

Como se mencionó en el capítulo 3, una parte importante de este proyecto fue el diseño e implementación de una metodología mediante la cual las agencias se comuniquen con los directores que existen en la biblioteca digital, como lo son el Director de Agentes (para el usuario), el Director de Agencias (para el administrador) o cualquier otra herramienta o sistema que se interrelacione con las agencias.

Cada agencia que ingrese a la biblioteca digital deberá cumplir con ciertos requisitos, que aunque no son vitales para su buen funcionamiento, sí son importantes para que otras aplicaciones puedan interactuar con ellas.

El estándar de comunicación que se definió es bidireccional, es decir se usa para que los directores o un sistema se comuniquen con la agencia y para que ésta se comuniquen con los directores; esto permite que tanto las agencias como los directores tengan un cierto dominio sobre el otro, aunque no total.

Los directores de agentes, el UAD y el AAD, son herramientas que facilitan el control de los agentes y de sus respectivas agencias en la biblioteca digital. Y aunque su presencia no es indispensable, se recomienda su uso para una simple administración de los agentes. Estos directores poseen una serie de rutinas específicas mediante las cuales se comunican con las agencias.

Aunque este estándar de comunicación puede ser usado por otros sistemas para comunicarse con las agencias, en este caso nos enfocaremos más a los directores UAD (de agentes) y AAD (de agencias) ya que estos también son parte importante del proyecto. La forma en la que funciona el estándar es la siguiente, cuando un usuario solicita administrar sus agentes a la interfaz de un director (ya sea UAD o AAD), la interfaz se comunica con su servidor, quien realiza las operaciones necesarias con su base de datos, y también éste se comunica con el servidor de la agencia. Una vez que el servidor de la agencia recibe la petición, realiza las operaciones pertinentes (sobretudo con su base de datos) y regresa una respuesta al servidor del director, quien a su vez notifica a su interfaz. Ver figura B.1

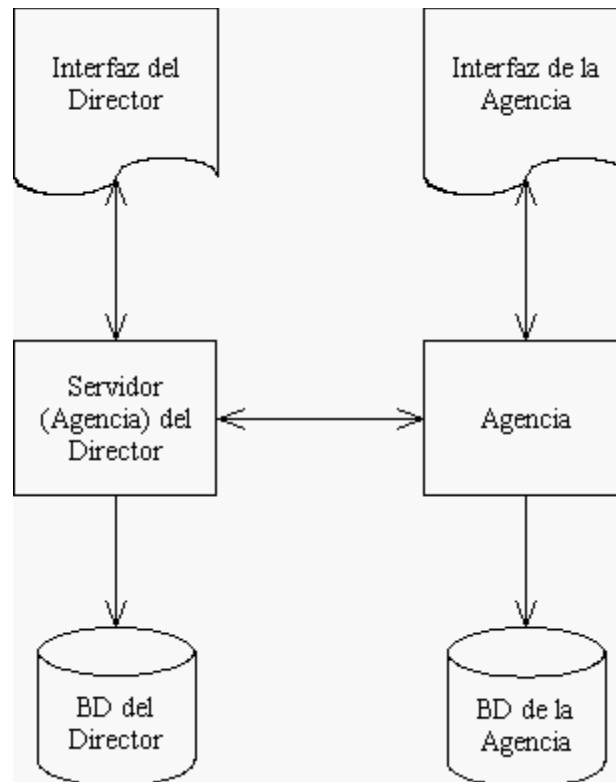


Figura 4.1 Estándar de comunicación

Este proceso también puede invertirse, es decir una agencia puede hacer peticiones al servidor del UAD, este caso surge cuando un usuario accesa a la interfaz de la aplicación y no al UAD y solicita un agente nuevo o cambiar el estado de su agente. Cuando el usuario le solicita a la agencia un nuevo agente, ésta se comunica con el servidor del UAD para que le proporcione un número único con el cual se identificará el agente en toda la biblioteca.

Además de la administración de los agentes, una agencia también puede suspenderse ella misma para evitar que otros usuarios creen más agentes o administren los agentes existentes. Para hacer esto, la agencia le solicita al Director la registre como no disponible, esto previa autenticación para evitar que alguna agencia suspenda a otra.

El proceso de autenticación es requerido para que el servidor del Director realice los cambios pertinentes, incluso las interfaces deben acreditarse como tales cuando se conectan con el servidor del Director.

