Capítulo 5.
Implementación del Agente Móvil SAM

En este capítulo se explica de manera detallada la implementación del agente móvil SAM. Primero se describe la integración de SAM a la arquitectura existente en FDL. Después se presentan las tecnologías usadas pare su desarrollo, las componentes del sistema, su funcionamiento y algunos de los algoritmos utilizados pare la implementación del mismo.

5.1 Incorporación de SAM a FDL

El agente SAM se integra al esquema de FDL al ser registrado como una clase de agente disponible, sin embargo eso no es todo, hay varios aspectos de comunicación que necesitan tomarse en cuenta. El primer paso para integrar a SAM en FDL es registrarlo como una clase de agentes a través del UAM, de este modo cuando se entre a la sección del UAD el agente SAM aparecerá como una de las opciones a elegir.

Una vez que se tiene el agente, se le presenta al usuario pare que lo pueda comenzar a usar. Si el usuario lo elige entonces el UAD le presenta una forma en la que debe llenar los campos que corresponden a los atributos de la clase, asimismo se le presenta otra forma para dar de alta sus preferencias. Después de esto, el UAD le pide a ALiS que registre la nueva instancia del usuario, y cuando se ha realizado esta acción se le muestra al usuario la información y el estado actual de su nuevo agente, finalmente se presenta una forma a través de la cual el usuario active al agente SAM, completando con esto la tarea del UAD. La integración de SAM a FDL se ilustra en los Apéndices E-F.

En concreto se desarrollaron programas utilizando la interfaz CGI (Common Gateway Interface) que combinan C, HTML (HyperText Markup Language) y SQL de I llustra, y que son accesibles a través del Web por medio de formas electrónicas creadas en HTML. Dos de esos programas son para el control del archivo de preferencias y uno es para el acceso y actualización de las bases de datos involucradas en el proceso de instanciar un agente. Se decidió trabajar con CGI´s y HTML, debido a que estas son las herramientas que se usaron paraconstruir la arquitectura FDL, de esta manera se logró que los archivos creados se acoplaran de manera natural a FDL.

5.2 Herramientas Utilizadas para la Implementación de SAM

Para implementar el agente SAM, se tomo en cuenta el ambiente distribuido de la biblioteca digital FDL, el DBMS con que fue creada y la característica de movilidad que debe mantener el agente.

Así que para el desarrollo de SAM se eligió *Aglets Workbench,* debido a que es un lenguaje multiplataforma, funcional en ambientes distribuidos, basado en Java y con características como JDBC y RMI entre otras. El hecho de haber implementado al agente en esta plataforma Java, permitió la integración de SAM con el resto de las aplicaciones ya incorporadas a FDL como son: Chrysalis [Lopez 1997], FRA [Cabrera 1997] y MUTANT [Flores 1997]. Debido a que los datos de la biblioteca FDL, estan dentro de una base de datos de Illustra, se opto por seguir con este mismo DBMS. Finalmente, se uso *mijdbc* como el Comnon Gateway Interface. HyperText Markup Language. controlador JDBC para conectarse a la base de datos. A continuacion se justifica el uso de cada una de estas herramientas.

5.2.1 Aglet, WorkBench

Después de estudiar los lenguajes existentes para el desarrollo de agentes móviles (tales como Agent Tcl, Ara, Odyssey y Tacoma, entre otros descritos en el capítulo *2,* se decidió utilizar Aglets WorkBench (AWB) para el desarrollo de SAM debido a que esta plataforma esta basada en el lenguaje de programación Java, y por consecuencia ofrece las ventajas de este lenguaje: es orientado a objetos, portable y cuando se invoca a un programa de manera remota, el código del programa en cuestión se envía del servidor al cliente para que se ejecute ahí, evitando con esto que el servidor se sobrecargue de trabajo, ofrece también los conceptos de clases, instancias y métodos. Cuenta con un ambiente de desarrollo visual, que permite combinar componentes gráficos y no gráficos pare la personalización de los agentes. AWB ofrece varios paquetes pare acceso de datos como JDBC/DB2 y JoDax [Lange y Chang 1996]. Cuenta también con *ATP* (Agent Transfer Protocol), un protocolo estándar a nivel de aplicación para sistemas de información distribuida basada en agentes. ATP ofrece un protocolo independiente de plataforma y uniforme para la transparencia de agentes entre nodos de una red. Además de lo ya mencionado, AWB cuenta con *Tahiti,* un manejador visual que permite monitorear y controlar los aglets. AWB incluye también un disparador de agentes llamado *Fiji,que* permite ejecutar un aglet desde un página HTML como se haría con un apples normal. Finalmente, AWB soporta un modelo de seguridad por capes (ver Apéndice C).

