

# Descripción de un Sistema Manejador de Objetos Web

**José L. Aguilar,**

Centro de Microcomputación y Sistemas Distribuidos (CEMSID)

Escuela de Ingeniería de Sistemas, Facultad de Ingeniería

Universidad de los Andes

Mérida – Venezuela

aguilar@ula.ve

y

**Juan Vizcarrondo**

Postgrado de Computación. Facultad de Ingeniería

Universidad de los Andes

Mérida – Venezuela.

## **Abstract**

The number of applications, systems and services developed for the Web is very large. In some cases, there are not supports at the level of the operating systems for them. One alternative is to develop a model of operating system, called SOW, which supports and manages a set of services in a heterogeneous and dynamic environment like Internet. One of the subsystems of this operating system must be the Management System of Web Object. It manages the web objects migration and the web objects replication on the SOW. In this paper we present the design of this subsystem.

**Keywords:** Operating Systems, Web Objects Migration, Web Objects Replication, Management System of Web Object.

## **Resumen**

La cantidad de sistemas, servicios y aplicaciones desarrolladas para la web han crecido considerablemente. En algunos casos, el soporte por parte de los sistemas operativos existentes a cada uno de ellos no es el esperado. Como alternativa de solución a esta necesidad, se plantea un modelo del sistema operativo denominado SOW, el cual soporta y maneja un conjunto de servicios en un contexto heterogéneo, dinámico y adaptativo, bajo el enfoque de reconfiguración de las aplicaciones. El SOW esta conformado por cuatro subsistemas que llevan a cabo una serie de funciones coordinadas. Uno de estos subsistemas es el Subsistema Manejador de Objetos Web, el cual es el encargado de la migración y replicación de los objetos web existentes en la SOW. En este trabajo presentaremos dicho subsistema.

**Palabras claves:** Sistemas Operativos Web, Migración de Objetos, Replicación de Objetos, Manejo de Objetos Web.

## 1. INTRODUCCIÓN

Dada la amplia variedad de servicios inimaginables que surgen cada día en la web, resulta difícil diseñar un sistema operativo que apoye y use a cada uno de esos servicios. De éstas necesidades surge un área llamada Sistema Operativo Web (SOW), que tiene como objetivo principal proveer una plataforma que permita a los usuarios beneficiarse del potencial computacional ofrecido en la web, a través del compartimiento de recursos y de la resolución de los problemas de heterogeneidad y adaptabilidad dinámica presentes en la misma [1, 6, 9, 11, 13, 14]. Así, para alcanzar un rendimiento óptimo en un ambiente dinámico de recursos distribuidos como la Internet, el SOW debe ser configurable y capaz de adaptarse a los cambios en cuanto a la disponibilidad de recursos (de software y de hardware). Teniendo en cuenta esas consideraciones, el modelo de SOW presentado en [1] propone una serie de aspectos para proveer servicios que se adecuen a esos rasgos especiales de la web. Así, nuestro SOW presenta un diseño que cuenta con las herramientas asociadas para permitir el uso transparente e interactivo de los recursos accesibles a través de la red, en cualquier momento que un usuario lo requiera. Esos servicios pueden ser hardware, software, o una combinación de ambos. El usuario sólo necesita comprender la interfaz del SOW, sin importarle como su solicitud es satisfecha.

Existen varias propuestas para el manejo e integración de los recursos computacionales disponibles en la Web. Quizás el proyecto más general sea el *WOS<sup>TM</sup>* [11], ya que permite el manejo e integración de los recursos tratando el problema de la heterogeneidad y volatilidad en la Web. Este proyecto, al igual que nuestra propuesta de SOW, está basado en la idea del uso de versiones como solución a esos problemas. En [11] se ha planteado la creación de una interfase llamada WOSMPI, que le permitirá al *WOS<sup>TM</sup>* migrar objetos en Internet, para que puedan ser usados por aplicaciones de calculo científico, donde se requiere el movimiento de enormes cantidades de datos. Otra propuesta para el manejo de recursos en la Web parecida, es la arquitectura *Jini* de SUN Microsystems [12]. *Jini* provee servicios de localización de recursos (conocidos como *lookups*), y aplica la idea de agrupar recursos en federaciones.

El objetivo del Subsistema Manejador de Objetos Web (SMOW) es proveer al SOW de mecanismos que le permiten adaptarse a la naturaleza dinámica de los recursos en Internet. Para esto, nuestro SOW se adapta a la volatilidad de los recursos e intenta disminuir la latencia debido a la distancia entre los nodos, tratando de tener más cerca de los usuarios la información que requieren, valiéndose de mecanismos como la replicación y migración de objetos. Así, la *replicación de objetos* es usada para colocar cerca de los usuarios la información que requieren valiéndose de caches en la web. Las caches en la Web tienen la capacidad de almacenar objetos cerca de los usuarios, para que no sea necesario desplazar un objeto de su ubicación original cada vez que se requiera [2, 3, 15]. Por otro lado, la *migración de objetos* provee un mecanismo que permite mejorar la disponibilidad de los recursos, al equilibrar la carga de trabajo entre los diferentes nodos migrando los objetos hacia nodos ociosos, y al reducir las comunicaciones entre los diferentes nodos al colocar los objetos en los sitios de mayor demanda [4, 7, 17]. En Internet existen dos tipos de objetos [5]: objetos estáticos que son colocados en un servidor (fotos, textos, archivos HTML, etc.) y objetos dinámicos que son construidos por programas que se ejecutan cuando son llamados. La migración de los objetos estáticos resulta fácil de implementar, ya que solo es mover un archivo a un nodo [5]. La movilidad de objetos dinámicos por Internet puede realizarse de 2 formas [5]: la primera se refiere a la migración de pequeños fragmentos de código del objeto (procedimientos) a sitios remotos que lo requieran, la segunda describe la migración total del código del objeto a otros sitios.

