Universidad Central “Marta Abreu” de Las Villas
Facultad de Matemática, Física y Computación

Trabajo para optar por Título de
Licenciado en Ciencia de la Computación

Sistema Predictor de Enfermedades en los Cultivos (SiPEC)

Autor
Leandro Montenegro Torres

Tutor
Dr. Daniel Gálvez Lio

Consultante
Dr. Orlando Saucedo Castillo

2011-2012

Declaración Jurada

El que suscribe, Leandro Montenegro Torres, hago constar que el trabajo titulado Sistema Predictor de Enfermedades en los Cultivos fue realizado en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de los estudios de la especialidad de Licenciatura en Ciencias de la Computación, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

_____

Firma del autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

_____ _____

Firma del tutor Firma del jefe del Laboratorio

28-6-2012

Fecha

Agradezco sinceramente:

- A mi tutor Dr. Daniel Gálvez Lio por ser mi ejemplo profesional a seguir.
- A toda mi familia por su ánimo y confianza, en especial a mi mamá, a mi papá y a mi novia que hicieron la carrera conmigo, a mi tío Jorge, a mi tía Ely y a mis abuelos Francisco y Caridad.
- A mi chiquitico mi eterno agradecimiento por hacerme una mejor persona, por tenerme una sonrisa lista apenas despertara el día, por confiar en mí cuando todo parecía conspirar contra nosotros, por tu ternura y comprensión, gracias mi ángel.
- A María y José por acogerme como un hijo más.
- A Becerra y Alba por acogerme como a un nieto y por apoyar siempre mi relación con su nieta.
- A mis hermanos: Yunieski, Pepito, Abel, Ariel y Héctor.
- A los amigos de la carrera por soportarme.
- A los amigos de la universidad, especialmente al Piti y Oscar.
- A los profesores que se empeñaron en hacer de mí y de mis compañeros un buen profesional.

Dedicatoria

A mi mamá, a mi papá y a mi novia.

"Si el problema tiene solución es inútil preocuparse

y si el problema no tiene solución es inútil preocuparse"

Aristóteles

RESUMEN

Uno de los factores que más influye en el rendimiento de los cultivos agrícolas son las enfermedades fungosas que atacan a los mismos, que traen como consecuencia una baja producción del cultivo afectado. La determinación a priori de estas enfermedades; resulta de gran conveniencia teniendo en cuenta los costos monetarios, de tiempo y de recursos humanos en la agricultura.

Para facilitar un diagnóstico rápido y preciso de estas enfermedades con el progreso de esta investigación se ha desarrollado una aplicación que realiza dicho proceso de forma automática.

En este trabajo se aborda el concepto y la utilidad de la Agrometeorología, además se describe detalladamente como se desarrolló el sistema creado y como se debe operar con el mismo en el momento de su instalación y utilización.

ABSTRACT

One of the factors that more influences the yield of the agricultural cultures they are the fungous illnesses that attack the same ones, which bring as consequence a low production of the affected culture. The determination to priori of these illnesses results from big expediency bearing in mind the monetary costs, of time and of human resources in the agriculture.

To facilitate a rapid and precise diagnosis of these illnesses with the progress of this investigation there has developed an application that realizes the above mentioned process automatically.

In this work there is tackled the concept and the utility of the Agrometeorology, also there is described elaborately how the created system developed and as it is necessary to operate with the same one in the moment of his installation and use.

Tabla de contenido

INTRODUCCIÓN 13
Objetivo general 14
Objetivos específicos 14
Justificación 14
Capítulo 1: “La Agrometeorología y las técnicas de programación utilizadas” . 15
1.1 Agrometeorología 15
1.1.1 Definición. 15
1.1.2 Elementos dentro de la Agrometeorología 16
1.2 Descripción del Problema 17
1.2.1 Flujo de los datos 18
1.3 Herramientas y tecnologías disponibles 19
1.3.1 Lenguaje de programación JAVA 19
1.3.2 NetBeans 20
1.3.3 Principales Bibliotecas Utilizadas 21
1.3.4 SQL-Server 2000 25
1.3.5 Prolog 26
1.4 Módulos de un sistema basado en conocimiento 28
1.5 Ejemplos de SBR para el diagnóstico de enfermedades en cultivos. 29
Capítulo2: “Sistema Predictor de Enfermedades en Cultivos (SIPEC)” 32
2.1 Descripción del Sistema. 32
2.1.1 Descripción del Actor. 33
2.1.2 Descripción de los casos de usos del sistema. 33
2.1.3 Diagrama de Clases 39
2.2 Gestión de los datos. 40
2.3 Implementación del Sistema. 44
2.3.1 Módulo para el Diagnóstico de Enfermedades por Razonamiento Basado en Reglas. 45
2.3.2 Módulo de Visualización de Datos. 47
2.3.3 Gestión de la Base de Conocimientos 49
2.3.4 Módulo de Gestión de Datos y Estaciones Meteorológicas. 50
2.4 Diagrama de despliegue. 51

2.5 Configuración de las herramientas para el desarrollador. 52
2.5.1 Conexión de “SIPEC.jar” con la Base de Datos 52
2.5.2 Conexión de “SIPEC.jar” el Prolog 53
Capítulo 3: “Manual de Usuario”. 54
3.1 Requerimientos e Instalación de los Componentes del Sistema. 54
3.1.1 Instalación de la “Base de Datos”. 54
3.1.2 Requerimientos e Instalación del “SIPEC.jar” 54
3.2 Manejo Óptimo del “SIPEC.exe”. 55
3.2.1 Datos Meteorológicos. 55
3.2.2 Predicciones. 66
3.2.3 Análisis Gráfico. 70
3.2.4 Estaciones. 73
Conclusiones 78
Recomendaciones 79
Referencias Bibliografías 80
Anexos 81
Anexo 1: “Ejemplo de Base de Conocimientos.” 81

INTRODUCCIÓN

Las enfermedades constituyen actualmente una de las principales limitantes en la potencialidad del rendimiento de los cultivos. La principal herramienta con que cuenta el productor para luchar contra esta problemática es la prevención en el manejo racional de las enfermedades. Este trabajo preventivo es realizado al evaluar un grupo de reglas a partir de un conjunto de datos meteorológicos. La experiencia alcanzada por expertos en el trabajo preventivo ha determinado que dicha tarea se torna lenta y poco precisa dado por las dimensiones de las reglas y el gran volumen de datos entrantes en el proceso.

En las últimas décadas, el aumento de la tecnología ha posibilitado que se pueda pronosticar gran cantidad de sucesos a partir del estudio de las variables climatológicas, entre estos se encuentra el pronóstico de plagas y enfermedades en los cultivos. Entre las principales ventajas de predecir cuándo se puede desarrollar una enfermedad en una planta, o ser atacada por una plaga se encuentran las siguientes:

- Reducir la contaminación que provoca la utilización de productos químicos en el terreno.
- Minimizar la contaminación química de los alimentos que se estén cultivando.
- Ayudar a la economía con la reducción del uso de los productos que se utilizan para contrarrestar las enfermedades o plagas.

El sistema que sea capaz de pronosticar la aparición de una enfermedad, debe estar compuesto por una serie métodos que controlen la información meteorológica diaria y puedan clasificar si un día es o no favorable para la aparición de una plaga o enfermedad en la planta. Entre las enfermedades que trataremos en este trabajo están el tizón tardío, el moho azul y la sigatoka.

Objetivo general

Desarrollar un sistema que permita el pronóstico de algunas plagas en varios cultivos, utilizando variables climatológicas como rasgos de entrada.

Objetivos específicos

- Diseñar un sistema que solucione el problema planteado.
- Crear una Base de Datos permita almacenar los datos que utilizará el software.
- Implementar la Base del Conocimiento que evaluará las reglas que se obtuvieron a partir de la experiencia de los expertos.

Justificación

Los países del tercer mundo que cuentan con pocos recursos económicos deben siempre intentar reducir los costos de la producción al mínimo, nuestro país está incluido dentro de los ya mencionados, por tal razón se pretende implementar un sistema que sea capaz de predecir cuándo será afectado un cultivo por alguna de las plagas estudiadas en este trabajo. Esto será de gran ayuda para la economía porque no se utilizarán recursos para contrarrestar las enfermedades, a no ser que se pronostique que en un momento dado una plaga atacará un cultivo; este trabajo también ayudará a reducir la contaminación que recibe la tierra después que se utilizan productos químicos para evitar las plagas y curar las plantas, esta reducción de la cantidad de químicos utilizados en las plantas también ayudará a que los alimentos presenten menos contaminación y se afecte menos la salud de los consumidores.

Capítulo 1: "La Agrometeorología y las técnicas de programación utilizadas"

En este capítulo se presenta el planteamiento del problema investigativo, así como aspectos teóricos relacionados con las técnicas utilizadas para dar solución efectiva al mismo. Se describe brevemente que es la agrometeorología y se muestran ejemplos de sistemas predictores.

1.1 Agrometeorología

Como lo indica su nombre, la agrometeorología resulta de la intersección entre dos disciplinas, las ciencias de la atmósfera y la agronomía. Necesita entonces de principios físicos y matemáticos de la primera, y de los principios físicos, biológicos, económicos y sociales propios de la agronomía.

1.1.1 Definición.

La Agrometeorología es la ciencia que trata sobre las características físicas del medio ambiente en el que se efectúa el crecimiento de las plantas y los animales. Se relaciona con el estudio de los procesos físicos que tienen lugar en este medio ambiente y también con la utilización e influencia de estas características físicas y procesos, en interés de la agricultura.

Esta definición de Agrometeorología se basa en uno de los conceptos fundamentales de la Biología moderna, el cual considera al organismo y su medio ambiente formando una unidad con caracteres inter-dependientes. Los estudios teóricos y experimentales en la ciencia de la Biología y en la Agricultura han confirmado la validez de este concepto.

Los trabajos sobre Agrometeorología han sido realizados por los más famosos biólogos y agrónomos entre los cuales podemos citar a: P. A. Kóstichev, A. A. Izmail'ski, K. A. Timiriázev, N. A. Maksímov, y otros. Debemos señalar que

estos trabajos continúan y a su vez, los nuevos aportes, enriquecen constantemente la ciencia de la Agrometeorología.(Marín 2005; Ruiz 2006)

1.1.2 Elementos dentro de la Agrometeorología

La Agrometeorología estudia la atmósfera, particularmente la capa en la cual crecen y se desarrollan las partes aéreas de las plantas, especialmente el régimen térmico de esta capa y la relación con la capa superficial de suelo. También estudia los movimientos verticales y horizontales del aire en la referida capa de la atmósfera. Estudia además, la energía solar recibida por las plantas cultivadas, así como la utilización técnica de dicha energía.

El objetivo de la Agrometeorología no es solamente el estudio de la capa de atmósfera próxima al suelo –los dos primeros metros–, su estado y los procesos físicos que tienen lugar en ella, sino, también, encontrar métodos para influir en el clima y el estado físico de esta capa como vía para combatir la escarcha, la sequía, los vientos secos y otras condiciones desfavorables. Si es posible cambiar el estado físico del aire cerca del suelo, se está en posición de mejorar el clima en una localización dada y crear condiciones favorables para las plantas.

Otra tarea importante de la Agrometeorología es el estudio de la aeración del suelo, así como el balance térmico y de la humedad de su superficie, en relación con su composición, relieve, microclimas latitud y otros factores. Otra cuestión importante está referida a la prevención de plagas y/o enfermedades que inciden en las plantas, ya que muchas de ellas se forman y desarrollan a partir de las condiciones atmosféricas.

Otro objetivo es el desarrollo, en las granjas pertenecientes a las estaciones experimentales y otras instituciones agrícolas, de métodos nuevos en Agrometeorología.

Se debe puntualizar que la Agrometeorología no está relacionada con las propiedades o los procesos químicos que tienen lugar en este medio ambiente, ya que ambos pertenecen a la Agroquímica y Bioquímica, respectivamente.

La Agrometeorología no debe confundirse con la Meteorología General, ciencia esta, que se ocupa del estudio de la atmósfera como un todo y que tiene como una de sus principales tareas el pronóstico del tiempo.

La Agrometeorología se divide en Fitometeorología –que es la Meteorología aplicada a las plantas de cultivo con inclusión de las plantas forestales– y la Zoometeorología –que es la Meteorología aplicada a la crianza de animales–. (Marín 2005; Ruiz 2006; Olivares and Chirinos 2009)

1.2 Descripción del Problema

Los agricultores actuales, día a día, tratan de que las producciones de los cultivos que atienden sean cada vez mayores y de mejor calidad. Para lograr este propósito, no solo deben hacer una selección adecuada del suelo y la semilla o postura a utilizar en dependencia de lo que vallan a sembrar, también deben aplicar tratamientos biológicos con el propósito de ayudar a las mismas en su desarrollo, así como tratamientos químicos para combatir plagas y/o enfermedades.

Es de mucho interés para los productores, aplicar dichos tratamientos en el momento preciso, esto garantiza la protección y ayuda a la planta, además no crea afectaciones severas al medio ambiente.

En los momentos actuales los productores dan tratamiento a los cultivos siguiendo dos criterios fundamentales: usando la experiencia acumulada durante años de trabajo y transmitida de generación en generación, o tratando el campo cuando este ya ha sido afectado por alguna enfermedad. Por esto se hace necesario tener un diagnóstico a priori acerca de las enfermedades que pudieran incidir en el campo para poder aplicar tratamientos antes de que el

mismo sea afectado directamente, contribuyendo de esta manera a la conservación del suelo y del medio ambiente en general.

Para realizar el diagnóstico se cuenta con bibliografía que describe como llevar a cabo dicha tarea, la misma se realiza evaluando un numeroso grupo de reglas a partir de datos meteorológicos con especificaciones propias de esta ciencia, que en ocasiones requieren pre-procesamiento y en general tornan el proceso un poco engorroso para los productores agrícolas.
Está demostrado (según la práctica) que este proceso se torna lento y poco preciso por el gran volumen de datos manipulados, sin dudas esto constituye un problema prioritario. Esta dificultad conlleva a desarrollar una herramienta que sea capaz de hacer este trabajo predictivo de forma rápida y precisa, ayudando así a una mejor administración de productos biológicos y químicos a las plantas.

1.2.1 Flujo de los datos

Los datos meteorológicos se reciben en la Estación Experimental Agrícola “Álvaro Barba Machado”, estos se reciben en un fichero texto con la estructura que se muestra en la Figura 1.2.1.1. Es en este centro donde se agregan al sistema los datos para que este los procese y muestre un resultado. La Figura 1.2.1.2 muestra para el caso de uso Iniciar Predicción cómo se comporta el flujo de los datos desde que se introducen al sistema hasta que se obtiene un resultado.

```
datos de sanidad
04/03/12.
estac. tmax tmed tmin tmax tmed tmin hmax hmed hmin horas HR lluvia vto  drc
78308  31.0 24.7 19.0 20.8 19.9 19.0 91   66   32   02.00 89  0.0   3.5  E.
78326  33.5 24.8 16.7 20.0 18.9 17.8 94   75   42   07.10 94  0.0   4.6  SW.
78338  33.0 24.4 16.6 18.5 17.6 16.6 93   73   32   06.30 93  0.0   4.1  v.
78343  32.6 25.4 18.2 21.0 20.0 18.9 99   67   27   06.30 98  0.0   7.8  SE.
78348  29.4 24.7 20.0 22.7 21.4 20.0 93   77   53   05.30 91  0.0   1.0  V-C.
```

Figura 1.2.1.1 Estructura con que se reciben los datos.

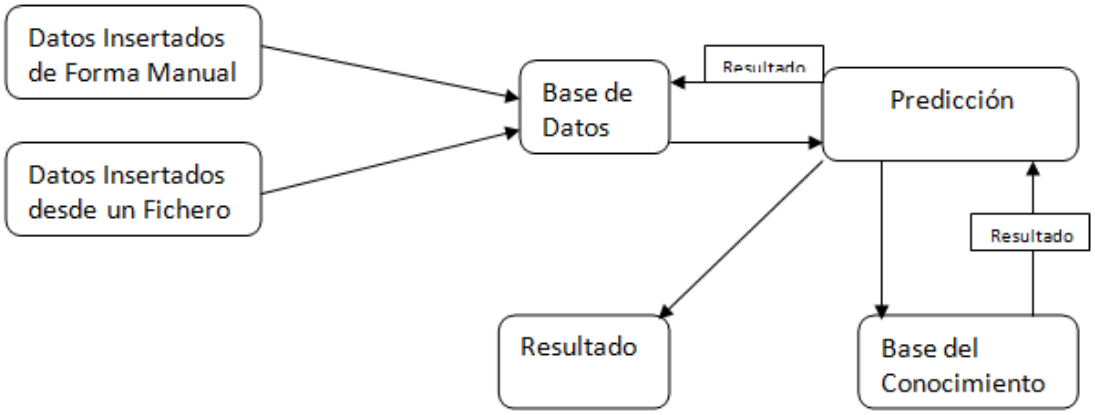


Figura 1.2.1.2 Flujo de datos para el caso de uso Iniciar Predicción.

1.3 Herramientas y tecnologías disponibles

Abordamos a continuación un conjunto de herramientas para el desarrollo de aplicaciones que serán utilizadas para la implementación computacional del sistema.

