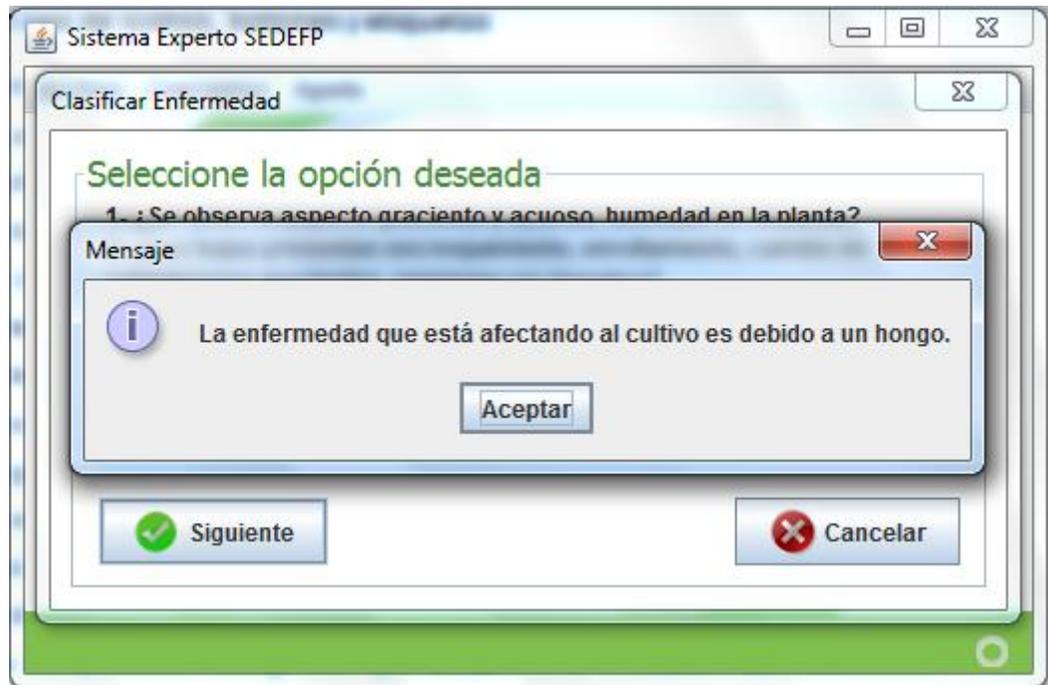


Figura 5: Uso de botones en SEDEFP

3.1.4 Otros Contenedores

JPanel es sólo una de las varias clases de contenedores que se pueden utilizar. Existen otros contenedores de propósito especial.

Los paneles o JPanel fueron de gran utilidad, pues juegan un papel más visible e interactivo en el GUI de un programa, siendo un eslabón importante en el diseño del software. Los **JTextField** fueron utilizados para que el especialista introduzca la palabra de la cual desea saber su concepto.

