

# CAPÍTULO 1

## Introducción

### 1.1 Sistemas basados en conocimiento (Knowledge Based Systems)

Los sistemas basados en conocimiento (Knowledge Based Systems. KBS) son aplicaciones que generan soluciones o respuestas satisfactorias a problemas que requieren un razonamiento por computadora que involucre conocimiento de algún tipo. Consideramos conocimiento como los datos e información que pueden mejorar la eficiencia y la eficacia de una solución referente a un problema. Algunos tipos de conocimiento pueden ser hechos (que expresen proposiciones válidas) o reglas. (Hayes, 94)

Los KBS construyen su razonamiento para resolver problemas concatenando afirmaciones y reglas en líneas de razonamiento. Estas líneas de razonamiento nos muestran cómo un conjunto de suposiciones y un conjunto específico de afirmaciones y de reglas producen

Algunas características básicas de los KBS pueden ser la representación implícita del conocimiento, la capacidad de razonamiento independiente de la aplicación espe capacidad de explicar sus conclusiones, el proceso de razonamiento, etc. Los KBS basan su rendimiento en la cantidad y calidad de conocimiento de un dominio específico y no tanto en las técnicas de solución de problemas. (Flores, 00)

Como elementos importantes de un sistema basado en conocimiento podemos citar las siguientes partes:

- 1)Base de Conocimientos (conocimientos y hechos):** Representación de el conocimiento de el dominio que compete al sistema, junto con el conjunto de información invariable en una u otra solución (hechos).
- 2)Máquina de Inferencia:** Proceso de razonamiento a partir de datos de entrada, tomando como base la base de conocimientos. Esta máquina es genérica en el sentido de que puede aplicarse a diferentes dominios con solo cambiar la base de conocimientos.
- 3)Interfaz con el Usuario:** Entradas y salidas del sistema incluyendo generalmente mecanismos de preguntas y explicaciones.

El verdadero valor de los KBS se debe a que a diferencia de los programas convencionales que utilizan algoritmos para resolver problemas, los KBS resuelven problemas donde las soluciones algorítmicas no existen o son muy costosas para ser implementadas, es decir implementan procesos heurísticos antes que algorítmicos. (Flores, 00)

Los KBS han aumentado de valor en entornos industriales y comerciales en comparación por ejemplo con otros sistemas de computadora en parte porque generalmente ofrecen interfaces gráficas, interactivas y auxiliares las cuales apuntan a una nueva generación de sistemas cooperativos en la cual se vislumbra quizás una sociedad formada por trabajadores del conocimiento los cuales trabajan de manera conjunta para concebir desarrollar y producir servicios. Cada fase de dicho proceso y cada tarea funcional dentro de esta sociedad será beneficiada por la acumulación de conocimiento ejecutable y por la directa manipulación de dicho conocimiento por los KBS.

## **1.2 Prolog**

### **1.2.1 Una breve descripción de Prolog**

Prolog, cuyo nombre se deriva de PROgramming in LOGic, es el lenguaje de mayor impacto en el paradigma de la programación lógica. La base de Prolog son las nociones matemáticas de relaciones y de inferencia lógica. Prolog es un lenguaje declarativo, lo que significa que en vez de describir cómo computar una solución, un programa consiste en una base de datos con hechos y relaciones lógicas que describe las relaciones relacionadas con una aplicación dada. Por lo que en vez de correr el programa para obtener una solución, el usuario hace una pregunta a dicha base. Una vez que se genera una pregunta, el sistema busca a través de la base de datos para determinar (por deducción lógica) la respuesta.

Algunas veces puede que haya mas de una manera para deducir la respuesta o que haya más de una respuesta, en esos se puede solicitar al sistema por más soluciones, por lo que se genera un proceso de “backtracking” para encontrar soluciones alternas.

No existe una estructura definida para crear un programa en Prolog. no hay un procedimiento principal y no existen las anidaciones de definiciones. Todas las reglas y hechos son globales y una variable se resuelve por el hecho o regla donde esta aparezca y la o lo haga válida.

Como se indicó anteriormente un programa en Prolog consiste en una base de datos con reglas y hechos que responde a solicitudes.