1.3.1 Lenguaje de programación JAVA

Java es un lenguaje de programación de alto nivel de última generación que se basa en el paradigma orientado a objetos. Por sus características, es un lenguaje que permite disminuir el tiempo de desarrollo y a la vez aumentar la manejabilidad del código.(Herrera 2007)

Fue creado en Sun Microsystems, en el año 1991 inicialmente estaba desinado a equipos electrodomésticos. Como lenguaje de programación para computadores, Java se introdujo a finales de 1995.

Debido a la existencia de distintos tipos de procesadores y ordenadores los ingenieros de Sun llegaron a la conclusión de que era muy importante conseguir un software que no dependiera del tipo de procesador utilizado; por lo que se puede decir que una de sus características más importante es su independencia de la plataforma. Al compilar un programa en Java, se genera

un pseudocódigo llamado “bytecode” para una máquina virtual que es la encargada de compilar el pseudocódigo generado en tiempo de ejecución. Es decir, al contar con un intérprete de este tipo de código para una computadora particular, cualquier programa compilado en Java corre sin ninguna modificación.

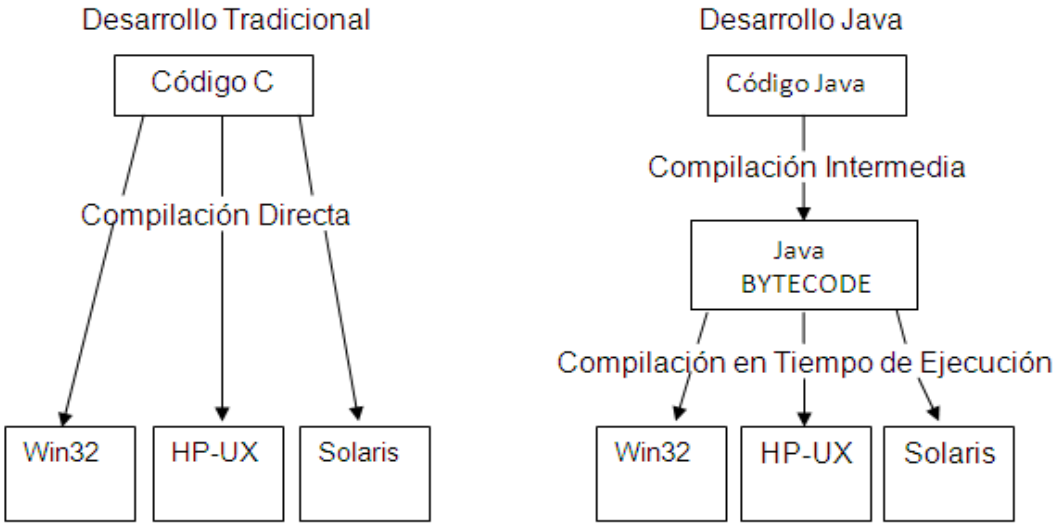


Figura 1.3.1.1 Comparación entre la compilación de un programa hecho en Java y uno en C. (tomado de (McGovern, Adatia et al. 2003))

La compañía Sun describe el lenguaje Java como “simple, orientado a objetos, distribuido, interpretado, robusto, seguro, de arquitectura neutra, portable, de altas prestaciones, multitarea y dinámico”. Todo ello describe bastante bien el lenguaje Java, aunque en algunas de esas características el lenguaje sea todavía bastante mejorable. (Jalón, Rodríguez et al. 1999; McGovern, Adatia et al. 2003)

1.3.2 NetBeans

El NetBeans es una aplicación del tipo IDE (Interface Development Environment) esta tecnología está dotada de muchas facilidades y ventajas; fue desarrollado por Sun Microsystems. La rapidez con que se pueden desarrollar las aplicaciones es un aspecto a tener muy presente, pues es de vital importancia para los usuarios tener el producto final lo más rápido posible en la mayoría de los casos. Entre los lenguajes que compila se encuentra el java, este es software libre por lo que se presta para apegarse a la política que están

siguiendo la mayoría de las empresas del país de usar tecnologías desarrolladas con software libre. Otras de las ventajas que ofrece esta tecnología es su rapidez en el procesamiento de datos, su alta capacidad para soportar estructuras de almacenamiento complejas, la posibilidad de construir Interfaces Gráficas de Usuario utilizando la llamada Programación Visual. En lugar de escribir el código puro en Java se crean aplicaciones utilizando herramientas visuales de manera que el código a escribir es mínimo, el rehúso de código fuente, así como la disponibilidad de la biblioteca Swing que dota al diseñador de la interfaz de usuario de las aplicaciones de muchas componentes que enriquecen y embellecen dichas interfaces, además de proporcionarle a los programadores otras componentes que implementan muchas funcionalidades reusables. (Cerda)

1.3.3 Principales Bibliotecas Utilizadas

En este epígrafe se abordan brevemente las bibliotecas de java utilizadas en la solución computacional.

1.3.3.1 Java.sql.*

Implementada por la API (Interfaz de Programación de Aplicaciones) de JDBC. Es una biblioteca importada por Java para hacer consultas en BD, si se usa únicamente esta biblioteca, entonces podemos realizar una conexión básica y simple, sirve para acceder a bases de datos SQL mediante una llamada a un método de la biblioteca Java.sql.*. Esta biblioteca contiene su propia estructura y hace llamado de sus propias clases.

Breve descripción de algunas de sus clases:

- Driver: Permite conectarse a una Base de Datos, cada gestor de Base de Datos requiere un Driver distinto.
- DriverManager: Permite gestionar todos los Drivers instalados en el sistema.
- DriverPropertyInfo: Proporciona diversa información acerca de un Driver.

- Connection: Representa una conexión con una Base de Datos. Una aplicación puede tener más de una conexión a más de una Base de Datos.
- DatabaseMetadata: Proporciona información acerca de una Base de Datos, como las tablas que contiene, etc.
- Statement: Permite ejecutar sentencias SQL sin parámetros.
- PreparedStatement: Permite ejecutar sentencias SQL con parámetros de entrada.
- CallableStatement: Permite ejecutar sentencias SQL con parámetros de entrada y salida.
- ResultSet: Contiene las filas o registros obtenidos al ejecutar un SELECT.
- ResultSetMetadata: Permite obtener información sobre un ResultSet, como el número de columnas, sus nombres, etc.

1.3.3.2 java.jpl.*

JPL es una biblioteca de uso de la interfaz SWI-Prolog exterior y de la interfaz Java JNI, esta proporciona un interfaz bidireccional entre Java y Prolog que se puede utilizar para incrustar Prolog en Java, así como para incluir Java en Prolog.

Objetivos de la biblioteca:

- Permitir que aplicaciones del Prolog exploten las clases, instancias, etc de Java.
- Permitir que las aplicaciones Java manipulen las bibliotecas de Prolog estándar, predicados, etc.
- Permitir híbridos entre las aplicaciones Prolog a las de Java, se diseñó e implementó con el fin de aprovechar al máximo los lenguaje y que sean comprobables, depurables, mantenibles, etc
- Capacidad de despliegue de mínimo impacto: soporte de ejecución para aplicaciones Java + Prolog independiente de la posición, sin necesidad de cambios a los registros, bibliotecas del sistema, archivos de configuración del sistema, etc

- Capacidad de despliegue de dependencia mínima: como la máquina virtual de java, el Prolog y el Java en tiempo de ejecución no debe depender de ningún elemento que no se pueda dar por hecho en las instalaciones del sistema operativo.
- Capacidad de despliegue de vulnerabilidad mínima: el Prolog y el Java en tiempo de ejecución deben ser inmunes a las variaciones legítimas en su entorno (la configuración de PATH, otras aplicaciones y bibliotecas, incluyendo otras aplicaciones de Prolog en Java).

1.3.3.3 javax.swing.*

Originalmente la creación de interfaces gráficas en Java se realizaba con la biblioteca AWT (Abstract Window Toolkit). Pero esta tenía serías limitaciones como el número de pequeños programas, sus posibilidades o la gestión de eventos. Era realmente complejo el lograr funcionalidades que se obtenían de forma sencilla con otras bibliotecas.

Todo esto cambió con la llegada de Swing la cual ha mostrado una calidad y unas funcionalidades superiores a AWT. De hecho, Swing es una biblioteca repleta de funcionalidades que permiten la creación de interfaces gráficas muy ricas tanto estéticamente como en funcionalidades. Usando esta biblioteca se podrán construir los tres tipos de programas o aplicaciones que son JFRAME, JWINDOW y JAPPLET.

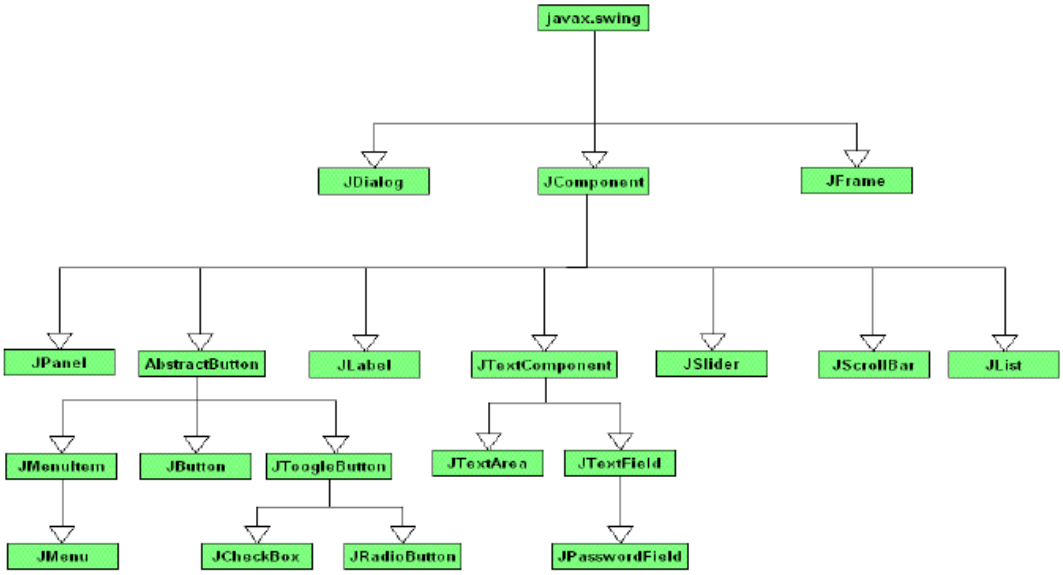


Figura 1.3.3.1 Clases que contiene la biblioteca JAVA SWING

1.3.3.4 jfree.chart.*

JFreeChart es un marco de software de código abierto para el lenguaje de programación Java, el cual permite la creación de gráficos complejos de forma simple.

También trabaja con GNU Classpath, una implementación en software libre de la norma estándar de biblioteca de clases para el lenguaje de programación Java.

Esta biblioteca es compatible con una serie de gráficas diferentes, incluyendo cuadros combinados. Algunos de estos tipos de gráficos con los que es compatible son:

- Gráficos XY (línea y dispersión). Es posible usar un eje del tiempo.
- Gráfico circular.
- Diagrama de Gantt.
- Gráficos de Barras (horizontal y vertical, apiladas e independientes). También tiene incorporado un dibujador de histogramas.
- Varias gráficas específicas (tabla de viento, gráfica polar, burbujas de diferentes tamaños, etc).

1.3.4 SQL-Server 2000

SQL Server es un sistema administrador para bases de datos relacionales basadas en la arquitectura cliente/servidor. Transact-SQL es el lenguaje que se emplea para mandar peticiones entre el cliente y el servidor. Es un lenguaje exclusivo de SQL Server, pero basado en el lenguaje SQL estándar, utilizado por casi todos los tipos de bases de datos relacionales que existen. SQL Server otorga a los administradores una herramienta potencialmente robusta, provista de las herramientas suficientes que le permiten mantener un óptimo nivel de seguridad en la utilización de los recursos del sistema y de la base de datos, que en este camino van cogidos de la mano. La finalidad de SQL Server 2000 es analizar y administrar datos, dar mayor escalabilidad, disponibilidad y seguridad a las aplicaciones de análisis y los datos empresariales y potenciar las aplicaciones dando una mayor prestabilidad.

SQL Server 2000 está diseñado para trabajar con dos tipos de bases de datos:

- OLTP (OnLine Transaction Processing): Son bases de datos caracterizadas por mantener una gran cantidad de usuarios conectados concurrentemente realizando ingreso y/o modificación de datos. Por ejemplo: entrada de pedidos en línea, inventario, contabilidad o facturación.
- OLAP (OnLine Analytical Processing): Son bases de datos que almacenan grandes cantidades de datos que sirven para la toma de decisiones, como por ejemplo las aplicaciones de análisis de ventas.

Los entornos Cliente / Servidor están implementados de tal forma que la información se guarde de forma centralizada en un computador central (servidor), siendo el servidor responsable del mantenimiento de la relación entre los datos, asegurarse de su correcto almacenamiento, establecer restricciones que controlen la integridad de datos, etc.

Del lado cliente, este corre típicamente en distintas computadoras las cuales acceden al servidor a través de una aplicación, para realizar la solicitud de datos, los clientes emplean el SQL (Structured Query Language), este lenguaje tiene un conjunto de comandos que permiten especificar la información que se desea recuperar o modificar.(Enriquez 2007)

1.3.5 Prolog

La programación lógica, junto con la funcional, forma parte de lo que se conoce como programación declarativa, hace búsquedas dirigidas por objetivo y a ciegas (primero en profundidad), además utiliza el mecanismo de regreso a atrás (*backtracking*).

En los lenguajes tradicionales, la programación consiste en indicar cómo resolver un problema mediante sentencias; en la programación lógica, se trabaja de una forma descriptiva, estableciendo relaciones entre entidades, indicando no cómo, sino qué hacer. La ecuación de Robert Kowalski (Universidad de Edimburgo) establece la idea esencial de la programación lógica: *algoritmos* = *lógica* + *control*.

Es decir, un algoritmo se construye especificando conocimiento en un lenguaje formal (lógica de primer orden), y el problema se resuelve mediante un mecanismo de inferencia (control) que actúa sobre aquél.(ROSSEL)

¿Cuándo surge el PROLOG?

En 1970 Colmerauer y su grupo de la Universidad de Marsella (Francia), desarrollan un probador de teoremas especializado, escrito en FORTRAN, el cual utilizaron para implementar sistemas de procesamiento del lenguaje natural.

Un programa PROLOG consta de un conjunto de aserciones, las cuales se consideran que constituyen los axiomas propios de una teoría junto con un conjunto de cláusulas objetivos, a los que se consideran teoremas que hay que probar.(Bramer 2005)

Beneficios:

- Permite crear programas de la Inteligencia Artificial (IA) mucho más rápido y más fácil.
- Es ideal para implementar sistemas expertos y procesamiento del lenguaje natural.
- Los mecanismos de inferencia como primero en profundidad y sus procedimientos son partes de él (built-in).

Prolog cuenta con dos tipos de estructuras: términos y sentencias. Los términos pueden ser constantes, variables o functores. Las sentencias son reglas o cláusulas.

Ventajas del Prolog:

1. Se destaca la elegancia en la terminación de la aplicación. La facilidad de hacer interfaces con otros lenguajes de programación.
2. Facilidades de manipulación de base de datos.
3. Otro aspecto de mucha importancia está ligado al hecho de que las aplicaciones Prolog son compiladas, lo que hace que el código de sus bases de conocimiento esté oculto a curiosos que le puedan ocasionar daños y mal funcionamiento al sistema.
4. Los sistemas que se entregan de forma independiente (stand-alone) son más rápidos y ocupan menos espacio en disco, ya que no necesitan la máquina de inferencia ni sus herramientas auxiliares, tales como: su compilador, algún editor de texto que pueda poseer, etc.
5. Mejor control sobre la inferencia a partir de los mecanismos de cortes, tijeras y repeat...fail hacen fácil controlarla.
6. Problemas complejos se resuelven con facilidad.
7. Los programas se mantienen sin dificultad en su ciclo de vida.
8. Obliga al programador a especificar una descripción bien estructurada del problema.
9. Tiempo corto de desarrollo.
10. Fácil de aprender, de leer y de modificar.
11. Fácil manipulación de estructuras de datos complejas (listas, árboles, grafos, etc.) (Marín 2005)

1.4 Módulos de un sistema basado en conocimiento

Base de conocimientos: Es la parte del sistema experto que contiene el conocimiento sobre el dominio en el que se desempeña. Hay que obtener el conocimiento del experto y codificarlo en la base de conocimientos.

Existen diversas formas de representar el conocimiento en un sistema basado en reglas como son: las reglas de producción, los frames, etcétera.
Generalmente la decisión de cual usar está acorde con el propósito del mismo, la forma de obtención del conocimiento, entre otros. Una de las formas más comunes y sencillas son las reglas de producción. Una regla es una estructura condicional que relaciona lógicamente la información contenida en la parte del antecedente con otra información contenida en la parte del consecuente.

Motor de inferencia: El mecanismo de inferencia es la unidad lógica con la que se extraen conclusiones de la base de conocimientos, según un método fijo de solución de problemas que está configurado imitando el procedimiento humano de los expertos para solucionar problemas.

Subsistema de explicación: Una particularidad de los sistemas expertos es su capacidad para explicar su razonamiento. Usando el módulo del subsistema de explicación, un sistema experto puede proporcionar una explicación al usuario de por qué está haciendo una pregunta y como ha llegado a una conclusión. Este módulo proporciona beneficios tanto al diseñador del sistema como al usuario. El diseñador puede usarlo para detectar errores y el usuario se beneficia de la transparencia del sistema.

