

**IMPLEMENTACIÓN DE UN SERVICIO WEB
PARA GEOCODIFICAR UNA DIRECCIÓN**

**Presentado por
Nuvia Esperanza Villamizar Pereira
Curso 2001/2002**

**FUNDACIÓN POLITÉCNICA DE CATALUÑA
Master en Sistemas de Información Geográfica
Barcelona Septiembre de 2003**

IMPLEMENTACIÓN DE UN SERVICIO WEB PARA GEOCODIFICAR UNA DIRECCIÓN

INTRODUCCIÓN.....	3
1. WEB SERVICES.....	4
1.1. 1.2 Ventajas del desarrollo de <i>web services</i>	4
1.2. Los web services: un estándar.....	4
1.3. Tecnologías de desarrollo de los web services.....	4
1.4. Modelo de Arquitectura de un web services.....	5
2. DESCRIPCIÓN DEL CASO DE USO.....	6
2.1. De lado del servidor.....	7
2.1.1. Descripción de la cartografía utilizada.....	7
2.1.2. Crear el archivo Barcelona.axl.....	7
2.1.3. Aplicar la función de geocodificación.....	7
2.1.4. Crear un MapService.....	8
2.1.5. Crear una aplicación web.....	8
2.1.6. Instalar el conector <i>arcims_jconnector.jar</i>	9
2.2. Crear un cliente.....	9
3. CONCLUSIONES.....	11
4. BIBLIOGRAFÍA.....	12

INTRODUCCIÓN

Este trabajo pretende un acercamiento teórico y práctico a los conceptos y a la tecnología que permite el desarrollo de servicios web, implementando un caso de uso con componentes de software particulares para desarrollar un cliente del servicio web de geocodificación de una dirección.

En virtud de lo anterior, la memoria se divide en dos partes: en la primera se definen los conceptos utilizados; y en la segunda se describe la aplicación de dichos conceptos y la puesta en funcionamiento del caso uso. Por último, se incluye un apartado de conclusiones.

1. WEB SERVICES

Un web service es un componente de *software* que puede ser accedido sobre el World Wide Web para el empleo en otras aplicaciones en ambientes distribuidos¹.

1.1. 1.2 Ventajas del desarrollo de web services

Las ventajas de desarrollar web services se miden en los ahorros significativos en tiempo de desarrollo, en el costo, y en recursos de ordenador, ya que el tratamiento de la información se realiza en los ordenadores donde el Servicio Web es localizado y devuelve los resultados al uso local. Para el caso de los **Sistemas de Información Geográfica**² un usuario no necesita mantener herramientas de aplicación SIG o los datos geográficos asociados sobre su sistema local.

Los web services ayudan a la computación de sistemas heterogéneos, pudiendo compartir información y código; es decir, logran la interoperatividad³ y proporcionan herramientas que ayudan al desarrollo y mantenimiento de aplicaciones⁴.

1.2. Los web services: un estándar

Inicialmente la interoperatividad se logró con el uso de formatos de datos propietarios, por lo tanto, el desarrollo estaba sujeto al modo de trabajar del fabricante. Por esta razón, se busco avanzar en el desarrollo de software, convirtiendo a los web services en estándares a partir de la portabilidad de datos que permite XML y la portabilidad de código que permite Java.

1.3. Tecnologías de desarrollo de los web services

XML, el lenguaje de marca extensible ha permitido la evolución de otras tecnologías. Las herramientas que brinda XML no son muy útiles sin lenguajes de programación como

¹Se define un sistema distribuido como aquel en el que los componentes están localizados en computadores en red, y comunican y coordinan sus acciones mediante el paso de mensajes. COLOURIS, George "*Sistemas Distribuidos, Conceptos y Diseño*", Ed. Prentice Hall.

² OTERO PASTOR, Isabel. "*Paisaje Teledetección y SIG Concepto y Aplicaciones*", Capítulo III Sistemas de Información Geográfica, Fundación Conde del valle de Salazar, Pags. 339 a 371.

³ La interoperatividad significa que los sistemas pueden trabajar entre sí con interfaces bien definidas para el intercambio de información. Por ejemplo, una aplicación desarrollada sobre un computador Unix puede interactuar con aplicaciones sobre OS/2 y mainframes IBM.

