

Previo a la implementación o desarrollo de cualquier proyecto o tarea, independientemente si es de computación o no, debe de existir un plan, una estrategia, un esquema que indique los procedimientos necesarios para el desarrollo de dicho proyecto. En el desarrollo de software a este plan se le llama diseño. El diseño es la parte fundamental para el éxito de todo proyecto, no solo de software sino de cualquier actividad en general [Pressman 1998 citado en Proal 2000]. El diseño posee muchas ventajas para el desarrollador de software, le ahorra tiempo, dinero, esfuerzo, así como permite un mejor aprovechamiento de recursos humanos y económicos.

Como se mencionó en el capítulo de trabajo relacionado, SerGen tiene dos predecesores principales: AGS [Sánchez 1996] y MICK [Barceinas 1997]. Ambos usan una misma arquitectura de biblioteca digital y mismo protocolo de comunicación para los agentes que trabajan en este contexto.

### 3.1 Términos Utilizados

### 3.2 Diseño

### 3.3 Modelo de Datos

### 3.4 Protocolo de Comunicación

### 3.5 Componentes del Proyecto

### 3.1 TÉRMINOS UTILIZADOS

Antes de comenzar con la descripción del diseño, es necesario explicar las palabras que se utilizan en este capítulo para evitar confusiones. En este documento se usan varios términos que se usan como sinónimos:

Agencia, Servicio, Aplicación.- Es un sistema que proporciona agentes cuando se le solicita. Pueden existir agencias de varios tipos, dependiendo de las tareas que puedan realizar sus agentes. Dentro del contexto de SerGen una agencia y una aplicación son lo mismo.

Agente, Instancia de Agente.- una entidad independiente que realiza tareas para un usuario en específico. Los agentes son provistos por las agencias. Se dice también instancia de agente porque se genera a partir de una agencia, es decir se crea un nuevo agente de una agencia.

Protocolo.- Estándar de Comunicación.- una metodología mediante la

cual se comunican los diferentes componentes de la biblioteca digital en el contexto de agentes.

Servidor.- Como su nombre lo indica, un sistema que recibe peticiones de los componentes de la biblioteca digital.

### 3.2 DISEÑO

A grandes rasgos, cuando se empezó a diseñar el proyecto se tomaron en consideración algunos aspectos importantes que repercuten en la forma de desarrollar el proyecto

El sistema no es una aplicación independiente ( *stand alone* ), sino que es parte de una biblioteca digital (U-DL-A) y por ende interactuará con las demás aplicaciones existentes dentro de ella.

El sistema deberá ser independiente de arquitectura o plataforma de la máquina que se esté usando tanto por el usuario como por los servidores de la biblioteca digital.

El sistema usará Internet como su medio principal de interacción con el usuario final. Esto quiere decir que el usuario interactuará con el sistema vía un navegador de web.

La forma de comunicación entre el Director de Agentes de Usuario (UAD), el Director de Agencias (AAD), las agencias y las aplicaciones relacionadas con éste debe ser estándar para que en dado caso que se anexe una nueva agencia, ésta pueda comunicarse o ser contactada de una manera transparente.

El sistema seguirá los estándares y convenciones estipulados por el Laboratorio de Tecnologías Interactivas y Cooperativas ICT.

El sistema propondrá un protocolo de comunicación que facilitará la interacción entre los componentes de la biblioteca digital. Ver figura 1.1

#### 3.2.1 Arquitectura de U-DL-A

Como se mencionó anteriormente, la arquitectura de U-DL-A posee componentes diversos los cuales tienen la necesidad de interactuar entre sí. La Figura 1.1 muestra la arquitectura de U-DL-A, una arquitectura que innova en muchos aspectos pero que debido a esto se dificulta el proceso de comunicación. Entre sus componentes. U-DL-A es:

