



Arquitecturas para SIG distribuidos

Trabajo Fin de Máster

Máster Universitario de Investigación en Ingeniería de Software y
Sistemas Informáticos

Alumno: D. Juan Carlos González González

Directora: Dra. Dña. Elena Ruiz Larrocha

Itinerario: Ingeniería de Software

Código de la asignatura: 105128

Curso académico: 2014/2015

Convocatoria de defensa: Febrero (extraordinaria)

Trabajo Fin de Máster

Máster Universitario de Investigación en Ing. de Software y Sistemas Informáticos
Universidad Nacional de Educación a Distancia

Itinerario	Ingeniería de Software
Código asignatura	105128
Título	Arquitecturas para SIG distribuidos
Tipología	B (trabajo específico propuesto por el alumno)
Alumno	D. Juan Carlos González González
Directora	Dra. Dña. Elena Ruiz Larrocha

Calificaciones

Autorización

Autorizo a la Universidad Nacional de Educación a Distancia a difundir y utilizar, con fines académicos, no comerciales y mencionando expresamente a su autor, la memoria de este Trabajo Fin de Máster.

Firmado:

D. Juan Carlos González González

Resumen

La sociedad actual se ha convertido en ávida consumidora de información geográfica, gracias a la cual es posible ubicar las diferentes actividades desarrolladas por el ser humano y por las colectividades que forma y, lo que es si cabe más importante, a partir de dichos emplazamientos poder planificar futuras actuaciones que faciliten el desarrollo de las actividades. Este escenario es, sin género de duda, muy prometedor, pero también pone de manifiesto importantes problemáticas a las que es preciso dar solución, entre las cuáles cabe destacar la capacidad para:

- Gestionar de forma estructurada grandes volúmenes de información heterogénea y con tasas de crecimiento muy elevadas.
- Ofrecer soluciones tecnológicas que permitan la creación de herramientas informáticas diseñadas de forma adecuada para distintos perfiles de usuario.

En el ámbito de la informática se ha venido desarrollando una tipología de sistema de información (SI), denominada Sistema de Información Geográfica (SIG), cuya principal virtud es la capacidad para gestionar información geográfica y ofrecer herramientas que permiten la modelización y análisis de diferentes variables dentro de un contexto espacial. Sin embargo, los SIG han tenido que experimentar una importante evolución, de la mano de las revoluciones acaecidas en diferentes ámbitos tecnológicos –informática, telecomunicaciones, etc.-, para poder hacer frente a las crecientes necesidades de sus usuarios. En la actualidad, los retos planteados anteriormente han motivado que los SIG vean en el ámbito de las arquitecturas propias de los sistemas distribuidos la solución a los mismos.

Centrado el contexto en el que se enmarca el presente Trabajo Fin de Máster, el contenido del mismo aborda el estudio de los principales paradigmas arquitectónicos utilizados en la actualidad en la implantación de sistemas distribuidos: Arquitectura Orientada a Servicios y Arquitectura Orientada a Recursos. A partir de este estudio se realiza un trabajo de investigación encaminado a la formulación de una arquitectura híbrida que permita, tal y como se ha reseñado anteriormente, hacer frente a los retos indicados.

Palabras clave:

SIG, SOA, ROA, OGC, WebGIS

Abstract

Today's society has become avid consumer of geographic information, through which it is possible to locate the different activities of the human being and the communities they form, and what is even more important, taking into account these locations, the possibility of planning future actions that facilitate the development of many types of activities. This scenario is, undoubtedly, very promising, but also highlights important challenges that need to be faced, among which must be remarked the capacity to:

- Manage in a structured manner very large volumes of heterogeneous data with very high rates of growth.
- Provide technical solutions to enable the creation of computer tools properly designed for different user profiles.

In the field of computer science, many people are committed to develop a typology of information system, called Geographic Information System (GIS), whose main virtue is the ability to manage geographic information and provide tools for modelling and analysing different variables within a spatial context. However, GIS have had to undergo a major change, with the help of technological revolutions experienced in different areas –computer science, telecommunications, etc.–, in order to meet growing user needs. At present, the above challenges have led GIS technologies to see in the world of the distributed information systems the architectural solution to face successfully these challenges.

Focusing the context in which this Master's Final Study is framed, its content describes major architectural reference models currently used in the implementation of distributed systems: Service Oriented Architecture and Resource Oriented Architecture. From this point is carried out a research with the aim of developing a hybrid architecture that, as has been reported previously, addresses the identified challenges.

Keywords:

GIS, SOA, ROA, OGC, WebGIS

Índice de contenidos

1 INTRODUCCIÓN	1
2 ESTADO DEL ARTE	5
2.1 <i>SISTEMAS DE INFORMACIÓN GEOGRÁFICA</i>	5
2.1.1 <i>Cartografía por ordenador (1960-1969)</i>	5
2.1.2 <i>SIG departamentales (1970-1989)</i>	6
2.1.3 <i>SIG corporativos (1990-1996)</i>	8
2.1.4 <i>SIG comunitarios (1997-actualidad)</i>	11
2.2 <i>RETOS ACTUALES</i>	13
3 ARQUITECTURAS DISTRIBUIDAS	19
3.1 <i>ARQUITECTURAS ORIENTADAS A SERVICIOS</i>	19
3.1.1 <i>Instrumentación de servicios mediante software</i>	19
3.1.2 <i>Computación orientada a servicios</i>	20
3.1.3 <i>Plataforma arquitectónica</i>	21
3.1.4 <i>Capas de la arquitectura</i>	24
3.2 <i>ARQUITECTURAS ORIENTADAS A RECURSOS</i>	26
3.2.1 <i>Fundamentos de REST</i>	26
3.2.2 <i>Arquitectura</i>	30
3.3 <i>ONTOLOGÍAS Y WEB SEMÁNTICA</i>	32
3.4 <i>COMPOSICIÓN DE SERVICIOS</i>	35
4 ARQUITECTURA PARA UN SIG DISTRIBUIDO	39
4.1 <i>MARCO REGULATORIO</i>	40
4.1.1 <i>Geoservicios OGC</i>	40
4.1.2 <i>Infraestructuras de datos espaciales</i>	66
4.1.3 <i>INSPIRE</i>	66
4.2 <i>ESCENARIO DE IMPLANTACIÓN</i>	78
4.2.1 <i>Geoservicios publicados</i>	78
4.2.2 <i>Arquitectura existente</i>	80
4.2.3 <i>Retos planteados</i>	81
4.3 <i>ESPECIFICACIÓN FUNCIONAL</i>	86
4.3.1 <i>Servicios de descubrimiento</i>	86
4.3.2 <i>Servicios de visualización</i>	87
4.3.3 <i>Servicios de descarga</i>	94
4.3.4 <i>Servicios de geoprocesamiento</i>	102
4.4 <i>DISEÑO ARQUITECTÓNICO</i>	113
4.4.1 <i>Modelo de referencia</i>	114
4.4.2 <i>Servicio de visualización WMS</i>	117
4.4.3 <i>Servicio de visualización WMTS</i>	134
4.4.4 <i>Servicio de descarga WFS</i>	144
4.4.5 <i>Servicio de descarga WCS</i>	154
4.4.6 <i>Servicio de descarga SOS</i>	160

4.4.7 Servicio de geoprocesamiento WPS.....	169
4.4.8 Seguridad.....	177
4.4.9 Composición	179
4.4.10 Estructura de capas	179
5 LÍNEAS FUTURAS DE INVESTIGACIÓN.....	185
6 CONCLUSIONES.....	187
7 AGRADECIMIENTOS	189
8 BIBLIOGRAFÍA	191
9 GLOSARIO	199
10 ACRÓNIMOS	203
11 ANEXO - GEOSERVICIOS DEL ICGC	209

Índice de figuras

Figura 2.1 Arquitectura SIG departamental aislada [fuente: propia].	7
Figura 2.2 Arquitectura SIG multidepartamental [fuente: propia].	8
Figura 2.3 Arquitectura SIG corporativa, parcialmente centralizada [fuente: propia].	10
Figura 2.4 Arquitectura SIG corporativa, totalmente centralizada [fuente: propia].	11
Figura 2.5 Arquitectura SIG comunitaria [fuente: propia].	12
Figura 2.6 Las cuatro características del <i>big data</i> [IBM13].	15
Figura 3.1 Relación entre servicios, clientes y proveedores [Gar08].	20
Figura 3.2 Marco básico de funcionamiento de la SOC [Gar08].	23
Figura 3.3 Modelo básico en SOC y sus lenguajes y estándares básicos [Gar08].	23
Figura 3.4 Descomposición en capas de SOA [fuente: propia].	25
Figura 3.5 Diagrama <i>UML</i> de secuencia con la mediación de la capa de integración [fuente: propia].	26
Figura 3.6 Derivación de <i>REST</i> a partir de las restricciones arquitectónicas [ACM02].	28
Figura 3.7 Descomposición en capas de ROA [fuente: propia].	31
Figura 3.8 Estructura del primer nivel de OWL-S [fuente: propia].	35
Figura 4.1 Interacciones entre un cliente y un servidor WMS [fuente: propia].	48
Figura 4.2 Interacciones entre un cliente y un servidor WMTS [fuente: propia].	51
Figura 4.3 Diagrama <i>UML</i> de transición de estados de los bloqueos en WFS [Vre10].	53
Figura 4.4 Interacciones entre un cliente y un servidor WFS [fuente: propia].	54
Figura 4.5 Disposición modular del estándar WCS [OGC14b].	56
Figura 4.6 Ejemplo de petición WCS con recorte [IGN14].	57
Figura 4.7 Ejemplo de petición WCS con rebanado [IGN14].	57
Figura 4.8 Interacciones entre un cliente y un servidor WCS [fuente: propia].	58
Figura 4.9 Concepto de Web de sensores [Bot07].	59
Figura 4.10 Conceptos relacionados con una observación [OGC11c].	60
Figura 4.11 Interacciones entre clientes y un servidor SOS [fuente: propia].	63
Figura 4.12 Interacciones entre un cliente y un servidor WPS [fuente: propia].	65
Figura 4.13 Arquitectura técnica de INSPIRE [NSD08].	68
Figura 4.14 Habilitación de servicios no compatibles con INSPIRE [NSD08].	71
Figura 4.15 Servicios WFS publicados por IDEG [IET14].	76
Figura 4.16 Cartografía catastral superpuesta al MTC 1:5.000 [fuente: propia].	79