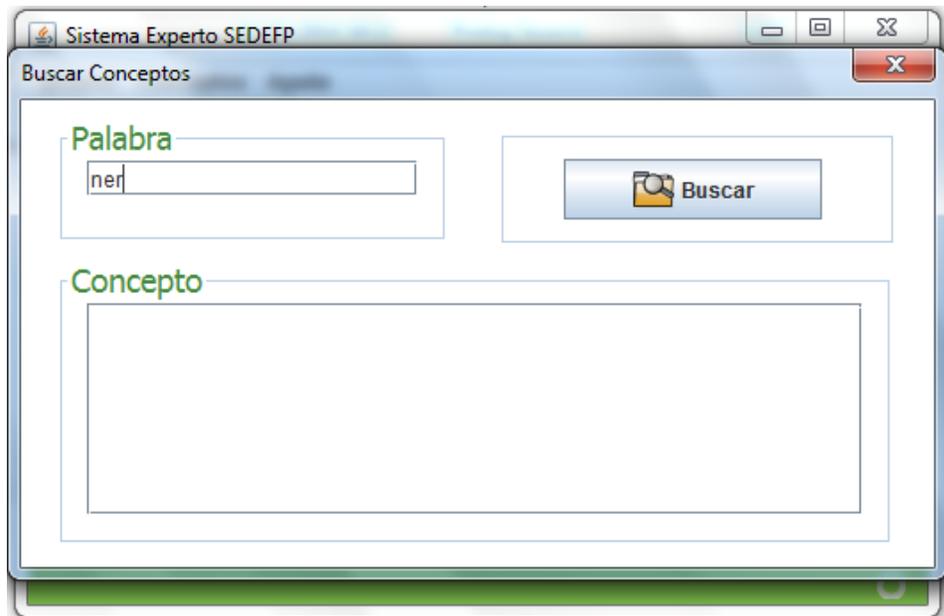


Figura 6: Uso de JTextField en SEDEFP

3.1.5 Barra de menú para un JFrame

Los menús desplegables tienen detalles interesantes de implementación. Uno es que cada menú tiene un menú desplegable apropiado. Cuando el menú se activa, utiliza su menú desplegable para mostrar sus ítems de menú. Otro detalle es que un propio menú desplegable utiliza otro componente para implementar la ventana que contiene los ítems del menú. En dependencia de las circunstancias bajo las que se muestre el menú desplegable, se implementaron "ventanas" utilizando un componente de peso pesado (Window).

Para añadir un JMenu a un JMenuBar, se utiliza el método add (JMenu), mientras que para adicionar ítems de menú y submenús a un JMenu, se utiliza el método add (JMenuItem).

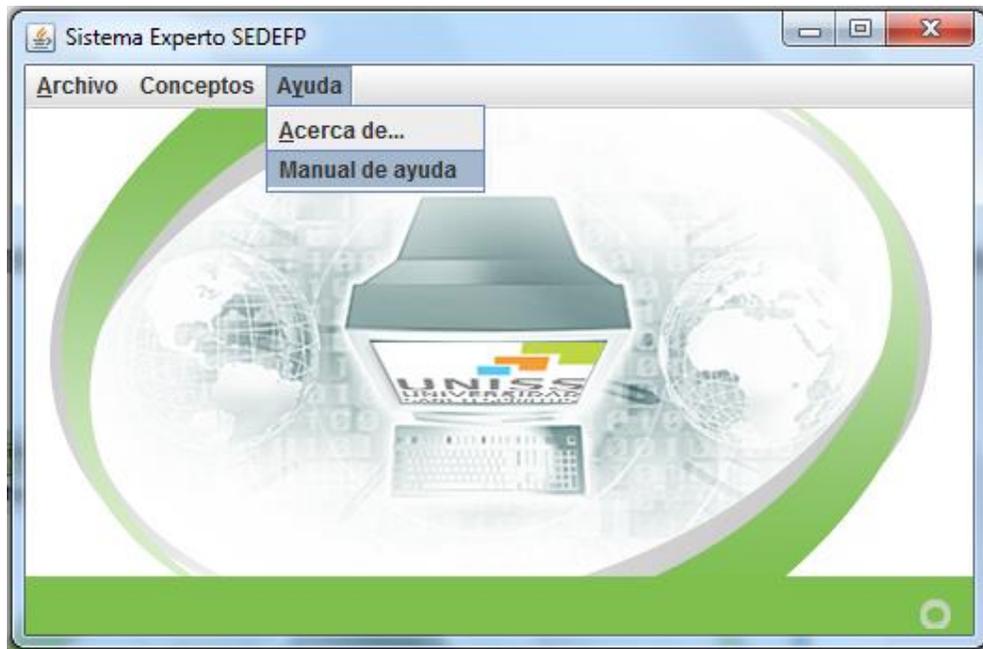


Figura 7: Barra de menú en SEDEFP

3.1.6 Evento actions

El Evento action es el más utilizado en los SE. Los más sencillos y comunes manejadores de eventos. Se implementa un oyente de action para responder a una indicación del usuario de que alguna acción dependiente de la implementación debería ocurrir. Envía un mensaje **actionPerformed** a través del método **void actionPerformed (ActionEvent)** que se le llama justo después de que el usuario informe al componente escuchado de que debería ocurrir una acción.