Finalmente, existen varias técnicas para localizar objetos después de que han sido migrados. La primera se refiere a enviar un mensaje a todos los nodos de la red anunciando la nueva dirección cada vez que se realiza una migración, la otra se refiere a enviar un mensaje global de búsqueda cada vez que se encuentre una referencia no válida [4, 7, 17], pero estas dos técnicas son muy costosa en sistemas de gran escala como Internet, al enviar un mensaje a cada nodo del sistema. Para solventar esta situación existe una técnica denominada Lazos de Persecución [4, 7, 17], la cual consiste en que cuando un objeto migra se deja en su sitio de origen una indicación de la nueva localidad, facilitando así la localización del objeto. Cuando llega un mensaje solicitando el objeto este es redireccionado a su nueva posición. El problema de esta técnica es en sistemas muy grandes donde un lazo puede ser muy largo, o no ser consultado nuevamente debido a que todos los que tienen vínculos con este objeto ya actualizaron su nueva dirección (o por lo menos parte del lazo). En este artículo nosotros proponemos un mecanismo para resolver este problema que llamamos *trazas de objetos*.

Los SMA han representado una solución para el diseño de sistemas muy grandes donde no se puede realizar un control centralizado y sus ambientes son heterogéneos, abiertos y cambian dinámicamente, como Internet [16]. Por estas razones se propone realizar el Diseño del SOW basado en la teoría de agentes. En este artículo presentamos nuestra propuesta de SMOW usando SMA. Comenzamos realizando una breve introducción del SOW. En la siguiente sección presentamos el diseño del SOW usando SMA. Finalmente, en la última sección definimos los agentes que componen el SMOW.

## 2. ARQUITECTURA DEL SISTEMA OPERATIVO WEB (SOW)

El SOW propuesto posee las siguientes características [1]:

- *Distribuido y Versionado*: Diferentes Aplicaciones Distribuidas y varias versiones de los servicios están corriendo simultáneamente sobre los diferentes sitios de la red.

- *Dinámico*: La web es una entidad que está evolucionando permanentemente, por lo tanto, el SOW debe adaptarse a los cambios imprevistos. Es por ello que nuestro SOW tiene incorporado cierto dinamismo en los subsistemas que lo integran. Por ejemplo, a través de la configuración dinámica de los servicios que ofrece; a través de la actualización permanente de la información sobre los recursos locales y remotos disponibles en cada nodo; a través de la agrupación dinámica de los nodos existentes de acuerdo con ciertas características, y a través de la migración, replicación y seguimiento de objetos web.
- *Abierto*: Nuestro SOW se caracteriza como un sistema abierto desde dos puntos de vista. En primer lugar, aceptará que diversas tecnologías sean usadas en todos los niveles de la red (heterogeneidad). En segundo lugar, permitirá que cualquier nodo de Internet pueda ser incorporado al sistema.
- *Inteligente*: Cada uno de los subsistemas del SOW tendrá algún nivel de inteligencia para el desarrollo de alguna de sus funciones con el fin de optimizar su funcionamiento.

Así, nuestro SOW es un sistema operativo versionado e inteligente que se autoconfigura dinámicamente para permitir un acceso fácil y transparente a los recursos distribuidos sobre la Internet. Nuestro SOW utilizará un Middleware orientado a objetos que debe proveer la estructura necesaria para lograr la descripción de todos los recursos en Internet. Nuestra arquitectura de SOW propuesta en [1] está compuesta por un conjunto de 5 subsistemas:

- *Subsistema Manejador de Recursos (SMR)*: Este subsistema maneja todos los requerimientos hechos al SOW, este funciona como un sistema reactivo de manejo de demandas a través del cual se administran los recursos de un ambiente computacional heterogéneo como la Web.
- *Subsistema Manejador de Repositorios*: En el subsistema es donde se depositan todos los recursos con los que cuenta un nodo, este a su vez se divide en dos subsistemas:
  - a) *Subsistema Manejador de Repositorios Local (SMRL)*: Funciona como un catálogo de los recursos y versiones del dominio local
  - b) *Subsistema Manejador de Repositorios Remotos (SMRR)*: En este subsistema se almacena información sobre los recursos remotos más utilizados por el sistema local, así como los objetos replicados.
- *Subsistema Manejador de Objetos Web (SMOW)*: Es el encargado de realizar la migración y la replicación de objetos en el SOW, para aumentar la disponibilidad del sistema al situarlos en los sitios de mayor demanda o equilibrar la carga. Así, el SMOW debe realizar la selección de los objetos a migrar o replicar, del nodo destino donde se almacenarán, y debe tener implementado los mecanismos necesarios que permitan localizar los objetos migrados, y que garantice la migración y replicación de los objetos.
- *Subsistema Manejador de Comunidades (SMC)*: Comisionado para agrupar los diferentes nodos que componen el SOW en grupos que presenten las mismas características.

### 3. PROPUESTA DE NUESTRO SOW COMO SISTEMA MULTIAGENTES

El diseñar un SOW conlleva tratar con el problema de heterogeneidad presentes en los múltiples servicios que se ofrecen en la web, tanto a nivel de hardware como de software presentes en los diferentes sitios, y con el problema de adaptabilidad debido a la volatilidad de los recursos. Diseñar un sistema de control centralizado para solucionar estos problemas resulta imposible. Los SMA han representado una solución para el diseño de sistemas muy grandes donde no se puede realizar un control centralizado y sus ambientes son heterogéneos, abiertos y cambian dinámicamente como Internet [16]. Por estas razones se propone realizar el diseño de nuestra propuesta de SOW usando SMA. Los SMA proveen métodos dinámicos y mecanismos de inteligencia, que permiten el diseño del SOW como un sistema robusto, donde todos sus componentes asumen sus propios roles y cooperan entre sí para la solución de problemas.

