

En este capítulo se describen los detalles de la implementación del proyecto, basados en el diseño conceptual del capítulo anterior. Aquí se describen la forma en la que este proyecto integra a la arquitectura de U-DL-A, las herramientas que se usaron para su implementación, así como una descripción detallada de los módulos del sistema.

### 5.1 Integración con U-DL-A

### 5.2 Herramientas Utilizadas

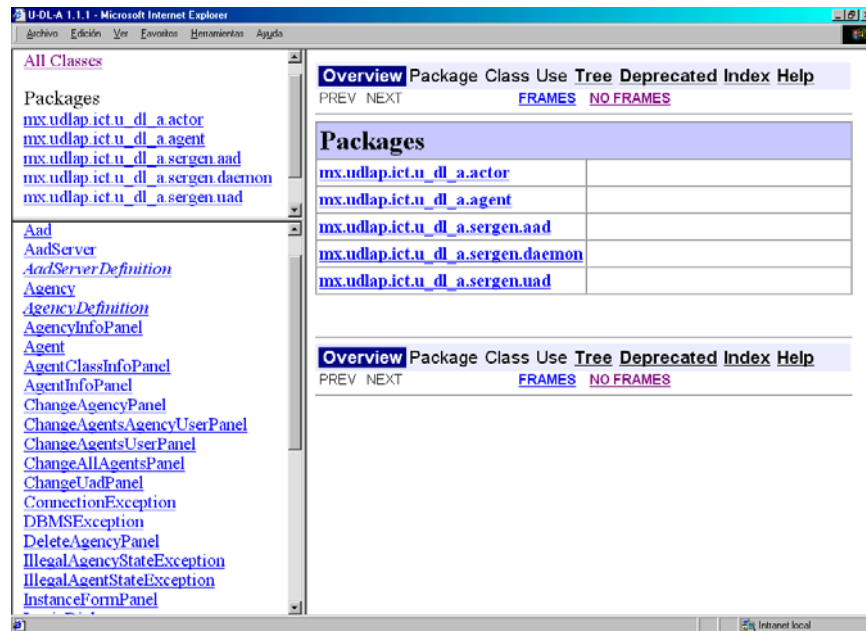
### 5.3 Módulos del Proyecto

#### 5.1 INTEGRACIÓN CON U-DL-A

Como anteriormente se había dicho, este proyecto fue planteado como un mecanismo de la biblioteca digital U-DL-A, para la implantación de un estándar dentro de la misma que facilite la comunicación entre las agencias y los directores de la biblioteca digital (el UAD y el AAD).

Para efectos de estandarización, se llegó al acuerdo de crear un paquete donde se encuentren ubicadas las aplicaciones hechas en Java que estarán disponibles para el usuario de la biblioteca digital. El nombre de este paquete es `mx.udlap.ict.u_dl_a`, dentro de éste se encuentran una serie de subcategorías que corresponden a las aplicaciones basadas en Java que existen en la biblioteca digital. Dentro del Laboratorio de Tecnologías Interactivas y Cooperativas (ICT) se creó un estándar de programación para todas aquellos proyectos a aplicaciones en desarrollo. Por esa razón el idioma en que se programó fue inglés. En adición a eso se establece el uso de Javadoc (una herramienta que genera la documentación de los programas de Java) como obligatorio para mostrar los métodos y funciones de los sistemas. SerGen consta principalmente de 3 paquetes: `mx.udlap.ict.u_dl_a.sergen.uad` en donde se encuentran las clases necesarias del UAD, su servidor, su agencia y su interfaz. De la misma forma ocurre con el AAD, su paquete `mx.udlap.ict.u_dl_a.sergen.aad` contiene el servidor y la agencia del mismo. El tercer paquete, `mx.udlap.ict.u_dl_a.sergen.daemon` contiene los procesos *daemon* que se encargan de verificar que las agencias y servidores de los directores se encuentren disponibles. Además de los paquetes relacionados con SerGen, se crearon paquetes de uso genérico para todas los servicios que participen en U-DL-A, esto forma parte del protocolo de comunicación puesto que se debe estandarizar la forma en la que se manejan las clases y objetos en U-DL-A. El paquete `mx.udlap.ict.u_dl_a.actor` clases relacionadas con los usuarios tales como

clases que almacenan sus datos personales como **mx.udlap.ict.u\_dl\_a.actor.Person** . Dentro del paquete **mx.udlap.ict.u\_dl\_a.agent** se encuentran clases relacionadas con agentes, las estructuras de datos de los agentes de usuario, estados ilegales de agentes ( *Exceptions* en Java). Así como las definiciones del protocolo de comunicación del UAD con las agencias llamado AgencyDefinition (Ver Figura 5.2 ). En la Figura 5.1 se ilustra los paquetes de U-DL-A generados con Javadoc.