5.2.2 Illustra

Se decidió trabajar con el DBMS Illustra, primero pare estandarizar este sistema con el resto de los ya desarrollados para FDL y segundo debido a que Illustra es un manejador de bases de datos que soporta de manera confiable la cantidad de transacciones que supone una biblioteca digital, además de que conserva la integridad de la información en todo momento y ofrece el manejo de un amplio rango de tipos de datos [Cabrera 1997]. Finalmente permite varias formas de interactuar con la información.

5.2.3 El controlador mijdbc

Se eligió el controlador JDBC *mijdbc* debido a sus características, mijdbc se conecta directamente con el servidor de la base de datos Illustra ya sea a través de una conexión *socket* o a través de un "demonio" puente en el servidor. Permite enviar declaraciones SQL a la base de datos como inserciones, actualizaciones, eliminaciones y selecciones. Recibe resultados en una forma que es conveniente en un ambiente Java. Es un controlador que esta completamente escrito en Java y no llama a ningún otro controlador o software intermedio del lado del cliente [Kalvelagen 1997]. Una vez que ya se han descrito y justificado las tecnologías que se usaron pare el desarrollo del agente, se van a describir cada uno de los módulos y elementos que utilice este.

5.3 Tablas del Agente

El agente para su operación necesita almacenar cierta información de manera permanente, por lo que utiliza una pequeña extensió a la base de datos. En lustra de la biblioteca digital floristica. Se crearon 4 tablas adicionales que son:

Ag_configuration,

> para almacenar los atributos de la configuración del agente.
> Ag_nodes, para almacenar la lista de nodos que forman el itinerario del agente.

Ag_queries,

> para guardar los términos a buscar .

Keyword_weights, que contiene las estructuras morfológicas que son de interés para el usuario, así como el "peso" asociado a cada una de ellas. Referirse al Apéndice D, pare ver a detalle, los atributos que conforman a cada una de estas tablas.

5.4 Condiciones Preliminares

Una vez que se instancia el agente SAM a traves del UAD de FDL, éste despliega la interfaz mostrada en la Figura 5.3. Esta interfaz y los demás módulos se explican en las siguientes secciones.

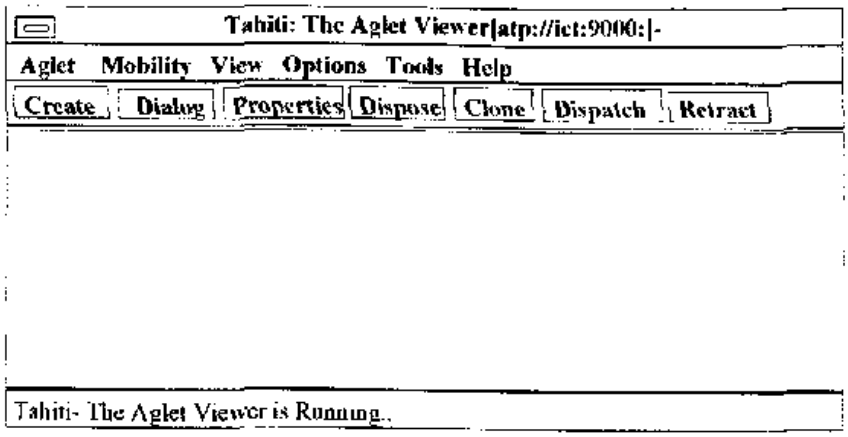

Figura 5.1 Aplicación Tahiti de IBM.