Realizar el diseño del SOW basado en SMA facilita el desarrollo de cada uno de los subsistemas del SOW por separado, logrando reducir la complejidad requerida para solucionar el problema de gestión a nivel global. Así, al diseñar nuestro SOW como un SMA, visto desde el nivel más alto el SOW puede ser considerado como un solo componente, denotado por  $L_0$ , donde la información interna y los procesos en él no son especificados (se ocultan las comunicaciones y los procesos internos del SMA). En el próximo nivel de abstracción  $L_1$ , el componente  $L_0$  puede ser visto como una colección de subsistemas que cumplen con los objetivos del SOW (ver figura 1).

Sistema Multiagentes (L0)	Esta compuesto por (L1)	Satisface Objetivos	Interrelacionado con
SOW	SMR	Administrar e integrar recursos computacionales presentes en el SOW	Usuarios, SMRL, SMRR y SMC
	SMRL	Buscar información en los RL	SMR, SMRR y SMOW
		Mantener coherente la información existente en los RL Y rr	
		Administrar el uso eficiente de los recursos en los RL y RR	
	SMRR	Buscar Información en los RR	SMR, SMRL, SMOW
		Mantener actualizada y coherente la información en los RL	
SMOW	Replicar objetos en los RR	SMRL, SMRR y SMC	
	Migrar objetos hacia nodos de mayor demanda		
SMC	Agrupar nodos con afinidades en comunidades	SMR, SMOW	

Figura 1. Descomposición del SOW en 5 SMA

#### 4. PROPUESTA DEL SUBSISTEMA MANEJADOR DE OBJETOS WEB (SMOW)

En nuestra propuesta de SOW, las replicas son almacenadas en los RR y los objetos web migrados en los RL. El rol desempeñado por el SMOW en lo que respecta a replicación, es el de proveer los mecanismos necesarios para transportar las replicas en Internet hacia los RR, cada vez que es solicitada por el SMRR, además de mover la información necesaria hacia los RL, cada vez que sea requerida por el SMRL, para garantizar la coherencia de los objetos cuando un objeto ha pasado a ser incoherente por la modificación de una replica.

A nivel de la migración de objetos, este es el acto de transferir un objeto entre dos máquinas (entre el nodo origen y el nodo destino), buscando maximizar algún objetivo, como por ejemplo, el balance de la carga entre los nodos. Para esto es necesario proveer un mecanismo que defina que objeto migrar y hacia donde. En nuestra propuesta es necesaria la migración de objetos para aumentar la disponibilidad de los objetos del sistema, además de reducir las comunicaciones entre los diferentes nodos. El SMOW es el encargado de realizar la selección de los objetos y el nodo destino donde se almacenaran, además de la implementación del mecanismo que garantice el transporte de los objetos por los distintos nodos del SOW. Cuando se realiza la migración de objetos, se debe garantizar la localización de estos cada vez que un objeto abandona un nodo y se almacena en otro, para esto en nuestra propuesta de SOW se ha propuesto una estructura basada en trazas, las cuales van desapareciendo a medida que transcurre el tiempo.

Las características a tomar en cuenta en el diseño del SMOW del SOW son:

- Cada Comunidad de nodos del SOW posee un SMOW que administra el movimiento de los objetos y de las replicas en toda la comunidad.
- El SMOW provee los mecanismos para la creación de replicas de objetos y suministrárselos a todos los RR que los soliciten.
- El SMOW provee mecanismos que le permiten realizar la migración de los objetos web en todo el SOW, el cual realiza por petición del SMC. Además, es el encargado de la creación de trazas de objetos en los RL cada vez que se produce la migración de objetos.

A continuación se definen las bases fundamentales de nuestra propuesta de diseño del SMOW, precisando los mecanismos necesarios para realizar la replicación, migración y facilitar la localización de objetos cada vez que son migrados.

##### 4.1. Replicación de Objetos

En nuestro SOW, el SMOW recibe peticiones de replicar un objeto cada vez que algún SMRR necesite almacenar una nueva replica en el RR. El encargado de la selección de los objetos que se van a replicar en el RR es el SMRR, para esto, cada vez que el SMRR necesita hacer una nueva replica se la solicita al SMOW. Cada vez que el SMRR solicita que le sea suministrada la replica, el SMOW revisa el objeto en el RL para verificar que no se encuentra en uso en ese momento, lo bloquea, y transfiere una copia de su contenido al RR. En caso que el objeto se encuentre en uso, el SMOW responde al SMRR que intente más tarde la solicitud.

## 4.2. Migración de Objetos

Para poder realizar la migración de un objeto es necesario conocer previamente la selección del objeto a ser migrado, el sitio de destino, y el mecanismo que permita la localización del objeto [10, 17]. En nuestra propuesta los objetos son migrados para colocarlos en los sitios de mayor demanda, migrando aquellos objetos web dinámicos. Para esto, el SMC cada vez que requiere la migración de un objeto lo solicita al SMOW, el SMOW realiza una comparación entre la demanda en el SMRL origen y el posible destino, y si es mayor la demanda en el SMRL destino se migra el objeto hacia ese nodo. Existen muchos mecanismos para realizar la migración de los objetos [4, 7, 17], en este trabajo se ha seleccionado el mecanismo de la *migración impaciente*, ya que el objeto y las comunicaciones recibidas son transferidas completamente hacia el nodo destino para de esta forma no hacer necesaria que la comunicación con el nodo origen sea restablecida nuevamente, al contrario de lo que ocurre con la migración perezosa y de la copia previa, las cuales solo transportan inicialmente una parte del objeto hacia el nodo destino.