## Hechos

Un hecho es como el nombre lo dice un hecho. Un hecho en cualquier lenguaje de hoy en día es una proposición como **hace calor** o **es verano** por lo que en Prolog estos hechos podrían quedar representados como:

```
hace_calor.  
es_verano.
```

## Queries

Un query en Prolog es la acción de preguntarle al programa acerca de cierta información contenida en su base de datos. Por lo que los queries se llevan a cabo de una manera interactiva. Por lo general cuando un programa es cargado se muestra el siguiente prompt:

```
?-.
```

con lo cual uno puede preguntar al sistema acerca de cierta información. Usando la base de datos de arriba se podría preguntar de la siguiente manera:

```
?-hace_calor.
```

A lo cual el sistema respondería:

```
yes.
```

## Reglas

Las reglas extienden las capacidades de un programa lógico. Ellas le dan a Prolog la habilidad de actuar a través de su proceso de toma de decisiones. El siguiente programa por ejemplo contiene dos reglas de temperatura. La primera se lee como **es caluroso si es verano y hace calor** y la segunda corresponde a **es helado si es invierno y hace frío**.

```
hace_calor.  
es_verano.  
es_caluroso:-es_verano, hace_calor.  
es_helado:-es_invierno, hace_frío.
```

Por lo que un query de la forma:

```
?-es_caluroso.
```

contestaría:

**yes.**

y un query de la forma:

**?-es\_helado.**

contestaría:

**no.**

Finalmente podemos mencionar que los tipos de datos simples más comunes son: boolean, integer, real y atom; y como un tipo compuesto tenemos las listas donde dentro de corchetes se ponen los elementos que conforman dicha lista, por ejemplo:

**[perro,gato,ratón]**

### **1.2.2 Bases de conocimiento en Prolog**

Proponemos el lenguaje de programación Prolog para el desarrollo de bases de conocimiento. Prolog es usado en aplicaciones de inteligencia artificial tales como interfaces de lenguaje natural, sistemas de razonamiento automático y sistemas expertos. Los cuales consisten generalmente en una base de datos con hechos y reglas junto con una interfaz que accesa dicha base como al final será este el caso.

Tenemos que hablar ahora de la representación de la gramática de algún lenguaje como una base de conocimientos en Prolog ya que el desarrollo que se presentará utiliza dicha forma de representación para el lenguaje Japonés, por lo que para el manejo de el lenguaje, se debe de tener una representación adecuada de dicho idioma que permita una interacción lo más fluida y natural posible por parte del sistema con relación al estudiante. Es decir el tener dicha representación de el vocabulario y gramática de dicho idioma de la manera más lógica posible para construir las oraciones y estructuras gramaticales necesarias de la mejor manera. Por lo cual se ha decidido implementar el manejo de el idioma tanto Japonés como los hechos para el lenguaje de programación Prolog. Se eligió dicho lenguaje lógico debido a sus características tales como el manejo de "variables lógicas" lo que significa que éstas se comportan como variables matemáticas, un poderoso recurso de "pattern-matching", una estrategia de "backtracking" para búsqueda de soluciones, estructuras de datos uniformes, etc.(Aaby, 97).

### **1.3 Educación e Internet**

Debido al aumento creciente en la cantidad de información disponible que se viene y se ve en el mundo actual, y que ahora gracias a la "globalización" se hace presente en nuestras vidas con mayor fuerza día con día; el facilitar el acceso a la educación deberá ser una obligación moral de la sociedad. En el futuro las personas, quizás no tengan que ir "físicamente" a la escuela para recibir una educación, pero deberán poder acceder a conocimientos que les permita adquirir destrezas laborales específicas; ya que los nuevos empleos exigirán conocimiento, imaginación e inteligencia intensiva (Educadis, 2000). En países desarrollados ya se ha empezado la migración de un modelo de escuela común y centralizada, a uno flexible y descentralizado que mejora la dinámica social y facilita el acceso al conocimiento (Educadis, 2000). Por lo que la educación a distancia debe comenzar a cobrar mayor importancia en países en desarrollo que indudablemente se encuentran en cierta "desventaja" en relación al acceso directo de fuentes de conocimiento, además de presentar dificultades de tiempo, distancia y dinero, que con la educación a distancia si no se resuelven del todo se mejoran. Además es importante el incluir elementos importantes tales como un contacto inmediato con estudiantes de distintas escalas sociales, culturales y económicas, o la posibilidad de "asistir" a eventos como conferencias que se lleven a cabo en lugares remotos.