Figura 4.17 Cartografía catastral superpuesta al MTC 1:5.000 –escala de grises– [fuente: propia].	80
Figura 4.18 Arquitectura de los geoservicios del ICGC [fuente: propia].	81
Figura 4.19 Representaciones cartográficas de los diferentes productos del ICGC [ICG14].	91
Figura 4.20 Visualización 3D –dcha.– del Lac Redon en el Pirineo catalán –izda.–	92
Figura 4.21 Aproximaciones y contexto de representaciones 3D basadas en <i>WS</i> [Coo12].	93
Figura 4.22 Componentes de la interfaz <i>RDF</i> para <i>WFS</i> diseñada en [Zha08].	96
Figura 4.23 Problemas de sincronización –superior BTC 1:50.000 e inferior BTC 1:25.000– [ICG14].	100
Figura 4.24 Jerarquía de clases de geometría según <i>OGC</i> [Her11].	103
Figura 4.25 Operaciones de la clase geometría según el <i>OGC</i> [Her11].	104
Figura 4.26 Matriz de combinaciones del modelo <i>DE-9IM</i> [fuente: propia].	105
Figura 4.27 <i>HTTP</i> como protocolo de transporte –sin opción <i>SOAP</i> – / aplicación [fuente: propia].	120
Figura 4.28 BTC 1:5.000 –simbología aleatoria–[fuente: propia].	122
Figura 4.29 BTC 1:5.000 –filtrada–[fuente: propia].	124
Figura 4.30 Mapa Topográfico de Cataluña (MTC) 1:5.000 [ICG14].	124
Figura 4.31 Interfaz de administración de <i>ArcGIS Server</i> [fuente: propia].	130
Figura 4.32 Tiempo de respuesta en función del número de usuarios concurrentes – <i>WMS/REST</i> –.	133
Figura 4.33 Rendimiento en función del número de usuarios concurrentes – <i>WMS/REST</i> –.	133
Figura 4.34 <i>HTTP</i> como protocolo de transporte –con opción <i>SOAP</i> – / aplicación [fuente: propia].	136
Figura 4.35 Conjunto de matrices [fuente: propia].	137
Figura 4.36 Conjunto de teselas que forman una matriz (n,m) [fuente: propia].	138
Figura 4.37 Tiempo de respuesta en función del número de usuarios concurrentes – <i>WMS/WMTS</i> –.	142
Figura 4.38 Rendimiento en función del número de usuarios concurrentes – <i>WMS/WMTS</i> –.	143
Figura 4.39 Conjunto de celdas que forman un índice espacial de resolución (n,m) [fuente: propia].	154
Figura 4.40 Sintaxis del parámetro <i>subset</i> en notación <i>EBNF</i> [OGC11b].	156
Figura 4.41 Fragmento de respuesta <i>GetCapabilities</i> de un servicio <i>WPS</i> [IGN14].	172
Figura 4.42 Modelo arquitectónico propuesto para SIG distribuidos [fuente: propia].	180

Índice de tablas

Tabla 4.1 Clasificación de las especificaciones de estándares OGC.....	41
Tabla 4.2 Primera formulación admitida por los estándares OGC.....	43
Tabla 4.3 Segunda formulación admitida por los estándares OGC.....	43
Tabla 4.4 Tercera formulación admitida por los estándares OGC.....	44
Tabla 4.5 Número de geoservicios agrupados por tipología y catálogo.....	47
Tabla 4.6 Porcentaje de geoservicios de visualización WMS por catálogo.....	47
Tabla 4.7 Tiempo de inoperatividad para diferentes disponibilidades y periodos [IOC11b].....	73
Tabla 4.8 Resultados de búsqueda de conjuntos de datos en IDEE e IDEG.....	75
Tabla 4.9 Resultados de búsqueda de servicios en IDEE e IDEG.....	75
Tabla 4.10 Resultados de búsqueda de conjuntos de datos en INSPIRE e IDEE.....	76
Tabla 4.11 Resultados pruebas de rendimiento de geoservicios del ICGC.....	85
Tabla 4.12. Listado de contenidos temáticos de INSPIRE [Eur14].....	88
Tabla 4.13 Relación de funciones básicas seleccionadas.....	104
Tabla 4.14 Relación de funciones de carácter topológico.....	106
Tabla 4.15 Relación de funciones de análisis espacial.....	107
Tabla 4.16 Comparativa de los protocolos arquitectónicos de SOA y ROA.....	116
Tabla 4.17 Protocolos y codificaciones admitidas en las operaciones WMS.....	117
Tabla 4.18 Relación de parámetros asociados a una petición GetMap WMS.....	118
Tabla 4.19 Relación entre recursos, representaciones y métodos WMS.....	121
Tabla 4.20 Cumplimiento de los principios REST en el estándar WMS.....	127
Tabla 4.21 Propuesta de recursos y operaciones asociadas a la interfaz REST del estándar WMS...	128
Tabla 4.22 Resultados de la comparativa de interfaces WMS y REST de ArcGIS for Server.....	132
Tabla 4.23 Protocolos y codificaciones admitidas en las operaciones WMTS.....	135
Tabla 4.24 Relación entre recursos, representaciones y métodos WMTS.....	136
Tabla 4.25 Cumplimiento de los principios REST en el estándar WMTS.....	141
Tabla 4.26 Resultados de la comparativa de interfaces WMS y WMTS de Mapproxy.....	142
Tabla 4.27 Protocolos y codificaciones admitidas en las operaciones WFS.....	144
Tabla 4.28 Relación entre recursos, representaciones y métodos WFS.....	147
Tabla 4.29 Cumplimiento de los principios REST en el estándar WFS.....	150
Tabla 4.30 Equivalencias entre operaciones HTTP y KVP para WFS.....	151

Tabla 4.31 Propuesta de recursos y operaciones asociadas a la interfaz <i>REST</i> del estándar <i>WFS</i>	153
Tabla 4.32 Protocolos y codificaciones admitidas en las operaciones <i>WCS</i>	155
Tabla 4.33 Relación entre recursos, representaciones y métodos en <i>WCS</i>	156
Tabla 4.34 Cumplimiento de los principios <i>REST</i> en el estándar <i>WCS</i>	159
Tabla 4.35 Protocolos y codificaciones admitidas en las operaciones <i>SOS</i>	161
Tabla 4.36 Relación entre recursos, representaciones y métodos en <i>SOS</i>	164
Tabla 4.37 Cumplimiento de los principios <i>REST</i> en <i>SOS</i>	166
Tabla 4.38 Equivalencias entre operaciones <i>HTTP</i> y <i>KVP</i> para <i>SOS</i>	167
Tabla 4.39 Propuesta de recursos y operaciones asociadas a la interfaz <i>REST</i> del estándar <i>SOS</i>	169
Tabla 4.40 Protocolos y codificaciones admitidas en las operaciones <i>WPS</i>	170
Tabla 4.41 Cumplimiento de los principios <i>REST</i> en el estándar <i>WPS</i>	173
Tabla 4.42 Equivalencias entre operaciones <i>HTTP</i> y <i>KVP</i> para <i>WPS</i>	175
Tabla 4.43 Propuesta de recursos y operaciones asociadas a la interfaz <i>REST</i> del estándar <i>WPS</i>	176
Tabla 11.1 Relación de geoservicios del ICGC.	214

1 Introducción

El filósofo, científico, jurista, orador, ensayista y autor inglés Sir Francis Bacon recogió en su obra [Bac97] la cita “*knowledge is power*” -el conocimiento es poder-, de la cual se derivó otra que ha cobrado mayor presencia mediática en nuestra sociedad: “la información es poder”. Sea cual fuera el origen de esta última cita, es cierto que tradicionalmente se ha asimilado el hecho de poseer información con el de tener mayor poder y esto es cierto, en líneas generales, si se tiene en cuenta que para tomar decisiones es necesario contar con información. Sin embargo, si se pretende que estas decisiones sean acertadas, la información de la que se parte ha de ser fiable y precisa, en relación al contexto en el que se esté trabajando.

Centrando el estudio en el ámbito de la información geográfica (IG), cabe reseñar que ésta es especialmente compleja, dado que requiere manejar dos variables: el qué y el dónde. Desde los albores de la humanidad el hombre se ha interesado por representar el territorio, utilizando para ello diferentes técnicas a lo largo de los tiempos. Sobre esta representación –que hoy en día denominaríamos mapa o plano– que nos permite determinar la variable “dónde”, el hombre dibuja un conjunto de elementos –variable “qué”– que escapan a la pura descripción del territorio; se trata de localizar los elementos más reseñables desde el punto de vista de la actividad desempeñada. En este sentido, a título ilustrativo, los marinos se interesan por representar en sus cartas náuticas o portulanos, aspectos relacionados con la navegación –puertos, corrientes, caladeros, etc.–, las empresas hacen lo propio con sus redes de distribución –proveedores, clientes, etc.–.

Más recientemente, el conocimiento del marco geográfico en el que el hombre desarrolla sus actividades ha ampliado su ámbito de acción al análisis de la IG. Esta nueva perspectiva marca un punto de inflexión hacia una era en la que la interpretación de la IG se realiza mediante la combinación de diferentes capas o variables temáticas, hecho que permite caracterizar las relaciones complejas de tipo espacial que se derivan de dicha información. En definitiva, se pasa de una perspectiva descriptiva –qué y dónde– a una prescriptiva –qué y por qué–, dando como resultado la generación de un conjunto de herramientas de análisis geográfico.

Sin embargo, un análisis concienzudo de una determinada problemática puede implicar aspectos que van más allá de los puramente topológicos, es decir, de la relación existente entre los objetos geográficos que forman parte de la representación. En este sentido, es muy habitual contar con información alfanumérica que describe los propios objetos, aportando nuevas variables que permiten realizar razonamientos más complejos. Retomando los ejemplos anteriores, sobre los puertos se puede recabar datos que hagan referencia al calado de sus dársenas, sobre los proveedores datos relacionados con los productos que suministran, etc. La panoplia de posibilidades analíticas, cuya finalidad última es ayudar a los gestores en la toma de decisiones, permite dar respuesta a aspectos como la gestión de los puntos de atraque en función de las dimensiones de los barcos y de las mercancías que estos transporten y, en el ámbito de la distribución, la determinación de las rutas óptimas para el reparto de los productos a los clientes.

Se puede colegir de todo lo expuesto que existen dos aspectos determinantes a la hora de facilitar la toma de decisiones: información y modelos matemáticos –vehículos para la formulación de los análisis–.

La primera actúa como materia prima de los segundos y, retomando las citas planteadas al inicio, esta combinación permite pasar de la mera información a lo que realmente importa, el conocimiento.

A partir de los años 60 del pasado siglo se ha producido un incremento, que podríamos caracterizar como exponencial, en la disponibilidad de información –geográfica y alfanumérica–, a la vez que también se han desarrollado numerosos modelos matemáticos para dar respuestas a la necesaria planificación de las actividades acometidas en el territorio por las organizaciones. Como el lector podrá fácilmente deducir, se enfrentan dos retos fundamentales:

- Disponer de entornos computacionales que faciliten las labores de gestión de volúmenes de datos ingentes, heterogéneos y con elevadas tasas de crecimiento, lo cual trasciende la tarea nada desdeñable de representación cartográfica de la información.
- Habilitar soluciones informáticas que permitan dar respuesta a las necesidades funcionales de una amplia variedad de perfiles de usuarios que superan los ámbitos tradicionales de los SIG en los que se cuenta con técnicos especializados –geógrafos y topógrafos, principalmente–.

Ligando ambas problemáticas, la necesidad de compartir dentro y fuera de las organizaciones productoras de IG los enormes volúmenes de información que permiten obtener los sensores actuales, así como la integración de los mismos en procesos analíticos, muestra un escenario en el que las arquitecturas de sistemas distribuidos juegan un papel vital a la hora de dar una respuesta adecuada a los retos planteados.