## 4.3. Localización de Objetos Migrados

Existen varias técnicas para localizar objetos después de que han sido migrados [10, 17]. En este trabajo se propone un nuevo mecanismo basado en *trazas de objetos*, inspirado en los Sistemas de Colonias de Hormigas, los cuales son modelos que emulan el comportamiento de colonias de hormigas reales [8]. Las hormigas son capaces de seguir la ruta más corta en su camino de ida y vuelta entre la colonia y una fuente de alimento. Esto es debido a que las hormigas pueden "transmitirse información" entre ellas gracias a que cada una de ellas, al desplazarse, va dejando un rastro de una sustancia llamada feromona (traza) a lo largo del camino seguido, la cual es detectada por otras hormigas, tendiendo a seguir la traza con mayor cantidad de feromona. A su vez, éstas van dejando su propia feromona a lo largo del camino recorrido, y por tanto, lo hacen más atractivo puesto que se ha reforzado el rastro de feromona. Por otro lado, la feromona también se va evaporando con el paso del tiempo provocando que el rastro de feromona sufra cierto debilitamiento. La noción de trazas de feromona dejada por los objetos, es la base de nuestra propuesta de localización de los objetos a implementar en el SOW. El objeto al moverse va dejando el rastro de sus movimientos, lo que podría utilizarse para encontrarlo al ser requerido por un RR. Pero por otro lado, el rastro dejado por los objetos al desplazarse debe ir desapareciendo a medida que transcurre el tiempo.

Nuestra propuesta es que cada vez que un objeto es migrado de algún SMRL de nuestro SOW, es colocado en el RL un lazo de persecución, que permita a los objetos encontrarlos en su ubicación actual. El problema de implementar los lazos de persecución es que a medida que transcurre el tiempo, se van convirtiendo en información no coherente o no son utilizados [17], ya que los objetos a que hacen referencia han migrado a otros sitios o no son utilizados por las caches ya que han actualizado el apuntador al objeto. Para solventar esta situación es que nosotros usamos la noción de las trazas de las hormigas. Los lazos de persecución dejados en el SMRL, consisten en una estructura que contiene la intensidad de la feromona, la cual se va evaporando a medida que transcurre el tiempo hasta desaparecer. Cuando un nodo accede a este lazo de persecución para actualizar la ubicación del objeto, la cantidad de feromona es reforzada (aumentada) por el SMRL, ya que si un SMRR ha actualizado la ubicación en un momento, es posible que otros SMRR aun no lo hayan hecho. Si este lazo no es consultado nuevamente, la cantidad de feromona que contiene el lazo se va evaporando, hasta desaparecer, eliminando de esta manera los lazos basura que no van a ser consultados nuevamente. En nuestro SOW propuesto, el encargado de crear las trazas es el SMOW, y los encargados de actualizarlas son tareas tanto del SMRL como del SMRR.

## 4.4. Roles Desempeñados por el SMOW en el SOW

Los roles en los que se encuentran involucrado el subsistema SMOW del SOW se muestran en la Figura 2.

<b>Acción</b>	<b>Decide</b>	<b>Solicita</b>	<b>Implementa</b>	<b>Almacena</b>	<b>Busca</b>	<b>Actualiza</b>
<i>Migración</i>	SMOW	SMC	SMOW	SMRL destino	-	-
<i>Replicación</i>	SMRR	SMRR	SMOW	SMRR	-	SMRR y SMRL
<i>Trazas</i>	SMOW	SMOW	SMRR y SMRL	SMRL origen	SMRR	SMRR y SMRL

Figura 2. Roles Desempeñados por el SMOW

La migración en el SOW permite ubicar los objetos web en los sitios donde más se les demanda en Internet, para esto el SMOW recibe las peticiones de migración de objetos del SMC y luego realiza la inferencia de si es necesario migrar el objeto al nuevo destino o no. En caso de decidir la migración, el SMOW la implementa almacenando el objeto en el SMRL destino y eliminándolo del SMRL origen. Una vez que el objeto ha sido migrado, la intensidad de las trazas son almacenadas en el SMRL origen, por solicitud del SMOW. La actualización de la traza es responsabilidad del SMRL al recibir la solicitud de acceso al objeto por parte del SMRR, o por el proceso de evaporación de ella.

La replicación de objetos le permite al SOW manejar la disponibilidad de los recursos, además de disminuir el tráfico en la red. Las replicas son almacenadas en los SMRR del SOW, por tanto, estos son los encargados de realizar la solicitud de replicas en el SOW. Además, debe administrar el espacio y la coherencia de las replicas que se encuentran en los RR. El SMOW recibe la solicitud de replicación por parte del SMRR y la implementa replicando el Objeto Web solicitado en el RR. Además, la asignación y desasignación de replicas es tarea del SMRR, la cual realiza por solicitud del SMR.

## 5. DESCRIPCIÓN DE LOS AGENTES DEL SMOW USANDO SMA

Realizar el diseño del SOW basado en SMA facilita el desarrollo de cada uno de los subsistemas del SOW por separado, logrando reducir la complejidad requerida para solucionar el problema de gestión a nivel global. Particularmente, la identificación y descripción de los agentes que intervienen en el SMOW se realiza en base a los actores y casos de usos definidos para el SMOW. Además, es necesario definir las estructuras de datos que usara el SMOW. A continuación se presenta la arquitectura propuesta para el SMOW.

### 5.1. Arquitectura del SMOW

En el SMOW del SOW las operaciones son coordinadas por dos componentes fundamentales (Figura 3), éstos son: el Replicador de Objetos y el Migrador de Objetos. Estos dos componentes constituyen las unidades funcionales básicas de la arquitectura del SMOW.