Interfaz de usuario: La interacción entre un sistema experto y un usuario se realiza en lenguaje natural. También es altamente interactiva y sigue el patrón de la conversación entre seres humanos. Para conducir este proceso de manera aceptable para el usuario es especialmente importante el diseño del

interfaz de usuario. Un requerimiento básico del interfaz es la habilidad de hacer preguntas. Para obtener información fiable del usuario hay que poner especial cuidado en el diseño de las cuestiones. Esto puede requerir diseñar el interfaz usando menús o gráficos.

1.5 Ejemplos de SBR para el diagnóstico de enfermedades en cultivos.

Se han desarrollado una gran gama de SE para el diagnóstico de enfermedades en cultivos agrícolas principalmente en el exterior, no así en nuestro país, todos ellos han contribuido a un mejor rendimiento de los cultivos y apreciables ahorros monetarios, sin dejar por alto la protección del medio ambiente.

El CITPATH, Sistema de diagnóstico y descripción de enfermedades en cítricos, fue desarrollado en Florida en el año1995, consiste de dos piezas de software: un sistema experto para el diagnóstico de las enfermedades (dados los síntomas) y un texto en hypertext que describe las enfermedades y las ilustra por medio de diapositivas que se despliegan en la pantalla. El SE fue elaborado para la predicción de enfermedades en cítricos, para ello usa un SBR que evalúa un árbol de reglas a partir de preguntas que son respondidas por el usuario del sistema evidenciando los síntomas que presenta el cultivo.(Marín 2005)

CALEX, es un sistema experto que fue desarrollado para la diagnosis de los problemas de la planta del melocotón y de la nectarina por la universidad de California. CALEX es un SBR y utiliza factores de certeza, la base de conocimiento del sistema de diagnóstico de CALEX contiene aproximadamente 600 reglas para la diagnosis de 120 problemas de melocotones y de nectarinas, representando la mayoría de los problemas de la producción en California. (Marín 2005)

VITIS, es otro ejemplo de SBR para el diagnóstico de enfermedades en la uva, fue desarrollado en 1992 con la cooperación de los especialistas de Pensylvania, de New York, de Ohio, y de Michigan.(Marín 2005)

Dos programas de diagnóstico computarizados, CIT-Xpert y TFRUIT- Xpert, fueron desarrollados para asistir rápidamente a productores comerciales en el diagnóstico de enfermedades en cultivos agricolas, El CIT-Xpert se centra en la fruta cítrica, mientras que el TFRUIT-Xpert se centra específicamente en el aguacate, carambola, mango y papaya. Los sistemas fueron desarrollados con la cooperación con los especialistas de la investigación y de la extensión con maestría en el área de las enfermedades a diagnosticar. La metodología de los sistemas reproduce el proceso de diagnóstico del razonamiento de estos expertos. (Marín 2005)

En nuestro país se desarrolló en la provincia de Cienfuegos un software llamado Pronostizón, el cual se encarga de pronosticar Tizón Tardío en la papa. Según los especialistas que hemos podido entrevistar por su experiencia en el uso de este programa, aseguran que los resultados obtenidos por el mismo son alentadores.(Marín 2005)

Se quiere abordar el siguiente software el cual fue construido para dar solución a un problema muy similar al que se presenta. CASTOR 2.0 es un programa para el manejo de datos meteorológicos y la predicción del Tizón Tardío en la papa y en el tomate. Fue desarrollado en el 2002 en la Universidad de Florida, Estados Unidos, este consta con un módulo basado en reglas, el cual es evaluado automáticamente a partir de datos meteorológicos obtenidos de estaciones meteorológicas automáticas, dando como salida las predicciones realizadas usando el método BliteCast. (Marín 2005)

Ahora nos centraremos en un software realizado en la Universidad Central de Las Villas en el 2005, este se construyó con el objetivo de predecir dos enfermedades específicas, el Tizón Tardío en la papa y el tomate y el Moho Azul en el tabaco. A partir de la información climatológica introducida al

software este es capaz de brindar un pronóstico diciendo si se van o no a desarrollar estas plagas en el cultivo. (Marín 2005)

En julio 2010. En la UNAM, el investigador Nicolás Kemper Valverde, del Centro de Ciencias Aplicadas y Desarrollo Tecnológico (CCADET), ha desarrollado una red de sistemas inteligentes que alerta a los productores sobre el posible surgimiento de enfermedades y plagas más frecuentes en cultivos de limón, durazno, aguacate, fresa, chile, guayaba, jitomate y mango. El sistema utiliza una computadora con datos especializados sobre los principales daños a cada plantío. Mediante un convenio de colaboración entre esta casa de estudios y la Fundación Produce Michoacán, la tecnología se aplica en ese estado de la República, para proteger sembradíos. (Valverde 2010)

Capítulo2: “Sistema Predictor de Enfermedades en Cultivos (SIPEC)”

En este capítulo se aborda el desarrollo y funcionamiento del sistema ante las características del problema a resolver, se describen detalladamente los componentes y módulos del mismo, así como las tecnologías computacionales que se usaron para la implementación.

2.1 Descripción del Sistema.

Los valores meteorológicos son obtenidos por una serie de estaciones meteorológicas que se encuentran distribuidas en distintos lugares de la provincia de Villa Clara. Estas estaciones envían los datos diariamente a una estación central ubicada en nuestra universidad, la Estación Experimental Agrícola “Álvaro Barba Machado”. Y es en este lugar donde se realiza el procesamiento de estos datos; o sea, donde se evalúan las reglas para obtener las predicciones de enfermedades que incidirán en los cultivos en cada zona en específico. El sistema será instalado en esta estación con el fin de que esta tarea se realice de forma rápida y efectiva.

Son varios los casos de uso del sistema extraídos del problema a resolver, dando cumplimiento en la implementación de los mismos se ha logrado la integridad del sistema en este sentido. Al referirse a la gestión de los datos, el sistema posee un módulo que se encarga de la inserción manual y/o inserción usando ficheros, modificación, eliminación, y muestra de los mismos, dando así al usuario facilidades de manipulación y almacenamiento de estos en una base de datos. El sistema está preparado para eliminar, modificar atributos e insertar nuevas estaciones meteorológicas proveedoras de datos meteorológicos. El proceso de predicción constituye el caso de uso donde se ha hecho un énfasis especial por la gran importancia del mismo, ya que

constituye la base principal y razón de desarrollo de esta investigación, para dar cumplimiento a esta tarea se ha implementado un módulo que realiza el diagnóstico apoyándose en la base de conocimientos. La figura 2.1.1 muestra el diagrama de actores y casos de uso del sistema:

2.1.1 Descripción del Actor.

Actor	***Descripción***
Usuario	Este actor tiene la posibilidad de realizar todas las acciones del sistema, como son administrar la base de datos y realizar las predicciones.

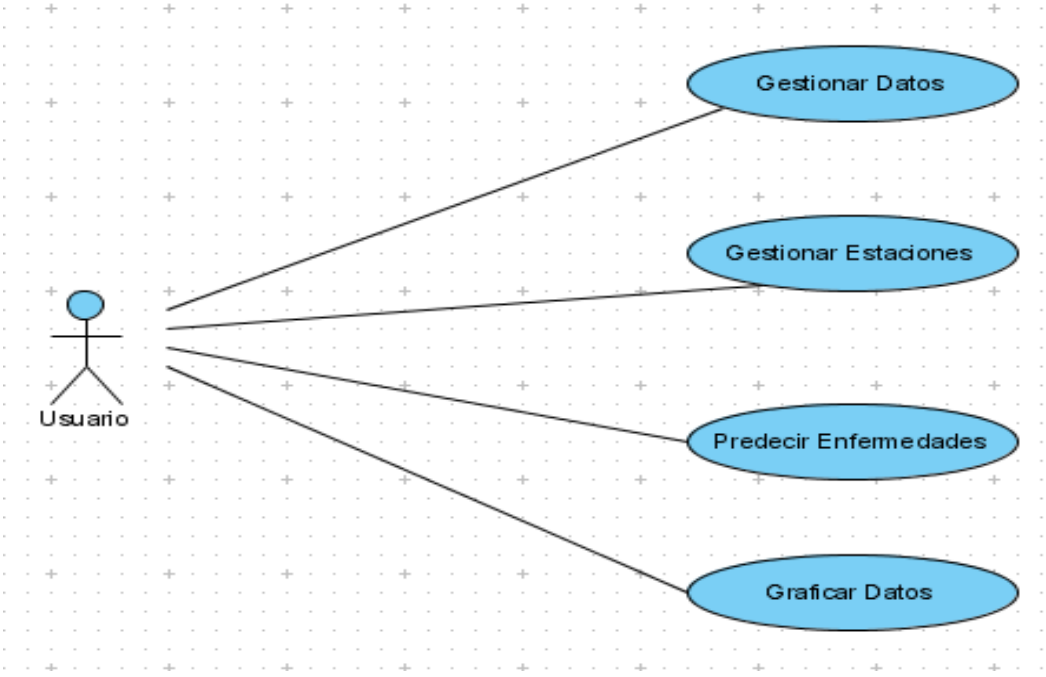


Figura 2.1.1: Diagrama de actores y casos de uso.

2.1.2 Descripción de los casos de usos del sistema.

En este epígrafe se realizará una breve descripción de los casos de uso más significativos del sistema.

2.1.2.1 Caso de uso "Gestionar Datos"

Caso de Uso:	Gestionar Datos
Actor:	Usuario
Propósito:	Gestionar los datos climatológicos en la Base de Datos.
Resumen:	El usuario selecciona la opción Gestionar Datos, escoge el escenario que desea, introduce los datos y la BD se modifica.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción Gestionar Datos del menú Datos Meteorológicos.	2. Muestra una serie de escenarios: a) Insertar Datos Manualmente. b) Insertar Datos de un fichero txt. c) Insertar Datos de la Sigatoka. d) Modificar Datos. e) Mostrar Datos. f) Insertar Humedad por Horas. g) Eliminar Datos.
3. El usuario elige el escenario a)	4. Muestra la ventana Insertar Datos que tiene una serie de campos: 1. Día 10. TMIN 2. Mes 11. HMAX 3. Año 12. HMED 4. Estación 13. HMIN 5. TMAX 14. Horas 6. TMED 15. HR 7. TMIN 16. Lluvia 8. TMAX 17. Viento 9. TMED 18. DViento

5. Llena los campos correspondientes y selecciona el botón Insertar.	6. Comprueba que no existan campos sin datos.
	7. Comprueba que en la Base de Datos no exista ya una fila con la misma fecha y la misma estación.
	8. Almacena los datos introducidos en la BD, evalúa las reglas de la BC y almacena en la BD los resultados obtenidos.
	9. Notifica al usuario el resultado de la operación.
3. El usuario elige el escenario c)	4. Muestra la ventana Insertar Datos de la Sigatoka con los siguientes campos: 1. Día 2. Mes 3. Año 4. Estación 5. Hora 1 …… 28. Hora 24 29. Evaporación
5. Llena los campos correspondientes y selecciona el botón Insertar.	6. Comprueba que no existan campos sin datos.
	7. Comprueba que en la Base de Datos no exista ya una fila con la misma fecha y la misma estación.
	8. Almacena los datos introducidos en la BD, evalúa las reglas de la BC y almacena en la BD los resultados obtenidos.
	9. Notifica al usuario el resultado de la operación.

3. El usuario elige el escenario g)	4. Muestra la ventana Eliminar Datos con los siguientes atributos: 1. Día 2. Mes 3. Año 4. Estación
5. Llena los campos correspondientes y selecciona el botón Eliminar.	6. Comprueba que no existan campos sin datos.
	7. Comprueba que en la Base de Datos exista una fila con la información que se introdujo.
	8. Elimina de la BD la fila correspondiente.
	9. Notifica al usuario el resultado de la operación.

2.1.2.2 Caso de uso "Predecir Enfermedades"

Caso de Uso:	Predecir Enfermedades
Actor:	Usuario
Propósito:	Predecir cuales enfermedades se van a desarrollar en una zona determinada y mostrar las predicciones realizadas.
Resumen:	El usuario selecciona la opción Predecir Enfermedades y seguidamente elije un escenario.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción Predecir Enfermedades del menú Predicciones.	Muestra un serie de escenarios: a) Iniciar Predicción b) Mostrar todas las Predicciones.

	c) Mostrar Predicciones por Estación.
3. El usuario elige el escenario a).	4. Muestra la ventana Iniciar Predicción que tiene los siguientes campos: 1. Día 2. Mes 3. Año 4. Estación
5. Llena los campos correspondientes y selecciona el botón Iniciar.	6. Comprueba que no existan campos sin datos.
	7. Comprueba que en la Base de Datos exista una fila con la información que se introdujo.
	8. Chequea la información de la BD, necesaria para realizar la predicción.
	9. Almacena los resultados obtenidos en la BD.
	10. Muestra al usuario el resultado de la operación.
3. El usuario elige el escenario c)	4. Muestra la ventana Mostrar Datos por Estaciones que tiene el siguiente campo: 1. Estación
5. El usuario llena el campo y selecciona el botón Mostrar.	6. Comprueba que no existan campos sin datos.
	7. Comprueba que en la BD exista la información introducida.
	8. Selecciona los datos necesarios.
	9. Muestra la ventana Mostrar Predicciones.

2.1.2.3 Caso de uso “Graficar Datos”

Caso de Uso:	Graficar Datos
Actor:	Usuario
Propósito:	Mostrar al usuario gráficamente como se han comportado las variables climatológicas.
Resumen:	El usuario selecciona la opción Análisis Grafico, introduce los datos que se le piden, selecciona el botón Aceptar, se muestra el resultado al usuario.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción Graficar Datos del menú Análisis Grafico.	2. Muestra la ventana Iniciar Predicción con los siguientes atributos: 1. Día 2. Mes 3. Año 4. Estación 5. Variable 6. Días
3. Llena los campos correspondientes y selecciona el botón Aceptar.	4. Comprueba que no existan campos vacios.
	5. Comprueba que existan en la BD una fila con la fecha y la estación introducida.
	6. Muestra al usuario una grafica con el resultado obtenido.

2.1.3 Diagrama de Clases

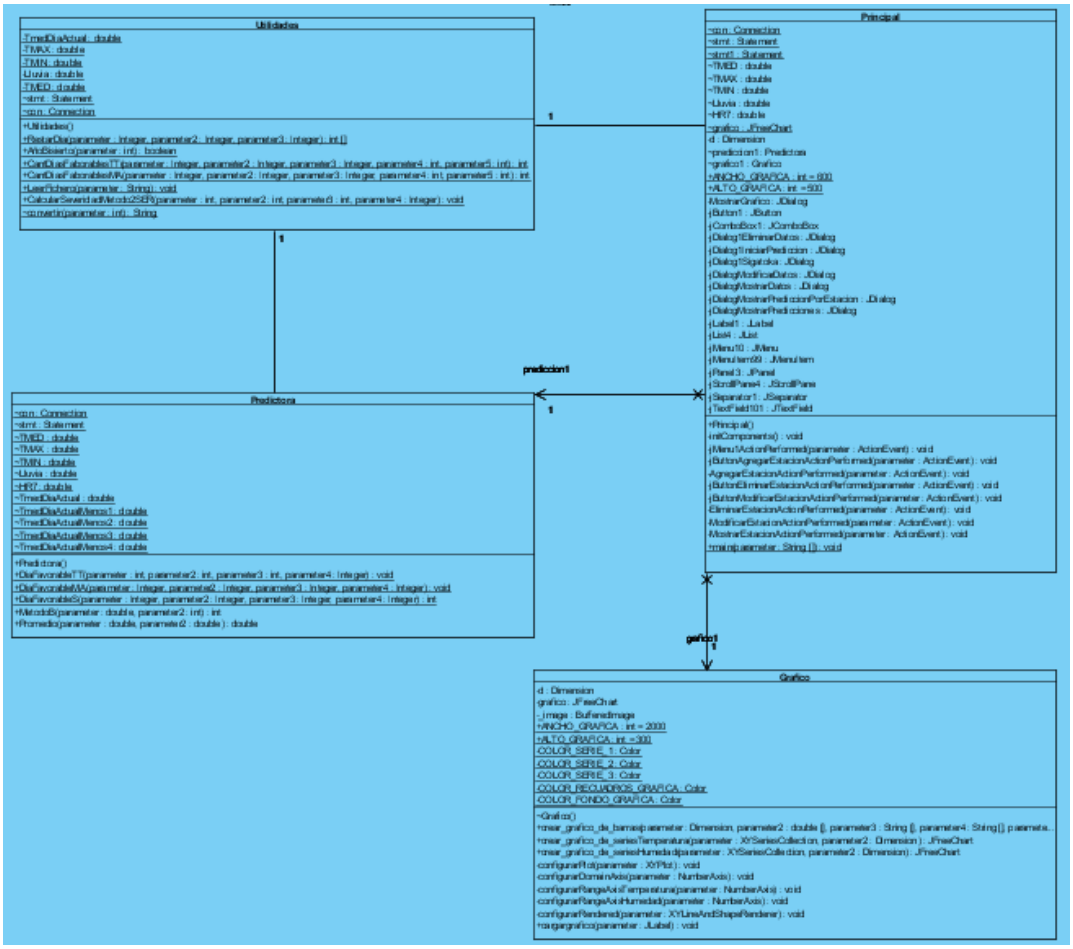


Figura 2.1.3.1Diagrama de Clases.