El Internet es por ahora el vehículo idóneo para la transmisión y construcción de conocimientos. Esto debido al conjunto de información diversa dentro de un mismo ente y el aumento de su participación en la capacitación de la gente en diversas áreas en donde la educación como presencia física dentro en un espacio común se transforma ahora en una educación vía Internet en donde la distancia ya no es un obstáculo para la transmisión de información.

### **1.4 Los clientes delgados**

En la necesidad de construir medios que permitan acercar a la gente con la información a distancia, los sistemas basados en clientes delgados cobran aquí una importancia fuerte debido entre otras cosas a su bajo costo y el poco mantenimiento necesario que requieren. Como tal, el concepto de cliente delgado se tiene que ver reflejado en una rápida respuesta por parte del ambiente hacia el estudiante, así como una mínima participación en la actividad, y en general de una mínima dependencia por parte del estudiante en relación a las necesidades técnicas que el ambiente requiera sin perder optimalidad por parte del sistema cliente.

No se está hablando de la necesidad de poseer un hardware delgado por parte del cliente para poder hacer uso de la tecnología que ha continuación se presentará, sino de la necesidad de construir dicha tecnología sobre la filosofía de los clientes delgados a manera de liberar al cliente de la necesidad de tal o cual aspecto necesario para hacer uso de dicho programa y así delegar responsabilidad al servidor que provee tal servicio.

La tecnología de servlets nos permitirá implementar la noción de clientes delgados en este trabajo. El trabajo fuerte de el manejo de el idioma y despliegue de resultados se hará en la máquina servidor mientras que el cliente solo recibirá resultados y hará requisiciones al servidor.

## **1.5 Definición del problema**

Se requiere desarrollar un conjunto de clases, para permitir al programador en Java el desarrollo de aplicaciones a través del Internet que permitan tener un fácil manejo de una base de conocimientos en Prolog.

Este trabajo de tesis propone la infraestructura necesaria para el desarrollo de la versión Internet del ambiente GRACILE (GRAMmar Collaborative Intelligent Learning Environment) el cual fue implementado en Prolog en una red de área local (Appletalk). (Ayala, 96)

El software desarrollado se probará al implementar GRACILE en Java con una interfaz permitir el acceso a la base de conocimientos de la gramática de el idioma Japonés. De esta manera el usuario podrá conocer el significado en Inglés y análisis gramatical de nombres, adjetivos y verbos del Japonés, así como preguntar por sustantivos en Japonés y conjugar además adjetivos y verbos.

El desarrollo del proyecto seguirá la línea de trabajos que la Universidad de Tokushima y el grupo de ambientes de aprendizaje del CENTIA.

## **1.6 Objetivo General**

El desarrollo del conjunto de clases en Java que permita el desarrollo de sistemas basados en conocimiento en Prolog, aplicándolo en la implementación de un ambiente de aprendizaje del Japonés en Internet (Gramática y Vocabulario) a partir de la base de conocimientos de GRACILE.

## **1.7 Objetivos Específicos**

1)Diseño en UML e implementación del paquete easyprolog como un conjunto de clases en Java que permitan la manipulación de una base de conocimientos en Prolog desde algún componente en Java.

2)Diseño en UML y programación de clases genéricas en Java para la implementación de interfaces para ambientes de aprendizaje de la gramática y el vocabulario del Japonés utilizando la base de conocimientos de GRACILE. La gramática del segundo idioma se encontrará como un conjunto de reglas y el diccionario como un conjunto de hechos, ambos en Prolog.

3)Determinar e implementar la forma de presentar caracteres japoneses en el navegador en la máquina cliente.

## **1.8 Alcances y Limitaciones**

Se obtendrá el paquete easyprolog como un conjunto de clases referente de una base de conocimientos en Prolog en el servidor a través de un componente en Java.

El ambiente a desarrollar será accesible a través de cualquier navegador que cuente con los componentes necesarios para manipular el idioma Japonés.

Algunos detalles de diseño e implementación estarán sujetos a el trabajo que se haga en conjunto con el equipo de trabajo en la Universidad de Tokushima en Japón.

## **1.9 Hardware y Software a utilizar**

- 1)Terminal de Enterprise 450.
- 2)Software de desarrollo Java: JDK 1.2/JSWDK 1.0.1/HTML.
- 3)Navegadores Web (Netscape y Microsoft Explorer).
- 3)Editor de texto: GNU Emacs, Microsoft Word.
- 4)Lenguaje de Programación de Prolog SICStus.