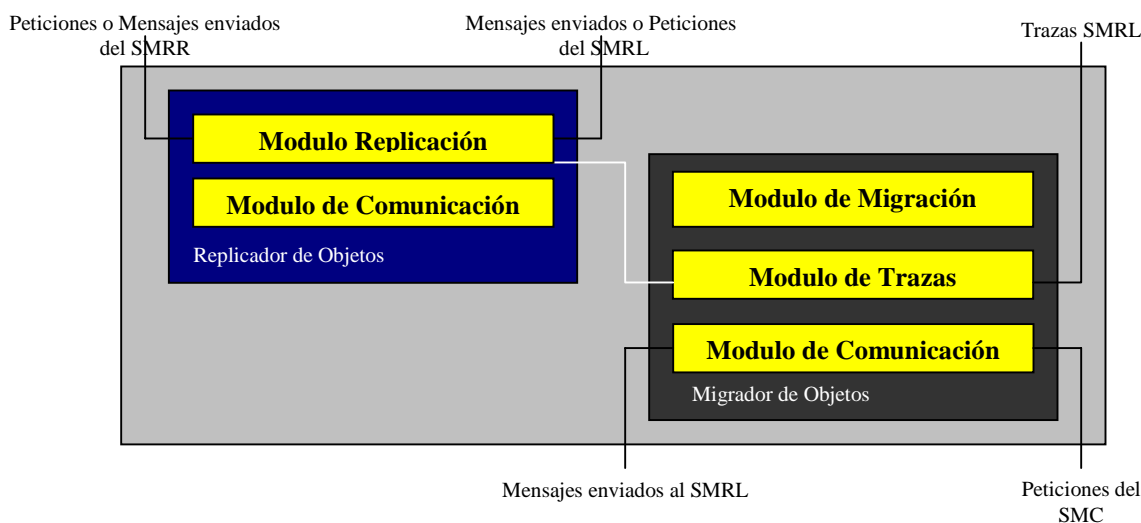


Figura 3. Componentes del SMOW

A continuación se presenta la descripción de cada uno de estos componentes:

- *Replicador de Objetos:* Esta unidad tiene como objetivo el traslado de las replicas de los objetos en el SOW. Para esto utiliza dos módulos, el *Modulo de Replicación*, que le permite realizar las replicas; y el *Modulo de Comunicación*, que le permite la coordinación con el SMRL fuente y el SMRR destino.
- *Administrador de Traslado de Objetos:* Esta unidad tiene como objetivo la migración de objetos web en el SOW y la creación de trazas. Para esto utiliza tres módulos, el *Modulo de Migración de Objetos*, que le permite realizar el traslado de Objetos de un SMRL a otro SMRL, el *Modulo de Trazas*, que le permite la creación de trazas cuando un objeto es migrado; y el *Modulo de Comunicación*, que le permite la interacción con el SMRL fuente en el momento de iniciar la migración.

### 5.2. Estructura de Datos Utilizada por el SMOW

Para facilitar el manejo de los objetos Web es necesario la creación de estructuras de datos. Para esto, el SMOW posee 2 estructuras de datos que le permiten manejar los objetos en el RL y RR. La figura 4 se refiere a la estructura de datos utilizada por el SMOW para manejar los objetos en el RL, la cual le concede el poder de bloquear y cuantificar la demanda de los objetos en el RL. La estructura de datos de la figura 5 le permite al SMOW reiniciar los objetos en el RR una vez que han sido migrados.

Recurso	Versión	Frecuencia	Estado
---------	---------	------------	--------

Figura 4. Estructura de datos utilizada para el manejo de objetos en el RL.

El significado de los campos que conforman la estructura de datos que utiliza el SMOW para el manejo de los objetos Web en el RL es el siguiente:

*Frecuencia:* Se refiere a la cantidad de veces que es requerido un objeto Web en el RR. Es un vector que almacena cantidad de demandas y origen de la misma.

*Recurso:* Identificador del objeto Web.

*Estado:* Se refiere al estado actual del objeto Web, lo cual es usado para la coherencia de estos. Dichos estados pueden ser:

- Modificado.
- Invalido.
- Exclusivo.
- Compartido.

El **Estado** del objeto es usado para bloquear el objeto en el RL, para que no pueda ser modificado por el SMRL, mientras se realizan operaciones de replicación o migración sobre él. El campo **Recurso** facilita la identificación de los objetos en el RL. La **Frecuencia** del objeto se refiere a la demanda del objeto en el RL, el cual le permite al SMOW tomar decisiones de migración sobre los objetos del RL, particularmente sobre si hay que hacerla y hacia donde.

Frecuencia	Recurso	Estado	Versión	Ubicación
------------	---------	--------	---------	-----------

Figura 5. Estructura de datos utilizada para el manejo de las replicas en el RR.

El significado de los campos que conforman la estructura de datos que utiliza el SMRR para el manejo de las replicas en el RR es el siguiente:

*Frecuencia:* Se refiere a la cantidad de veces que es requerido un objeto Web en el RR desde el sitio local.

*Recurso:* Identificador del objeto Web.

*Estado:* Se refiere al estado actual del objeto Web, lo cual es usado para su coherencia. Dichos estados pueden ser:

- Modificado.
- Invalido.
- Exclusivo.
- Compartido.

*Versión:* Versión del objeto web.

*Ubicación:* Sitio Remoto en donde se encuentra localizado el objeto replicado.