⁴ El desarrollo de aplicaciones es facilitado por los toolkits web services, que son un juegos de herramientas para servicios web, tales como Microsoft .NET, el "*The Mind Electric GLUE*" que esta actualmente en la versión 4.0 o el paquete de desarrollo JWSDP que esta en la versión 1.2. Simplifican la puesta en práctica de Servicios Web porque *el protocolo de comunicación es manejado automáticamente*.

Java, que fijan el estándar de cómo utilizar XML en las aplicaciones. “Primero se analizan los datos y después se manipulan”⁵.

Una vez el documento XML es analizado, la información esta disponible para ser utilizada por un cliente. Dado que XML es usado para la presentación, la comunicación y la configuración, un cliente de un servicio web puede ser una aplicación, un web service o un humano.

1.4. Modelo de Arquitectura de un web services

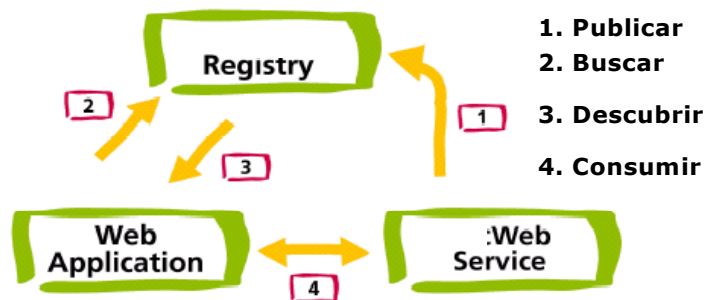


Fig: 1 Modelo de Arquitectura

En este modelo se identifican las acciones que afectan el funcionamiento de un web service, así:

1. Una vez desarrollado el web service, se registra. Usando **UDDI, WSDIL**
2. Un cliente, en este caso una aplicación web, lanza una solicitud para conseguir una lista de los web services que satisfagan las condiciones de búsqueda.
3. El cliente recibe información a partir de la cual puede determinar la petición que más se ajuste a su necesidad.
4. El cliente lanza la petición al web service y recibe la respuesta.

Los web services como sistemas abiertos⁶, logran su funcionamiento adoptando tecnologías basadas en XML como:

SOAP: Simple Object Access Protocol. Un protocolo para el intercambio de información en un ambiente distribuido.

⁵ McLAUGHLIN, Brett, “Java y XML”, Ed. Anaya Multimedia, Madrid, 2001. Cap 3 Analizar XML.

⁶ Se define un sistema abierto como aquel en el que los protocolos o especificaciones son públicos, estandarizados y permiten diferentes implementaciones interoperables entre sí. COLOURIS, George “Sistemas Distribuidos, Conceptos y Diseño”, Ed. Prentice Hall.

WSDL : Web Services Description Language

Este documento sirve para realizar la descripción funcional y técnica del web service. Al *describir* un Web Service el *cliente sabe lo que el servicio hace*, define que métodos están disponibles, los tipos de parámetros y que sus parámetros estén.

Un programador debe saber como esta constituido un web service para poder codificar un cliente capaz de invocarlo, debe conocer su API, que servicios ofrece y que tipo de protocolo de codificación se usará. Si un web service se encuentra disponible al público, su WSDL también estará a la vista de cualquiera. Estos documentos XML pueden ser encontrados en registros UDDI o en web especializadas que funcionan como un buscador normal de Internet, como por ejemplo: www.xmethods.net

EL wsdl no es obligatorio, sin embargo su uso permite ventajas como:

- ❖ Reducir el costo de mantenimiento en clientes.
- ❖ Facilitar la estructuración y construcción de objetos.
- ❖ Automatizar la construcción de clientes volviendo atractivo el uso de web services.
- ❖ La generación del WSDL también puede automatizarse, con herramientas que existe para ello.

UDDI: Universal Description Discovery and Integration.

Los Web Services son publicados sobre el registro UDDI Descubrimiento e integración. Es Una base de datos universal de servicios web, donde los desarrolladores pueden buscar cualquier sitio para descubrir servicios. El servicio de interés pasa a través de una interfaz web o a llamadas SOAP. Es el protocolo usado para la identificación del WSDL.

WSDIL: Web Service Inspection Language.