3.2 Aspectos a tener en cuenta para la interacción de SWI-Prolog y java

Para la interacción de SWI-Prolog y java se necesitan los siguientes programas:

- SWI Prolog 5.4.7 o superior
- Netbeans IDE 6.9 o superior
- J2SE-JDK 6 o superior

Para hacer la interfaz se necesita una Base de Conocimiento, o sea un archivo fuente Prolog.

Aspectos fundamentales a seguir:

1. Crear un proyecto tipo Java con el Netbeans IDE
2. Copiar la Base de Conocimiento (nombre.pl) para el directorio raíz del proyecto
3. Incorporar al proyecto la biblioteca jpl.jar ubicada en: C:\Archivos de programa\p\lib\jpl.jar

El éxito de la interfaz entre SWI Prolog y Java en gran medida depende de la configuración actual de su sistema operativo. Por tanto es preciso agregar a la variable de entorno path seguida de los datos que trae por defecto un punto y coma y la siguiente línea:

- C:\Archivos de programa\Intel\DMIX;C:\Archivos de programa\Java\jdk1.6.0\bin;C:\Archivos de programa\Java\jdk1.6.0\lib\tools.jar;C:\Archivos de programa\Java\jdk1.6.0\jre\lib\rt.jar;C:\Archivos de programa\p\bin;C:\Archivos de programa\p\lib\jpl.jar;

En principio las condiciones están creadas para comenzar a inferir sobre el conocimiento, sin embargo es recomendable acatar las siguientes sugerencias:

1. Cerrar todas las aplicaciones, especialmente el NetBeans si lo tuviera abierto.
2. Limpiar el registro con alguna herramienta aparente como el CClenear o TuneUp Utilities.
3. Reiniciar la máquina.

De esta manera queda enlazado el archivo fuente nombre.pl con el proyecto Java. Para interrogar a la base de conocimiento se utiliza una instancia de la clase Query:

- `Query query = new Query(String pathFile)`: Se llama una sola vez al inicializar la aplicación para cargar en memoria la Base de Conocimiento.
- `Query query = new Query(String consult)`: Se llama cada vez que se desea hacer una nueva consulta Prolog.

La clase `Query` tiene muchas funciones o métodos que permiten extraer inferencias del conocimiento representado en la base de conocimiento Prolog. Sin embargo con solo cinco funciones es posible implementar la interfaz de Prolog con Java de una manera flexible y cómoda. Estas funciones se describen a continuación:

- **`boolean hasSolution()`**: Método que permite saber si una consulta Prolog puede ser resuelta con el conocimiento disponible.
- **`Hashtable oneSolution()`**: Este método devuelve un objeto `Hashtable` con la información referente a una solución de la consulta Prolog.
- **`boolean hasMoreSolutions()`**: Método que permite saber si la consulta Prolog tiene más soluciones.
- **`Hashtable nextSolutions()`**: Si la consulta Prolog tiene más de una solución, este método permitirá encontrar la siguiente solución.
- **`Hashtable [] allSolutions()`**: Retorna un arreglo de objetos tipo `Hashtable` con la información referente a todas las soluciones existentes.

Estas funciones utilizan internamente predicados de segundo orden, por lo que es conveniente interrogar al conocimiento Prolog utilizando sentencias simples con el objetivo de alcanzar una implementación lo más óptimo posible.

3.3 Requerimientos del Sistema

Para que el software funcione correctamente se necesitan un mínimo de requerimientos técnicos tanto de hardware como de software, siendo un software desarrollado en Java que ofrece la ventaja de ser

multiplataforma, pero el instalador de SWI-Prolog está hecho para ser utilizado en Windows.

Requerimientos de hardware:

- Al menos 256 MB de memoria RAM
- La instalación básica necesita de 132 MB de espacio disponible en disco más 150 MB para la instalación del Java RunTime Environment (JRE) Versión 6 si no está instalado, aunque pudiera requerir más dependiendo del tamaño que pueda tener la base de conocimiento a medida que se vaya actualizando.
- Computador Pentium de 512MHz o superior.