La **frecuencia** permite cuantificar la demanda de una determinada replica, la cual es usada por el SMRR para la toma de decisiones de cuales replicas reemplazar y cuales replicas mantener en el RR. Los campos **Recurso** y **Versión** facilitan la identificación de las replicas en el RR. Por ultimo, los campos **Estado** y **Ubicación** son utilizados por el SMRR para el manejo de coherencia de las replicas, y por el SMOW para reiniciar la nueva replica en el RR. El **Estado** de un objeto se refiere al estado actual en que se encuentra la replica en ese momento y la **Ubicación** almacena el apuntador del nodo donde se encuentra ubicado el objeto replicado.

El almacenamiento de las trazas de los objetos Web en el RL se realiza en el formato que se muestra en la Figura 6.

Recurso	Versión	Tasa de evaporación	Intensidad Traza	Ubicación
---------	---------	---------------------	------------------	-----------

Figura 6. Estructura de datos utilizada para almacenar las trazas de los objetos en el RL.

El significado de los campos que conforman la estructura de datos que utiliza el SMOW para almacenar las trazas de los objetos Web es el siguiente:

*Recurso:* Identificador del objeto Web.

*Versión:* Versión del objeto Web.

*Tasa de Evaporación:* La tasa de evaporación que permite eliminar la traza en el transcurso del tiempo.

*Intensidad Traza:* La cantidad de feromona del objeto.

*Ubicación:* Ubicación en donde se encuentra el objeto.

Los campos **Recurso**, **Versión** y **Ubicación** tienen el mismo significado de los campos de las estructura de datos utilizadas para el manejo de los objetos y las replicas en el RL y RR. El campo **Intensidad Traza** contiene la intensidad de la feromona que se encuentra en el lazo de persecución, y la **tasa de Evaporación** contiene la velocidad con que se evapora el lazo.

### 5.3. Descripción de los Actores y Casos de Uso

Los actores son descritos por el papel que desempeñan en el sistema, identificando posteriormente los agentes que en él se desenvuelve y las colaboraciones que realizan entre ellos. En el subsistemas SMOW es posible la descripción de dos actores, los cuales se muestran en la Figura 7.

Actor	Descripción	Casos de Uso
Replicador de Objetos	Provee mecanismos que le permiten realizar replicas.	- Replicar Objeto
Migración de Objetos	Provee mecanismos que le permiten realizar la migración de objetos.	- Migrar objetos - Crear Traza

Figura 7. Descripción de los actores que intervienen en los subsistemas SMOW

A continuación se muestra la descripción de los casos de uso de los actores anteriormente mencionados:

### 5.3.1. Casos de uso del Actor Replicación de Objetos

Los casos de uso del actor Replicador de Objetos se muestran en la Figura 8.

<p><b>Caso de Uso:</b> Replica de Objeto.  <b>Resumen:</b> El actor Replicador de Objetos recibe la solicitud de la creación de una replica por parte de un SMRR, y luego realiza la replica del objeto para almacenarla en el SMRR del nodo solicitante.  <b>Actores:</b> SMRR, SMRL, Replicador de Objetos.  <b>Precondición:</b> Haber recibido una solicitud de Replica.  <b>Excepción:</b> No se puede realizar la replica del objeto.</p>
---

Figura 8. Casos de uso del actor Replicador de Objetos.

### 5.3.2. Casos de uso del Actor de Migración de Objetos

Los casos de uso del actor Migración de Objetos se muestran en la Figura 9.

<p><b>Caso de Uso:</b> Migración de Objeto.  <b>Resumen:</b> El actor de Migración de Objetos realiza la migración de objetos web, la cual efectúa hacia los sitios de mayor demanda.  <b>Actores:</b> Migración de Objetos, SMC y SMRL.  <b>Precondición:</b> Haber recibido una solicitud de migración.  <b>Excepción:</b> No se puede realizar la migración del objeto.</p>	<p><b>Caso de Uso:</b> Crear Traza.  <b>Resumen:</b> El actor de Migración de Objetos solicita al SMRL origen del objeto la creación de la traza de objeto cuando se realiza la migración de un objeto.  <b>Actores:</b> SMRL, Migración de Objetos.  <b>Precondición:</b> Objeto migrado.  <b>Excepción:</b> No se puede crear la traza en el SMRL origen.</p>
--	---

Figura 9. Casos de uso del actor de Migración de Objetos.

## 5.4. Identificación de los Agentes del SMOW

La identificación y descripción de los agentes que intervienen en el sistema SMOW se realiza en base a los actores y casos de usos definidos. Según los roles definidos hasta esta fase, tenemos que el SMOW provee las siguientes funcionalidades: Replicar Objetos, Migrar Objetos y Crear Traza. Estas funciones corresponden a los roles de cada uno de los actores identificados. El SMA aquí propuesto para el SMOW consta de dos agentes, los cuales corresponden a los actores identificados anteriormente: Agente Replicador de Objetos (ARO) y Agente Migrador de Objetos (AMO). El agente ARO contendrá mecanismos que le permitirán realizar las replicas, la cual realiza por petición del SMRR. El agente AMO proveerá mecanismos que le permitirán inferir los sitios de mayor demanda para un objeto, además de ser el encargado del traslado de los objeto hacia dichos sitios. Cuando un objeto es migrado, el agente AMO crea una traza en el SMRL donde se encontraba el objeto.

Las tareas que ejecutara el SMOW se derivan de los actores y casos de uso realizados por el subsistema en el SOW. Así, las tareas identificadas se muestran en la Figura 10.



Agente	Tareas	Descripción
<b>Administrador de replicación de Objetos (ARO)</b>	Replicar Objetos	El agente ARO recibe la solicitud de replicar un objeto por parte de SMRR, para esto crea una replica del objeto en el SMRL, y luego la traslada al SMRR solicitante.
<b>Administrador de Migración de Objetos AMO</b>	Migrar Objetos	El agente AMO recibe peticiones de mover un objeto hacia otro nodo, este decide si migrar el objeto hacia el destino sugerido en base a las demandas de los dos nodos y si el nodo origen presenta problemas. Si el agente AMO decide migrar el objeto, este mueve el objeto hacia el SMRL destino.
	Crear Traza	El agente AMO necesita crear trazas cada vez que realiza la migración de un objeto.