En el diagrama de clases se muestran cuatro de las clases con que cuenta el sistema, ellas son Principal, Predictora, Utilidades y Grafico. Faltan las clases jDialog que crea el NetBeans para mostrar la parte visual del proyecto.

El sistema usa tecnologías cliente-servidor. Este al ocuparse de la gestión de datos se comporta como un cliente de la base de datos, esta última al actuar como servidor atiende las consultas del Sistema.

La figura 2.1.2 muestra la relación que existe entre las componentes del sistema y los datos que fluyen entre las mismas:

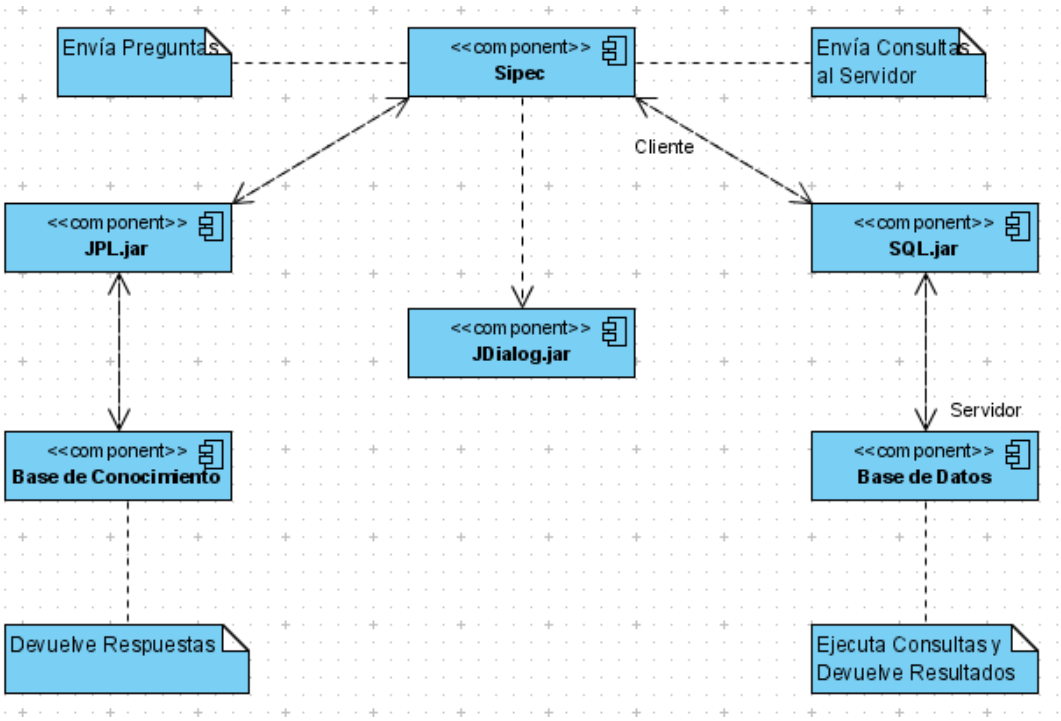


Figura 2.1.2: Diagrama de componentes.

2.2 Gestión de los datos.

El almacenamiento de los datos no solo se limita a los valores meteorológicos recibidos de las distintas estaciones meteorológicas, también hay que tener en cuenta, los datos particulares (nombre, identificador) de cada estación, datos acerca de las enfermedades que se tratan, las predicciones hechas con sus diagnósticos, las fechas de inicio para comenzar a predecir una enfermedad en una estación en particular, así como los mensajes de aplicación emitidos para las distintas estaciones.

Las variables meteorológicas cuyos valores son objeto de almacenamiento en nuestra base de datos son descritas a continuación con el objetivo de que se tenga una idea de la información más relevante tratada.

Nombre	Unidad de Medida	Descripción	Valores Límites	
			Min	Max
tMax	ºC	temperatura máxima	-20	50
tMed	ºC	temperatura media	-20	50
tMin	ºC	temperatura mínima	-20	50
tMaxHR	ºC	temperatura máxima con la humedad relativa por encima del 90%	-20	50
tMedHR	ºC	temperatura media con la humedad relativa por encima del 90%	-20	50
tMinHR	ºC	temperatura mínima con la humedad relativa por encima del 90%	-20	50
hMax	%	humedad relativa máxima	0	100
hMed	%	humedad relativa media	0	100
hMin	%	humedad relativa mínima	0	100
HR	%	humedad relativa a las 7:00 AM	0	100
Horas	Horas y Minutos	cantidad de horas que la humedad relativa estuvo por encima del 90%	0.00	24.00
Lluvia	Mm	cantidad de milímetros de lluvia	0	5000

		caída		
Viento	Km/h	velocidad del viento	1	400
DireccionViento	Cadena de tamaño 3 caracteres como máximo. Ej (ENE)	dirección del viento	-	-

Como resultado del análisis cuidadoso de toda la información a tratar, se desarrolló un modelo que respondiera a las exigencias de dicho análisis. Un primer diseño de la base de datos a construir se obtuvo usando el Modelo Entidad Relación de Peter Chen. A partir del modelo anterior, se desarrollo otro modelo, el relacional o lógico, mostramos las tablas y relaciones más significativas de este último en la figura 2.2:

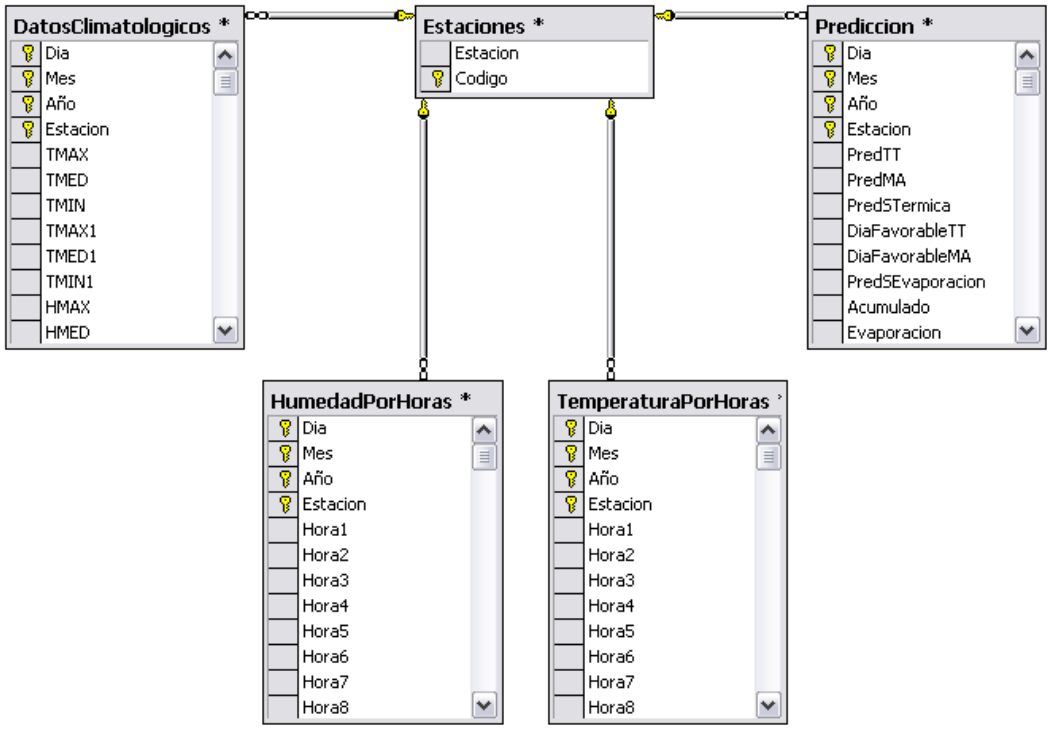


Figura 2.2: Modelo Relacional de la Base de Datos.

Existen numerosas tecnologías donde implementar el modelo anterior con posibilidades de éxito, entre las más destacadas se encuentran Oracle, SQLite, MySQL, MS SQL SERVER, entre otras. Oracle es una tecnología muy fuerte,

entre sus muchas características y facilidades, se puede mencionar la alta seguridad que implementa, lo cual no constituye una gran ventaja para el desarrollo de nuestro sistema. SQLite, es poco utilizada en nuestro país, esto hace que el conocimiento previo sobre la misma sea muy escaso. MySQL es una tecnología de las más usadas a nivel mundial, consta con muchas facilidades, aunque no implementa funciones y procedimientos almacenados, cualidad que ayuda en gran medida a la rapidez en el proceso de gestión de datos, tampoco brinda posibilidades de relaciones entre tablas en sus versiones más usadas. MS SQL SERVER es una tecnología dotada de muchas bondades, las cuales facilitan las exigencias de nuestro problema, a continuación se aborda un poco más sobre esta.

Por sus características, la tecnología escogida para la implantación del modelo relacional fue MS SQL SERVER 2000. Este es un servidor de bases de datos de los más populares actualmente, debido a su gran potencialidad. Brinda muchas posibilidades de gestión de la información y para la administración de las bases de datos que alberga, tanto de los datos que estas contienen como de las operaciones, usuarios, procesos, etc. Brinda servicio de transformación de datos y servicio de gerencia (el cual incluye posibilidad de ver en cualquier momento los procesos que están ejecutándose y datos de estos) entre otros.

Para el manejo de la seguridad de las bases de datos que hospeda, brinda la posibilidad de creación de usuarios. A estos se les puede asignar roles, que son privilegios de seguridad a nivel de servidor, y se le puede dar permisos en cada base de datos por separado, lo que determina las operaciones que puede realizar en cada una de ellas. Permite además, tener usuarios de Windows como usuarios suyos; o sea, realiza el proceso de autentificación de los usuarios verificando contraseñas suyas y de Windows. Como servicios de soporte tiene un Coordinador de Transacciones Distribuidas y un servicio denominado Full Text Search (Búsqueda completa de texto), que posibilita hacer búsquedas con mayores facilidades en las consultas de selección de datos.

Para el trabajo con cada base de datos en específico aporta muchas facilidades. Entre las más relevantes se pueden mencionar la posibilidad de tener funciones y procedimientos almacenados; estos son gestiones de datos, o sea, combinaciones de consultas de selección, inserción, modificación o eliminación en una o más tablas de la base de datos. La gran ventaja de estos procedimientos es que ya se encuentran compilados en el servidor SQL y en el momento de llamarlos desde cualquier aplicación, el tiempo de obtención de los resultados se reduce al tiempo de ejecución del procedimiento almacenado, lo cual no sucede con las consultas creadas y ejecutadas en tiempo real, que necesitan ser compiladas primeramente.

A continuación se enuncian y describen de una forma breve los procedimientos almacenados más relevantes que fueron implementados para garantizar una óptima gestión de los datos.

- `dbo.insert_DatosClimatologicos_1`: inserta valores meteorológicos pasados como parámetros en la tabla DatosMeteorologicos.
- `dbo.insert_Estaciones_1`: inserta una nueva estación, sus atributos son pasados como parámetros.
- `dbo.insert_Prediccion_1`: inserta valores en la tabla Prediccion, los cuales le son pasados como parámetros.
- `dbo.insert_HumedadPorHoras_1`: inserta los valores de la humedad que le son pasados como parámetros.
- `dbo.insert_TemperaturaPorHoras_1`: inserta los valores de la temperatura que le son pasados como parámetros.

2.3 Implementación del Sistema.

Para la construcción del Sistema, se analizó primeramente toda la bibliografía explicada en el Capítulo 1. Posteriormente se pasó a escoger las tecnologías para implementar el mismo, el lenguaje Prolog para describir las reglas, o sea, para crear la base de conocimientos.

Para escribir la base de conocimientos en lenguaje Prolog se usó particularmente el Amzi Prolog, ya que este ambiente se torna cómodo y eficaz para los propósitos perseguidos.

El sistema experto no solo se ocupa de realizar los diagnósticos, sino también han sido implementados otros módulos para apoyar la gestión y visualización de datos y predicciones en un ambiente favorable para el usuario, para la realización de estos módulos se utilizó el NetBeans 6.0 y el lenguaje de programación fue el Java.

Después de haber desarrollado la implementación de los requerimientos del sistema, se obtuvo como producto final el SIPEC.jar, este brinda solución al problema inicialmente planteado, posibilita el uso de otros servicios que se fueron incorporando con el avance y desarrollo del mismo.

Para modelar el problema se usó el Modelo Orientado a Objetos (MOO), este brinda muchas bondades de incorporación de nuevas entidades en el diseño inicial, así como modularidad y claridad del modelo en cuestión.
A continuación se muestran los distintos módulos implementados para el funcionamiento óptimo de SIPEC.

2.3.1 Módulo para el Diagnóstico de Enfermedades por Razonamiento Basado en Reglas.

Para la implementación de este módulo se crearon distintas clases que apoyan el proceso de diagnóstico. Este proceso debe de realizarse de forma rápida y precisa, ya que estas son algunas de las dificultades que poseía el mismo antes de que se pensara en un Sistema que realizara dicha tarea. La relación que existe entre las mismas es mostrada en la figura 2.3.1.1:

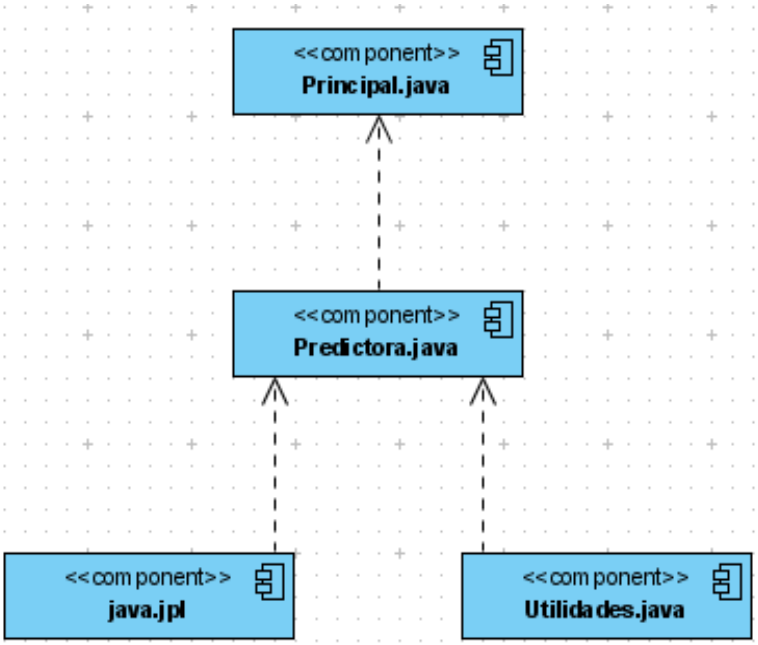


Figura 2.3.1.1: Relación Entre Clases del Módulo de Diagnóstico.

Como puede observarse la clase Principal.java utiliza a Prediccion.java, la cual a su vez usa a java.jpl y a MetodosNecesarios.java.

El módulo realiza las predicciones a través de las funcionalidades implementadas en Principal.java, básicamente lo que se hace es utilizar la biblioteca Java.sql para extraer de la base de datos los valores meteorológicos de cada una de las estaciones meteorológicas respectivamente, para esto es usada la clase Prediccion.java que es la encargada de conectarse con la base de datos y realizarle consultas, esta misma clase evalúa los valores obtenidos de la base de datos en las reglas descritas en la base de conocimientos apoyándose en la biblioteca JPL para realizar dicha tarea. La clase Prediccion.java ofrece objetos para tomar los valores meteorológicos y mandarlos a evaluar, si el resultado de la evaluación es verdadero estamos en presencia de una predicción positiva de alguna enfermedad, entonces se puede mandar a almacenar este resultado, posteriormente se siguen analizando los juegos de datos restantes. Si al terminar el proceso de

predicción no se ha hecho ninguna inferencia, el sistema emite un mensaje declarando esta situación.

El diagrama de actividad que se muestra en la figura 2.3.1.2 brinda una descripción más detallada del proceso de predicción.