Mantiene listas de enlaces a direcciones donde se encuentra un WSDL.

UDDI y WSDIL Se pueden usar para la *publicación y localización* de servicios web, la diferencia esta en que UDDI almacena un registro centralizado, y WSDIL almacena listas distribuidas.

Sin SOAP, como es el caso de uso un cliente tiene que crear una petición de XML, enviarlo en un mensaje POST a un URL sobre HTTP, luego analizar la petición usando XML parser.

2. DESCRIPCIÓN DEL CASO DE USO

En este caso de uso se muestra paso a paso cómo instalar y configurar el software del lado del servidor; cómo crear un map service, cómo implementar un algoritmo de geocodificación, cómo crear una aplicación web y cómo desarrollar un cliente que use un servicio web de geocodificación de una dirección.

2.1. De lado del servidor

Se utilizo un producto de la firma ESRI, el ArcIMS 3.1

Sistema Operativo: Windows 2000
Servidor: ArcIMS 3.1
Servlet⁷: Tomcat 4.0.2.
Java Runtimes: Java 2 SDK 1.3 o superior
Este servidor necesita de un Servlet y java Runtimes para funcionar.

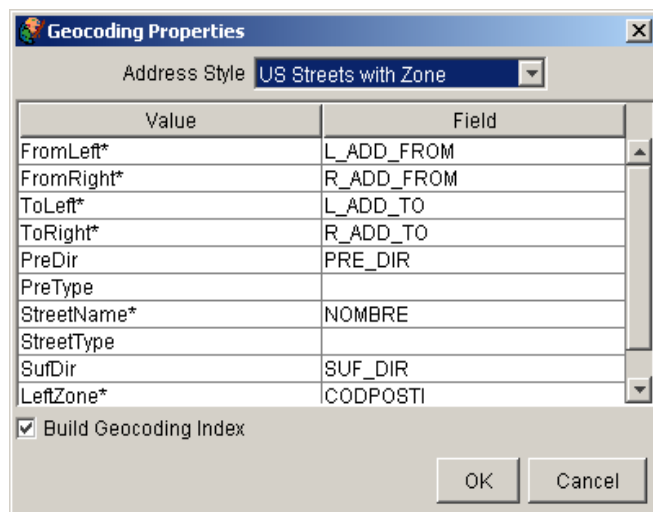
2.1.1. Descripción de la cartografía utilizada

Se instalaron tres capas en formato SHP que comprende una zona de la ciudad de Barcelona y están en coordenadas UTM. Una capa llamada ejes contiene los ejes de vía que son necesarios para aplicar la función de geocodificación.

2.1.2. Crear el archivo Barcelona.axl

Este archivo es un documento XML que debe cumplir las especificaciones de ARXML⁸. Para su creación se puede utilizar la herramienta author de ArcIMS que permite la visualización y edición de las capas facilitando la creación del archivo al guardar la capas visualizadas como un proyecto.

2.1.3. Aplicar la función de geocodificación



Para geocodificar dirección existen algoritmos que calculan de coordenadas y generan un archivo de índices teniendo en cuenta el estilo de codificación que se desee. Para este caso de uso, se eligió U.S. streets with zone, que construye un índice de geocodificación.

Este estilo de geocodificación, considera las direcciones de calle con información de la zona tal como el código postal o nombre de las ciudades. Las características de la calle, son representadas por dos

⁷ Un servlet es una pequeña extensión conectable a un servidor que aumenta su funcionalidad; permiten a los desarrolladores extender y personalizar cualquier Servidor Java disponible –Servidor Web, Servidor de Correo, Servidor de Aplicaciones o cualquier Servicio Web. Con un alto grado de portabilidad, flexibilidad y facilidad. Este corre dentro de una maquina virtual de java JVM sobre el servidor. En otras palabras, un servlet es una clase de java que puede ser cargada dinámicamente para expandir la funcionalidad de un servidor. HUNTER Jason, CROWFORD William, *Java Servlet Programming*, Ed. Benchmark Productions, Inc. 1998.

⁸ ARXML es el protocolo de comunicaciones con el Servidor espacial de arcIMS que provee la funcionalidad de acceder y envolver datos y mapas dentro de un formato apropiado.

intervalos de números de casas, uno por el lado izquierdo de la calle y otro por el lado derecho de la calle. Acordando la paridad_par/impar de cada intervalo. El número de inicio del intervalo puede ser menor o mayor que el número de fin de intervalo.