Figura 10. Identificación de las tareas del SMOW

### 5.5. Interacciones de los agentes del SMOW con el resto de los subsistemas del SOW

Los agentes propuestos para el SMOW deben mantener las mismas interacciones propuestas para el subsistema. Así, los agentes del SMOW deben interactuar con los demás subsistemas del SMOW para alcanzar sus objetivos de replicación y migración de objetos. El encargado de realizar la replicación de objetos es el agente ARO, para esto, recibe la solicitud de creación de una replica por parte del SMRR, luego bloquea el objeto, cambiándole su estado para que el SMRL origen no pueda realizar operaciones sobre él, para después transportarlo hacia el RR solicitante. Las interacciones de los actos de habla de la conversación Replicar Objeto en el RR se presentan en el diagrama UML de la figura 11.

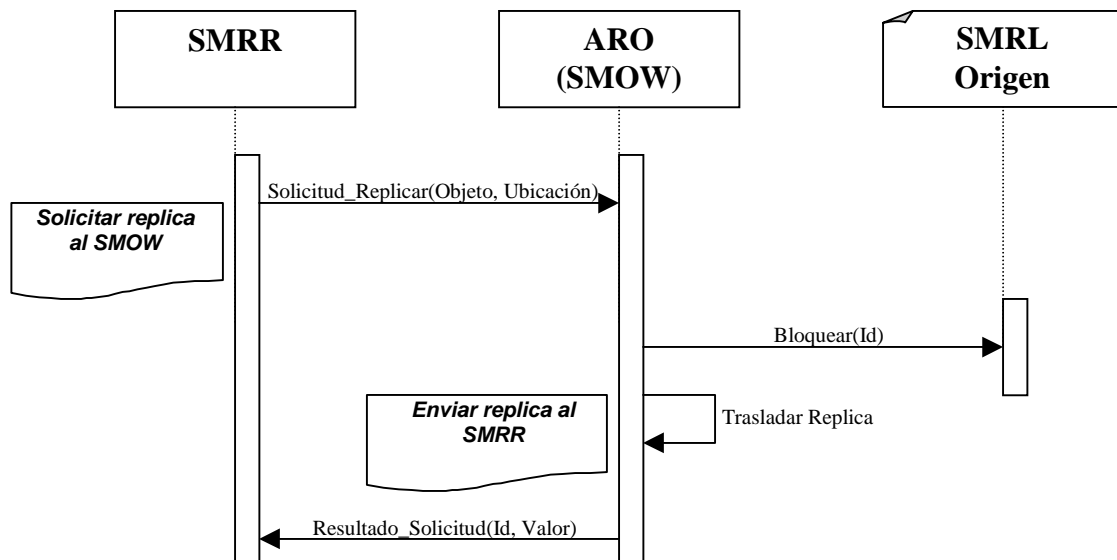


Figura 11. Diagrama UML de la conversación Replicar Objeto.

El agente AMO es el encargado de colocar los objetos presentes en el SOW en los sitios de mayor demanda. Así, el agente AMO recibe la solicitud de migración de un objeto por parte del SMC, solicita permiso para migrar el objeto al destino, bloquea el objeto en el RL origen, y lo migra al SMRL destino, después solicita al Replicador de Objetos que cree la traza del objeto en el SMRL origen. Las interacciones de los actos de habla de la conversación Búsqueda de Objeto en el RR se presenta en el diagrama UML en la figura 12.

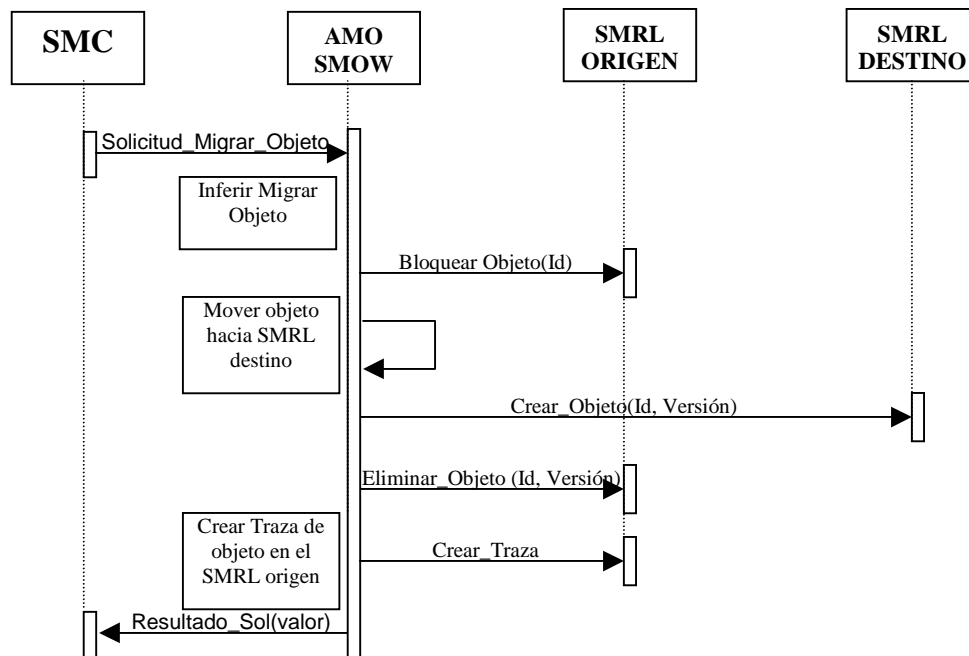


Figura 12. Diagrama UML de la conversación Replicar Objeto.