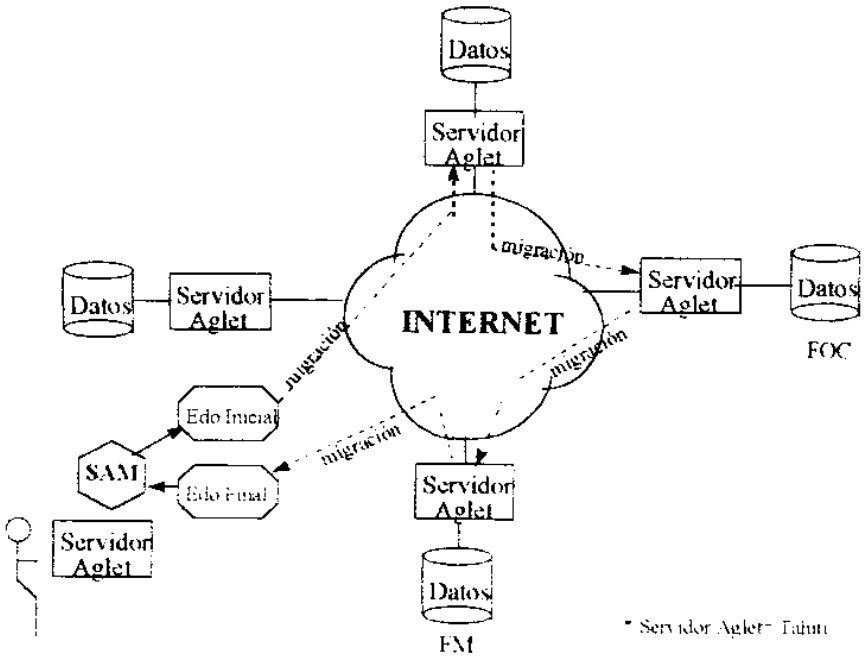

Figura 5.2 Descripción del funcionamiento

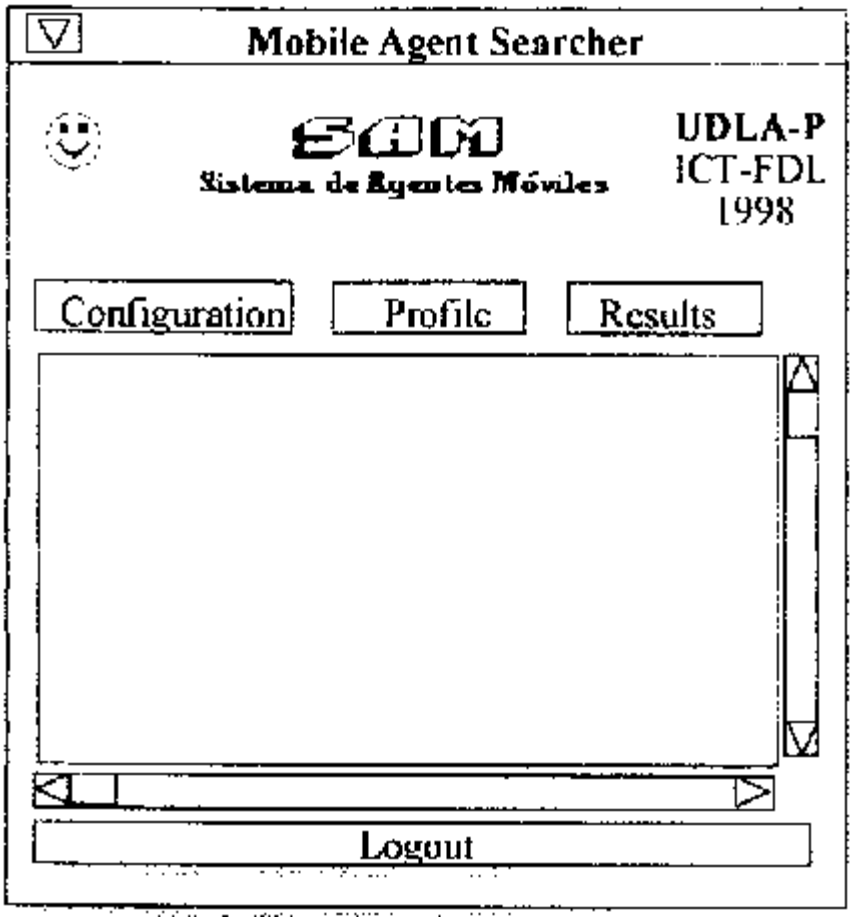

Figura 5.3 Interfaz del sistema SAM.