**Altamente Distribuida** .- puesto que tanto los componentes así como los

acervos digitales no se encuentran en un solo servidor. U-DL-A propone eliminar limitaciones físicas tanto para el usuario que necesita la información, como para sus componentes. Esto significa que mientras sus servicios estén ejecutándose en un servidor, los acervos digitales que usan estos servicios y los administradores de estos, se encuentran en otro dominio. Como ejemplo se puede mencionar a SyReX [Ramírez 2001], una agencia que proporciona agentes recomendadores para el usuario. Esta agencia además de proporcionar los servicios mencionados, tiene que ser administrada por la biblioteca digital, es decir los directores de Agencias y de Agentes. Debido a que la agencia es independiente de los administradores y de los demás componentes de la biblioteca digital, el proceso de comunicación entre ellos es un tanto difícil, puesto que no poseen un lenguaje común de comunicación. El verdadero problema de esto surge cuando se tienen diversas agencias en la biblioteca digital, y todas deben de ser administradas por los directores. Dando como resultado un grave problema de comunicación, pues el lenguaje que maneja cada agencia es muy diferente entre sí.

**Extensible** .- U-DL-A esta diseñada de forma tal que pueda ser extendida sin problema alguno, esto quiere decir que si una agencia nueva se quiere agregar a la Biblioteca Digital su adición debe ser transparente para ésta. Debido a que la mayoría de las agencias son desarrolladas independientes, no poseen los mismos métodos o funciones de comunicación que las demás, provocando un problema para los administradores de la biblioteca digital, pues habría que comunicarse de manera diferente con cada agencia. La falta de un protocolo de comunicación provoca que el administrador genere un nuevo módulo de comunicación para cada agencia nueva que llega.

Debido a lo anterior, SerGen surge como una solución a estos problemas de comunicación. En esta tesis se propone un protocolo de comunicación para las agencias, los agentes y sus directores respectivos, mediante el cual se logre una comunicación efectiva y que permita la extensibilidad de U-DL-A. En Adición al protocolo de comunicación, esta tesis también se enfoca a los directores de agentes y agencias, indispensables para controlar a los servicios de agentes.

Enfocándose a los aspectos técnicos, en vista de que se requería que la aplicación estuviera disponible vía Internet, se tomó la decisión de desarrollar el sistema en el lenguaje de programación Java, este lenguaje además de poseer una utilería de ventanas por ejemplo Java Swing que hacen que una aplicación desarrollada en Java se adapte a la plataforma del usuario de tal manera que aparente ser una aplicación más; también posee un estándar en cuestiones de base de datos llamado JDBC (Java Database Connectivity) que permite una conexión más rápida a la base de datos, esto gracias al uso de puertos lógicos. Otra ventaja de usar JDBC es que es independiente del manejador de bases de datos (DBMS), lo cual quiere decir que el desarrollador no tendrá que lidiar con rutinas o

métodos exclusivos de un DBMS en específico.

### 3.3 MODELO DE DATOS

El modelado de datos o modelado de información es una técnica que se enfoca exclusivamente en los datos, representando una "red de datos" que existe en un sistema determinado. Este modelado provee al analista y diseñador de base de datos una visión en detalle de los datos y de las relaciones que los gobiernan [Pressman 1992].

Para poder crear las tablas del sistema, dentro de las cuales se almacenará la información pertinente para que el sistema funcione correctamente se requirió realizar un diagrama entidad - relación para que nos dé una vista panorámica de los datos a utilizar. Una vez realizados los procesos de normalización, se obtuvo el siguiente modelo. Ver Figura 3.2

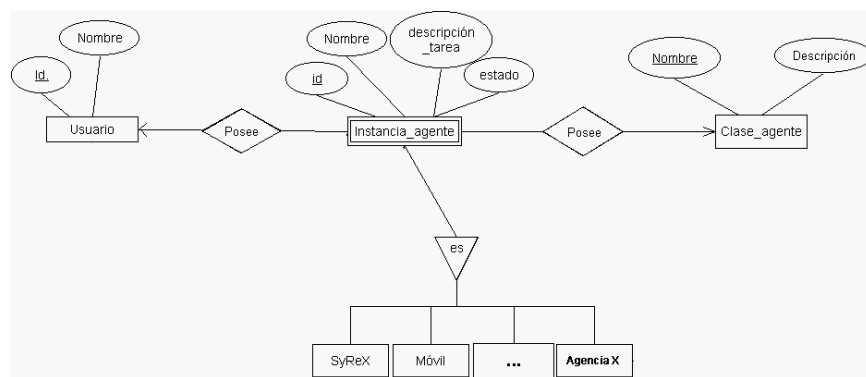


Figura 3.2 Modelo Entidad-Relación (Se omiten atributos por simplicidad)

### 3.4 PROTOCOLO DE COMUNICACIÓN

Una característica importante de U-DL-A es que es una biblioteca digital altamente distribuida, esto quiere decir que puede o no estar ubicada físicamente en una misma máquina, sino que puede hacer uso de varios servidores para cumplir con su cometido.