Requerimientos de software

- Sistema Operativo Windows 2000, XP, 2003, Vista (x86, x64), Windows 7 (x86, x64).
- Máquina virtual de Java (JRE) en su versión 6 o mayor.
- SWI-Prolog, versión 5.6.33 o superior.

3.3.1 Facilidades que brinda al usuario

El software SEDEFP ofrece una interface gráfica de usuario amigable, que constituye una herramienta que facilita una rápida utilización del SE. Se puede realizar inferencias de forma sencilla obteniéndose en pocos segundos los resultados esperados.

Descripción de la aplicación.

SEDEFP posee una ventana principal. Esta se muestra al iniciar la aplicación, la cual contiene una barra de menú donde se encuentran agrupadas las opciones que se le brinda a cada uno de los usuarios.



Figura 8: Pantalla principal de SEDEFP.

3.3.2 Menús

La barra de menú está compuesta por una serie de menús desplegables que se muestran según el usuario que consulte el sistema.

Menú Archivo:

- El menú **Clasificar Enfermedad** permite al especialista diagnosticar un cultivo afectado.
- El menú **Salir** permite abandonar el sistema cuando el usuario lo desee.

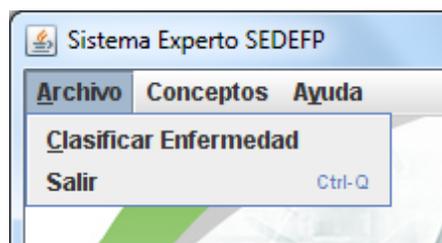


Figura 9: Menú Archivo para Clasificar Enfermedad.

Menú Conceptos:

- El menú **Buscar Conceptos** brinda la posibilidad de que el usuario introduzca la palabra de la cual desea conocer su significado.

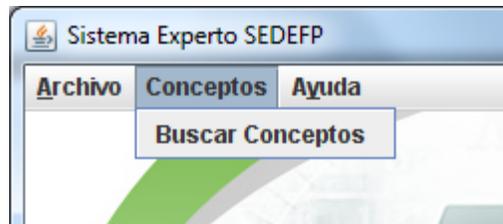


Figura 10: Menú Conceptos.

Menú Ayuda:

- El menú **Acerca de** permite que los usuarios accedan a la descripción del sistema.
- El menú **Manual de Ayuda** permite que los usuarios conozca cómo trabajar con el sistema.



Figura 11: Menú Ayuda para SEDEFP.

Principios de diseño.

El diseño de las interfaces de las aplicaciones está altamente condicionado por el medio para el cual se confeccionen. Los estándares existentes para los sistemas *stand alone*, difieren grandemente de las aplicaciones en ambiente Web, estos influyen notablemente en el éxito o fracaso de una aplicación. A continuación se describen los principios de diseño seguidos para el desarrollo de SEDEFP.

3.4 Estándares en la interfaz de la aplicación

Para esta aplicación se desarrolló el diseño de la interfaz, basándose en el estándar de ventanas de Windows, de forma amigable y sencilla para posibilitar a los usuarios un ambiente conocido y por tanto fácil a la hora de utilizarlo. De modo general se utilizaron colores agradables a la vista, siguiéndose el mismo prototipo para todas las ventanas. La familia de fuentes utilizada en el sistema es la Arial de tamaño 12 puntos, logrando de esta forma uniformidad, claridad y buena visibilidad a la hora de mostrar las explicaciones, un estilo apropiado y un entorno agradable para el usuario.

3.4.1 Concepción general de la ayuda

El sistema cuenta con una ayuda capaz de explicar al usuario cómo es la manipulación y funcionamiento del software. La ayuda está formada por un menú interno que contiene todas las funcionalidades del sistema. El usuario podrá navegar por cada uno de ellos y consultarlo en cualquier momento ya que se encuentra ubicada en el menú principal de la aplicación.