La lista de métodos que propone el estándar de comunicación (ver capítulo 3) son los siguientes, cabe señalar que estos métodos se hicieron en inglés de acuerdo a los estándares del Laboratorio de Tecnologías Interactivas y Cooperativas ICT, y los estándares de U-DL-A; además de que el idioma inglés posee una cierta universalidad en el mundo y por esa razón permite una mejor comunicación de conocimientos.

Mensajes de los Servidores del UAD y AAD a las Agencias.

`getAgentStatus()`.- obtiene el estado de un agente
`isAlive()`.- permite comprobar si la agencia esta viva
`newAgent()`.- crea un nuevo agente
`resumeAgent()`.- reactiva un agente previamente suspendido
`resumeAllAgents()`.- reactiva todos los agentes del usuario
`setKey();`.- almacena una clave de autenticación para la agencia
`summonAgent()`.- invoca a la agencia o servicio de agentes
`suspendAgent()`.- suspende un agente
`suspendAllAgents()`.- suspende todos los agentes de un usuario
`terminateAgent()`.- elimina a un agente
`terminateAllAgents()`.- elimina a todos los agentes del un usuario

Mensajes de las Agencias a la Agencia del UAD.

`addAgency()`.- agregar una agencia
`deleteAgency()`.- eliminar una agencia
`getAgencyId()`.- obtiene el id de una agencia
`getAgencyStatus()`.- obtiene el estado de la agencia
`getAgentDescription()`.- obtiene la descripción de un agente
`getAgentStatus()`.- obtiene el estado de un agente
`getName()`.- obtiene el nombre de la agencia del UAD
`getPort()`.- obtiene el puerto de la agencia del UAD
`getUrl()`.- obtiene el URL de la agencia del UAD
`isAlive()`.- permite saber si esta viva la agencia del uad
`newAgent()`.- crea un agente nuevo
`resumeAgent()`.- reactiva un agente
`resumeAllAgents()`.- reactiva todos los agentes de un usuario

setAgencyStatus().- cambia de estado una agencia
setAgentDescription().- cambia la descripción de un agente
suspendAgent().- suspende un agente
suspendAllAgents().- suspende todos los agentes de un usuario
terminateAgent().- elimina un agente
terminateAllAgents().- elimina a todos los agentes

Mensajes de la Interfaz al Servidor del AAD.

AddAgency().- agregar una agencia
AddUad().- agregar un UAD nuevo
deleteAgency().- eliminar una agencia
deleteUad().- eliminar un UAD
getAgencyId().- obtiene el id de la agencia
getAgencyStatus().- obtiene el estado de la agencia
getAgentDescription().- obtiene la descripción de la agencia
getAgentStatus.- obtiene el estado del agente
getName().- obtiene el nombre del servidor
getPort().- obtiene el numero de puerto del servidor
getUrl().- obtiene el URL del servidor
isAlive().- permite saber si el servidor esta vivo o no
resumeAgent().- activa un agente
resumeAllAgents().- activa todos los agentes de la biblioteca digital
resumeAllAgents(user).- activa todos los agentes de un usuario
resumeAllAgents(user, agency).- activa todos los agentes de un usuario, por una agen
especifico
setAgencyStatus().- cambia el estado de una agencia
setUadStatus().- cambia el estado de un UAD

suspendAgent().- suspende un agente
suspendAllAgents().- suspende todos los agentes de la biblioteca digital
suspendAllAgents(user).- suspende todos los agentes de un usuario
suspendAllAgents(user, agency).- suspende todos los agentes de un usuario
terminateAgent().- termina un agente
terminateAllAgents().- termina todos los agentes de la biblioteca digital
terminateAllAgents(user).- termina todos los agentes de un usuario
terminateAllAgents(user, agency).- termina todos los agentes de un usuario por agente

Mensajes de la Interfaz al Servidor del UAD.

getAgentDescription().- obtiene la descripción de la agencia
getAgentStatus.- obtiene el estado del agente
getName().- obtiene el nombre del servidor
getPort().- obtiene el número de puerto del servidor
getUrl().- obtiene el URL del servidor
isAlive().- permite saber si el servidor está vivo o no
resumeAgent().- activa un agente
resumeAllAgents(user).- activa todos los agentes de un usuario
suspendAgent().- suspende un agente
suspendAllAgents(user).- suspende todos los agentes de un usuario
terminateAgent().- termina un agente
terminateAllAgents(user).- termina todos los agentes de un 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.