Para que el Director de Agentes de Usuario (UAD) y el Director de Agencias (AAD) se puedan comunicar con las diversas agencias que existen y existirán en U-DL-A se creó un protocolo de comunicación mediante el cual las agencias (aplicaciones basadas en agentes), el UAD, el AAD, el administrador y el usuario final de la biblioteca digital

interactúan entre sí.

El mecanismo de comunicación se basa en la arquitectura propuesta por [Sánchez 1996] en donde esencialmente ésta se compone de tres partes, el módulo de Servicios de Biblioteca Activa (ALiS), el Administrador de Agentes de Usuario (UAM) y el Director de Agentes de Usuario (UAD). Las relaciones entre estos componentes se ilustran en la Figura 3.3.

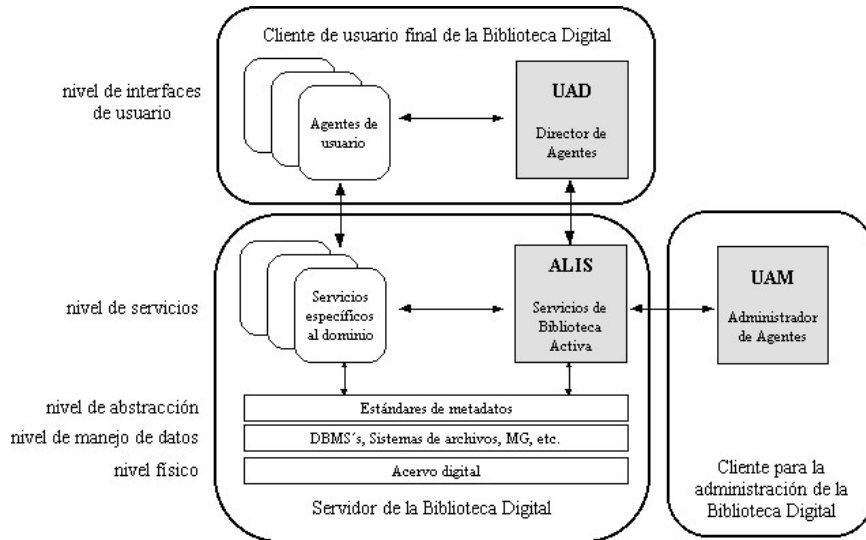


Figura 3.3 Arquitectura de los servicios de agentes (adaptada de [Chevalier 2000])

ALiS se encarga de almacenar las representaciones de las abstracciones de los objetos de agentes en el momento de la ejecución, en cada servidor de la biblioteca digital existe una base de datos que almacena a los diferentes tipos de agentes. Posee diversos métodos que permiten administrar las instancias de agentes, las clases de agentes y las agencias. Este módulo es el más importante de los servicios de agentes. En el contexto de SerGen, ALiS no existe como un módulo aislado, sino que es incluido dentro cada uno de los servidores de SerGen, esto porque cada servidor requiere de cierta funcionalidad que otros no necesitan. Esto permite que las interfaces, servidores y procesos *daemon* funcionen correctamente.

El UAM (Administrador de Agentes de Usuario) es una interfaz para el administrador de la biblioteca digital, por medio de la cual se le da mantenimiento a los servicios de agentes de usuario de la biblioteca digital. Por medio de ella se puede agregar o eliminar una agencia, además de verificar si los agentes existentes no tienen algún problema de funcionamiento. Dentro del contexto de SerGen al UAM se le conoce como AAD (Director de Agencias). La razón por la cual se le cambió el nombre es porque provoca cierta confusión con el UAD, haciendo pensar que el UAM y el UAD realizan la misma tarea..