2

Presentado el contexto, este Trabajo Fin de Máster (TFM) opta por realizar un estudio de investigación cuyo **objetivo principal es el de diseñar conceptualmente una arquitectura distribuida que dé respuesta a las necesidades funcionales actuales y presente unas características, en términos de escalabilidad funcional y operativa de los SIG, que permita su crecimiento futuro.**

La consecución de este objetivo implica una necesaria revisión de otros aspectos que se conforman como **objetivos secundarios** de este trabajo:

- **Identificación del abanico funcional** que permita cubrir las demandas de la actual sociedad de la información –geográfica–.
- **Análisis comparativo de los principales modelos de arquitecturas distribuidas** existentes, bajo el enfoque de las necesidades funcionales propias del ámbito geoespacial.
- **Estudio de los estándares geoespaciales** que dan respuesta a las necesidades funcionales identificadas, desde la perspectiva de los modelos arquitectónicos analizados y los protocolos o interfaces subyacentes.

La respuesta a los retos planteados se plantea articulando este trabajo en los siguientes apartados:

- **Estado del arte:** presenta una disertación sobre la evolución experimentada por los SIG que permitirá al lector entender los retos planteados en la actualidad dentro de este ámbito.

- **Arquitecturas distribuidas:** estudia las características arquitectónicas de las dos plataformas de sistemas distribuidos que copan las implementaciones en el mercado mundial: Arquitectura Orientada a Servicios (SOA) y Arquitectura Orientada a Recursos (ROA).
- **Arquitectura para un SIG distribuido:** analiza desde un punto de vista funcional la adecuación de las arquitecturas estudiadas anteriormente a los retos planteados. Para ello, se estudian las soluciones arquitectónicas que han promulgado algunos organismos de referencia en el ámbito de la IG y los SIG, tras lo cual se formula el diseño de una arquitectura híbrida que concilia una solución para los retos indicados anteriormente. Dada la imprescindible concisión del marco en el que se realiza este trabajo de investigación, no se plantea el desarrollo de un prototipo real que plasme físicamente la arquitectura resultante.
- **Líneas futuras de investigación:** resume el conjunto de variables, extraídos del análisis realizado en los capítulos anteriores, que es preciso monitorizar de cara a ir adaptando la plataforma propuesta a los nuevos requerimientos que surgirán en el futuro.
- **Conclusiones:** se cierra el estudio con un análisis del grado de cumplimiento de los objetivos planteados inicialmente –principal y secundarios–.

Espera el autor que la lectura de esta memoria resulte del interés del lector, captando su atención por un entorno tecnológico que está cobrando, día a día, una mayor importancia en la toma de decisiones en múltiples ámbitos de la sociedad.

2 Estado del arte

Entender los retos a los que se enfrentan los SIG implica echar una mirada al pasado para conocer las circunstancias que han marcado su evolución. De esta forma, podrá caracterizarse la situación actual y poner en relación con la misma las demandas existentes en la actualidad.

2.1 *Sistemas de Información Geográfica*

Aunque existen diversas definiciones sobre el concepto de SIG, una de las más reseñables es la enunciada en el "Core Curriculum in GIS" del *National Center for Geographic Information Systems and Analysis (NCGIA)* [Goo97] que define un SIG como "un sistema de hardware, software y procedimientos diseñado para realizar la captura, almacenamiento, manipulación, análisis, modelado y representación de datos referenciados espacialmente para la resolución de problemas complejos de planificación y gestión".

La mayoría de las actividades humanas pueden circunscribirse a un ámbito espacial y existen numerosas referencias que apuntan a una cita según la cual el 80% de la información que se gestiona en el ámbito de las administraciones tiene una componente geográfica –por ejemplo la recogida en [Han92]–, aunque no existe ningún estudio que avale científicamente dicha afirmación –de hecho la investigación recogida en [Bur11] no es concluyente–. Este hecho, más allá de la precisión de la cifra reflejada, debe mover al lector a reflexionar sobre la enorme importancia que tiene poder manejar información geográfica. En los siguientes subapartados se analiza la evolución que han vivido los SIG, recogiendo las variables contextuales que han caracterizado cada una de las etapas. La delimitación de los periodos de cada una de ellas es meramente orientativa, puesto que no existen unos límites precisos.

2.1.1 *Cartografía por ordenador (1960-1969)*

Como en cualquier otro ámbito en el que se trata con una tecnología incipiente o emergente a mediados del siglo XX, los primeros pasos fueron dados dentro del ámbito de los centros de investigación vinculados a empresas y universidades norteamericanas. Este hecho implica que exista muy poca documentación al respecto y acceder a la existente implica una ardua tarea de búsqueda en los archivos de dichos centros. Afortunadamente, esta labor fue acometida meritoriamente por los autores de [Cop91], sirviendo de referencia su estudio para lo que aquí se presenta.

Este periodo, se caracteriza por una creciente necesidad de automatización de los procesos de producción cartográfica. Las numerosas críticas recibidas por los autores de la publicación del *Atlas of Great Britain and Northern Ireland* [Bic65], con respecto a su falta de actualización y su compleja gestión, hicieron ver a uno de sus promotores –Bickmore– que sin la ayuda de los computadores no se podría disponer de un mecanismo rentable de comprobación, edición y clasificación de la información. Este hecho se veía reforzado por la continua reducción de costes de los computadores y, a la vez, el aumento de sus capacidades de procesamiento. En este último aspecto, sirva como referencia la aparición en 1964 del modelo de ordenador 360/65 de *IBM*, el cual multiplicaba por 400 la velocidad de procesamiento y en 32 la capacidad de memoria de su antecesor, el *IBM 1401* –1959–.

Abundando en el aspecto computacional, es preciso indicar que se está en una era caracterizada por ordenadores de enormes dimensiones y capacidades de cálculo –extremadamente limitadas desde la perspectiva actual– especialmente orientadas a la ejecución de rutinas administrativas y gestión de información, tanto en el ámbito público como privado, así como para finalidades científicas que requiriesen cálculos extensivos –matemáticas, física, química– y con mínimas –o inexistentes en la mayoría de los casos– capacidades de representación gráfica. Esta categoría de computadores se denominaron *mainframes*.

El salto al ámbito de la gestión de información numérica vino promovido por la necesidad de gestionar volúmenes ingentes de datos catastrales, siendo especialmente relevante el trabajo realizado por el *District of Columbia* y el *Nassau County* de Nueva York. A ellos se sumó el *Bureau for the Census* del gobierno de Estados Unidos con un proyecto de automatización de geocodificación de direcciones y el *New Haven County* que, en 1967, realizó las primeras demostraciones de geocodificación, generación de cartografía por ordenador –*mapping*– y análisis de datos en pequeñas áreas.

Paralelamente, la creciente disponibilidad de computadores en las universidades, conllevó en este periodo una revolución en el estudio de la geografía, enfatizando el tratamiento estadístico de datos geográficos y su modelado. En este sentido, cabe remarcar la creación del *Harvard Laboratory for Computer Graphics and Spatial Analysis* –1965– por parte de Howard Fisher, centro de investigación que desarrolló *SYMAP*, paquete de generación de cartografía con capacidad para el cálculo de isolíneas, mapas de coropletas y análisis de proximidad. El éxito fue tal que se llegaron a distribuir 500 licencias de este software (tanto en el ámbito universitario como en el empresarial).

6

La aparición del primer SIG, propiamente dicho, tuvo lugar bajo la batuta de R. Tomlison, el cual lideró el proyecto de creación del *Canada Geographic Information System (CGIS)* para el gobierno canadiense en el año 1966, aunque el mismo no estuvo completamente operativo hasta 1971.

En líneas generales, cabe indicar que existía muy poco contacto entre los equipos de desarrolladores de los diferentes proyectos que se estaban desarrollando coetáneamente. Esta característica provocó que se produjera una importante duplicación de esfuerzos y recursos, generando productos con características similares en diferentes organizaciones.

Desde un punto de vista organizativo y pese al éxito comercial de *SYMAP*, es incuestionable que el coste de implantación de un SIG y las limitaciones funcionales propias de los inicios de esta tecnología, fueron los principales factores que impidieron una mayor proliferación. Por otro lado, la complejidad de sus interfaces de usuario requería disponer de personal altamente cualificado y escaso en el mercado. Estas circunstancias provocaron que en las organizaciones en las que se implantaron, el SIG fuera administrado por un conjunto muy reducido de expertos, a los cuales se remitían todas las peticiones que pudieran surgir dentro de la organización.

2.1.2 SIG departamentales (1970-1989)

Los últimos años de la década de los 60 del pasado siglo vieron nacer una nueva tipología de computadores que se denominaron “estaciones de trabajo” –*workstations*–. Estos nuevos ordenadores se caracterizaban por tener unas capacidades computacionales más discretas que los *mainframes*, pero suficientes para el ámbito de la ingeniería y la educación, gracias a lo cual se abarataba notablemente su

precio. Por otro lado, su reducido tamaño hizo que por primera vez se hablase de micro-computadores, destinados a ser utilizados por un único usuario.

El primer representante de esta categoría fue comercializado por el fabricante *IBM* en el año 1965 y se trataba del modelo 1130. Sin embargo, la madurez de este mercado se consiguió durante la década de los 70, con la incorporación de nuevos modelos desarrollados por *IBM*, *Digital Equipment Corporation* y *Xerox*. Posteriormente, a finales de los 80 las *CPU* comenzaron a diseñarse con microprocesadores que disponían de instrucciones más pequeñas y simples –arquitecturas *RISC*– que, consecuentemente, tardaban menos tiempo en ejecutarse. Al calor de esta nueva arquitectura aparecen nuevos fabricantes como *Hewlett Packard*, *Sun Microsystems* o *Silicon Graphics*, todos ellos implementando variantes propietarias del sistema operativo *UNIX*.

Esta evolución tecnológica permitió gestar grandes avances en el ámbito de los SIG. La primera referencia importante la constituye *ODISSEY*, primer SIG capaz de gestionar información vectorial y que fue desarrollado en el centro de investigación de la *University of Harvard* reseñado anteriormente. *ODISSEY* fue concebido en el año 1975 por N. Chrisman y D. White como una *suite* de programas conectados y con una interfaz de usuario común. En el año 1982, la última versión de este SIG incluía siete programas para la realización de análisis geográficos. Aunque no triunfó comercialmente, sí marcó el devenir de las futuras generaciones de SIG.

Sin embargo, más allá de los proyectos de investigación auspiciados en el ámbito universitario, empieza a consolidarse un tejido empresarial en el área del desarrollo y comercialización de productos SIG. La mayoría de las empresas que nacieron en esa época, como *Intergraph*, *ComputerVision* y *Synercom*, provenían del ámbito del diseño asistido por ordenador (*CAD/CAM*); pero también se crean otras de nuevo cuño. Entre ellas encontramos la empresa que ha liderado el mercado de los SIG prácticamente desde su creación en 1969: *Environmental Systems Research Institute (ESRI)*. Después del desarrollo de productos como *GRID* y el módulo de visualización tridimensional (3D) del mismo, *ESRI* lanzó al mercado en 1982 el SIG que marcó una auténtica revolución: *Arc/INFO*. Se trataba del primer SIG multiplataforma, capaz de funcionar en *mainframes*, estaciones de trabajo y, posteriormente, en ordenadores personales.