**Figura 5.1** Paquete `mx.udlap.ict.u_dl_a`

Cada vez que una nueva aplicación hecha en Java se agregue a la biblioteca digital, deberá seguir este estándar de paquetes para permitir una manipulación y comunicación transparente con los demás módulos de U-DL-A.

Además de seguir el estándar de paquetes, las aplicaciones basadas en agentes deben de cumplir con otro estándar de comunicación, mediante el cual el UAD y el AAD se puedan comunicar con ellas. Esto es que para que una agencia se comunique de manera transparente con los servidores del UAD y del AAD se requiere que la agencia posea ciertas rutinas o métodos mediante los cuales los servidores se comuniquen con ellas. Por el otro lado, los servidores ya poseen ciertas rutinas mediante las cuales logran la comunicación con las agencias, esto es que cuando una agencia nueva llega a la biblioteca digital, ésta debe de dar por hecho que algunos métodos específicos ya están implementados en los servidores, más específicamente los relacionados con la administración de los agentes. Ver Figura 4.2.

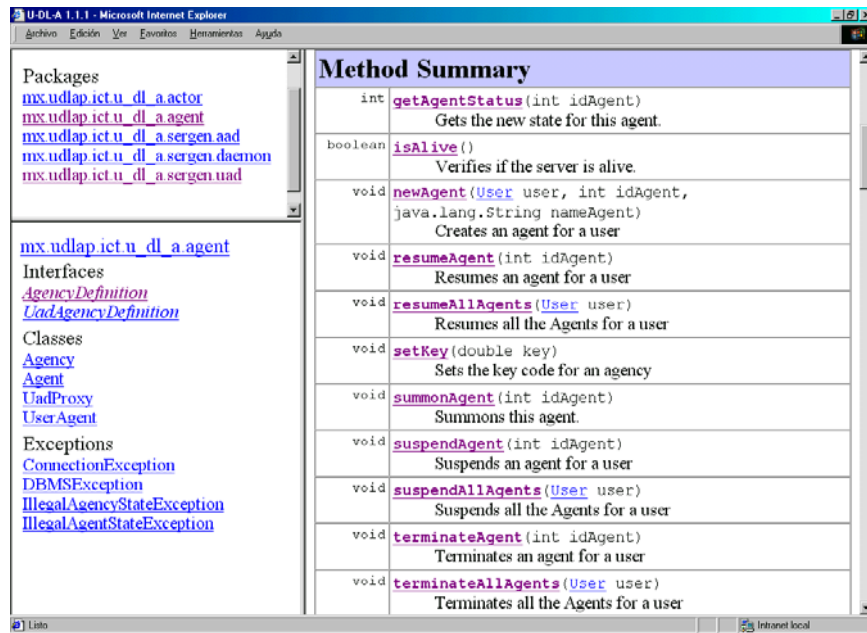


Figura 5.2 Estándar de comunicación AgencyDefinition

En la Figura 4.2 , se muestran una serie de rutinas o métodos con los que debe contar la agencia para que el UAD y el AAD puedan comunicarse con ella. Cada vez que el usuario desee manipular a sus agentes a través del UAD, éste se comunicará por medio de estas rutinas con la agencia.

## 5.2 HERRAMIENTAS UTILIZADAS

Una parte fundamental para el éxito o fracaso de un sistema son las herramientas con las que fue desarrollado, se tiene que tomar en cuenta una serie de situaciones antes de escoger el lenguaje de programación bajo el que se va a implementar un sistema.

La Biblioteca Digital U-DL-A se basa principalmente en Internet para poder funcionar, todos sus servicios están disponibles vía Web, por lo que las aplicaciones deben hacer uso de Internet para lograr su objetivo.