El UAD es la interfaz para el usuario final de la biblioteca, mediante ella el

usuario administra, crea, elimina o cambio el estado de sus agentes. Debido a que U-DL-A es una biblioteca digital con sistemas multiagentes se ha propuesto esta interfaz para que el usuario no se confunda entre tantas aplicaciones existentes. Al usuario se le presenta un listado de sus agentes disponibles mediante la interfaz principal del UAD.

Toda agencia se comunicará con el UAD y el AAD mediante un estándar de comunicación. Además de dicha comunicación, también existe la posibilidad de que el usuario solicite los servicios de usuarios desde otros ambientes o aplicaciones (por ejemplo un espacio personal [Carballo 2000]), por lo que este protocolo también se aplica.

En las siguientes tablas se listan los métodos de los que dispone el protocolo de comunicación para la comunicación entre los sistemas de agentes y los componentes de la arquitectura de agentes. En el siguiente capítulo se detalla más éste protocolo.

**Tabla 3.1 Mensajes del servidor del UAD y AAD a las Agencias.**

```
getAgentStatus()
isAlive()
newAgent()
resumeAgent()
resumeAllAgents()
setKey();
summonAgent()
suspendAgent()
suspendAllAgents()
terminateAgent()
terminateAllAgents()
```

Toda agencia debe de incluir estas funciones básicas mediante las cuales los servidores del UAD y AAD pueda informar a estas agencias las decisiones del usuario.

**Tabla 3.2 Mensajes de las Agencias a la Agencia del UAD.**

```
addAgency()
deleteAgency()
getAgencyId()
getAgencyStatus()
getAgentDescription()
getAgentStatus()
getName()
getPort()
getUrl()
isAlive()
newAgent()
resumeAgent()
resumeAllAgents()
setAgencyStatus()
setAgentDescription()
suspendAgent()
suspendAllAgents()
terminateAgent()
terminateAllAgents()
```

El UAD no es una interfaz indispensable, el usuario puede también a través de otras aplicaciones como MiBiblio [Carballo 2000] cambiar el estado de sus agentes.

Además de los protocolos arriba descritos, también se crearon protocolos de comunicación para las interfaces de los directores y sus respectivos servidores, esto para hacer independientes de interfaz al AAD y al UAD, de tal forma que las interfaces no influyen en el servidor. Las siguientes

Tablas muestran los métodos propuestos.

**Tabla 3.3 Mensajes de la Interfaz al servidor del AAD.**

AddAgency()
AddUad()
deleteAgency()
deleteUad()
getAgencyId()
getAgencyStatus()
getAgentDescription()
getAgentStatus
getName()
getPort()
getUrl()
isAlive()
resumeAgent()
resumeAllAgents()
resumeAllAgents(user)
resumeAllAgents(user, agency)
setAgencyStatus()
setUadStatus()
suspendAgent()
suspendAllAgents()
suspendAllAgents(user)
suspendAllAgents(user, agency)
terminateAgent()
terminateAllAgents()
terminateAllAgents(user)
terminateAllAgents(user, agency)



El servidor del AAD es una parte vital de la biblioteca digital, pues en el recae la mayor tarea de administración, se encarga de administrar agencias y agentes. Su interfaz le hace peticiones y el se encarga de comunicarse con las agencias correspondientes. A través de el se resuelven problemas de servicios caídos o no disponibles, agentes que no responden, entre otras tareas.