Esta nueva coyuntura del mercado SIG, marcada por el abaratamiento de los costes de implantación de los SIG, favorece una mayor difusión de los mismos. No obstante, siguen siendo aplicaciones complejas que requieren personal altamente cualificado y, aunque la forma de trabajar no varía sustancialmente con respecto a la descrita en el periodo anterior, sí que se aprecia un cambio en las políticas de gestión de la IG dentro de las organizaciones.

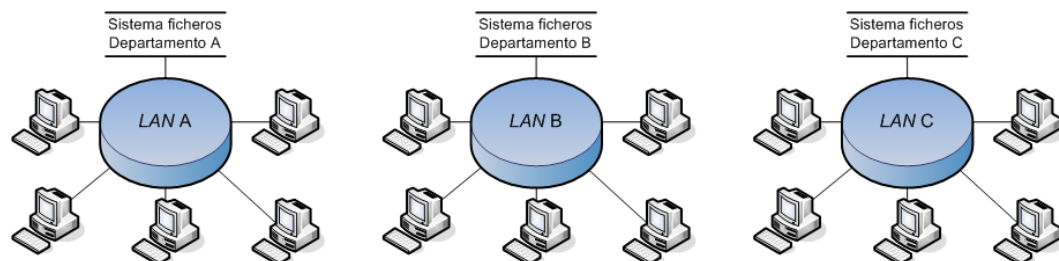


Figura 2.1 Arquitectura SIG departamental aislada [fuente: propia].

La existencia de varios usuarios con acceso a aplicaciones SIG y la necesidad de compartir información con la finalidad de mejorar el rendimiento y obtener mejores resultados, apoyada en el importante desarrollo de las redes de comunicaciones *Ethernet* –locales (*LAN*) y amplias (*WAN*)–, provoca que se produzca una migración desde instalaciones SIG con datos locales en cada uno de los puestos de trabajo hacia instalaciones SIG con datos centralizados en repositorios comunes. De ahí surge el concepto de SIG departamental con el que se titula este periodo. La Figura 2.1 permite apreciar un primer estadio en el que se comparte información dentro de los departamentos, mientras que la Figura 2.2 muestra una evolución en la que la información también se comparte entre los diferentes departamentos de la organización a través de redes de mayor alcance.

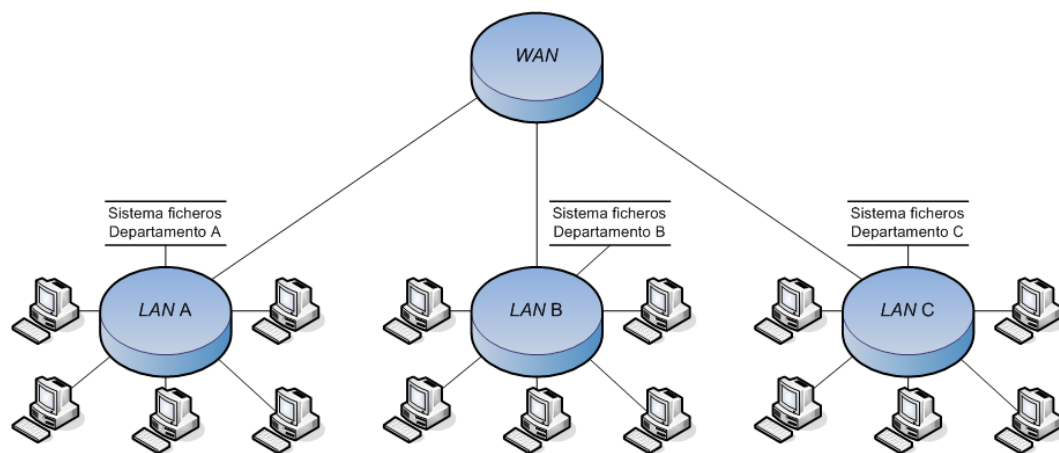


Figura 2.2 Arquitectura SIG multidepartamental [fuente: propia].

2.1.3 SIG corporativos (1990-1996)

Aunque la existencia de los ordenadores personales (*PC*) es anterior a la década de los 90, la irrupción en el ámbito de los SIG se produce en este periodo. Se trata de ordenadores que disponen de un único procesador –en un inicio–, que se benefician de los grandes avances tecnológicos vividos en el ámbito de las tecnologías de la información (*TI*) y del abaratamiento de los costes de fabricación. Estos factores favorecen su amplia difusión a segmentos de la población que en épocas anteriores habían quedado totalmente apartados de los computadores. Esta situación también se manifestó en el ámbito profesional, extendiéndose la presencia de computadores en las organizaciones, hasta el punto de que hoy en día son piezas imprescindibles en el equipamiento de prácticamente cualquier puesto de trabajo.

Las capacidades de estos computadores también centraron la atención de los fabricantes de SIG que veían en estas nuevas arquitecturas posibilidades para extender el uso de las tecnologías SIG en las organizaciones, independientemente del tamaño que éstas tuvieran. El máximo exponente de esta nueva era lo constituye el producto lanzado por *ESRI* en el año 1992: *ArcView* [Pet08]. Se trata del primer SIG que se cataloga como “*desktop mapping*” y se caracteriza por disponer de un abanico funcional muy reducido, en comparación con las soluciones existentes hasta ese momento. Esta simplificación se basaba, en términos de marketing, en el siguiente eslogan: “el 90% de los usuarios utiliza tan solo un 10% de las funciones de un SIG”. Partiendo de esta premisa, se priorizó el diseño de interfaces de usuario mucho más sencillas, en las cuales ese subconjunto de funciones habilitadas pudiera ser ejecutado por usuarios mucho menos expertos. Además, esa simplificación también pasa por poner en el

mercado versiones para una nueva plataforma, la que estaba liderando el mercado de los ordenadores personales: *Microsoft Windows*. La estela de *ESRI* también fue seguida por otros fabricantes que irrumpieron en este segmento, como *MapInfo* con su aplicación *MapInfo Professional* e *Intergraph* con *Geomedia Professional*.

Paralelamente, *ESRI* continúa con el desarrollo de *Arc/INFO* y comercializa la versión para *PC* – denominada *PC Arc/INFO*– que, además, permite el intercambio de información con las versiones para estaciones de trabajo mediante el formato de intercambio *Export Arc/INFO* –extensión *E00*–.

Pese a la simplificación de las interfaces y a la disponibilidad de herramientas para la personalización de los *desktop mapping*, también es importante destacar la demanda por parte de los integradores de SIG de crear soluciones empresariales personalizadas, con sus propias interfaces, distintas de las que proveen los fabricantes. *ESRI* dio respuesta a esta necesidad y comercializó *MapObjects* –1996–, librería de componentes para *Microsoft Visual Basic* –posteriormente también se distribuyó una versión para *Java*– que permitía a los desarrolladores integrar en las aplicaciones de negocio que desarrollaban un apartado dentro de la interfaz que pudiera mostrar mapas, así como disponer de un conjunto reducido de funciones de análisis y representación SIG.

Por otro lado, en esta época también se concentran dos nuevos focos de evolución tecnológica de los SIG. El primero responde a la necesidad de eliminar los “cuellos de botella” que se ocasionaban dentro de las organizaciones entre los técnicos delineantes –encargados de los procesos de creación de la información vectorial en formatos propios del ámbito *CAD* como *DWG*, *DXF* o *DGN*– y los técnicos SIG que debían cargar esta información en los SIG. Las numerosas inconsistencias que se presentaban ralentizaban mucho los procesos de migración y este hecho motivó que tanto *ESRI*, como posteriormente *Autodesk* –fabricante de la aplicación *CAD* más utilizada, especialmente entre el colectivo de arquitectos– y *Bentley* –aliado comercial en esos momentos de *Intergraph*, fabricante de *Microstation*, segunda plataforma *CAD* en cuanto a implantación, pero con presencia mayoritaria en los ámbitos de producción cartográfica– optaran por desarrollar extensiones SIG para los productos *CAD* que permitieran estructurar y validar topológicamente el contenido de los planos y mapas antes de ser pasados a los técnicos SIG. La solución de *ESRI*, desarrollada conjuntamente con *Autodesk*, se denominó *ArcCAD* –1993– y funcionaba sobre la plataforma *AutoCAD*. Pocos años más tarde, la alianza entre *ESRI* y *Autodesk* se rompe y esta última lanza al mercado el producto *AutoCAD Map*, el cual integra de forma transparente funcionalidades SIG dentro de la plataforma *CAD* reseñada –1997–. Paralelamente, *Bentley* lanza al mercado una solución homóloga sobre su plataforma *CAD*, denominada *Microstation Geographics*.

El segundo foco de evolución está relacionado con la gestión de la IG. En el periodo anterior ya se marca una clara tendencia a la compartición de la IG dentro de los departamentos. En este periodo esta tendencia se acentúa –teniendo en cuenta la proliferación de nuevos usuarios SIG, especialmente motivada por las nuevas plataformas “*desktop mapping*”–, puesto que cada vez cobra cada vez más importancia el hecho de estandarizar el proceso de creación y gestión de los datos, así como facilitar la gestión de la concurrencia de usuarios. Estas necesidades ya estaban cubiertas en el ámbito de la información alfanumérica con los Sistemas Gestores de Bases de Datos (*DBMS*), así que lo que se necesitaba era dotar a esos motores de gestión de capacidades para el almacenamiento de IG y de funciones de análisis espacial.

Con la finalidad de estandarizar los formatos de descripción de la IG, a finales de 1994 la Organización para la Estandarización Internacional (ISO) hizo oficial la extensión del estándar 13249 con un tercer apartado dedicado a la IG, el cual fue denominado *ISO 13249-3 SQL/MM Part 3: Spatial*. Los proveedores de *DBMS* del momento y todos los que surgieron posteriormente –tanto de tecnología propietaria como de código libre–, habilitaron estas opciones espaciales dentro de sus productos, siguiendo las directrices del estándar.

El primer fabricante que se dedicó a este cometido fue Oracle –principal proveedor mundial de *DBMS*– que, de forma conjunta con el *Canadian Hydrographic Service*, desarrolló una modificación en el núcleo de su *DBMS* homónimo para habilitar capacidades de gestión de IG. Esta modificación vio la luz, comercialmente hablando, con la aparición de la versión 7.3 de Oracle –1996– y bajo la denominación *Spatial Data Option (SDO)*, aunque en la actualidad se comercialice bajo el nombre comercial de Oracle *Spatial*.

En otras latitudes, la empresa australiana *Geographic Technologies Inc.* comenzó el desarrollo de un motor geográfico que denominó *Spatial DataBase Engine (SDE)*, el cual fue comercializado tras la compra de este proyecto por parte de *Salamanca Software Pvt Ltd.*, empresa que, a su vez, fue adquirida por *ESRI* en el año 1996. Tres años más tarde, *ESRI* anuncia el lanzamiento de *ArcSDE*, componente –middleware– que dotaba de habilidades geográficas a los principales *DBMS* –en 2012 era compatible con *IBM DB2*, *Informix*, *Microsoft SQL Server* y *Oracle*–, independientemente de la plataforma. Operativamente, *ArcSDE* permitía realizar migraciones entre bases de datos implementadas sobre cualquiera de los *DBMS* soportados, de forma transparente para los usuarios. Además, con respecto a *SDO*, disponía de una mayor funcionalidad.