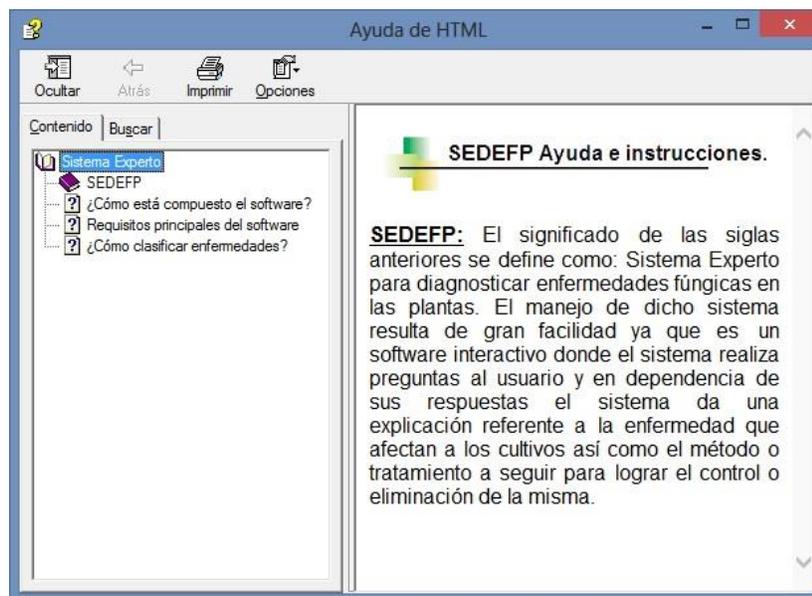


Figura 12: Ayuda de SEDEFP.

3.4.2 Tratamiento de excepciones

Una aplicación debe ser diseñada para evitar que se produzcan errores graves y que ante un error sea capaz de detectarlo y tratar de corregirlo. El correcto funcionamiento de todos los programas está dado en gran medida por la adecuada manipulación que se haga de los datos, teniendo en cuenta cada uno de los posibles errores que pueden ser introducidos en el sistema por los usuarios. Se dice que una aplicación bien diseñada minimiza el número y la peligrosidad de los errores, el control de estos, las validaciones necesarias y el tratamiento de excepciones, constituyen premisas que garantizan un buen desarrollo del software y la integridad de la información presentada.

Siempre que sea posible se debe recurrir a los controles de selección; como son: botones de opción (radio buttons), casillas de verificación (check box), entre otros, Por ejemplo para el trabajo con los botones de opción siempre habrá que seleccionar uno. De manera que si el usuario no selecciona ninguno y da aceptar dará margen al error.

SEDEFEP permite al usuario introducir datos, se verifica siempre que no se deje ningún campo en blanco, es decir, se tiene que introducir datos para continuar la interacción con la aplicación.

En cada uno de estos casos se le muestra un mensaje lo más claramente posible al usuario, informándole acerca de lo que acaba de ocurrir, o sea por qué surgió el error.