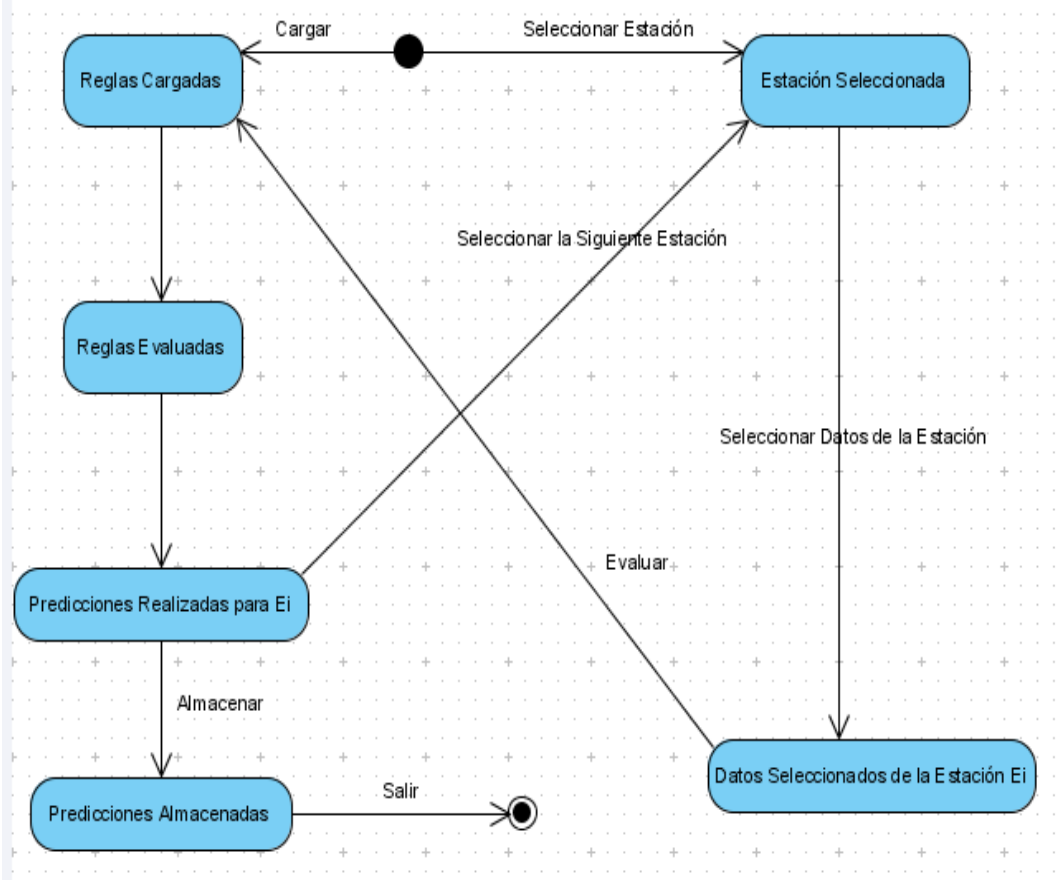


Figura 2.3.1.2: Diagrama Actividad del Proceso de Diagnóstico

2.3.2 Módulo de Visualización de Datos.

Este módulo está creado para brindar al usuario una información detallada relacionada con las predicciones realizadas y con algunos de los datos de la base de datos.

Esta información está descrita principalmente por

- en qué zonas se realizarán los ataques.

- predicciones realizadas.
- los datos de las estaciones que se encuentran almacenadas
- los datos climatológicos

Es importante destacar que la información que se expone en este módulo es de gran importancia para el usuario, no solo porque visualiza las predicciones que se han realizado en la temporada actual para que se tomen las decisiones pertinentes de como combatirlas; también se muestra un resumen de todas las predicciones realizadas a lo largo de la historia para una estación determinada, lo cual da una medida de cuan factible es cultivar una variedad determinada en una zona en particular, o sea, que riesgos se corren al sembrar dicha variedad y que ganancias se pueden obtener al ahorrar productos químicos y biológicos aplicables, así como mano de obra y medios de producción.

La figura 2.3.2 muestra las clases implementadas para este propósito y la relación que existe entre ellas.

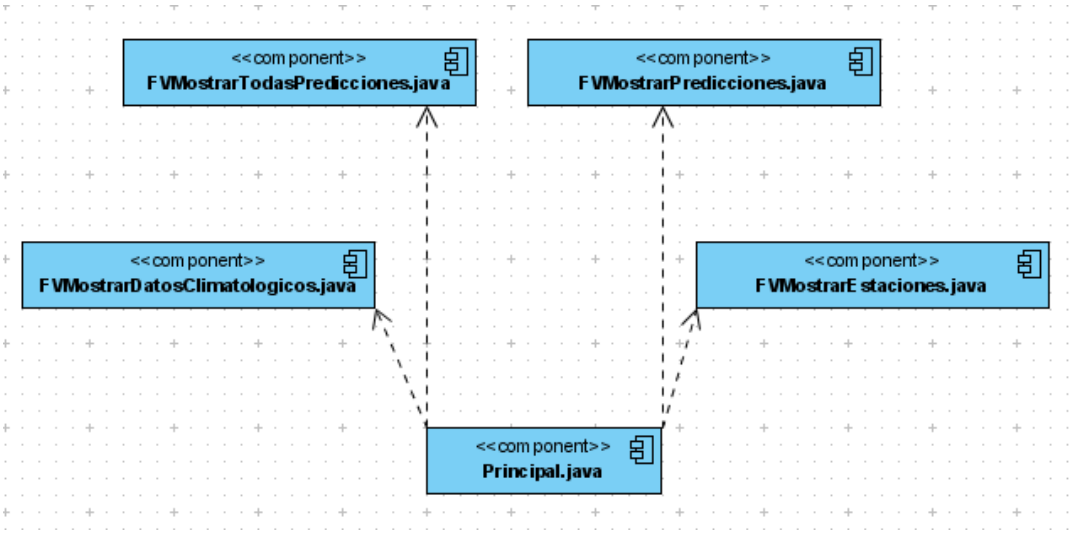


Figura 2.3.2: Relación Entre los Objetos del Módulo de Visualización.

MostrarTodasPredicciones.java y MostrarPrediccionesEst.java muestran visualmente las predicciones realizadas ordenadas descendentemente por fecha de predicción y las predicciones realizadas para una estación determinada también ordenada por fecha de predicción respectivamente. Para esto se auxilian de Principal.java que es la clase que posee las implementaciones relacionadas con el acceso y consultas a la base de datos.

2.3.3 Gestión de la Base de Conocimientos.

Después de un período de revisión de literatura especializada y de varias entrevistas con el experto en Agrometeorología se realizó un prototipo del sistema de reglas, este fue revisado en reiteradas ocasiones por el experto hasta llegar a una primera versión.

Las reglas han sido escritas en lenguaje Prolog y son analizadas por la máquina de inferencia (MI), la cual las carga y libera de memoria tantas veces sea necesario con el objetivo de optimizar esta última.

Según lo planteado en la bibliografía especializada en Agrometeorología; se dice que un “sistema” se ha cumplido, si se cumplen todas las reglas que describen al mismo durante un periodo de tiempo t (en días); y se dice que una “enfermedad” está lista para afectar a un cultivo x, si se cumple al menos uno de los sistemas que la describen.

Las clases Predictora y Utilidades son las encargadas de conectarse con la Base de Conocimientos y realizar la evaluación de las reglas. La primera clase a la que se hace referencia cuenta con tres métodos que son los encargados de evaluar las reglas que predicen el tizón tardío, el moho azul y la sigatoka. Dentro de la clase Utilidades se encuentra un método que evalúa el desarrollo del tizón tardío, controlando una regla diferente a la que controla la clase Predictora para la misma enfermedad.

La BC es un conjunto de reglas escritas en lenguaje Prolog. Las mismas son escritas siguiendo los patrones siguientes:

Las reglas de este tipo describen las enfermedades y plagas.

r1(NE, NM,GC,V1,V2, **. . .** , Vm) :- R1, R2, …, Rn.

Donde:

NE → Identifica el nombre de la enfermedad a la que contrala esta regla.
NM → Identifica el nombre del método a utilizar.
GC → Identifica el grado de las condiciones climatológicas para que se desarrolle la enfermedad que describe la regla.
V1 → Primera variable meteorológica.
V2 → Segunda variable meteorológica.
……
Vm → m-ésima variable meteorológica.
R1 → Primera regla que describe al sistema NM.
R2 → Segunda regla que describe al sistema NM.
….
Rn → n-ésima regla que describe al sistema NM.

Para ver un ejemplo de la BC remitirse al Anexo 1

2.3.4 Módulo de Gestión de Datos y Estaciones Meteorológicas.

Este módulo está encaminado a gestionar tanto los datos como las estaciones meteorológicas. Los datos llegan de las distintas estaciones diariamente, estos deben ser almacenados en la base de datos para ser procesados y mostrados posteriormente. A las estaciones debe de dársele un tratamiento especial también, pues estas pueden salir de servicio, modificar sus atributos, pueden

incorporarse otras nuevas, etc. Para esto se auxilia de varias clases, la relación existente entre las mismas se describe a continuación mediante la figura 2.3.5:

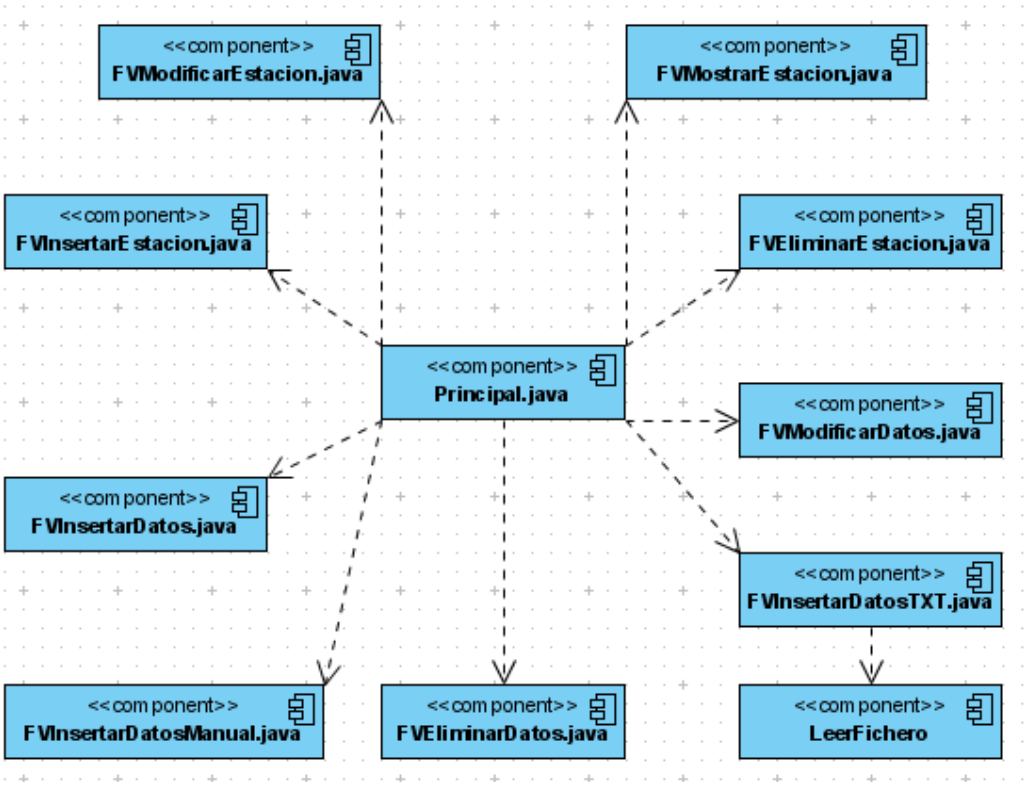


Figura 2.3.5: Relación Entre Clases del Módulo de Gestión de Datos y de Estaciones Meteorológicas.

Como se puede observar Principal.java utiliza a todas las clases, ella es la encargada de mandar a ejecutar los respectivos procedimientos almacenados y métodos que se encuentran en el servidor de base de datos y en la misma clase, los mismos realizan las operaciones que describen los nombres de los métodos.

En el caso particular de FVInsertarDatosTXT.java, usa a LeerFichero por las funcionalidades que el mismo brinda para procesar texto.

2.4 Diagrama de despliegue.

Los diagramas de despliegue muestran las relaciones físicas entre los distintos nodos que componen un sistema y la distribución de los componentes sobre

dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación.

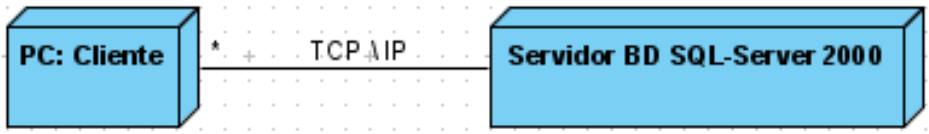


Figura 2.3.6: Diagrama de Despliegue.

2.5 Configuración de las herramientas para el desarrollador.

El sistema se conecta a un servidor de base de datos SQL-Server 2000 y para evaluar la base de conocimiento se conecta al SWI-Prolog, para realizar estas conexiones se deben hacer las modificaciones descritas en los epígrafes 2.5.1 y 2.5.2.

Para poder modificar el sistema se debe agregar al proyecto las bibliotecas necesarias para que se reconozcan todos los métodos que este utiliza. Estas bibliotecas son las siguientes:

- dnastructure-1.2.jar
- epsgraphics-1.0.0.jar
- itext-1.4.3.jar
- jcommon-1.0.9.jar
- jfreechart-1.0.5.jar
- toolkit-1.0.jar

Estas bibliotecas son para el trabajo con los gráficos.

- jpl.jar ——— Esta biblioteca es para el trabajo con prolog.
- liquidlnf.jar ——— Esta biblioteca es modificar la parte visual del proyecto.

2.5.1 Conexión de “SIPEC.jar” con la Base de Datos

1 – Crear el ODBC

- Panel de Control
- Herramientas Administrativas
- Orígenes de Datos (ODBC)
- Agregar

- SQL Server
- Finalizar
- Nombre: leo
- Servidor: <nombre del servidor de base de datos>
- Siguiente
- Siguiente
- Establecer la siguiente base de datos como predeterminada (Nombre de la base de datos que se utilizará)
- Siguiente
- Finalizar

2.5.2 Conexión de "SIPEC.jar" el Prolog

1 - Configurar las rutas del SWI-Prolog

- Propiedades de Mi PC
- Opciones avanzadas
- Variables de entorno

- Agregar variables

Nombre: CLASSPATH

Valor: C:\Program Files\Java\jre1.6.0\lib;C:\Program Files\pl\lib

---

Nombre: LIB

Valor: C:\Program Files\pl\bin;C:\Program Files\pl\lib

Nombre: PATH

Valor: C:\Program Files\pl\bin;C:\Program Files\pl\lib;C:\Program Files\Java\jre1.6.0\bin

Donde:

C:\Program Files\pl -> Es la ruta donde se instaló el SWI-Prolog.

C:\Program Files\Java\jre1.6.0 -> Es la ruta donde se instaló el JRE

Capítulo 3: "Manual de Usuario".

En este capítulo se explican los requerimientos de hardware, instalación, y uso óptimo de los componentes del sistema. (Base de datos, SIPEC.jar, Prolog).

3.1 Requerimientos e Instalación de los Componentes del Sistema.

Para instalar cada uno de los componentes del sistema se tienen que tener en cuenta cada uno de los requisitos específicos de cada uno de ellos por separado, ya que los mismos no necesariamente tienen que estar instalados en la misma computadora.

3.1.1 Instalación de la "Base de Datos".

Para instalar la Base de Datos y lograr un correcto funcionamiento de la misma se deben seguir los siguientes pasos:

1) Tener un servidor de Base de Datos MS SQL SERVER 2000.
2) Crear una nueva Base de Datos en el Servidor.
3) Restaurar la Base de Datos creada en el paso 2) usando el fichero "BACKUP RESTAURED" que se encuentra en la carpeta "SIPEC" de la instalación.

3.1.2 Requerimientos e Instalación del "SIPEC.jar"

Antes de pasar a la instalación del SIPEC.exe, se deben revisar los siguientes requisitos.

- Procesador Pentium o superior.
- Al menos 256 Megabytes de memoria.
- Versión Windows XP.
- Máquina Virtual de Java jre-6u16-windows o superior.
- SWI-Prolog 5.4
- Microsoft SQL Server 2000

Si ya se han verificado los requerimientos anteriores, se deben crear las conexiones necesarias con los componentes externos del sistema, sean estos

la base de datos y el prolog, para esto se debe seguir las indicaciones de los epígrafes 2.5.1 y 2.5.2. Después de hecho esto entonces se puede pasar a la sencilla instalación del SIPEC.jar.

3.2 Manejo Óptimo del "SIPEC.exe".

Al ejecutar el software, rápidamente se está en presencia de su portada inicial, esta es mostrada en la figura 3.2.1:

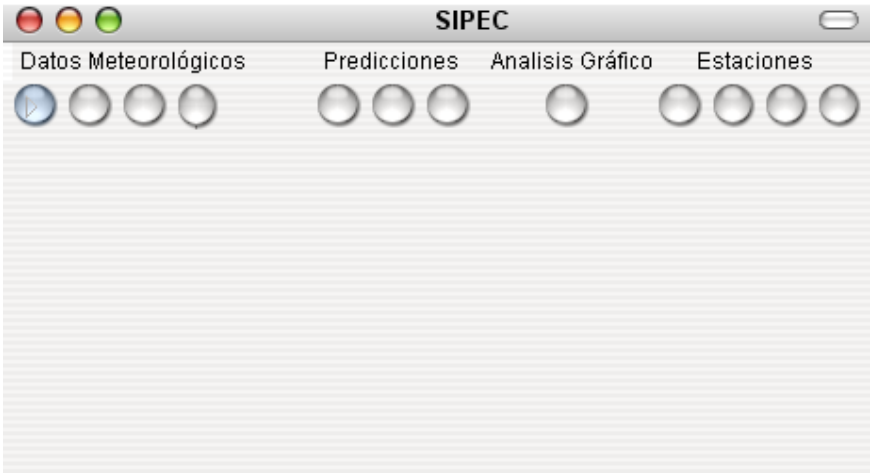


Figura 3.2.1: Portada del SIPEC.jar

A continuación se describen por separado cada una de las interfaces sobre las cuales se realizan las operaciones con el software.