### 5.2.1 Lenguaje de Programación

Para realizar este proyecto se consideró a Java, un lenguaje de programación creada por SUN Microsystems, como el idóneo para esta tarea, esto porque algunas de sus características o filosofías coinciden con las de este proyecto como son:

Independencia de plataforma.- Java es un lenguaje multiplataformas por lo que una aplicación desarrollada podrá ser visualizada en la mayoría de las plataformas de cómputo existentes. Esta característica también la

comparte este proyecto pues el usuario debe poder usar el sistema no importando si están en una Macintosh, una Sun o una PC.

*Look and Feel* .- Java posee una serie de librerías gráficas llamadas *Swing* que se encargan de crear una serie de objetos gráficos de tal manera que simulan ser una aplicación más de su sistema operativo, es decir, las aplicaciones que hacen uso del *look and feel* de Java poseen la cualidad de adaptarse gráficamente al sistema operativo del cliente, facilitándole el uso del sistema. Uno de los objetivos de este proyecto es crear una serie de interfaces amigables para el usuario, y este componente de Java es el idóneo para hacerlo.

Orientado a Red.- Java nació como un lenguaje orientado a la red, con la frase de "La Red es la computadora" Sun Microsystems quiso dar a entender que Java asume la red como un elemento más en el que se pueden hacer aplicaciones, de ahí que surgen las aplicaciones en Web como Applets, Servlets, entre otras. Como se mencionó anteriormente, U-DL-A es una biblioteca digital cuya funcionalidad está basada en Internet, por lo tanto los servicios que proporcione a sus usuarios también deben de estar orientados al Web.

### 5.2.2 Java RMI

Como se mencionó anteriormente, Java es un lenguaje orientado a la red, y por esa razón además de los métodos tradicionales de programación también se incluye la posibilidad de poder usar objetos remotamente, esto quiere decir, objetos que están ubicados físicamente en otra máquina, y que está haciendo uso de la red. A este módulo de Java se le conoce como Java RMI (Invocación de Métodos Remotos). RMI permite que un objeto de una máquina virtual de Java se comunice con otro que está siendo ejecutado en otra Máquina Virtual. Java RMI usa el *Java Remote Messaging Protocol (JRMP)*, un protocolo desarrollado única y exclusivamente para objetos remotos.

Una parte importante de este proyecto es el estándar de comunicación entre los directores y las agencias, éste estándar hace uso extensivo de RMI para lograr ese objetivo. Usando RMI, una aplicación cliente se puede comunicar con el servidor de otra independientemente de que este en otra máquina o en diferente máquina virtual de Java. Los métodos o rutinas que se incluyen en el estándar de comunicación se describen en los capítulos 3 y 4.

### 5.2.3 Java Swing

Java posee una serie de librerías gráficas que permiten la creación de interfaces de Usuario Gráficas (GUI) de una forma sencilla y transparente, haciendo uso de las JFC ( *Java Foundation Classes*) este módulo llamado Java Swing permite crear interfaces estéticas, funcionales e independientes de plataforma.

Otro beneficio de usar Java Swing es que a diferencia de la otra librería de gráfica de Java llamada AWT ( *Abstract Window Toolkit* ) posee la posibilidad de predefinir un estilo de ventanas semejantes al de un sistema operativo en especial, a esta cualidad se le conoce como *Look and Feel* porque hace sentir al usuario de la interfaz como si estuviera usando una aplicación normal de su sistema operativo como Microsoft Windows o Solaris. Existen varios tipos de apariencias de las interfaces de Swing como la de Windows, Macintosh, Motif y la metálica o estándar de java., pero lo más importante es que lo único que cambia es el aspecto de la interfaz porque la funcionalidad sigue siendo la misma. Swing se basa en un esquema de jerarquías de objetos para elaborar las interfaces, es decir, existe un objeto mayor que contiene a todos los demás.

El Director de Agentes (UAD) y el Director de Agencias (AAD) fueron desarrollados en Java Swing para obtener todos los beneficios anteriormente mencionados de Java Swing.