2.1.4. Crear un MapService

Para crear un MapServer se requiere tener configurado un servidor virtual de imágenes, luego se crea un Servicio de Mapas con ayuda de la herramienta "ArcIMS Administrator"

En esta instancia se especifica:

- Nombre del Servicio
- El Archivo map
- El servidor virtual
- La salida del servicio

2.1.5. Crear una aplicación web⁹.

Se crea la estructura de directorios adecuada. El nivel superior de un WAR¹⁰ es el documento raíz (*document root*) de la aplicación. El document root es donde se almacenan las paginas JSP, las clases y archivos del lado del cliente y recursos estáticos.

⁹ Una aplicación web es una extensión dinámica de un Servidor Web. Hay dos tipos de aplicaciones web:

Orientada a Presentación: Estas aplicaciones generan paginas web dinámicas, que contiene en respuesta a solicitudes varios tipos de lenguaje de marcas como HTML, XML, etc.

Orientada a Servicios: Estas aplicaciones implementan la url (endpoint) a un servicio web (web service) y frecuentemente son invocadas por las aplicaciones orientadas a presentación.

Los componentes web pueden ser:

Servlets: Son clases del lenguaje de programación java que dinámicamente procesan solicitudes y construyen respuestas.

Páginas JSP: Son documentos basados en texto que permiten la inserción de código servlet, cada bloque de código es llamado *scriptlet* y va encerrado en `<% %>`. Estos componentes son soportados por los servicios de una plataforma en tiempo de ejecución llamando a un contenedor web (p.e. Tomcat).

Los componentes, corren en tiempo de ejecución en la plataforma de llamado a un contenedor web como tomcat que suministra servicios tales como despacho de solicitudes, seguridad, concurrencia y administración de ciclos de vida.

¹⁰ Para distribuir una aplicación se debe empaquetar en un archivo *.war.

Web Application Archive, el cual es un archivo jar similar a los paquetes usados por las librerías de clases java. Un WAR puede contener del lado del servidor clases utilitarias que con frecuencia se amoldan a la arquitectura de componentes de javabeans. Un contenido de presentación web estática (p.e. html, imágenes, archivos de sonido). Clases del lado del cliente (applets y clases utilitarias). Una aplicación web puede correr desde un archivo war o desde un directorio desempaquetado en el mismo formato que un archivo WAR.

El documento raíz contiene un subdirectorio llamado WEB-INF el cual contiene los siguientes archivos:

- web.xml: El descriptor de despliegue de la aplicación web.
- Librería de etiquetas¹¹: Archivo descriptor de librerías de etiqueta.
- Classes : Un directorio que contiene clases del lado del servidor: Servlets, clases utilitarias y componentes JavaBeans.
- Lib : Un directorio que contiene una librería de archivos jar (librerías de etiqueta y librerías de utilidad llamadas del lado del servidor).

2.1.6. Instalar el conector `arcims_jconnector.jar`

El conector java es una implementación del lado del servidor que permite la comunicación entre el servidor de Aplicaciones ARCIMS y clientes JSP¹² o aplicaciones java standalone. Este es una suite de componente de software reutilizable que incluyen una librería modelo de objetos JavaBeans™¹³ y un conjunto de etiquetas JSP soportadas en la forma de una librería de etiquetas.

Las JSP Tag library son una colección de etiquetas JSP sobre el modelo de objetos del conector Java. Proporcionan una interfaz a la librería del modelo de objetos. Las solicitudes son escritas usando etiquetas JSP, la librería de etiquetas, convierte etiquetas a objetos que JavaBeans entiende. El Java Beans convierte la solicitud a ArcXML.