10

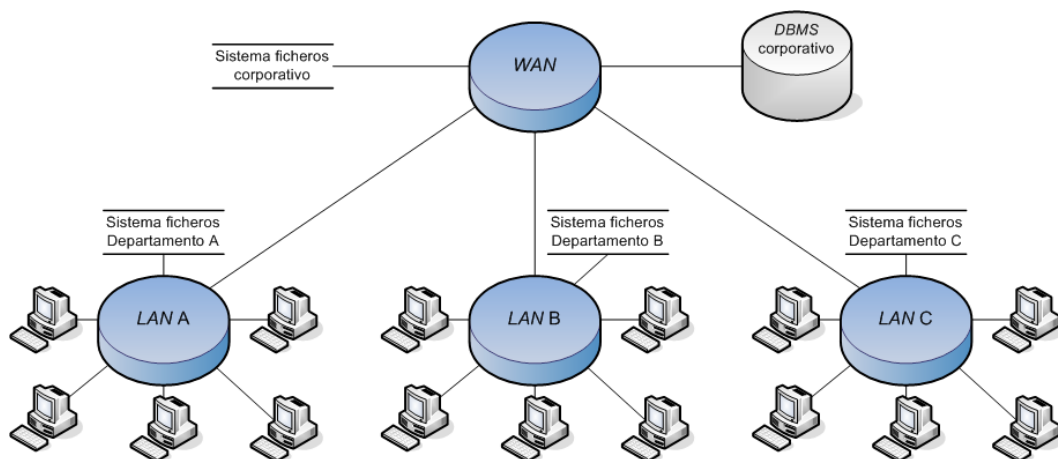


Figura 2.3 Arquitectura SIG corporativa, parcialmente centralizada [fuente: propia].

Este panorama, aparte de la proliferación de SIG en las organizaciones, extendiéndose por todos aquellos departamentos que tienen necesidades de gestión de IG y alfanumérica asociada, también origina un proceso de migración de los datos a repositorios centralizados para toda la organización, más allá de los repositorios departamentales propios de la época anterior, y soportados sobre *DBMS*. Se había entrado en la era de los SIG corporativos. Las figuras anteriores muestran dos posibles estados de dicha migración; en la Figura 2.3 se ha pasado a la base de datos espacial y al sistema central de ficheros –gestionados ambos por el departamento de TI– una parte de la información –normalmente

bases cartográficas de referencia–, mientras que la información restante sigue gestionándose a nivel departamental. En la Figura 2.4, por su parte, el proceso de migración abarca a todas las fuentes de información.

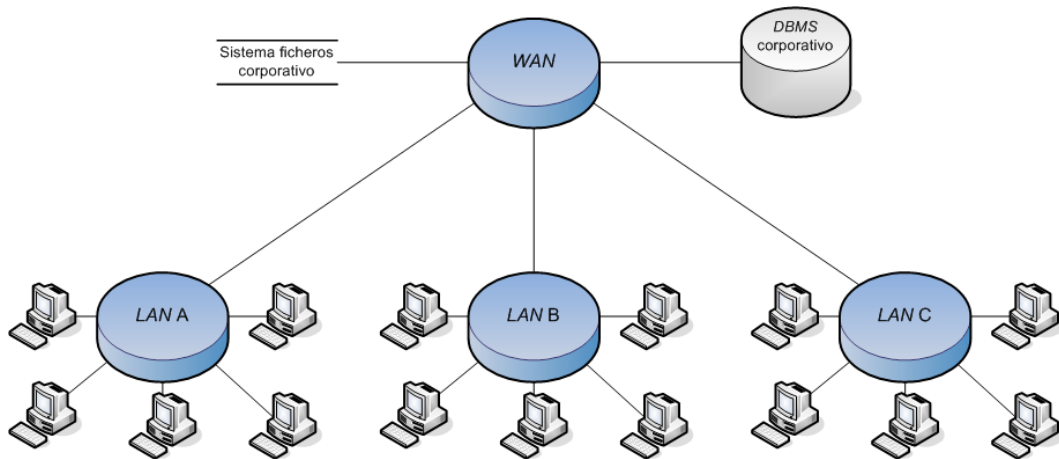


Figura 2.4 Arquitectura SIG corporativa, totalmente centralizada [fuente: propia].

2.1.4 SIG comunitarios (1997-actualidad)

Desde un punto de vista de TI, las postrimerías del pasado siglo muestran una revolución propiciada por Internet. La denominada “red de redes” pone de manifiesto las enormes posibilidades que este medio habilita gracias a poder compartir información entre comunidades, estados y organizaciones en todo el mundo. En primera instancia, la posibilidad de publicar puntos de acceso a contenidos geográficos que pudieran ser directamente utilizados por los usuarios, independientemente de su ubicación, simplemente con el condicionante de tener acceso a Internet, fue la que lideró los primeros desarrollos. En este sentido, el primer prototipo fue desarrollado en 1993 dentro del *Xerox PARC* y se denominó *Xerox PARC Map Viewer*. Comercialmente hablando, el primer producto comercializado fue *Argus Mapguide* de la empresa *Argus Technologies*, la cual fue adquirida por *Autodesk* a finales de 1996 y su solución comercializada en 1997 como *Autodesk Mapguide* –en la actualidad distribuido como *Mapguide Open Source* bajo licencia de código libre–. Poco tiempo después aparecieron las propuestas de *Intergraph* con *Geomedia WebMap* –1996– y *ESRI* con *ArcIMS* –1997, sustituida en 2004 por la plataforma *ArcGIS Server*–.

Las consecuencias son claras, empieza a existir un importante número de recursos –servicios Web (WS)- de acceso a IG en Internet, aunque existía también un hándicap muy importante: cada fabricante había diseñado un protocolo propietario de comunicación con su plataforma servidora. De esta forma, un integrador de una aplicación Web que necesitara utilizar recursos provenientes de diferentes plataformas tecnológicas, debía lidiar con diferentes protocolos, cosa que complicaba sustancialmente el proceso de desarrollo. Como consecuencia de esta nueva problemática –que en el fondo no es más que una reedición de los problemas existentes con la diversidad de formatos que siempre ha caracterizado a la IG–, surge una iniciativa en 1994 denominada *Open GIS Consortium (OGC)* –en la actualidad *Open Geospatial Consortium*– que aglutina a actores de diferentes ámbitos de la comunidad SIG, desde universidades a empresas, pasando por administraciones públicas y otras organizaciones; con la

finalidad de colaborar para consensuar el desarrollo e implementación de estándares abiertos para datos y servicios geográficos, análisis SIG y compartición de datos. Entre los estándares promulgados por OGC, los relativos a los servicios de publicación de cartografía son ampliamente adoptados por las empresas que comercializan servidores de mapas en Internet.

Este nuevo escenario, que no menoscaba que se sigan desarrollando plataformas tecnológicas en los ámbitos tratados en las etapas anteriores –a título ilustrativo, en 1999 *ESRI* lanza la plataforma que venía a sustituir a *Arc/INFO* y *ArcView*, denominada *ArcGIS Desktop* y que es el software SIG más extendido mundialmente–, provoca el nacimiento de una nueva era, la de los SIG comunitarios –simplificada en algunos casos con los términos “*Web mapping*” o “*Web SIG*”–, puesto que su alcance traspasa al de las propias organizaciones, abarcando otros colectivos de usuarios que son consumidores de los recursos producidos por las mismas. Comienzan a aparecer numerosos portales Web con interfaces para la visualización de cartografía y cada vez un mayor número de usuarios acceden a estas interfaces SIG, mucho más sencillas que sus “hermanas mayores”. Sin embargo, no solo hay que pensar en entornos Web, en las propias redes de las empresas también se desarrollan aplicaciones Web que permiten a usuarios no cualificados, desde el punto de vista SIG, acceder a contenidos concebidos para sus necesidades de una forma sencilla. Al calor de este movimiento, aparecen también plataformas de código libre –*open source*– tanto en el ámbito de publicación Web –*Mapserver* en 1997, *GeoServer* en 2001, etc.–, como en el de los tradicionales SIG –*QuantumGIS* en 2002, *gvSIG* en 2004, etc.–.

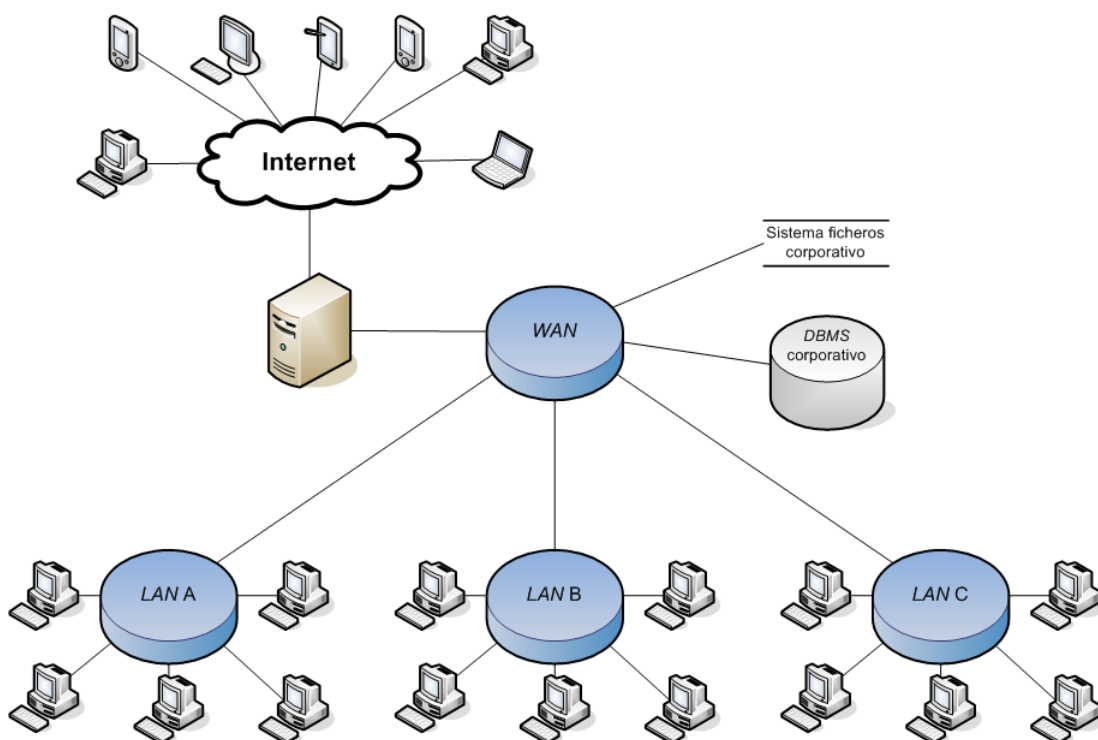


Figura 2.5 Arquitectura SIG comunitaria [fuente: propia].

Como consecuencia de todo lo expuesto, se produce una apertura al “exterior” de los entornos SIG corporativos gracias a la incorporación de un servidor de mapas por Internet (*IMS*) –véase Figura 2.5– que actúa como pasarela para que toda la panoplia de usuarios –en su sentido más amplio, puesto que

pueden ser otras organizaciones o cualquier ciudadano, dependiendo de las restricciones de acceso que pueda fijar la organización que actúa como proveedora— pueda acceder a la información de forma estandarizada mediante diferentes dispositivos.