#### 5.2.4 Java Servlets

Java Swing es una herramienta muy útil para desarrollar aplicaciones en Web, también llamadas *Applets* , sin embargo, para poder ejecutar un Applet se requiere que la máquina del usuario tenga una máquina virtual de Java esto quiere decir que previo a la ejecución de una aplicación es necesario realizar algunas descargas e instalaciones de ciertos componentes requeridos para que se pueda ejecutar el programa. Esto se debe a que la ejecución del programa se realiza en la máquina del cliente, lo cual puede llegar a ser un inconveniente para los usuarios de algunos sistema operativos. Para los usuarios que poseen un esquema de cuentas como en Unix, la instalación del aditamento de java para los navegadores no es posible mas que solo por el administrador de la red.

Por lo anterior, en ocasiones lo más conveniente es hacer que los procesos de Java se ejecuten en el servidor y no en el cliente, de ahí surgen los Servlets como una solución eficiente a este problema. [Hunter y Crawford 1998 citados en Chevalier 2000] mencionan algunas ventajas de los Servlets:

**Extensibilidad.** Los servlets extienden las capacidades del servidor.

**Elegancia.** El código generado en la implementación de los servlets es orientado a objetos, limpio, simple y modular.

**Seguridad.** Los servlets son altamente seguros ante los problemas de manejo y desperdicio de memoria, así como referencias a apuntadores inválidos.

**Portables.** Pueden ejecutarse en todas las plataformas que soportan Java y con la mayoría de los servidores de web disponibles.

Poderosos. Heredan el poder completo del corazón del API de Java: redes, acceso a URL's, multi-hilado, manipulación de imágenes, compresión de datos, conexión a bases de datos por medio de JDBC, internacionalización, invocación a métodos remotos (RMI), serialización de objetos, entre otros.

Debido a que usan HTML, los servlets pueden llegar a ser mucho más accesibles para algunos usuarios que las interfaces hechas en Swing. Es por esa razón que se decidió crear dos interfaces del UAD, una hecha en Java Swing (Ver Figura 5.3 ) y la otra en Java Servlets (Ver Figura 5.4). Aunque poseen aspectos diferentes, poseen la misma funcionalidad, de tal forma que el usuario las pueda usar indistintamente. Cabe señalar que ambas interfaces se comunican con el servidor del UAD para hacer las peticiones a las agencias.

### *5.2.5 Informix Universal Server*

El manejador de base de datos utilizado fue Informix Universal Server, esto con el objetivo de estandarizar el presente proyecto con el resto de los desarrollados en U-DL-A. Informix conserva la integridad de los datos y soporta una amplia gama de las transacciones que se llevan a cabo en la biblioteca digital. Este DBMS incluye un API (Interfaz para Programación de Aplicaciones) para comunicarse con Java por medio de JDBC (Java Data Base Connectivity). JDBC es una interfaz que sirve para acceder a bases de datos por medio de SQL (Lenguaje Estructurado de Consultas).

## 5.3 MÓDULOS DEL PROYECTO

SerGen consta principalmente de ocho partes (ver figura 3.4 ), a continuación se describen los detalles de implementación de estos módulos. Cabe señalar que la implementación se basó en el modelado UML descrito en el apéndice A.

### *5.3.1 Módulo UAD*

Dentro de este módulo se realizan todos los procesos y funciones relacionadas con el Director de Agentes de Usuario. Este módulo consta de 3 partes o módulos, la Interfaz, el servidor y la agencia del UAD.

#### *5.3.1.1 La interfaz del UAD*

Esta es la parte visual del módulo del UAD, una interfaz que interactúa con el usuario de la biblioteca digital. Desde esta interfaz, el usuario puede crear, suspender, terminar y reactivar sus agentes personales. Además puede observar las agencias que existen en la biblioteca digital así como el estado en que se encuentran (activas o suspendidas). De esta

interfaz se desarrollaron dos versiones una hecha en Java Swing (Ver figura 5.3 ) y la otra en Java Servlets (Figura 5.4). El objetivo de desarrollar dos interfaces del UAD es para llegar a la mayoría de usuarios posibles, ya que todavía existen algunos problemas con los Java Applets que usan Swing debido a que requiere de un aditamento para el navegador especial de Java. Además de Swing, esta interfaz hace uso extensivo de procesos de RMI y requieren un administrador de seguridad para realizar estos procesos. Para hacer que el Applet esté disponible vía Web se requiere que éste posea una firma digital que compruebe su origen y le garantice seguridad al usuario.