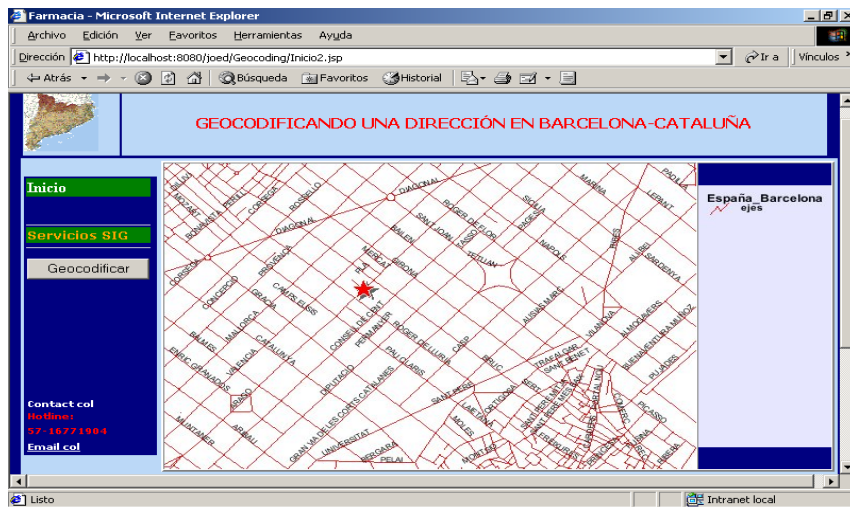
2.2. Crear un cliente

El desarrollo de un cliente permite digitar una dirección y solicitar el servicio de geocodificación, recibiendo como respuesta un mapa donde un punto identifica la localización de ésta dirección.

¹¹ Tag Library, la librería de etiquetas es una colección de etiquetas personalizadas *custom tag* que permiten en tecnología Java Server Page, la invocación de acciones. Las acciones son elementos que crean y acceden a objetos del lenguaje de programación y afectan la cadena de salida. <http://java.sun.com/products/jsp/tutorial/TagLibraries3.html>

¹² JSP JavaServer Pages es un archivo de texto con extensión *.jsp, usado en lugar de una página HTML, disponible como una extensión a el serverlet engine (p.e. Tomcat). Las páginas del Servidor, “en su mínima expresión” permite insertar etiquetas y códigos java lineales en otra página normal HTML. Cuando se solicita la página JSP se ejecuta el código resultante y los resultados se insertan en la salida HTML. Las JSP se precompilan en un servlet y un PrintWriteractually se manipula la salida.

¹³ JavaBeans es un modelo de componentes orientado a objetos para Java. Son un mecanismo para la construcción de aplicaciones partiendo de componentes de software reutilizable. MOHSENI, Piroz; STEWART, *Tom Guía de desarrollo de JavaBeans* Ed. Anaya Madrid 1997 pp48



Este desarrollo se basa en el uso de componentes de software que son implementaciones particulares de las tecnologías que soportan el desarrollo de web services como el conector java de ArcIMS.

La implementación del modelo de objetos de la especificación de ArcXML permite funciones de despliegue de mapas, interpretación y simbología, adicionar capas dinámicamente, ejecutar solicitudes y características espaciales, geocodificar direcciones, proyecciones y metadatos; y administrar y autenticar servicios.

Las operaciones de geocodificación se realizan con los objetos que ofrece el paquete: *com.esri.aims.mtier.model.map.layer.geocode* en donde: Un objeto *AddressMathcInputs* adiciona valores de dirección para procesar una solicitud de geocodificación. Un objeto *AddressMathc Results* representa el resultado de una colección de objetos geocodificados. Un objeto *Results* representa un resultado individual creado después de un proceso de geocodificación y contiene la puntuación, el valor, y un objeto punto que tiene las coordenadas del resultado.

Dentro de la página JSP se puede usar la librería de etiquetas o las clases java.

Si se usa la librería de etiquetas:

// El atributo uri apunta a la localización de la librería de etiquetas.

```
<%@taglib uri="arcims_taglib.tld" prefix="aims" %>
```

Si se usa las clases java se da la siguiente directiva

// Hace disponible a JSP la clase geocode

```
<%@page import="com.esri.aims.mtier.model.map.layer.geocode" %>
```

El código html Regresa la solicitud al cliente. La acción recae sobre la misma página lo que le permite ser refrescada, usando el mismo URL. La información es pasada por post.

```
<form action="geocat.jsp" method="post"><center>
```

El desarrollo de este cliente por ser una página JSP contiene dos tipos de texto:

Una plantilla estática de datos que esta expresada en formato HTML y elementos JSP que construyen el contenido dinámico. Implica elementos constructores de JSP como:

Directivas `<%.....%>`; Scriptlets `<%.....%>`; Expresiones `<%=.....%>`

3. CONCLUSIONES

Los web services no son nada nuevo, como se puede observar a través del caso de uso, implementaciones particulares usan este concepto, sin adoptar el uso de estándares.