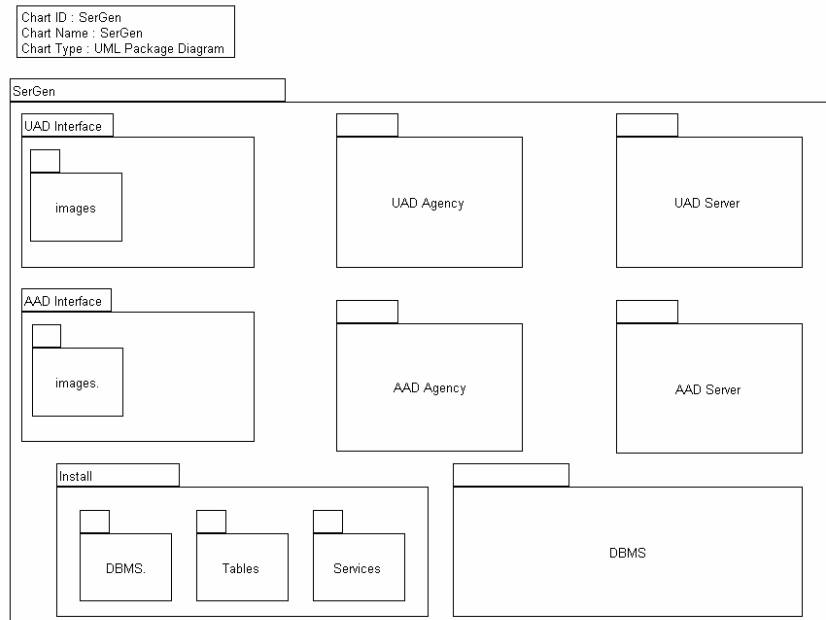
Por su lado, el servidor del UAD también es independiente de su interfaz y posee métodos muy similares a los del servidor del AAD.

**Tabla 3.4 Mensajes de la Interfaz al servidor del UAD.**

<code>getAgentDescription()</code>
<code>getAgentStatus</code>
<code>getName()</code>
<code>getPort()</code>
<code>getUrl()</code>
<code>isAlive()</code>
<code>resumeAgent()</code>
<code>resumeAllAgents()</code>
<code>resumeAllAgents(user)</code>
<code>suspendAgent()</code>
<code>suspendAllAgents()</code>
<code>suspendAllAgents(user)</code>
<code>terminateAgent()</code>
<code>terminateAllAgents()</code>
<code>terminateAllAgents(user)</code>

### 3.5 COMPONENTES DEL PROYECTO

Este proyecto se divide principalmente en ocho partes, las cuales se describen detalladamente en el Apéndice A. A continuación se describe brevemente cada uno de estos módulos. Ver Figura 3.4



**Figura 3.4 Componentes del Proyecto**

**UAD Interface** .- La interfaz principal del UAD, ésta interactúa directamente con el usuario de la biblioteca Digital. La interfaz se comunica con un servidor (Uad Agency) el cual hace se comunica y hace peticiones con las agencias existentes.

**AAD Interface** ,- Es el equivalente al UAM en la arquitectura de [Sánchez 1996]. Una interfaz para el administrador de las agencias de la biblioteca digital. Esta interfaz se comunica con un servidor (Aad Agency) el cual se comunica con las agencias y con el Uad. Realiza una serie de procesos administrativos en la Biblioteca Digital.

**UAD Agency** .- Parte de la funcionalidad de ALiS [Sánchez 1996] se desarrolla aquí. Un servidor que procesa peticiones del Uad y de las agencias. Las rutinas o métodos que maneja este servidor pertenecen al estándar de comunicación implantado, esto quiere decir que cualquier agencia debe de comunicarse con este servidor para hacer peticiones o hacer notificaciones al Uad.

**AAD Agency** .- Un servidor que procesa peticiones de la interfaz del Aad. Las rutinas o métodos que maneja este servidor pertenecen al estándar de comunicación implantado. Por medio de este servidor la interfaz realiza diversas acciones relacionadas con las agencias, como por ejemplo altas, bajas y cambios de estado en las agencias, así como administración de agentes tanto de un usuario como los de toda la biblioteca digital.

**UAD Server** .- También se desarrollan aquí algunos procesos de ALiS. Servidor que se encarga de los procesos internos del UAD y que verifica si el estado de las agencias existentes así como la del Uad en sí. Es el

intermediario entre las agencias y la interfaz de usuario.

**AAD Server** .- Un servidor que se encarga de los procesos internos del AAD y que administra las agencias de la biblioteca digital. Este servidor acepta únicamente las peticiones de la interfaz del AAD y se comunica con las agencias existentes, así como con el servidor del UAD. Al igual que el servidor del Uad, éste también es un intermediario entre las agencias y la interfaz de usuario.

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

Cocolezzi 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.