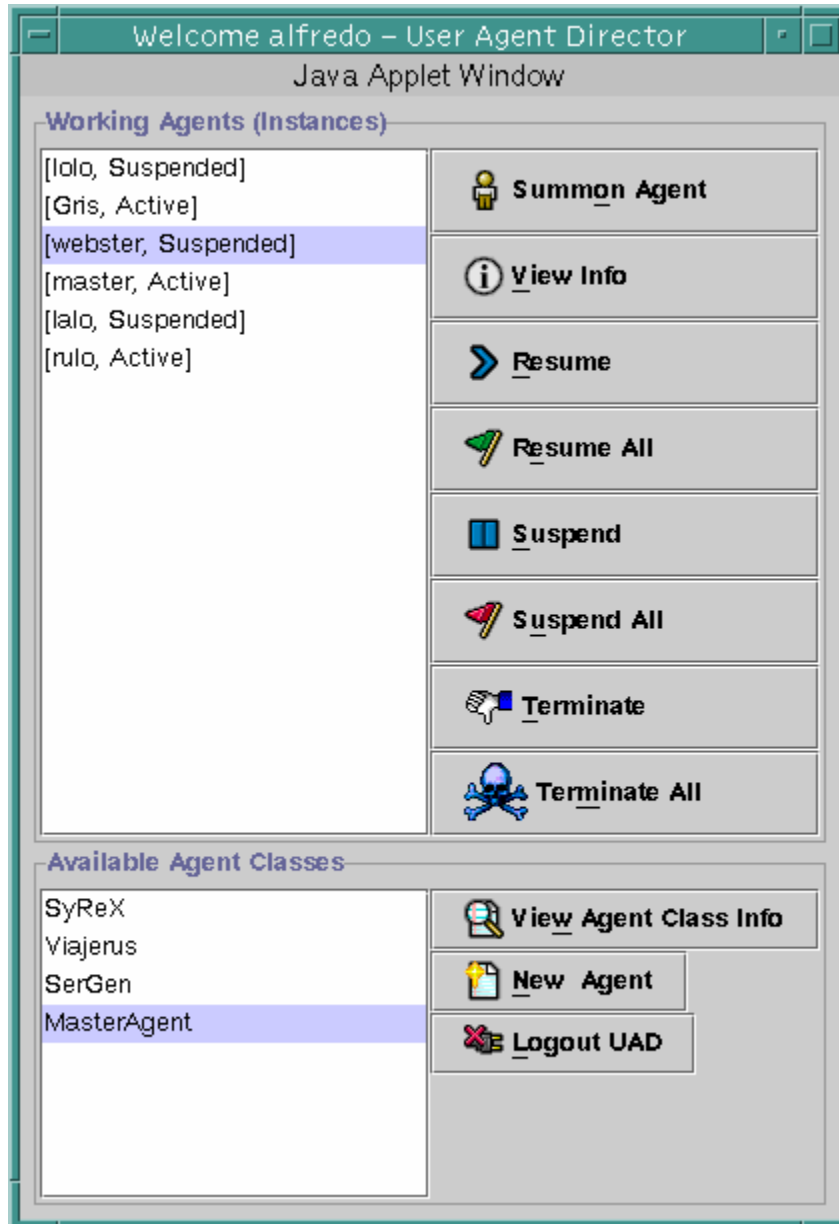


Figura 5.3 Interfaz del UAD en Java Swing

La interfaz de Swing se divide en dos partes principales, el de las

instancias de agentes y la de las agencias o clases de agentes disponibles. La primera muestra un listado de las instancias de agentes que posee el usuario, no importando de que clase sean, muestra también el estado en que se encuentra ese agente. El panel de la derecha del listado es una serie de botones que controlan a los agentes, ya sea uno por uno o todos de una sola vez. Entre las posibles acciones que el usuario puede realizar con sus agentes están:

Suspender un agente o todos los agentes del usuario

Terminar un agente o todos los agentes del usuario

Reactivar un agentes o todos los agentes del usuario

Ver la información relacionada a un agente en específico

El segundo panel de la interfaz es la de las clases de agentes disponibles en la biblioteca digital y la salida del sistema. Aquí se muestra un listado de las agencias que posee la biblioteca digital así como su estado actual (activa o suspendida). A la derecha del listado se encuentran los botones que permiten crear un nuevo agente, ver la información de una agencia en específico, así como salir del sistema. Cada vez que le usuario crea un nuevo agente, tiene la opción de darle un nombre a manera de personalización, de ahí que en el listado solo aparece el nombre que el usuario le puso a su agente.