2.2 Retos actuales

Pese a la enorme evolución acaecida en el ámbito de los SIG, dotándose de infraestructuras distribuidas a las que es posible conectarse de forma remota desde múltiples artefactos, la evolución experimentada por este mercado no ha sido suficientemente rápida como para dar respuesta a las crecientes necesidades vividas en los últimos años, de la mano de acontecimientos que dan lugar a una revolución tecnológica de dimensiones difícilmente imaginables. Esta revolución se fundamenta en el establecimiento de las primeras redes sociales y en la necesaria adaptación de la Web tradicional a las necesidades que éstas conllevan.

En su sentido más amplio, una red social es una estructura social constituida por personas u organizaciones que se conectan como consecuencia de la existencia de algún tipo de relación o interés común [Pon12]. Estas conexiones, por tanto, van más allá de un ámbito concreto y su existencia podemos considerar que es tan antigua como la del hombre. La antropología y la sociología han venido estudiando este fenómeno desde hace muchas décadas, caracterizando las relaciones que se definen dentro de diferentes grupos de población. La matemática y las ciencias de la computación, por su parte, han abordado el análisis de las mismas modelando el comportamiento observado por antropólogos y sociólogos, con la finalidad de intentar automatizar el proceso de deducción de información a partir de las relaciones existentes; valiéndose para ello de la representación de las redes mediante grafos, compuestos por nodos —cada uno de los cuales se asimila a un individuo u organización— y aristas —indican las relaciones existentes—, cuyo análisis se realiza aplicando las enseñanzas propias de la teoría de grafos. Pese a este notable bagaje, las redes sociales no adquieren una relevancia importante para la común de las personas hasta la irrupción de *Facebook* en el año 2004 y *Twitter* en 2006. Estas dos redes sociales, que aprovechan las capacidades de conexión de Internet, aglutinan a millones de usuarios y portan el estandarte de la revolución tecnológica vivida en paralelo a la irrupción de dispositivos móviles con extraordinarias capacidades de computación, —resultando especialmente reseñables los *smartphones*— y al aumento del ancho de banda disponible para las comunicaciones; aunque existen precedentes que llevarían al lector hasta mediados de la década de los 90 del siglo pasado, con la aparición de las primeras redes sociales en Internet —a título ilustrativo, *GeoCities* se fundó en 1994—. En cualquier caso, las capacidades de conexión y comunicación de información entre los distintos usuarios de las redes, porta aparejado un aumento exponencial de la demanda de contenidos entre los que, como no podía ser de otra forma, cada vez es mayor la presencia de aquellos que llevan asociada la localización —componente geográfica—.

Como se puede deducir, este nuevo escenario también implica un rol más predominante de los usuarios en lo que respecta a la formulación de los contenidos presentes en la red. Las páginas tradicionales se caracterizaban por tener contenidos mayoritariamente estáticos y que, además, no eran actualizados con frecuencia. El éxito de Internet pasaba por tender a generar páginas con mayor contenido dinámico, en los que los sistemas de gestión de contenidos debían servirse de repositorios —típicamente bases de datos— que proveyeran información actualizada y, especialmente, personalizada en función del perfil del

usuario que accediese a las páginas. Este hecho, junto con el diseño de interfaces más sencillas y atractivas para dichos usuarios, debía constituir la plataforma sobre la que se permitiera dar respuesta a las crecientes necesidades de las diferentes plataformas de contenidos y muy especialmente a las redes sociales. Todo ello dio lugar a un nuevo hito en la forma en la que se entiende la Web, acuñándose el término de Web 2.0 en referencia a una nueva generación de aplicaciones y páginas en Internet que utilizan la inteligencia colectiva para proporcionar servicios interactivos en red. El usuario pasa de ser un sujeto pasivo que recibe información a contribuir de forma activa a la generación de contenidos en la red –un claro ejemplo lo constituye el proyecto *OpenStreetMap (OSM)* para la creación de una base cartográfica libre con la participación directa de ciudadanos que crean y/o actualizan contenidos cartográficos de forma altruista–.

De la mano de estos dos fenómenos, resulta interesante mencionar otro concepto que tiene mucho en común con las demandas de la sociedad tecnológica actual: la “computación ubicua”. El término fue enunciado de manera visionaria en el año 1988 por el Dr. Mark Weiser [Bro99] y se basaba en los siguientes principios:

- El propósito de un ordenador es el de ayudar a realizar alguna tarea.
- El mejor computador es aquel que actúa como un sirviente tranquilo, invisible.
- Cuantas más cosas puedas hacer por intuición, más inteligente serás; el computador debe ampliar tu inconsciencia.
- La tecnología debe ser capaz de crear calma.

Bajo estos preceptos conceptuales, la visión que tuvo el Dr. Weiser ha fructificado en los últimos años con el auge de los dispositivos móviles –ordenadores portátiles, tabletas y *smartphones*– provocando una importante demanda de aplicaciones con contenidos y servicios SIG embebidos, tanto para su uso en ámbitos profesionales como de ocio. En el primer caso, estas nuevas plataformas permiten ampliar el abanico a usuarios, dentro de las organizaciones, que realizan trabajo de campo para inventariar y gestionar infraestructuras, requiriendo poder acceder a la información en tiempo real. En el segundo, se trata de disponer de contenidos cartográficos y funciones SIG muy sencillas orientadas a múltiples fines, como el de consumo de bienes o el cultural.

Esta proliferación de los dispositivos móviles se sustenta en unos importantísimos avances tecnológicos que han permitido la encapsulación de componentes *hardware* en recipientes cada vez más pequeños. Entre estos componentes encontramos diferentes tipos de sensores que también han permitido una importante evolución tecnológica en el ámbito de las ciudades, la cual ha llevado a que se acuñe el término “ciudad inteligente” –*smart city*–. Este concepto hace referencia a la posibilidad de crear un entorno urbano, basado en la sostenibilidad, que sea capaz de atender de forma óptima las necesidades de todos los actores presentes en este ámbito –instituciones, empresas, ciudadanos–. Para ello, se precisa disponer de un conjunto de sensores –indicadores de capacidad disponible de los contenedores, de disponibilidad de plazas de aparcamiento, etc.– que midan determinadas variables y envíen dicha información a unos repositorios centrales, para su análisis mediante modelos matemáticos que permitan conocer la situación de una determinada problemática –acumulación de residuos urbanos, vialidad

urbana, estado del transporte público, etc.– y tomar decisiones para solventar las posibles incidencias observadas, así como para planificar mejor la gestión de los servicios implicados.

Dentro de este contexto de las ciudades inteligentes, la variable más importante es la geográfica, puesto que todos los datos que se manejan están referenciados a unas coordenadas, de forma directa o indirecta –aquéllos que están unidos mediante un código identificador a algún elemento del que se conoce su ubicación–. Esta variable está presente en la totalidad de los procesos de análisis de la información y es aquí donde los SIG juegan un importante papel. El reto a enfrentar es característico del concepto anglosajón “*big data*”, el cual lleva implícita la necesidad de gestionar datos –ver Figura 2.6–:

- Voluminosos –*volume*–: la proliferación de sensores y la magnitud de las medidas recogidas en algunos casos, provoca la existencia de requerimientos de almacenamiento crecientes de forma dramática.
- En tiempo real –*velocity*–: en muchos casos los análisis implican tener acceso a los datos que se están recogiendo de forma prácticamente inmediata, para que los resultados obtenidos sean eficaces.
- Heterogéneos –*variety*–: la procedencia tan variada de la información provoca que los formatos en los que ésta es capturada sean muy variados. Poder manejar datos heterogéneos implica un coste añadido. En este ámbito, los procesos de estandarización juegan un papel principal.
- Veracidad –*veracity*–: no es suficiente disponer de mucha información, es preciso poder determinar la precisión de la misma para descartar aquélla que pueda contribuir a falsear los resultados obtenidos.

15

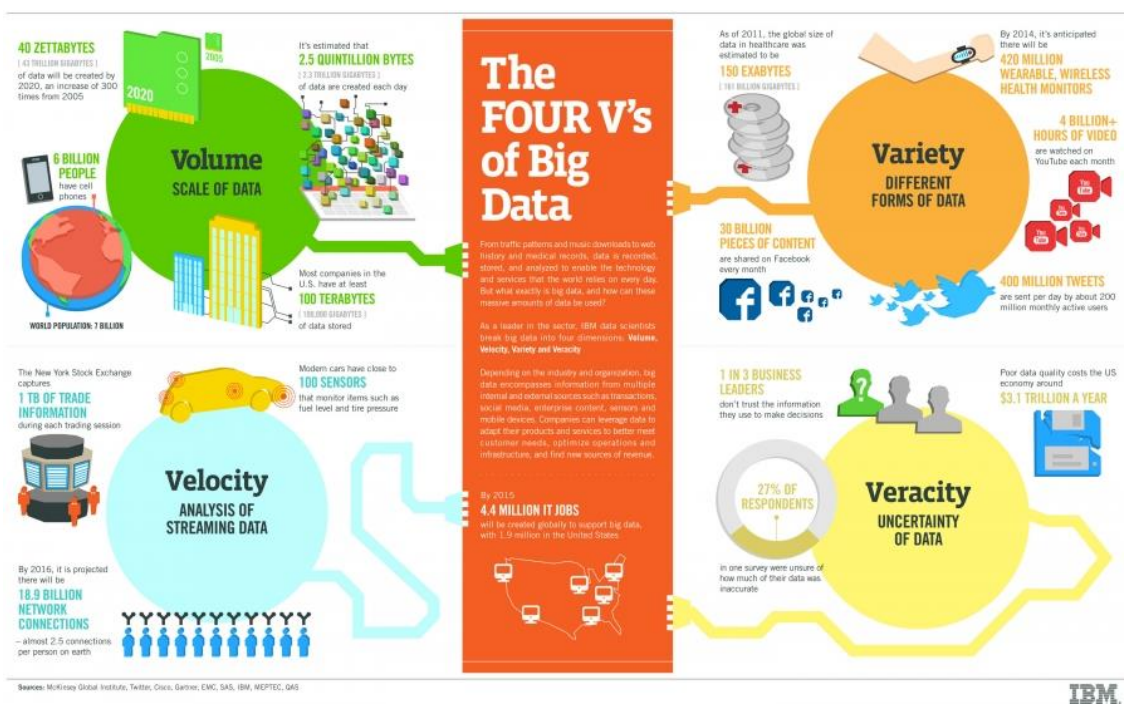


Figura 2.6 Las cuatro características del *big data* [IBM13].

Como consecuencia de todo lo expuesto, el nuevo panorama tecnológico y sociológico plantea una situación en la que la disponibilidad de IG en tiempo real, así como demanda de funciones de análisis de dicha IG combinada con aquella otra alfanumérica a la cual se encuentre vinculada, cobran un papel importantísimo al cual los SIG deben ser capaces de dar respuesta. Ésta deberá venir de la mano de la creación de arquitecturas distribuidas de publicación de WS que no solo permitan visualizar contenidos cartográficos, si no que aboguen por la presencia de servicios de análisis y procesado de datos, de forma que las plataformas computacionales amplíen sus capacidades funcionales.