5.5 Módulos de SAM

Esta versión de SAM implementa toda la funcionalidad planteada en el capítulo anterior. En el capítulo siguiente se describen los alcances y las limitaciones de SAM, además se mencionan las pruebas que se hicieron en este sistema, así como las ventajas y desventajas de esta implementación.

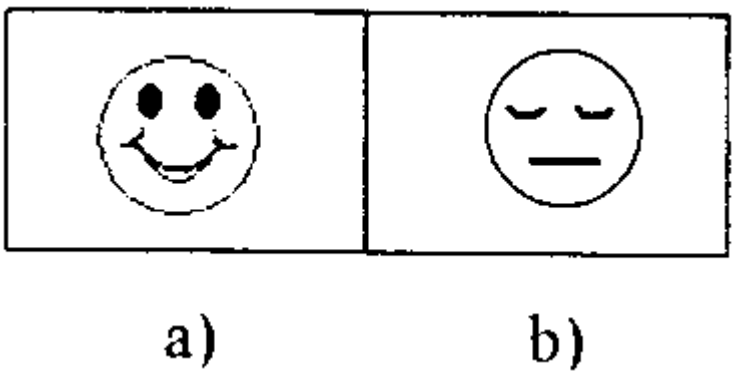

Figura 5.4 Caricaturas para indicar el estado del agente a)activo. b)suspendido

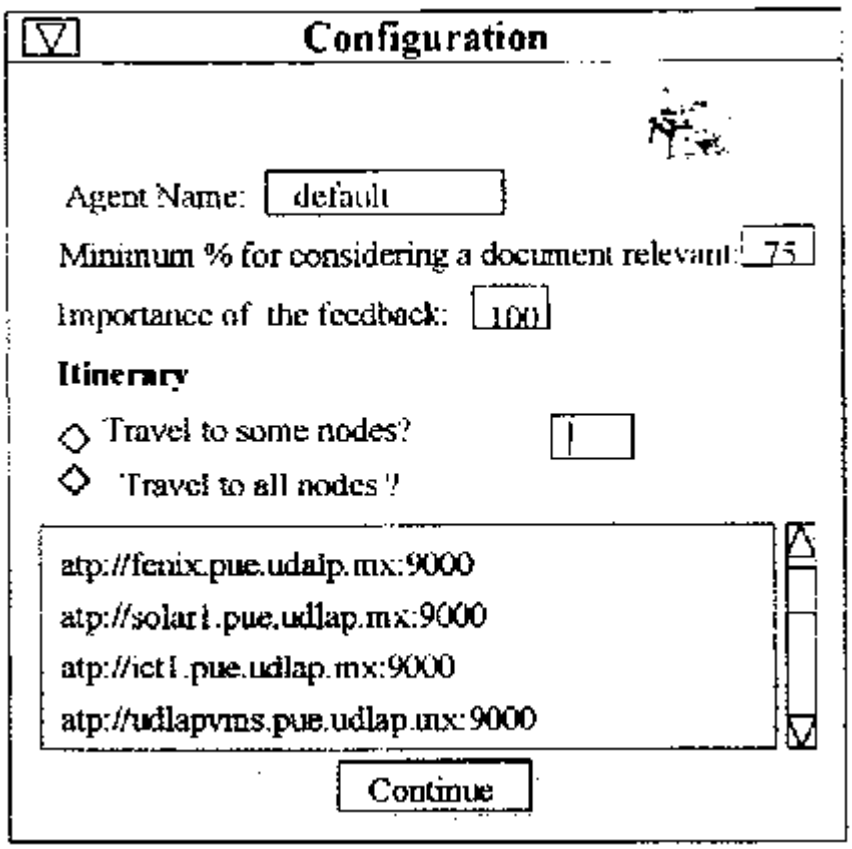

Figura 5.5 Caja de Diálogo para la configuración de SAM

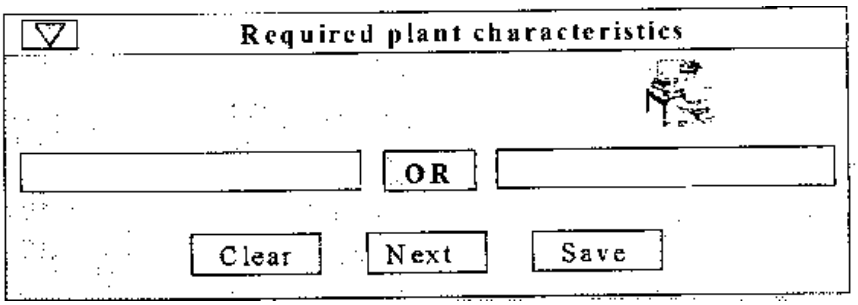

Figura 5.6 Caja de Diálogo pare la introducción de los términos a buscar.

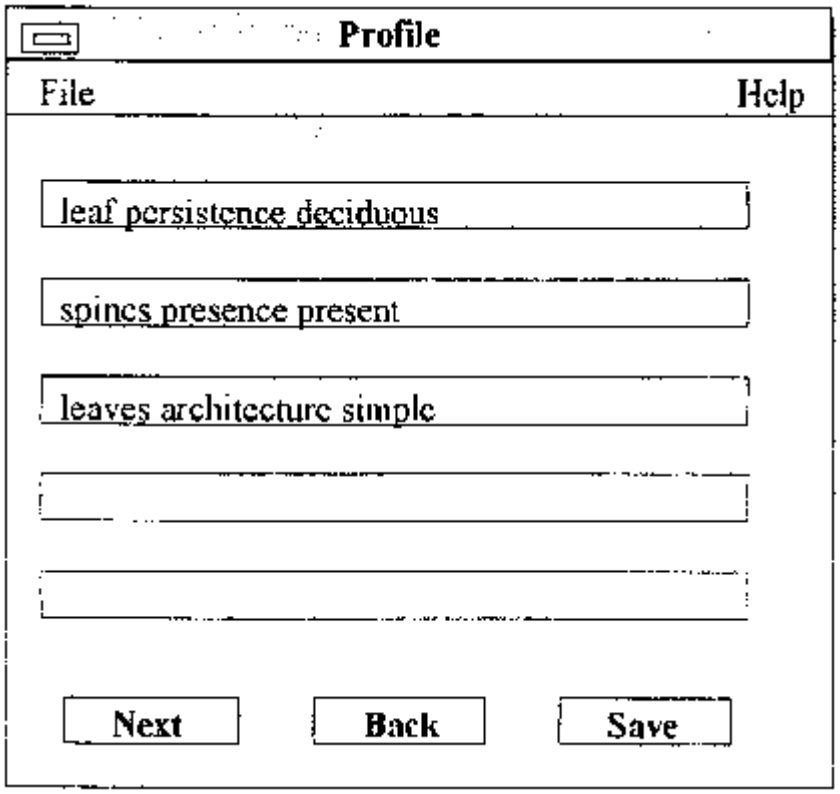

Figura 5.7 Pantalla del editor del perfil del usuario.