3.2.1 Datos Meteorológicos.

3.2.1.1 Insertar Datos Manualmente.

Esta opción permite que el usuario del sistema incluya nuevos datos en la BD.
Acceso:

1) Desde el menú.

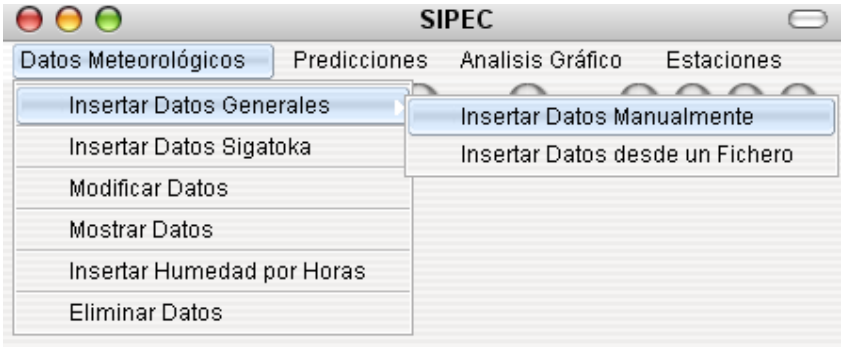


Figura 3.2.1.1.1: Acceso a la operación Insertar Datos Manualmente desde el menú.

2) Desde el botón de acceso rápido.

Figura 3.2.1.1.2: Acceso a la operación Insertar Datos Manualmente desde botón de acceso rápido.

Ventana Operativa:

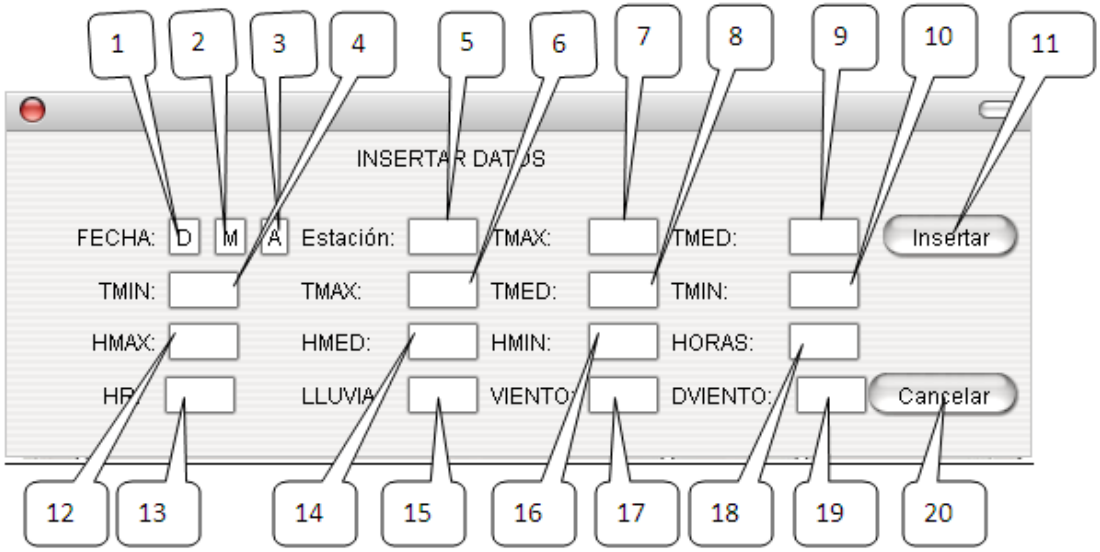


Figura 3.2.1.1.3: Descripción de la ventana Insertar Datos.

1) Insertar el día al que pertenecen los datos climatológicos que se insertarán.
2) Insertar el mes al que pertenecen los datos climatológicos que se insertarán.
3) Insertar el año al que pertenecen los datos climatológicos que se insertarán.
4) Insertar la temperatura mínima registrada.
5) Insertar el código de la estación a la cual pertenecen los datos que se insertarán.
6) Insertar la temperatura máxima registrada mientras la humedad estuvo por encima del 90%.
7) Insertar la temperatura máxima registrada.
8) Insertar la temperatura media registrada mientras la humedad estuvo por encima del 90%.
9) Insertar la temperatura media registrada.
10) Insertar la temperatura mínima registrada mientras la humedad estuvo por encima del 90%.
11) Botón para insertar los datos en la BD.
12) Insertar la humedad máxima registrada.
13) Insertar la humedad relativa.
14) Insertar la humedad media registrada.
15) Insertar los milímetros de lluvia registrada.
16) Insertar la humedad mínima registrada.
17) Insertar la velocidad del viento.
18) Insertar las horas que se mantuvo la humedad por encima del 90%.
19) Insertar la dirección de viento.
20) Botón Cancelar que regresa a la ventana principal.

3.2.1.2 Insertar Datos desde un Fichero.txt.

Esta opción permite que el usuario del sistema incluya nuevos datos en la BD.

El fichero que se añada debe tener el siguiente formato:

```
05/03/2012
estac  tmax tmed tmin tmax tmed tmin hmax hmed hmin horas HR lluvia  vto  drc
78308  31.0 24.7 19.0 20.8 19.9 19.0 91   66   32   02.00 89  03.0   3.5  E
78326  33.5 24.8 16.7 20.0 18.9 17.8 94   75   42   07.10 94  20.0   4.6  SW
78338  33.0 24.4 16.6 18.5 17.6 16.6 93   73   32   06.30 93  04.0   4.1  v
78343  32.6 25.4 18.2 21.0 20.0 18.9 99   67   27   06.30 98  30.5   7.8  SE
78348  29.4 24.7 20.0 22.7 21.4 20.0 93   77   53   05.30 91  17.0   1.0  V-C
```

Acceso:

3) Desde el menú.

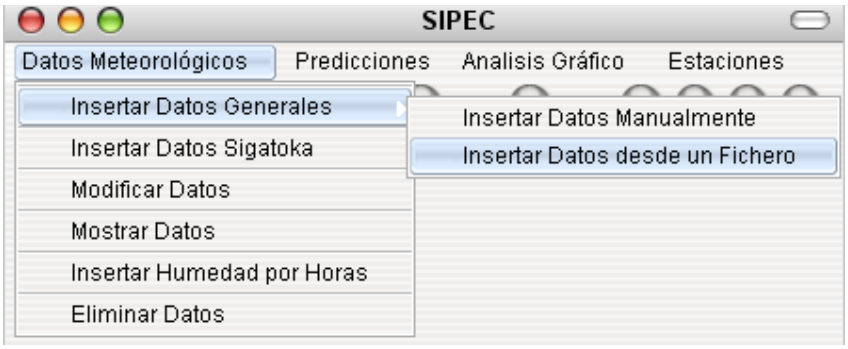


Figura 3.2.1.1.1: Acceso a la operación Insertar Datos desde un Fichero.txt desde el menú.

Ventana Operativa:

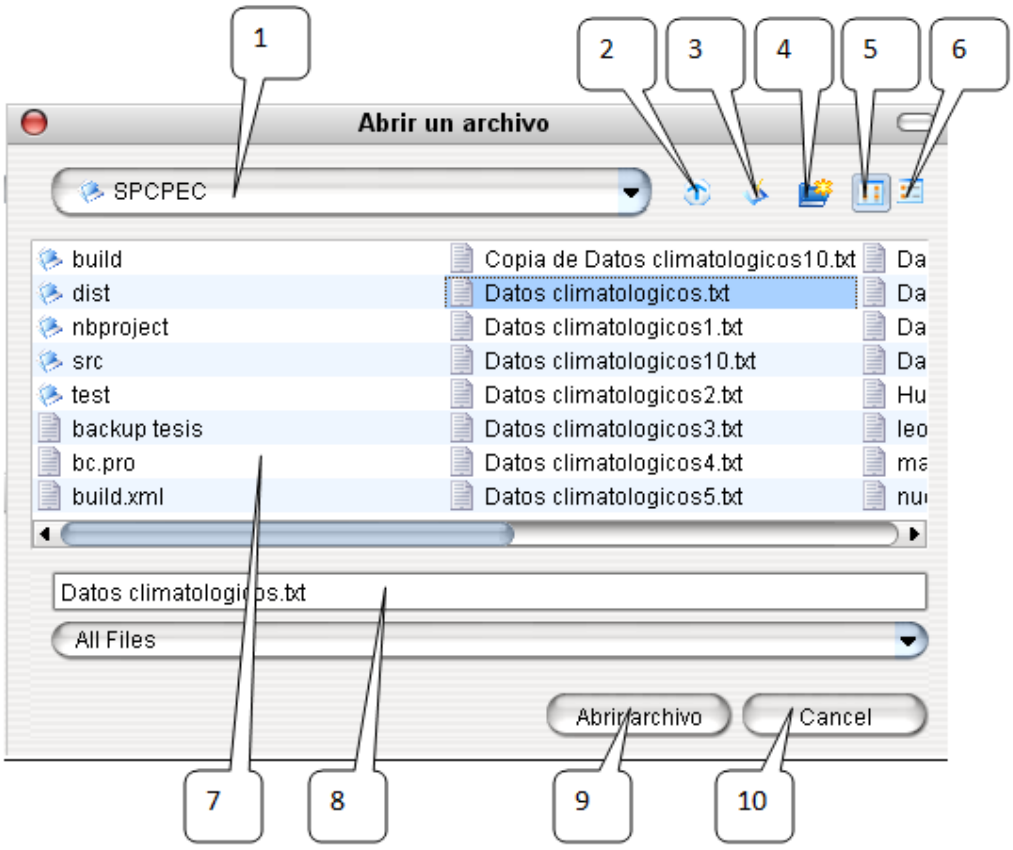


Figura 3.2.1.1.3: Descripción de la ventana Insertar Datos.

1. Para buscar la localización del fichero texto en la máquina.
2. Botón para regresar a la carpeta anterior.
3. Botón para ir al Escritorio directamente.
4. Botón para crear una nueva carpeta en el directorio actual.
5. Botón para cambiar la visualización del contenido del archivo (Lista).
6. Botón para cambiar la visualización del contenido del archivo (Detalles).
7. Área donde se visualiza el contenido de los archivos y donde se selecciona el fichero buscado.
8. Muestra el nombre del fichero seleccionado
9. Botón para abrir el fichero seleccionado.
10. Botón para cancelar la acción.

3.2.1.3 Insertar Datos de la Sigatoka

Esta opción permite insertar los datos climatológicos necesarios para predecir esta enfermedad.

Acceso:

1) Desde el menú.

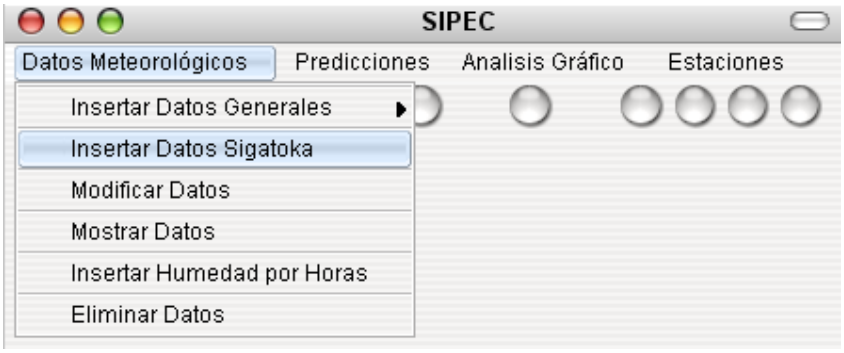


Figura 3.2.1.3.1: Acceso a la operación Insertar Datos de la Sigatoka desde el menú.

2) Desde el botón de acceso rápido.

Figura 3.2.1.3.2: Acceso a la operación Insertar Datos de la Sigatoka desde botón de acceso rápido.

Ventana Operativa:

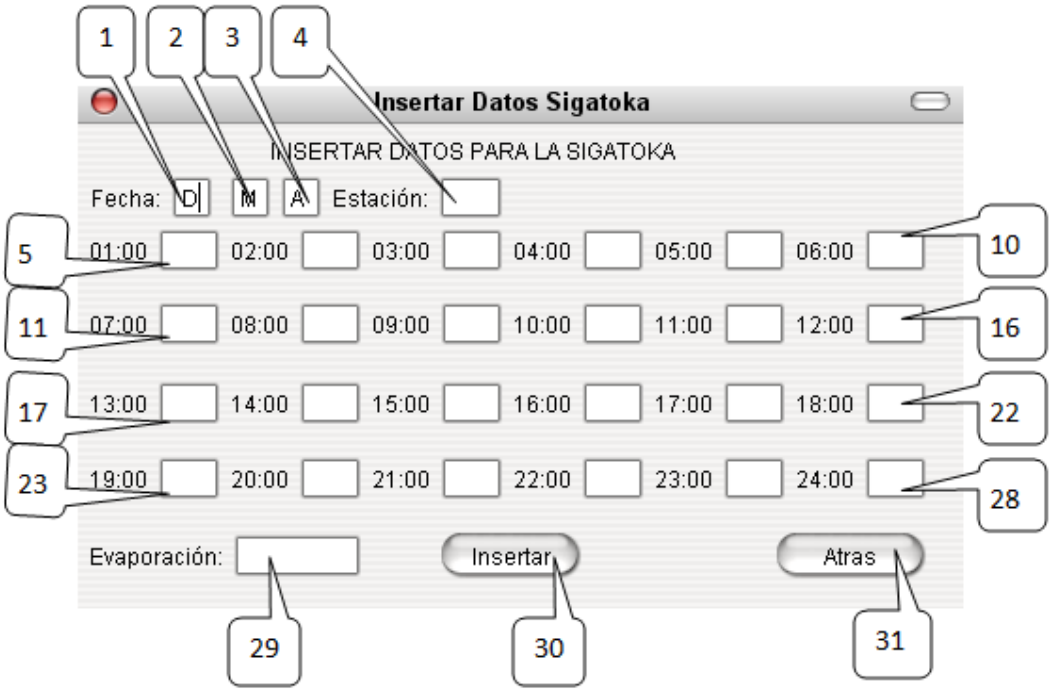


Figura 3.2.1.3.3: Descripción de la ventana Insertar Datos Sigatoka.

1. Insertar el día al que pertenecen los datos que se insertarán.
2. Insertar el mes al que pertenecen los datos que se insertarán.
3. Insertar el año al que pertenecen los datos que se insertarán.
4. Insertar la estación a la que pertenecen los datos que se insertarán.
5. Insertar el valor de la primera hora del día.
6. Insertar el valor de la segunda hora del día.
7. Insertar el valor de la tercera hora del día.
8. Insertar el valor de la cuarta hora del día.
9. Insertar el valor de la quinta hora del día.
10. Insertar el valor de la sexta hora del día.
11. Insertar el valor de la séptima hora del día.
12. Insertar el valor de la novena hora del día.
13. Insertar el valor de la décima hora del día.
14. Insertar el valor de la oncena hora del día.
15. Insertar el valor de la duodécima hora del día.
16. Insertar el valor de la decimotercera hora del día.
17. Insertar el valor de la decimocuarta hora del día.

18. Insertar el valor de la decimoquinta hora del día.
19. Insertar el valor de la decimosexta hora del día.
20. Insertar el valor de la decimoséptima hora del día.
21. Insertar el valor de la decimoctava hora del día.
22. Insertar el valor de la decimonovena hora del día.
23. Insertar el valor de la vigésima hora del día.
24. Insertar el valor de la vigesimoprimera hora del día.
25. Insertar el valor de la vigesimosegunda hora del día.
26. Insertar el valor de la vigesimotercera hora del día.
27. Insertar el valor de la vigesimocuarta hora del día.
28. Insertar el valor de la evaporación.
29. Botón para insertar los valores en la Base de Datos.
30. Botón para regresar a la ventana principal.

3.2.1.4 Modificar Datos.

Esta opción permite que el usuario del sistema modifique los datos de la BD.

Acceso:

1) Desde el menú.

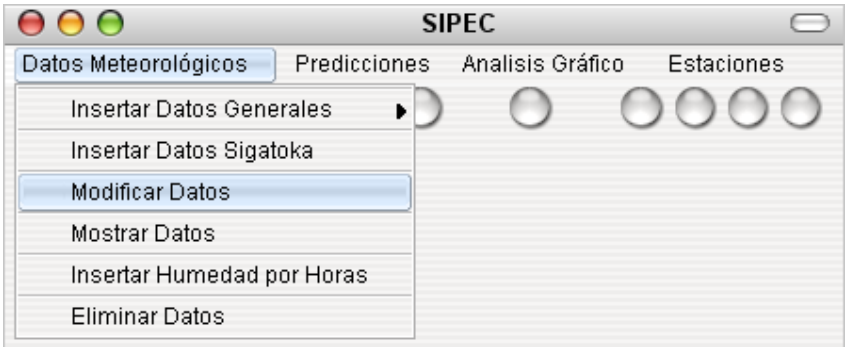


Figura 3.2.1.4.1: Acceso a la operación Modificar Datos desde el menú.

2) Desde el botón de acceso rápido.

Figura 3.2.1.4.2: Acceso a la operación Modificar Datos desde botón de acceso rápido.

Ventana Operativa:

Figura 3.2.1.4.3: Descripción de la ventana Modificar Datos.

Son los mismos componentes que en la ventana Insertar Datos, para que se activen las cajas de textos se debe insertar el día, luego el mes y seguidamente el año. Se deben poner todos los valores, porque el botón Modificar sustituirá en la BD toda la fila que tenga la misma fecha y estación por lo nuevos valores.