## 6. CONCLUSIONES

El SMOW provee al SOW de mecanismos que le permiten adaptarse a la naturaleza dinámica de los recursos en Internet. De esta manera el SOW se adecua a la volatilidad de los recursos e intenta disminuir la latencia debido a la distancia entre los nodos, tratando de tener más cerca de los usuarios la información que requieren, valiéndose de mecanismos como la replicación y migración de objetos.

El realizar el diseño del SOW basado en SMA permite el desarrollo de los subsistemas del SOW por separado, donde todos sus componentes asumen sus propios roles y cooperan entre sí para la solución de problemas, distribuyendo de esta manera, los objetivos del SOW entre los subsistemas que lo componen. Así, el diseño del SMOW fue desarrollado como un SMA separado del resto de los subsistemas, pero donde los agentes obtenidos cooperan e interactúan entre sí y con el resto de los subsistemas del SOW para alcanzar sus objetivos.

El SMOW propuesto consiste en un SMA compuesto por 2 agentes: el ARO y el AMO, los cuales cumplen con los roles y objetivos planteados para el SMOW, como son el permitir la migración y replicación de objetos Web. El agente ARO permitirá al SOW almacenar replicas objetos en los RR. Por otra parte el agente AMO proveerá al SOW de mecanismos que le permitirán migrar objetos hacia los nodos de mayor demanda del SOW, realizando la inferencia necesaria para selección del objeto a migrar y del nodo destino ha donde será enviado. Además, provee mecanismos que le permiten al SOW la localización de los objetos web una vez migrados. Para esto, en este artículo se propuso un mecanismo que permite la localización de objetos, que nosotros hemos llamado trazas de objetos, el cual utiliza la noción de trazas de feromonas usadas por las hormigas en su búsqueda de comida, para eliminar los lazos de persecución basura presentes en los RL.

### Reconocimiento

Este trabajo fue financiado por el Proyecto I-620-98-02-AA: "Gestión de Recursos en Redes de Estaciones de Trabajo mediante Sistemas MultiAgentes" del CDCHT de la Universidad de los Andes.

### Referencias

- [1] Aguilar J, Perozo N., Ferrer E., Vizcarrondo J. Arquitectura de un Sistema Operativo Web, *Revista Gerencia Tecnológica Informática*, N. 2, Vol 1, 13 páginas, Colombia Julio de 2003. [http://www.cidlisuis.org/aedo/RGTIN2V1/RGTI\\_01.pdf](http://www.cidlisuis.org/aedo/RGTIN2V1/RGTI_01.pdf), Sistema de Autoría Electrónica de Documentos.
- [2] Aguilar J. File Decomposition, Replication and Assignment Problems: Definition and Resolution methods. *International Journal of Engineering Intelligent Systems*, Vol. 8, N. 4, pp. 245-254, 2000.
- [3] Aguilar J., Leiss L. A Web Proxy Cache Coherency and Replacement Approach, *Lectures Notes in Computer Science*, Vol 2198, pp 75-94,2002.

- [4] Artsy Y., Finkcl R. Designing a Process Migration Facility: The Charlotte Experience. *IEEE Computer*, pp. 47-56, 1989.
- [5] Baker S., Bongki M. Distributed cooperative web servers. *Computer Networks*, Vol. 31, pp. 1215- 1229, 1999.
- [6] Baldeschwieler J., Blumofe R., Brew E. ATLAS: An Infrastructure for Global Computing. *Proceedings of 7th ACM Special Interest Group on Operating Systems, European Workshop on System Support for Worldwide Applications*, pp.165-172, 1.996.
- [7] Douglis F., Ousterhout J. Process migration in the Sprite operating system, *Proceedings of the 7th International Conference on Distributed Computing Systems*, IEEE, Berlin, West Germany, pp. 18-25, 1987.
- [8] Dorigo, M. Maniezzo V. Coloni A. The ant system: Optimization by a colony of cooperating agents, *IEEE Trans. Syst. Man, Cybern.*, Vol. 26, pp. 29-41, 1996.
- [9] Foster I., Kesselman C. Globus: A Metacomputing Infraestructure Toolkit. *Supercomputer Applications*, Vol. 11, N, 2, pp. 115- 128, 1998.
- [10]Kon F., Campbell R., Ballesteros F. 2k: A Distributed Operating System For Dinamic Heterogeneous Environments. *Proceedings of 9th IEEE International Symposium on High Performance Distributed Computing*, pp. 201-207, Pittsburgh, USA, 2000.
- [11] Kropf P. Overview of the WOS Project. *Proceedings of Advanced Simulation Technologies Conference*, pp. 350-356, 1999.
- [12]Morgan, S. Jini to the Rescue. *IEEE Spectrum*, Vol. 4, pp:44-49, 2000.
- [13] Vahdat A., Belani E., Eastham P., Yoshikawa C. WebOS: Operating System Services for Wide Area Applications. *Proceedings of 7th IEEE Symposium on High Performance Distributed Systems*, pp. 52- 63, 1998.
- [14]Van Steen M., Homburg P., Tanenbaum A. The Architectural Design of GLOBE: A Wide-Area Distributed System, *Technical Report IR-422*, Vrije Universiteit Amsterdam, 1997.
- [15]Wang Jin, A survey of Web Caching Schemes for the Internet, *ACM Computer Communication Review*, Vol 25, N. 9,pp 36-46, 1999.
- [16]Weiss G. Multiagent Systems. *The MIT Press*. Massachusets, USA, 1999.
- [17]Wheeler R. Migraction Process. *ACM Computing Surveys*, Vol. 32, No. 3 2000.