Los problemas planteados para *la integración de aplicaciones* intra empresa, inter empresa y con los clientes; para la *adaptación* o *flexibilidad* en los modelos de negocio y para reducir costos *reutilizando* código, han sido solucionado con la aplicación de los protocolos CORBA, COM (Microsoft), RMI (SUN), XML-RPC. Sin embargo, existen razones para usar web services:

- La utilización de XML y protocolos de Internet estándares.
- El apoyo de la comunidad que lo ha convertido en un estándar.
- La existencia de implementaciones en todos los lenguajes.

Existen en Internet potentes herramientas gratuitas que facilitan el desarrollo de servicios web, esta facilidad está sujeta al conocimiento del lenguaje XML y Java, ya que a partir de ellos se han desarrollado nuevas tecnologías, entre ellas servlets y JSP; lenguajes como WSDL y protocolos como UDDI y SOAP que han, aprovechando la portabilidad de datos con XML y la portabilidad de código con Java.

Reutilizar documentos xml que se encuentra en Internet tal como DTD, esquemas, hojas de estilo, información (p.e. Geográfica) para crear una aplicación web, requiere control, pues cambios o desaparición de esta información, pueden alterar el funcionamiento de la aplicación que en ellos se soporta. Esto supone mecanismos de control sobre el mantenimiento y permanencia tanto de documentos como de servicios web para asegurar la funcionalidad de aplicaciones que se basan en ellos.

El desarrollo del cliente como una página JSP Java Server Pages o páginas del servidor java se facilitó con el uso de una librerías de etiquetas. Sin embargo, se presentan la desventajas de que no hay una separación entre contenido y presentación, por tanto, el mantenimiento del cliente se dificulta.

Particularmente, la fase conceptual, la configuración del servidor, la creación del mapservice, la implementación de algoritmo de geocodificación y el desarrollo de un cliente JSP para comprender parte de la arquitectura de un web service, son el punto de partida para iniciar el desarrollo de web services con la implementación de estándares.

4. BIBLIOGRAFÍA

HUNTER, Jason, "JAVA Servlet Programming", Ed. Benchmark Productions, Inc. USA 1998.

McLAUGHLIN, Brett, "Java y XML", Ed. Anaya Multimedia, Madrid, 2001.

GOLDFARD, Charles, "Manual de XML", Ed. Printece Hall, Madrid, 1998.

RIBAS LEQUERICA, Joan, "Web Services" Ed. Anaya Multimedia, Madrid, 2003.

OTERO PASTOR Isabel, PAISAJE TELEDETECCIÓN Y SIG Concepto y Aplicaciones, Capítulo III Sistemas de Información Geográfica, Fundación Conde del valle de Salazar, pp339- 371, 1997.

ADELL HERNANI José Antonio y otros, *La Sociedad de la Información en España, Perspectiva 2001 - 2005*, Informe realizado por Telefónica, 1 Edición, Julio de 2001.

MOHSENI, Piroz; STEWART, Tom *Guía de desarrollo de JavaBeans* Ed. Anaya 1997 pp48

COLOURIS, George. "Sistemas Distribuidos, Conceptos y Diseño", Ed. Prentice Hall, Madrid, 2000.

<http://krypton.mnsu.edu/~spiral/eta/glossary/indxGlossOOxml.html>

Glosarios en Internet

http://arcweb.esri.com/services/arcwebonline/understanding_services.pdf

Generalidades sobre Web Services.

<http://atenea.ucauca.edu.co/~gramirez/archivos/AnotacionesRUP.pdf>

Proceso Unificado para el desarrollo de software_ RUP

http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/Tema3_2x1.pdf

Ciclo de vida del desarrollo del software

<http://www.etse.urv.es/EngInf/assig/sob/>

Sistemas Distribuidos

<http://java.sun.com/products/jsp/tutorial/TagLibraries3.html>

Librería de etiquetas

<http://java.sun.com/products/jsp/technical.html#syntax>

Sintaxis de elementos JSP.