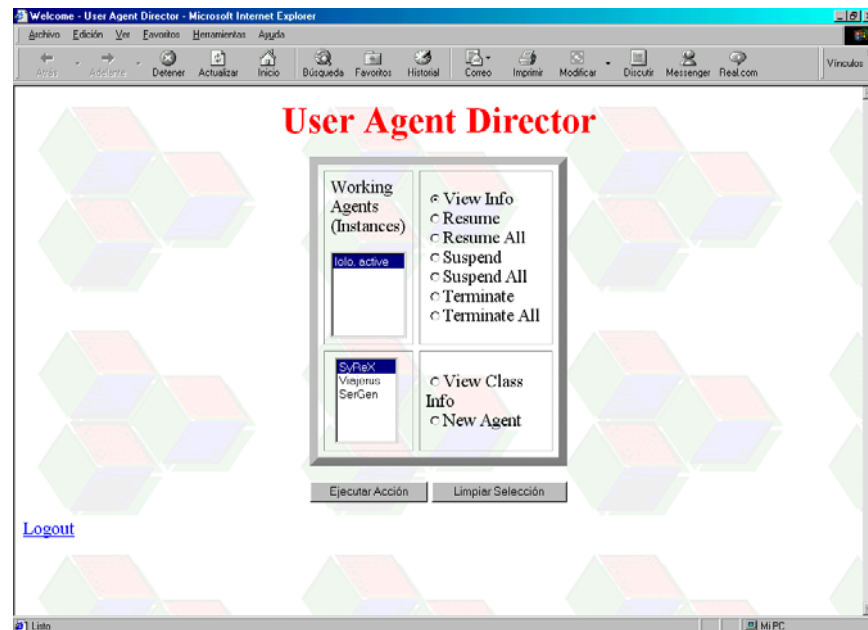


Figura 5.4 Interfaz del UAD en Java Servlets

La interfaz de Java servlets posee la misma funcionalidad que la de Swing, solo que ésta no requiere de aditamentos extras porque conserva la principal característica del Servlet: se ejecuta en el servidor. Esta interfaz



está hecha en HTML y la funcionalidad está implementada en Java Servlets.

#### *5.3.1.2 La Agencia del UAD*

Este es un servidor que recibe y manda peticiones. Recibe peticiones de la interfaz del UAD, de la Interfaz del AAD y de las agencias. También funciona de intermediario entre las interfaces y las agencias pues manda peticiones a estas últimas.

La agencia del UAD recibe peticiones de la interfaz del mismo cuando un usuario decide cambiar el estado de su agente o crear un nuevo agente para él, una vez que hace esto, la interfaz establece una conexión con el servidor del UAD, una vez hecho esto le pasa la petición, el servidor la procesa, realiza las operaciones que le pertenecen y después hace una conexión con la agencia respectiva para realizar la petición del usuario.

Además de la interfaz, las agencias también pueden hacer peticiones al UAD, esto debido al esquema de U-DL-A. Un usuario inicialmente tendrá acceso al UAD y a otras aplicaciones mediante un espacio personalizado [Carballo 2000]; además del UAD el usuario puede crear o cambiar el estado de sus agentes en las mismas agencias, pero a pesar de esa independencia del UAD, las agencias deben de notificarle al servidor del UAD las peticiones del usuario. Para el caso específico de la creación de un agente, el servidor es el que se encarga de generar el número de identificación del agente y se lo pasa a la agencia para que lo use.

#### *5.3.1.3 El servidor del UAD*

Este servidor se encarga de realizar algunos procesos internos del UAD, como actualización de la base de datos del UAD, verificación del status de las agencias y de la agencia misma del UAD. En caso de que alguna agencia no se encuentre disponible por diversas causas (mantenimiento, fallo en la red, o en la máquina donde reside la aplicación), el servidor se encarga de notificar a la interfaz del UAD que la agencia no se encuentra disponible por el momento. En caso de que el que no se encuentre disponible sea la agencia del UAD.