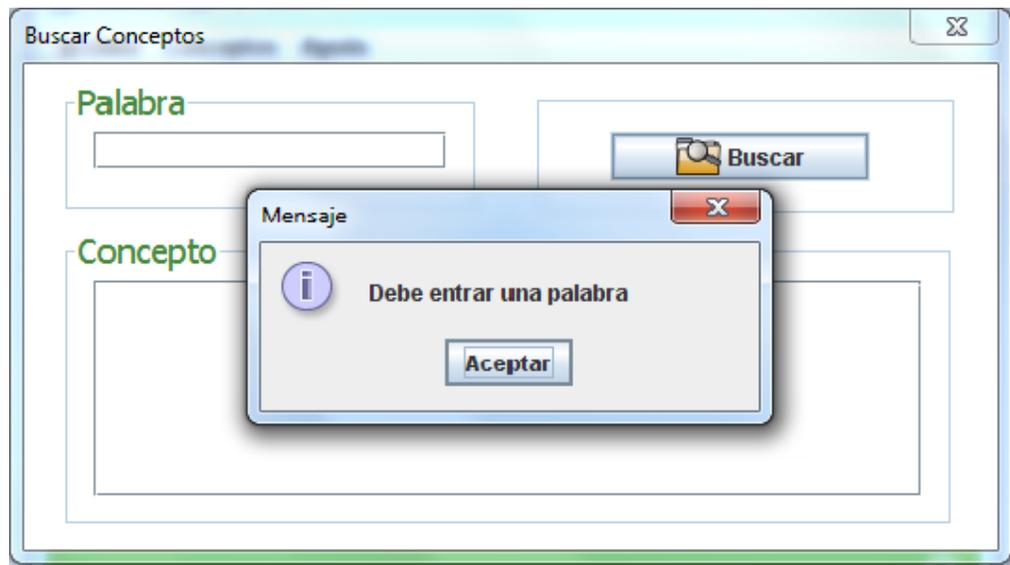


Figura 13: Tratamiento de excepciones en SEDEFP.

3.5 Conclusiones Parciales

En este capítulo se demostró que la unión de Java con Prolog es posible a través de las funciones que tiene implementada la clase Query de la biblioteca jpl. Además se eligieron los principales componentes para el diseño de SEDEFP: los ComboBox, los TextArea, List, RadioButtons, Label, Panel y Buttons que atienden al evento actionPerformed, entre otros unido a la barra de menú de la ventana principal. Además se realizó el manual de usuario de ayuda para trabajar con el sistema.

Conclusiones

Conclusiones

Se determinaron los fundamentos teóricos y metodológicos que permitieron desarrollar un SE para diagnosticar enfermedades fúngicas en los cultivos. Lo cual permitió determinar que el lenguaje de programación Prolog, resulta apropiado para la construcción del SE.

Se diseñó un SE utilizando la metodología del Dr. Mateo Lezcano (Lezcano Brito, 1995) para su construcción y de forma complementaria el Lenguaje Unificado del Modelo.

Se implementó un SE para diagnosticar enfermedades fúngicas en los cultivos de interés agrícola en la provincia de Sancti Spiritus, utilizando como herramienta para desarrollar la base de conocimiento SWI-Prolog y para la interfaz visual NetBeans 6.9.

Recomendaciones

Recomendaciones

- Ampliar el sistema para que incluya también la identificación de enfermedades víricas y bacterianas.
- Incorporarle otros cultivos al sistema.
- Agregar más palabras en la búsqueda de conceptos.

Bibliografía

Definición.De. (2008). Recuperado el 20 de mayo de 2014, de Definición de Inteligencia Artificial: definicion.de/inteligencia-artificial/

AGRIOS, G. .. (2005). *Plant Pathology*. 3ra edición.

Alfonso, W. G. (2011). *del sitio de Wilmer Garzón Alfonso*. Recuperado el 20 de mayo de 2014, de <http://ece.uprm.edu/~wgarzon/Material/2011-1/Diagnosis%20System.pdf>

Alfredo Durán Quirós, D. M. (2009). *Enfermedades Y Otros Problemas de Las Plantas*. Costa Rica: Universidad de Costa Rica Casa del Libro.

Almodóvar, W. I. (s.f.). *Clínica de diagnóstico de enfermedades y plagas en plantas*. Recuperado el 16 de 05 de 14, de academic.uprm.edu/walmodovar

Análisis y diagnóstico de enfermedades parasitarias de los cultivos hortícolas. (s.f.).

autores, C. d. (2000). *Aprenda Java como si estuvieras en primero*.

autores, C. d. (2011). *Hydro Environment*. Recuperado el 19 de mayo de 2014, de hydroenv.com.mx

Ávila, N. A. (23 de 8 de 2012). *Historia de la Inteligencia Artificial*. Recuperado el 20 de 5 de 2014, de www.slideshare.net/norma8a/

Balvantín C., I. A. (2009). "Biotecnología en el control de microorganismos fitopatógenos. *CienciAcierta*.

Bombelli, E. C. (15 de noviembre de 2011). *Modelado para la predicción de enfermedades en cultivos de alto valor comercial*. Recuperado el 19 de mayo de 2014, de <http://www.edutecne.utn.edu.ar>, edutecne@utn.edu.ar

Bonafio, M. (2008). *Historia de los Sistemas Expertos*.