Desde el punto de vista de las infraestructuras de TI, si en un mismo sistema se unifica la necesidad de disponer de una creciente oferta funcional –escalabilidad funcional–, junto con la de poder atender a un volumen creciente de clientes accediendo a dichas funciones y a los datos implícitos –escalabilidad operacional–, los costes operativos pueden dispararse y comprometer la propia viabilidad del sistema. En este contexto cobra vital importancia el fenómeno conocido como computación en la nube –*cloud computing*–. Se trata de un nuevo modelo de prestación de servicios de negocio y tecnología, en el que existen unos proveedores de infraestructuras de computación que ponen al alcance de las organizaciones que necesiten desplegar un conjunto de servicios, los recursos necesarios. De esta forma, estas organizaciones pueden responder de forma rápida y eficiente a necesidades temporales o permanentes de recursos de TI, gracias a que estas infraestructuras en la nube permiten un alto grado de automatización de las tareas de gestión, una rápida movilización de recursos, una elevada capacidad de adaptación para atender a una demanda variable, así como una virtualización avanzada y un precio flexible en función del consumo realizado.

A modo de corolario, el informe emitido por la prestigiosa firma *Gartner* [Cea14] en referencia a las diez tendencias tecnológicas más importantes en el año 2014, viene a corroborar todos los extremos apuntados. El resultado de este análisis pone de manifiesto que dichas tendencias vienen dominadas fundamentalmente por los dispositivos móviles y la computación ubicua, así como por el fenómeno del procesamiento en la nube; tal y como se recoge a continuación:

- Gestión de la diversidad de dispositivos móviles –*mobile device diversity and management*–: el enfoque tradicional de los sistemas de información choca en múltiples casos con los intereses de los usuarios, los cuales intentan utilizar sus dispositivos móviles y las aplicaciones en ellas instaladas para hacer que su vida personal y profesional sea más sencilla y productiva. Es necesario, por tanto, que los gestores de los entornos de TI incluyan estos dispositivos entre los terminales con los cuales los usuarios interactúan con el sistema, ofreciéndoles aplicaciones que les permitan ganar en productividad a la vez que no se ponga en riesgo la seguridad del sistema.
- Aplicaciones móviles y aplicaciones –*mobile apps and applications*–: la creciente pujanza que experimenta el ámbito de las aplicaciones móviles –también denominadas *Apps*– provoca necesarios ajustes en las tradicionales estrategias de diseño y gestión del ciclo de vida del software, así como en los modelos de financiación necesarios para acometer su desarrollo.
- Internet de todo –*Internet of Everything, IoE*–: Internet juega un papel cada día más central en el mundo actual, no solo en términos de infraestructura, sino también en la cultura, la sociedad y el comercio. La denominada Internet de las cosas –*Internet of Things, IoT*– extiende este protagonismo incrementando el número y la diversidad de los dispositivos y los canales de

comunicación, muchos de los cuales se producen esencialmente entre máquinas. Sin embargo, es preciso estructurar estos dispositivos y sus comunicaciones en un marco estable e inteligente – *smart*– que permita gestionar los servicios ofrecidos a los usuarios.

- Gestión en la nube híbrida y TI como corredor de servicios –*hybrid cloud and IT as service broker*–: los responsables de TI en las organizaciones deben ser capaces de reinventar los procesos de negocio como parte de sus estrategias de comercio digital. La emergencia de la *IoE* y de los dispositivos móviles y aplicaciones en manos de empleados, socios y clientes, conducen hacia la necesidad de utilizar servicios de proceso que combinen la capacidad de procesamiento local –en los propios servidores de las organizaciones– con el procesamiento en la nube.
- Nube/cliente –*cloud/client*–: la tradicional arquitectura cliente-servidor se ha visto afectada con la irrupción de los dispositivos móviles y con la forma en la que sus aplicaciones son distribuidas. En este entorno, la computación en la nube ofrece grandes ventajas en cuanto al proceso de distribución de las aplicaciones cliente.
- La era de la nube personal –*the era of personal cloud*–: las arquitecturas anteriores se focalizan en los modelos arquitectónicos subyacentes que son necesarios para la distribución de aplicaciones y contenido contextualizado para los distintos usuarios. La nube personal, por su parte, se centra en la experiencia personal del usuario cuando interactúa con estos sistemas, habilitándose espacios en la nube para el almacenamiento, sincronización, recuperación y compartición de sus datos de forma ubicua.
- Todo definido por software –*software defined everything*–: esta tendencia pone de manifiesto que la red es el epicentro de las actividades e implica que el software se aplica a todo, desde los servidores hasta el almacenamiento, pasando por los centros de datos y las redes de interconexión. Esto contrasta con una visión más tradicional en la que el *hardware* juega ese papel predominante, pero que no son adecuadas para manejar las necesidades de computación consecuencia de la computación ubicua.
- TI a escala Web –*Web-scale IT*–: es el resultado de dar respuesta a las crecientes necesidades de creación de servicios en la nube que permitan enfrentar la creciente complejidad de los entornos cliente, utilizando la automatización, la definición por software y modelos de explotación que permitan ganar en agilidad y velocidad.
- Máquinas inteligentes –*smart machines*–: los enormes avances en el ámbito de la computación, la disponibilidad de grandes volúmenes de información variada y la habilidad para distribuir resultados mediante servicios en la nube están dando cobertura a la emergencia de un mayor número de máquinas inteligentes, las cuales realizan tareas que hasta hace poco se pensaba que eran territorio reservado para las personas. Este escenario muestra posibilidades para mejorar la efectividad de los procesos que actualmente gestionan personas.
- Impresión 3D –*3D printing*–: los profesionales del ámbito de las TI deben ser capaces de gestionar los cambios que están provocando en el mundo de las pequeñas industrias y las cadenas de suministro esta técnica de impresión, anticipando las oportunidades de negocio que ofrece.

Por todo lo expuesto, queda definido un escenario en el que la capacidad de adaptación a las nuevas tendencias tecnológicas resultará clave para el éxito de las organizaciones.

3 Arquitecturas distribuidas

Como en otros muchos escenarios organizativos, en términos de TI, la solución a lógicas de negocio complejas se construye a partir de alguna tipología de arquitectura distribuida. Dado que los WS constituyen la vía más ampliamente utilizada para la estandarización de la distribución de servicios en Internet, el diseño conceptual de una plataforma SIG capaz de dar respuesta a las necesidades indicadas en el punto 2.2 implica realizar un estudio sobre la adaptación de los dos modelos principales de arquitecturas distribuidas –SOA y ROA– a la gestión de WS de carácter geoespacial.

3.1 Arquitecturas Orientadas a Servicios

En el ámbito del diseño y desarrollo de sistemas distribuidos, el principal paradigma arquitectónico es el constituido por SOA. Esta tipología arquitectónica persigue satisfacer los objetivos de negocio de los entornos en los que se aborda la implantación de un sistema distribuido, gracias a su:

- Facilidad y flexibilidad de integración con sistemas legados: los entornos corporativos suelen disponer de un conjunto de soluciones inconexas –muchas de ellas desarrolladas *ad hoc*–, que permiten resolver necesidades específicas, pero que no están preparadas para funcionar en un entorno distribuido. La integración de estas soluciones dentro del nuevo entorno corporativo es un aspecto fundamental a la hora de alcanzar con éxito la implantación del mismo.
- Convergencia con los procesos de negocio: aspecto que permitirá abordar una reducción de los costos de implementación y obtener una solución más ajustada a la realidad corporativa.
- Innovación en la oferta de servicios: permitiendo generar una cartera de productos más atractiva para los clientes de la organización, sean estos internos o externos.
- Adaptabilidad a los cambios: se agiliza sobremanera el proceso de cambio al que está subyugado cualquier sistema, permitiendo introducir mejoras en los productos ofrecidos de forma más ágil, con lo cual se mejora la imagen ante los clientes y se ayuda a crear una posición ventajosa frente a la competencia.

La consecución de estos objetivos pasa, bajo la perspectiva de este paradigma arquitectónico, por la instrumentación de los productos que se quieren ofertar en forma de servicios software; en lo que se denomina computación orientada a servicios (SOC) [Ben05].

3.1.1 Instrumentación de servicios mediante software

El primer aspecto a consensuar a la hora de analizar un entorno SOC es el del concepto de servicio. En el congreso *Internet Conference on Service Oriented Computing* de 2005 se definió un servicio entregado por un componente software como:

“Elementos de computación autónomos e independientes de la plataforma que se pueden describir, publicar, descubrir, orquestar y programar usando protocolos normalizados, con el propósito de construir redes de aplicaciones distribuidas que colaboran dentro y a través de las fronteras de las organizaciones”.

Como consecuencia, la ingeniería del software ha evolucionado para dar lugar a la ingeniería de servicios, siendo esta una disciplina en desarrollo. Entre los principios de esta nueva disciplina se hallan la fuerte encapsulación de los servicios y su encadenamiento para ofrecer funciones más complejas.

Los servicios se muestran como componentes –definidos mediante una interfaz que puede implementarse de múltiples formas– para la articulación de soluciones empresariales, pudiendo orquestarse en función de tres roles: el consumidor o cliente, el proveedor y el intermediario o *broker*. Por un lado, los servicios se ponen a disposición de los clientes sin necesidad de que éstos conozcan sus detalles. El cliente, a su vez, puede utilizar diferentes servicios y ofertarlos como uno nuevo, convirtiéndose a su vez en proveedor. El intermediario, por su parte, realiza la labor de facilitar a los usuarios la localización del servicio más adecuado a sus necesidades.

En este modelo juega un rol vital que la interfaz del servicio esté definida de forma correcta. Ésta permite que se firme un contrato entre el cliente y el proveedor, de ahí su importancia. La interfaz deberá recoger aspectos como la información para llamar al servicio, atributos de interés, coste, etc. Según lo expuesto, la Figura 3.1 muestra gráficamente la relación existente entre clientes, proveedores y servicios.

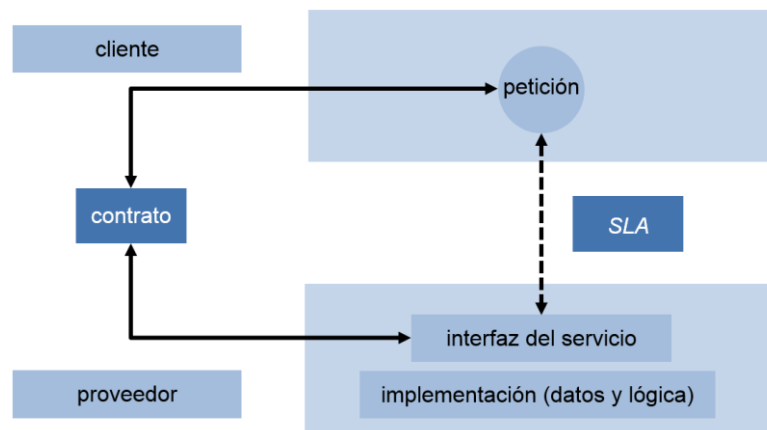


Figura 3.1 Relación entre servicios, clientes y proveedores [Gar08].

3.1.2 Computación orientada a servicios

Todo lo expuesto anteriormente ha llevado a la definición de la SOC como un nuevo paradigma que utiliza los servicios como elementos que soportan el desarrollo de aplicaciones distribuidas. Los servicios son elementos auto-descriptivos, aspecto que permite su combinación y ésta mueve a realizar las siguientes consideraciones [Gar08]:

- La composición debe realizarse de forma dinámica durante la ejecución de las aplicaciones. Dicho de otra forma, son los propios servicios los encargados de buscar e incorporar otros servicios que sean necesarios.