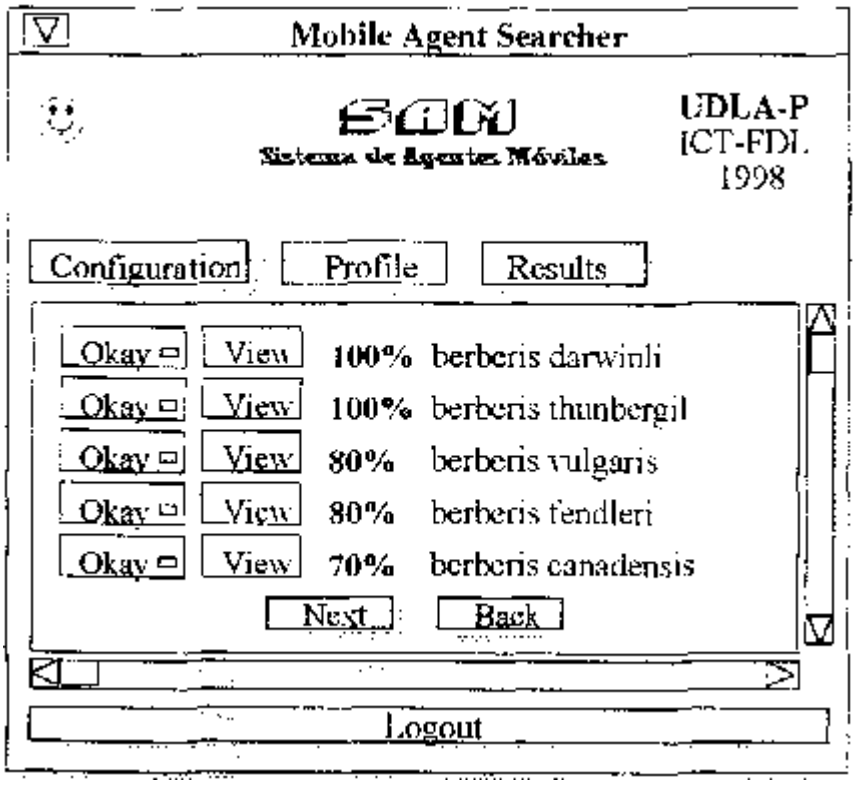

Figura 5.8 Pantalla de Resultados.

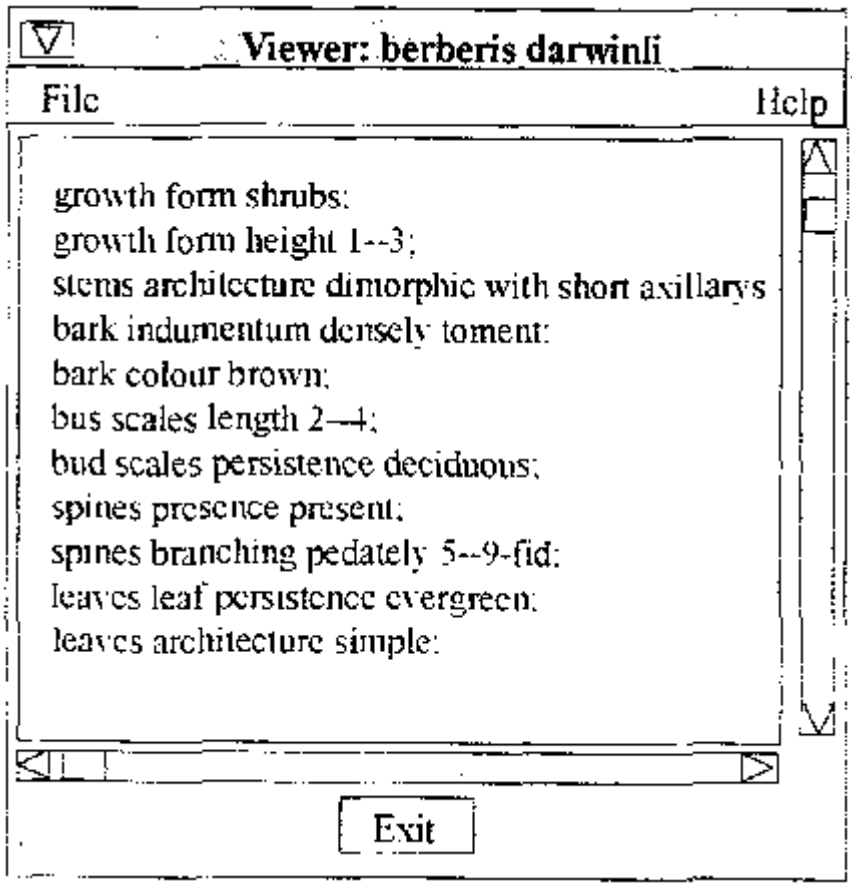

Figura 5.9 Documento Completo.

Figura 5.10 Pasos asignados para el proceso de retroalimentación

Calificador	Peso
Excellent	0.30
Good	0.15
Ok	0.05
Wrong	-0.15
Very Wrong	-0.30

Pérez Lezama, C. V. 1998. **Agentes Móviles en Bibliotecas Digitales**. Tesis Maestría. Ciencias con Especialidad en Ingeniería en Sistemas Computacionales. Departamento de Ingeniería en Sistemas Computacionales, Escuela de Ingeniería, Universidad de las Américas Puebla. Mayo. Derechos Reservados © 1998.