Briz, J. M. (2007). *Introducción a los Sistemas Expertos*. Obtenido de www.ingenieroseninformatica.org/recursos/tutoriales/sist_exp/index.php

Bustamante, E. (2007). Elementos e importancia del diagnóstico de problemas fitosanitarios. *Manejo integrado de Plagas*.

- Carpio, M. E. (2010). *Software para la gestión de la información relacionada con el proceso productivo de la Fábrica de Cemento Siguaney*. Sancti Spíritus.
- Castillo Ron, E. G. (1999). España: Real Academia de Ingeniería.
- Contreras, G. A. (16 de julio de 2004). *Desarrolloweb.com*. Recuperado el 20 de mayo de 2014, de Qué es Prolog: <http://www.desarrolloweb.com/articulos/1573.php>
- Echemendía, M. P. (2007). *Sanidad Vegetal*. Cuba.
- Elguea, J. (s.f.). *Inteligencia Artificial y psicología: la concepción contemporánea de la mente humana*. Obtenido de http://www.hemerodigital.unam.mx/ANUIES/itam/estudio/estudio10/sec_16.html
- Evleva E.V., R. T. (2006). Extracellular proteinases from the phytopathogenic fungus *Fusarium culmorum*. *Applied Biochemistry and Microbiology*.
- García Valdivia, Z. ,. (2000). *Introducción a la Inteligencia Artificial*. La Nogalera, Guadalajara, Jalisco: Pandora.
- Gonsalves, D. C. (2004). *Transgenic virus resistant papaya: from hope to reality for controlling Papaya ringspot virus in Hawaii*. Obtenido de <http://www.apsnet.org/publications/apsnetfeatures/Pages/papayaringspot.aspx>
- Harper G, H. R. (2010). Control de Plagas y Enfermedades en los cultivos.
- Jacobson, I. B. (2006). *El Proceso Unificado de Desarrollo de Software*. La Habana.
- Lambrecht, A. K. (2004). *Las Bacterias como Patógenos Vegetales*. Recuperado el 20 de mayo de 2014, de <https://www.apsnet.org/edcenter/intropp>
- Lambrecht, V. &. (2004). Diseases of Annuals and Perennials. 202.
- Lezcano Brito, M. (1995). *Prolog y los Sistemas Expertos*. Santa Clara: Ediciones Universidad Central de las Villas.
- Martín, M. (2010). *Sistema Experto para la clasificación temprana de Dengue Severo en el Instituto Pedro Kourí*. Ciudad de La Habana.
- Montecino, J. (2003). Lenguaje Prolog. *El Rincón del Vago*, 9.
- Norving, S. R. (1996). *Inteligencia Artificial: Un enfoque moderno*. México: Primera edición.

- Pacheco, A. (2009). Ventajas y Desventajas.
- Palacios, M. I. (2008). Evolución de la Inteligencia Artificial. *Inteligencia Artificial*.
- Quintanar, T. d. (2007). *SISTEMAS EXPERTOS Y SUS APLICACIONES*. PACHUCA DE SOTO, HIDALGO.
- Ramos, J. d. (2000). *Sistemas Expertos*. Obtenido de http://www.lafacu.com/apuntes/informatica/sist_expe
- Ramos, J. d. (2000). *Trabajos de administración de centros de cómputos*.
- Rendon, L. M. (14 de noviembre de 2012). Inteligencia Artificial.
- Roque, M. J. (26 de abril de 2014). *Análisis y diagnóstico de enfermedades en las plantas*. Recuperado el 19 de mayo de 2014, de Cómo Diagnosticar Las Enfermedades: <http://cvirtual.ual.es/dirweb/servlet/bin?id=99017328-2005-06&con>
- Samper, J. J. (2002). *Sistemas Expertos: El conocimiento al poder*.
- Sánchez, J. (2003). Java2, Swing, Threads, programación en red, JDBC y JSP/ Servelets.
- Vega, J. d. (2009). *Inteligencia Artificial*. Obtenido de <http://www.itlp.edu.mx/publica/boletines/actual/inteligencia.html>
- Wielemaker, J. (2012). *Softpedia*. Recuperado el 20 de mayo de 2014, de <http://www.softpedia.es/programa-SWI-Prolog-184315.html>