3.2.1.5 Mostrar Datos.

Esta opción permite que el usuario del sistema pueda ver los datos de la BD.

Acceso:

1) Desde el menú.

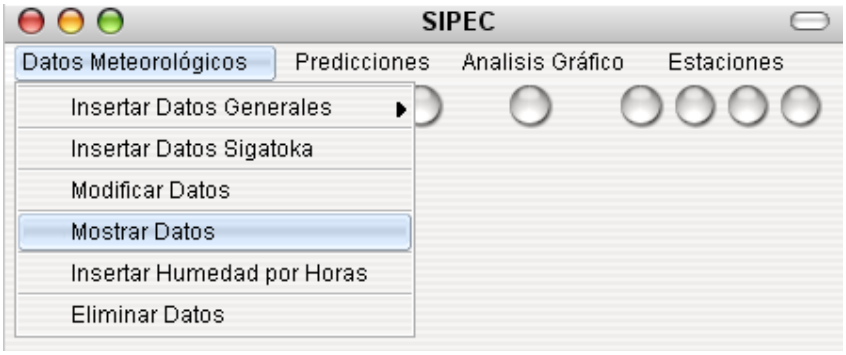


Figura 3.2.1.5.1: Acceso a la operación Modificar Datos desde el menú.

2) Desde el botón de acceso rápido.

Figura 3.2.1.5.2: Acceso a la operación Mostrar Datos desde botón de acceso rápido.

Ventana Operativa:

Figura 3.2.1.5.3: Descripción de la ventana Mostrar Datos.

Se insertarán el día, el mes, el año y la estación de la cual se quiera ver los datos y estos se mostrarán en las mismas posiciones que indica los textos.

3.2.1.6 Insertar Humedad por Horas.

Esta opción permite que se inserten los datos de la humedad cada una hora en la BD.

Acceso:

1) Desde el menú.

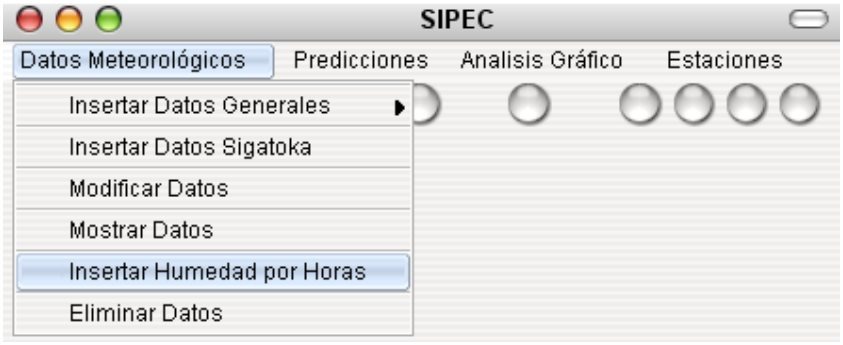


Figura 3.2.1.6.1: Acceso a la operación Insertar datos de la HR desde el menú.

Ventana Operativa:

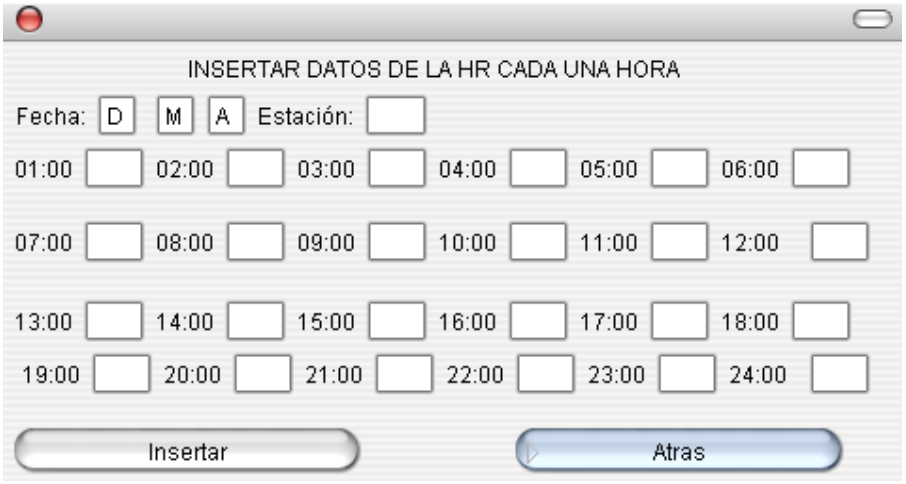


Figura 3.2.1.6.2: Descripción de la ventana Insertar datos de la HR.

Esta ventana tiene las mismas opciones que la de Insertar Datos de la Sigatoka.

3.2.1.7 Eliminar Datos.

Esta opción permite que se eliminen de la BD los datos meteorológicos de un día en una estación.

Acceso:

1) Desde el menú.

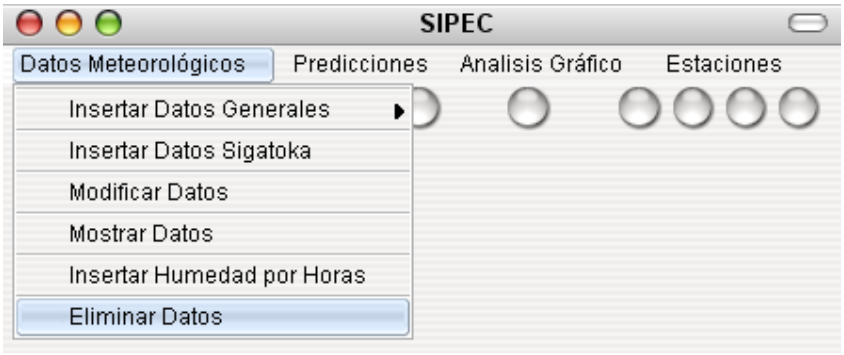


Figura 3.2.1.7.1: Acceso a la operación Eliminar Datos desde el menú.

Ventana Operativa:

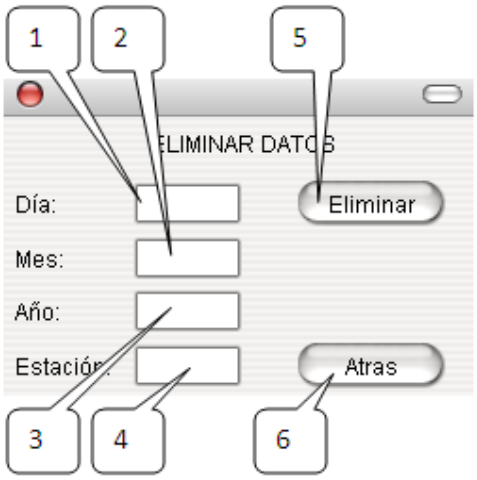


Figura 3.2.1.7.2: Descripción de la ventana Eliminar Datos.

3.2.2 Predicciones.

3.2.2.1 Iniciar Predicción.

Esta opción permite iniciar el pronóstico de las enfermedades en la estación seleccionada.

Acceso:

1) Desde el menú.

Figura 3.2.2.1.1: Acceso a la operación Iniciar Predicción desde el menú.

2) Desde del botón de acceso rápido.

Figura 3.2.2.1.2: Acceso a la operación Iniciar Predicción desde botón de acceso rápido.

Ventana Operativa:

Figura 3.2.2.1.3: Descripción de la ventana Iniciar Predicción.

1) Insertar día en el que se quiere iniciar la predicción.
2) Insertar mes en el que se quiere iniciar la predicción.
3) Insertar año en el que se quiere iniciar la predicción.
4) Insertar código de la estación en el que se quiere iniciar la predicción.
5) Botón que inicia la predicción.

6) Botón que regresa a la ventana principal.

3.2.2.2 Mostrar todas las Predicciones.

Esta opción permite mostrar las predicciones hechas hasta el momento.

Acceso:

1) Desde el menú.

Figura 3.2.2.2.1: Acceso a la operación Mostrar todas las Predicciones desde el menú.

2) Desde del botón de acceso rápido.

Figura 3.2.2.2.2: Acceso a la operación Mostrar todas las Predicciones desde el botón de acceso rápido.

Ventana Operativa:

Figura 3.2.2.2.3: Descripción de la ventana Mostrar todas las Predicciones.

En la ventana se muestran cinco columnas con la Fecha, Estación, Tizón Tardío, Moho Azul y Sigatoka que dejan ver las predicciones realizadas hasta el momento.

3.2.2.3 Mostrar las Predicciones por Estaciones.

Esta opción permite mostrar las predicciones realizadas en una estación específica.

Acceso:

1)-Desde el menú.

Figura 3.2.2.3.1: Acceso a la operación Mostrar por Estación desde el menú.

2) Desde del botón de acceso rápido.

Figura 3.2.2.3.2: Acceso a la operación Mostrar por Estación desde el botón de acceso rápido.

Ventana Operativa:

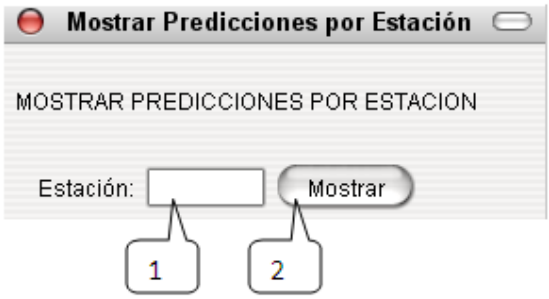
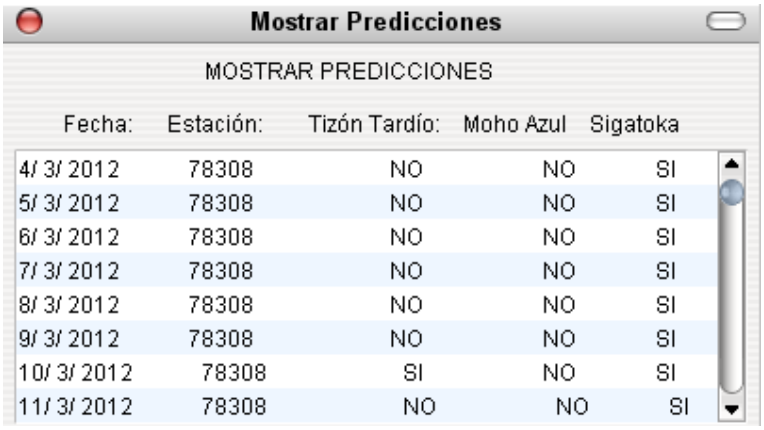


Figura 3.2.2.3.3: Descripción de la ventana Mostrar por Estación.

1) Insertar la estación.
2) Para que el botón se habilite se debe insertar el código de la estación primero, al presionarlo se visualizar las predicciones realizadas en dicha estación.

Ventana donde se muestran las predicciones



Fecha:	Estación:	Tizón Tardío:	Moho Azul	Sigatoka
4/ 3/ 2012	78308	NO	NO	SI
5/ 3/ 2012	78308	NO	NO	SI
6/ 3/ 2012	78308	NO	NO	SI
7/ 3/ 2012	78308	NO	NO	SI
8/ 3/ 2012	78308	NO	NO	SI
9/ 3/ 2012	78308	NO	NO	SI
10/ 3/ 2012	78308	SI	NO	SI
11/ 3/ 2012	78308	NO	NO	SI

Figura 3.2.2.3.4: Descripción de la ventana Mostrar Predicciones.

3.2.3 Análisis Gráfico.

3.2.3.1 Análisis Gráfico.

Esta opción permite visualizar gráficamente como se han comportado las variables meteorológicas (temperatura, lluvia y humedad) en un tiempo determinado en una estación especificada por el usuario.

Acceso:

1) Desde del menú.

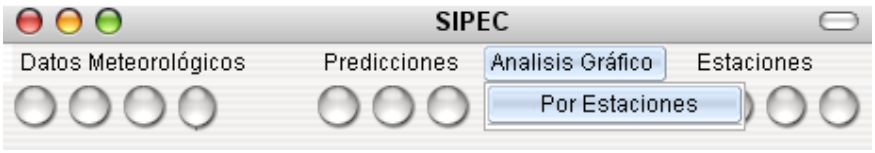


Figura 3.2.3.1.1: Acceso a la operación Análisis Gráfico por Estaciones desde el menú.

2) Desde del botón de acceso rápido.

Figura 3.2.3.1.2: Acceso a la operación Análisis Gráfico por Estaciones desde botón de acceso rápido.

Ventana Operativa:

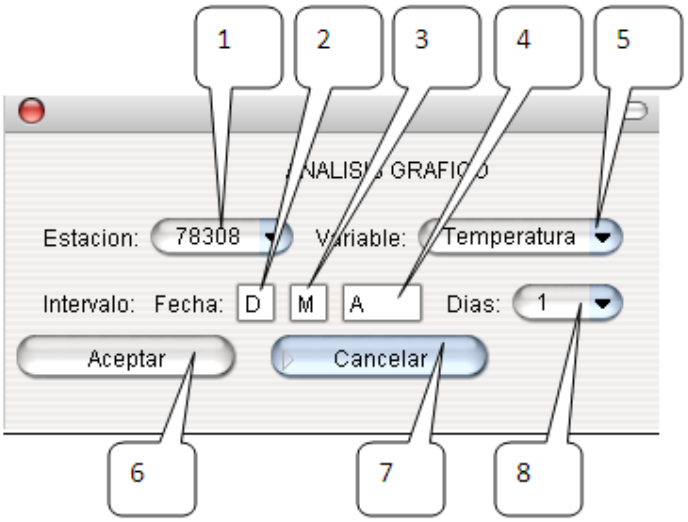


Figura 3.2.3.1.3: Descripción de la ventana Análisis Gráfico por Estaciones.

1) En este combo se elige la estación que se quiere analizar.
2) Se inserta el día final en que se analizará la variable.
3) Se inserta el mes final en que se analizará la variable.
4) Se inserta el año final en que se analizará la variable.
5) En este combo se elige la variable que se quiere analizar.
6) El botón Aceptar es el que muestra el análisis grafico.
7) El botón Cancelar muestra nuevamente la ventana inicial.
8) En este combo se elige la cantidad de días que se quiere realizar el análisis.

Ventana de visualización del grafico

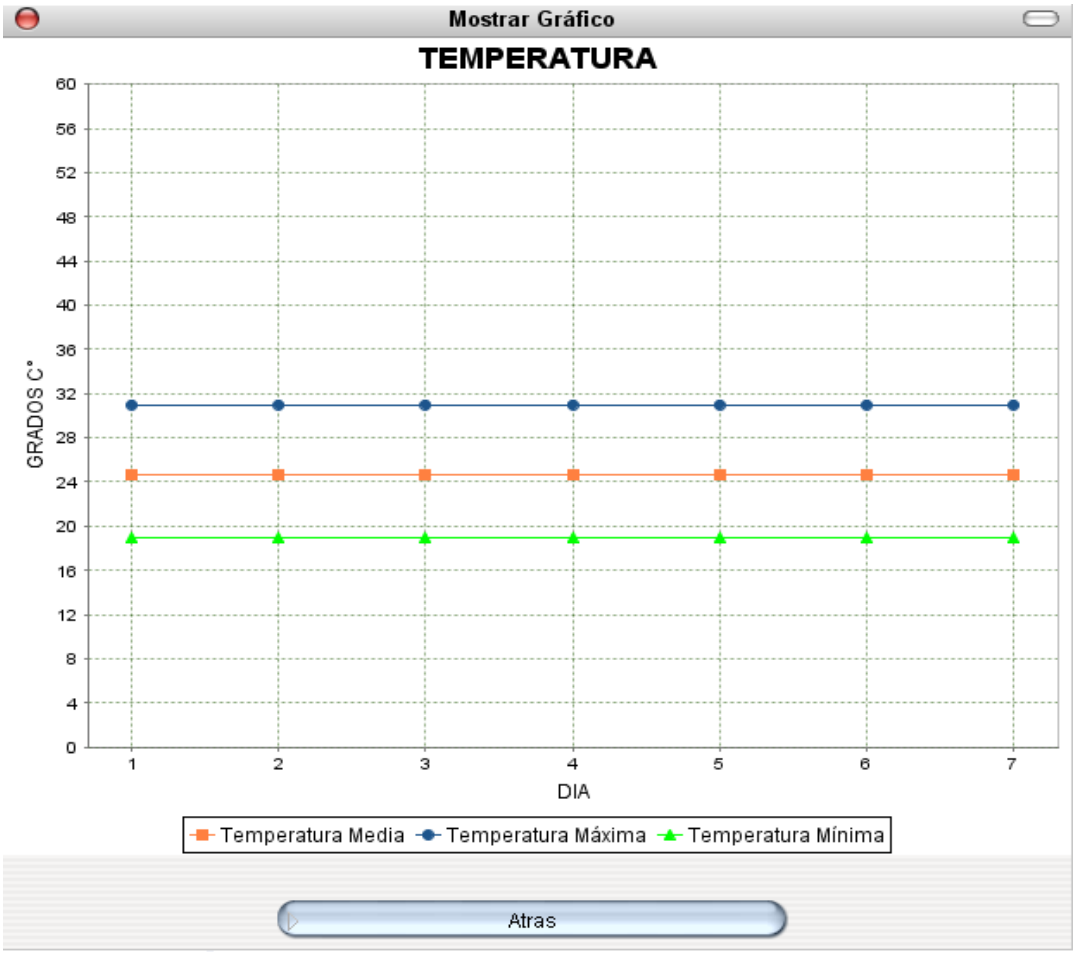


Figura 3.2.3.1.4: Ventana donde se muestra el grafico obtenido.