### *5.3.2 Módulo AAD*

Dentro de este módulo se realizan todos los procesos y funciones relacionadas con el Director de Agentes de Usuario. Este módulo consta de 3 partes o módulos, la Interfaz, el servidor y la agencia del UAD.

#### *5.3.2.1 Interfaz del AAD*

La interfaz del administrador de la biblioteca digital, por medio de esta interfaz, se puede dar mantenimiento a las agencias de la biblioteca digital. Cuando una nueva agencia se va a incorporar a la biblioteca digital, el administrador tiene que registrarla proporcionando algunos

datos básicos para que el UAD pueda establecer una comunicación con ella. Para el caso en que se tenga que dar mantenimiento a una agencia o ésta provoque algunos problemas, el administrador puede suspenderla para que no se creen mas agentes de ella, hasta que el problema este resuelto. Algunas de las funcionalidades de la interfaz del AAD son:

Cambio de estado de agencias

Cambio de estado de un agente en específico.- debido a que los agentes poseen un número de identificación único, se puede suspender a uno en particular

Cambio de estados de agentes por usuario.- es posible modificar el estado de todos los agentes de un usuario en particular.

Cambio de estados de agentes por agencia.- es posible modificar el estado de todos los agentes que pertenezcan a una agencia en particular. Por lo general, esto debido a mantenimiento de la agencia.

Alta de nuevas agencias

Visualización del UAD de un usuario en específico.- para el caso de las tareas de mantenimiento, el administrador puede solicitar se le presente el Director de Agentes de un Usuario en especial

Las herramienta que se usó para implementar la interfaz del AAD fue Java Swing. Ver Figura 5.5



Figura 5.5 Interfaz del AAD

### 5.3.2.2 La Agencia del AAD

Aunque en realidad no es una agencia como tal, su funcionalidad es muy parecida a la agencia del UAD, es una mediadora entre la interfaz del AAD, la interfaz del UAD y las agencias. La agencia del UAD es un servidor que acepta peticiones única y exclusivamente de la interfaz del AAD y hace peticiones a las agencias y a la interfaz del UAD.

El administrador de la biblioteca digital, por medio de la interfaz del AAD,

da mantenimiento a las agencias, cada vez que desea realizar algún cambio, ya sea cambio de estado de una agencia o incorporación de una nueva agencia, la interfaz se comunica con el servidor del AAD, el cual luego de haber realizado las operaciones que le competen, establece una comunicación con la agencia o agencias en cuestión para hacerles la petición.

### 5.3.2.3 El servidor del AAD

Al igual que el servidor del UAD, se encarga de los procesos internos del AAD, como actualización de la base de datos, verificación del estado de las agencias, del servidor del UAD, de la agencia del UAD y de la agencia del ADD. Digamos que este servidor se encarga de monitorear casi todo el sistema, a excepción de las interfaces del sistema.

### 5.3.3 Módulo de Instalación

Este módulo, como su nombre lo indica, se encarga de preparar el ambiente para que el sistema se ejecute sin problemas, en decir, se encarga de la instalación de la base de datos (creación de la base de datos, tablas, permisos apropiados, etc.), de los servidores, de los jar's ( *Java Archives* ) que contienen al sistema, entre otras cosas. Este módulo debe de ser ejecutado previo a la ejecución del sistema.

### 5.3.4 DBMS

Se encarga de ejecutar todos los procesos relacionados con la base de datos (consultas, manipulación de tablas, etc.). Debido a que este proyecto hace un uso extenso de base de datos, este módulo es muy importante, pues sin éste, el sistema no podría funcionar adecuadamente. Además de lo anterior, se requiere uso de la base de datos para recuperar la información que el sistema necesita de U-DL-A como por ejemplo los usuarios de la biblioteca digital.

índice   preliminar   figuras   tablas   1   2   3   4   5   6   7   A   referencias

Cocoletzi Moreno, H. 2001. [Servicios generales de una Biblioteca Digital](#). Tesis Licenciatura. Ingeniería en Sistemas Computacionales. Departamento de Ingeniería en Sistemas Computacionales, Escuela de Ingeniería, Universidad de las Américas-Puebla. Mayo.

Derechos Reservados © 2001, Universidad de las Américas-Puebla.