3.2.4 Estaciones.

3.2.4.1 Agregar

Esta opción permite introducir nuevas estaciones en la BD.

Acceso:

1) Desde menú.

Figura 3.2.4.1.1: Acceso a la operación Agregar una estación desde el menú.

2) Desde del botón de acceso rápido.

Figura 3.2.4.1.2: Acceso a la operación Agregar una estación desde el botón de acceso rápido.

Ventana Operativa:

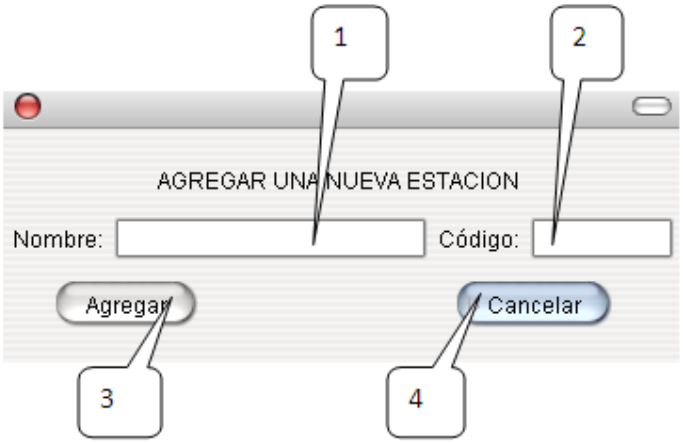


Figura 3.2.4.1.3: Descripción de la ventana Agregar una nueva estación.

1) Introducir el Nombre de la Estación.
2) Introducir el Código de la Estación.
3) Este botón agrega una nueva estación en la BD.
4) Este botón muestra nuevamente la ventana principal.

3.2.4.2 Eliminar

Esta opción permite eliminar una estación de la Base de Datos.

Acceso:

1) Desde menú.

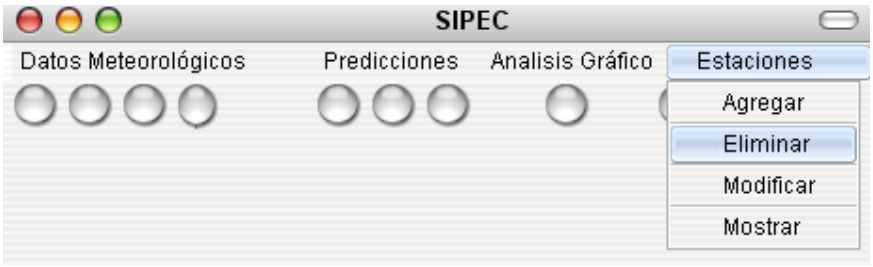


Figura 3.2.4.2.1: Acceso a la operación Eliminar una estación de la BD desde el menú.

2) Desde el botón de acceso rápido.

Figura 3.2.4.2.2: Acceso a la operación Eliminar una estación de la BD desde botón de acceso rápido.

Ventanas Operativas:

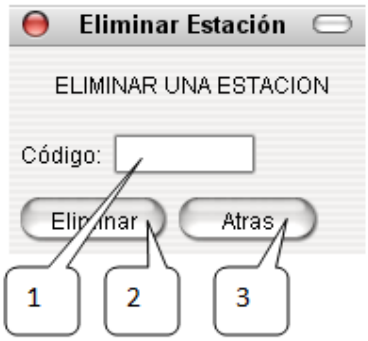


Figura 3.2.4.2.3 Descripción de la ventana Eliminar Datos

1) Introducir el Código de la Estación.
2) Este botón elimina una estación de la BD.
3) Este botón muestra nuevamente la ventana principal.

3.2.4.3 Modificar

Esta opción permite los valores de las estaciones almacenadas en la base de datos.

Acceso:

1) Desde menú.

Figura 3.2.4.3.1: Acceso a la operación Modificar desde el menú.

2) Desde el botón de acceso rápido.

Figura 3.2.4.3.2: Acceso a la operación Modificar desde botón de acceso rápido.

Ventana Operativa:

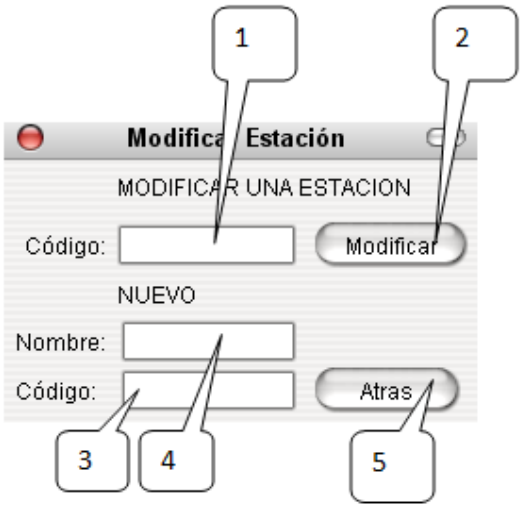


Figura 3.2.4.3.3: Descripción de la ventana Modifica los valores de las estaciones almacenadas en la Base de Datos.

1) Introducir el código de la estación que se va a modificar.
2) Este botón modifica los valores de la estación vieja por los nuevos.
3) Introducir el código que se desea que tenga la estación que se va a modificar.
4) Introducir el nombre que se desea que tenga la estación que se va a modificar.
5) Este botón regresa a la ventana principal.

3.2.4.4 Mostrar.

Esta opción permite visualizar los valores de todas las estaciones almacenadas en la base de datos.

Acceso:

1) Desde menú.

Figura 3.2.4.4.1: Acceso a la operación Mostrar desde el menú.

2) Desde el botón de acceso rápido.

Figura 3.2.4.4.2: Acceso a la operación Mostrar desde botón de acceso rápido.

Ventana Operativa:

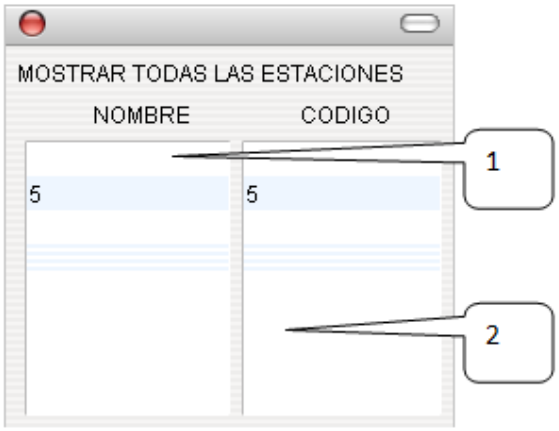


Figura 3.2.4.4.3: Descripción de la ventana Mostrar las estaciones.

1) Se muestra el nombre de todas las estaciones almacenadas en la Base de Datos.
2) Se muestra el Código de todas las estaciones almacenadas en la Base de Datos.

Conclusiones

Con este trabajo se logra como resultado la obtención de un Sistema Basado en Reglas que realiza predicciones de forma rápida y precisa, además de gestionar otras informaciones (valores meteorológicos, estaciones, etcétera).

1ro. Según las pruebas de funcionamiento realizadas al Sistema en la Estación Experimental “Álvaro Barba Machado”, el mismo es capaz de realizar diagnósticos de enfermedades en cultivos agrícolas de manera rápida, precisa y fiable apoyándose para dar cumplimiento a dicha tarea en una metodología previamente planteada por los expertos agropecuarios.

2do. Al concluir el desarrollo del sistema se ha hecho una base de datos para almacenar la información que necesita el sistema, también se construyó una base de conocimiento en lenguaje Prolog para evaluar las reglas que describen las enfermedades y se implementó un sistema que utiliza los componentes anteriormente mencionados para dar solución al problema planteado. Con esto se le ha dado cumplimiento a los objetivos planteados al inicio del trabajo.

Recomendaciones

Se recomienda continuar el desarrollo de este trabajo en función de:

1) Insertar los valores meteorológicos en la base de datos de manera automática, extrayendo estos de los correos electrónicos que arriban de las distintas estaciones meteorológicas.
2) Implementar un servicio que independice el proceso de predicción del resto de la aplicación y que el mismo se active automáticamente para analizar los valores meteorológicos cada vez que se realice una inserción de los mismos en la base de datos.
3) Desarrollar una alternativa para el envío de Señales de Aplicación vía correo electrónico de forma automática a las estaciones correspondientes.

Referencias Bibliografías

. "NetBeans." Retrieved 15/05/2012, from http://www.netbeans.org.

Bramer, M. (2005). Logic Programming with Prolog.

Cerda, F. NetBeans 6.5.

Enriquez, E. I. (2007). SQL Server 2000.

Herrera, I. G. (2007). Incorporación de técnicas de visualización al sistema Weka. Santa Clara UCLV.

Jalón, J. G. d., J. I. Rodríguez, et al. (1999). Aprenda Java como si estuviera en primero.

Marín, Y. C. (2005). Sistema Inteligente Predictor de Enfermedades en Cultivos Inteligencia Artificial. Santa Clara, UCLV.

McGovern, J., R. Adatia, et al. (2003). Java™ 2 Enterprise Edition 1.4 Bible.

Olivares, B. O. and J. I. Chirinos (2009). "Importancia de la agrometeorología en la sanidad vegetal."

ROSSEL, G. Programación lógica.

Ruiz, E. T. (2006). Agrometeorología.

Valverde, N. K. (2010). "DESARROLLAN SISTEMA INTELIGENTE QUE ALERTA A PRODUCTORES AGRICOLAS SOBRE EPIDEMIAS Y PLAGAS." Retrieved 10/4/2012, 2012, from http://universitam.com/academicos/?p=4295.

Anexos

Anexo 1: "Ejemplo de Base de Conocimientos."

```
r1(tizontardio,metodo1,TMAX,TMED,TMEDMENOS1,TMEDMENOS2,
   TMEDMENOS3,TMEDMENOS4,TMIN,LLUVIA):-TMAX < 30,TMED =< 25,
                TMEDMENOS1 =< 25,TMEDMENOS2 =<25,
                TMEDMENOS3 =< 25,TMEDMENOS4 =< 25,
                TMIN > 7,LLUVIA >= 24.5.
calcularindice1(HR,SOL):- HR > 96, HR < 100, SOL is 40.
calcularindice1(HR,SOL):- HR > 91, HR < 95, SOL is 35.
calcularindice1(HR,SOL):- HR < 90, SOL is 30.

calcularindice2(LLUVIA,SOL):- LLUVIA > 10,SOL is 15.
calcularindice2(LLUVIA,SOL):- LLUVIA > 6, LLUVIA < 10,SOL is 10.
calcularindice2(LLUVIA,SOL):- LLUVIA > 1,LLUVIA < 5, SOL is 5.
calcularindice2(LLUVIA,SOL):- LLUVIA =< 1, SOL is 0.

calcularindice3(TMIN,SOL):- TMIN > 12,TMIN < 15,SOL is 30.
calcularindice3(TMIN,SOL):- TMIN > 15.1,TMIN < 18,SOL is 25.
calcularindice3(TMIN,SOL):- TMIN > 18.1,TMIN < 21,SOL is 20.
calcularindice3(TMIN,SOL):- TMIN < 12,TMIN > 21,SOL is 0.

calcularindice4(TMAX,SOL):- TMAX < 25,SOL is 30.
calcularindice4(TMAX,SOL):- TMAX > 25,TMAX < 28,SOL is 25.
calcularindice4(TMAX,SOL):- TMAX > 28,SOL is 0.

calcularindice5(TMED,SOL):- TMED > 15,TMED < 19.9,SOL is 35.
calcularindice5(TMED,SOL):- TMED > 20,TMED < 22,SOL is 15.
calcularindice5(TMED,SOL):- TMED < 15,TMED > 22,SOL is 0.

r1(mohoazul,nc,HR,LLUVIA,TMIN,TMAX,TMED,INDICETOTAL):-
   calcularindice1(HR,SOL1),calcularindice2(LLUVIA,SOL2),
   calcularindice3(TMIN,SOL3),calcularindice4(TMAX,SOL4),
```

```
    calcularindice5(TMED,SOL5),
    INDICETOTAL = (SOL1 + SOL2 + SOL3 + SOL4 + SOL5),
    INDICETOTAL > 30, INDICETOTAL < 75.
r1(mohoazul,cf,HR,LLUVIA,TMIN,TMAX,TMED,INDICETOTAL):-
    calcularindice1(HR,SOL1),calcularindice2(LLUVIA,SOL2),
    calcularindice3(TMIN,SOL3),calcularindice4(TMAX,SOL4),
    calcularindice5(TMED,SOL5),
    INDICETOTAL = (SOL1 + SOL2 + SOL3 + SOL4 + SOL5),
    INDICETOTAL > 80, INDICETOTAL < 95.
r1(mohoazul,cmf,HR,LLUVIA,TMIN,TMAX,TMED,INDICETOTAL):-
    calcularindice1(HR,SOL1),calcularindice2(LLUVIA,SOL2),
    calcularindice3(TMIN,SOL3),calcularindice4(TMAX,SOL4),
    calcularindice5(TMED,SOL5),
    INDICETOTAL = (SOL1 + SOL2 + SOL3 + SOL4 + SOL5),
    INDICETOTAL > 100, INDICETOTAL < 125.
r1(mohoazul,caf,HR,LLUVIA,TMIN,TMAX,TMED,INDICETOTAL):-
    calcularindice1(HR,SOL1),calcularindice2(LLUVIA,SOL2),
    calcularindice3(TMIN,SOL3),calcularindice4(TMAX,SOL4),
    calcularindice5(TMED,SOL5),
    INDICETOTAL = (SOL1 + SOL2 + SOL3 + SOL4 + SOL5),
    INDICETOTAL > 130.

r1(sigatoka,termica,mf,DIA1,DIA2,DIA3,DIA4,DIA5,DIA6,DIA7,
  ACUMULADO):- ACUMULADO=(DIA1+DIA2+DIA3+DIA4+DIA5+DIA6+DIA7),
          ACUMULADO >= 11500.
r1(sigatoka,termica,f,DIA1,DIA2,DIA3,DIA4,DIA5,DIA6,DIA7,
  ACUMULADO):- ACUMULADO=(DIA1+DIA2+DIA3+DIA4+DIA5+DIA6+DIA7),
          ACUMULADO >= 11000,ACUMULADO < 11500.
r1(sigatoka,termica,pf,DIA1,DIA2,DIA3,DIA4,DIA5,DIA6,DIA7,
  ACUMULADO):- ACUMULADO=(DIA1+DIA2+DIA3+DIA4+DIA5+DIA6+DIA7),
          ACUMULADO >= 10000,ACUMULADO < 11000.
r1(sigatoka,termica,df,DIA1,DIA2,DIA3,DIA4,DIA5,DIA6,DIA7,
  ACUMULADO):- ACUMULADO=(DIA1+DIA2+DIA3+DIA4+DIA5+DIA6+DIA7),
```

```
        ACUMULADO < 10000.

r1(sigatoka,evaporacion,mf,VALOR):-VALOR < 22.
r1(sigatoka,evaporacion,f,VALOR):-VALOR >= 22,VALOR < 30.
r1(sigatoka,evaporacion,pf,VALOR):-VALOR >= 30,VALOR < 40.
r1(sigatoka,evaporacion,df,VALOR):-VALOR >= 40.

r1(tizontardio,metodo2,severidad0,HORAS,PROMTEMP):-
                (PROMTEMP>7.2,PROMTEMP<11.6,HORAS<15);
                (PROMTEMP>11.7,PROMTEMP<15.0,HORAS<12);
                (PROMTEMP>15.1,PROMTEMP<26.6,HORAS<9).
r1(tizontardio,metodo2,severidad1,HORAS,PROMTEMP):-
        (PROMTEMP>7.2,PROMTEMP<11.6,HORAS>16,HORAS<18);
        (PROMTEMP>11.7,PROMTEMP<15.0,HORAS>13,HORAS<15);
        (PROMTEMP>15.1,PROMTEMP<26.6,HORAS>10,HORAS<12).
r1(tizontardio,metodo2,severidad2,HORAS,PROMTEMP):-
        (PROMTEMP>7.2,PROMTEMP<11.6,HORAS>19,HORAS<21);
        (PROMTEMP>11.7,PROMTEMP<15.0,HORAS>16,HORAS<18);
        (PROMTEMP>15.1,PROMTEMP<26.6,HORAS>13,HORAS<15).
r1(tizontardio,metodo2,severidad3,HORAS,PROMTEMP):-
         (PROMTEMP>7.2,PROMTEMP<11.6,HORAS>22,HORAS<24);
         (PROMTEMP>11.7,PROMTEMP<15.0,HORAS>19,HORAS<21);
         (PROMTEMP>15.1,PROMTEMP<26.6,HORAS>16,HORAS<18).
r1(tizontardio,metodo2,severidad4,HORAS,PROMTEMP):-
         (PROMTEMP>7.2,PROMTEMP<11.6,HORAS>25);
         (PROMTEMP>11.7,PROMTEMP<15.0,HORAS>22);
         (PROMTEMP>15.1,PROMTEMP<26.6,HORAS>19).
```