- El desarrollo de aplicaciones puede ser rápido y con un coste limitado. Este hecho, sin embargo, plantea la necesidad de describir la calidad de los servicios para generar la confianza necesaria a la hora de su utilización.

Desde el punto de vista de los protocolos técnicos se puede clasificar las tipologías de servicio en:

- **Servicios Web:** sistema que soporta la interacción entre máquinas en una red. Para ello se describe la interfaz mediante algún lenguaje, siendo el más utilizado el Lenguaje de Descripción de Servicios Web (*WSDL*). Por su parte, la interacción dentro del sistema se realiza mediante mensajes definidos, preferiblemente, mediante algún estándar como el Protocolo de Acceso a Objetos Simples (*SOAP*).
- **Servicios *grid*:** sistema distribuido y paralelo que permite compartir, seleccionar y agregar recursos distribuidos geográficamente de forma dinámica y en tiempo de ejecución. La agregación estudia aspectos como la disponibilidad, capacidad, rendimiento coste o los requisitos de calidad.
- **Servicios *P2P*:** servicios ofrecidos por pares de nodos en una red entre pares o iguales (*P2P*). La interacción entre las partes puede realizarse mediante terceros.

En cambio, desde el punto de vista del uso funcional, se podrían clasificar en:

- **Servicios de información:** aquéllos que personalizan la información de acuerdo con un conjunto de requisitos aportados por los usuarios.
- **Servicios inteligentes y adaptados a un contexto:** se suministran teniendo en consideración las características del lugar en el que son prestados. Son componentes de la conocida como inteligencia ambiental.
- **Servicios que funcionan como intermediarios:** facilitan el trabajo con otros servicios, facilitando su localización y utilización.

3.1.3 Plataforma arquitectónica

El funcionamiento de la SOC necesita de una infraestructura básica conocida como SOA, la cual utiliza un enfoque en el que los recursos software –servicios– están disponibles en una red y en el que se toma como referencia una arquitectura de computación distribuida basada en protocolos estándar con un acoplamiento ligero. Esta utilización de estándares facilita la interoperabilidad, aspecto vital de una SOA.

Desde un punto de vista conceptual, SOA se caracteriza por atender los siguientes principios [Bia11]:

- **Estandarización:** es quizás el aspecto que más ha favorecido la amplia adopción de esta arquitectura en las soluciones empresariales. SOA hace un uso intensivo de diferentes protocolos que permiten estandarizar las diferentes capas que componen esta arquitectura. Este hecho favorece la interoperabilidad entre distintos componentes que han de coexistir en cada una de las capas, con la finalidad de lograr los objetivos de negocio planteados para el sistema.

- Bajo acoplamiento: esta característica facilita la reutilización y el mantenimiento de los diferentes componentes del sistema. De esta forma, un consumidor de un servicio no tiene por qué preocuparse de la tecnología concreta utilizada en el desarrollo de un servicio, para poder utilizarlo. Los cambios en los componentes que permiten la publicación no tienen que afectar a los clientes, siempre que se respeten las interfaces de comunicación.
- Reutilización: SOA persigue que los servicios publicados en una determinada implementación realicen funciones autocontenidas, las cuáles puedan ser combinadas para crear servicios de grano más grueso –mayor nivel– que permitan dar respuesta a procesos de negocio más complejos.
- Composición: la capacidad de adaptación a nuevas circunstancias en el entorno del sistema o en el propio sistema, por mor del mantenimiento correctivo y/o adaptativo del mismo, es fundamental. El objetivo final no es otro que el de poder cambiar piezas de un determinado proceso de negocio sin que ello ocasione un impacto en los consumidores del servicio compuesto que implementa el proceso en cuestión.
- Descubrimiento: en aras a favorecer la utilización de los servicios, es preciso que existan mecanismos de publicación de los mismos que faciliten su descubrimiento. Esta labor de descubrimiento suele realizarse durante la fase de diseño de un nuevo servicio compuesto, aunque la definición e implementación de mecanismos que faciliten el descubrimiento de forma dinámica, en tiempo de ejecución, es uno de los campos de investigación más activos dentro de SOA.

De cara a poder cumplir con los principios expuestos, la comunidad ha desarrollado un conjunto de patrones de diseño o conjunto de buenas prácticas basadas en la experiencia acumulada por los arquitectos, cuya observancia sigue de guía en el proceso de diseño e implementación de nuevos sistemas. En [Erl08] y [SOA14] se recogen una amplia colección de los mismos, entre los cuales se encuentran patrones apropiados para cada uno de los componentes comunes a un sistema basado en SOA: el bus de servicios, el servicio de registro y el repositorio de servicios, el sistema de intercambio de mensajes, el motor de procesos de negocio y las herramientas de monitorización y administración del sistema.

El marco básico de funcionamiento de la SOC –véase Figura 3.2– se basa en la presencia de tres módulos denominados cliente, proveedor y registro; y en tres fases denominadas publicar y suscribir, encontrar y enlazar [Gar08].

El proveedor tiene un servicio que puede ser útil a un cliente potencial y para ello lo hace público a través del registro. El cliente tiene una necesidad que puede ser cubierta por un servicio, para lo cual realiza una consulta al registro en la que especifica una serie de filtros. El motor que gestiona el registro se encarga de buscar en el repositorio aquellos servicios que cumplen con las características indicadas y los muestra al cliente. Si el cliente encuentra entre ellos un servicio que satisface sus necesidades, procede a suscribirse al mismo, de forma que proveedor y cliente quedan enlazados. A partir de ese momento, el cliente podrá conectarse a la plataforma del proveedor del servicio para su utilización.

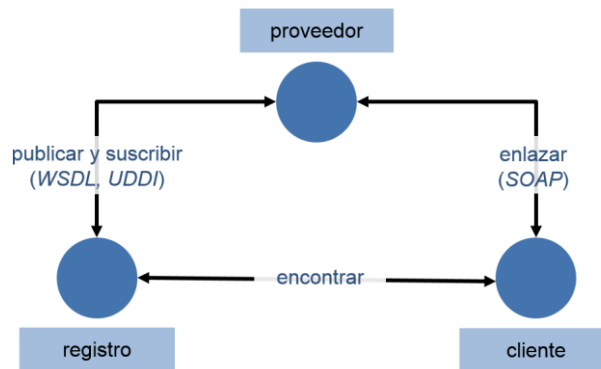


Figura 3.2 Marco básico de funcionamiento de la SOC [Gar08].

El marco descrito puede implementarse mediante una arquitectura organizada según un modelo de capas –véase diagrama izquierdo de la Figura 3.3–, en cuya base se encuentra la de transporte. Sobre ella, se dispone otra capa que gestiona los protocolos de comunicación de mensajes, gracias a los cuales se realiza la interacción entre los servicios. La siguiente capa contiene un proxy del cliente y un resguardo del servidor; por encima de la cual se introducen los lenguajes que permiten describir los servicios. Por último, en la cima, se halla el cliente y el servidor.

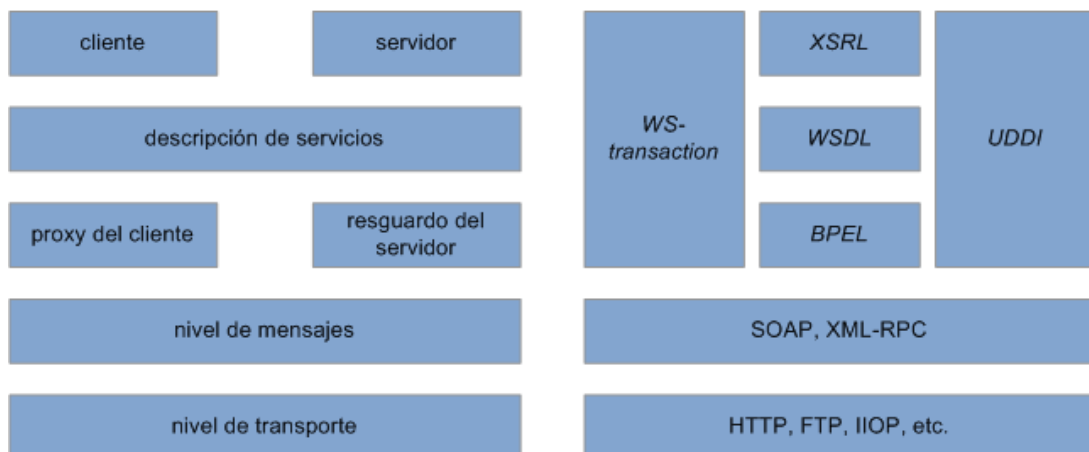


Figura 3.3 Modelo básico en SOC y sus lenguajes y estándares básicos [Gar08].

Cada una de las capas expuestas dispone de estándares que las implementan, tal y como se muestra en el diagrama derecho de la Figura 3.3. En el caso de la capa de transporte prevalecen los básicos de Internet –*HTTP* y *FTP*– así como *IIOP* –asociado a *CORBA*–, aunque es factible la utilización de otros protocolos. Por su parte, en la capa de mensajes, *SOAP* es sin duda el protocolo por excelencia. Su utilización, conjuntamente con otros estándares desarrollados por la *Web Services Interoperability Organization (WSIO)* –actualmente englobada dentro de la organización *OASIS*–, comúnmente referidos como pila *WS-I*, confiere a *SOA* una importante capacidad expresiva que permite la integración de múltiples y heterogéneos sistemas en una arquitectura común, favoreciendo la reutilización de componentes ya existentes, así como la transparencia e independencia de los clientes con respecto a la tecnología de los mismos.

Por otro lado, en la capa de descripción de servicios es el estándar Descripción Universal, Descubrimiento e Integración (*UDDI*) el que predomina. Se trata, en sí mismo, de un conjunto de protocolos orientados a aportar un directorio público en el que se puedan registrar y consultar en línea los *WS* y otros procesos de negocio. El registro se hace en el Lenguaje de Marcado Extensible (*XML*) y consta de tres partes:

- Páginas blancas: dirección, contacto y otros identificadores.
- Páginas amarillas: categorización industrial basada en taxonomías.
- Páginas verdes: información técnica sobre los servicios.

Junto a *UDDI*, se presentan en esta capa un conjunto de lenguajes que facilitan la creación de lenguajes sobre la base de la composición de servicios, como es el caso del Lenguaje de Ejecución de Procesos de Negocio (*BPEL*), *WSDL* o el Lenguaje *XML* de Peticiones a Servicios (*XSRL*). Además, para articular mecanismos que garanticen la integridad de la información, la especificación *WS-Transaction* define dos tipos de coordinación: atómica –transacciones con una única operación– y actividad de negocio –transacciones largas–.

3.1.4 Capas de la arquitectura

Tal y como ya se ha apuntado anteriormente, la implementación de una plataforma de SOC implica el desarrollo de una arquitectura que permita organizar las aplicaciones software y su infraestructura en forma de un conjunto de servicios que interactúan entre sí. Esta arquitectura debe partir conceptualmente del marco mostrado en la Figura 3.3 –diagrama izquierdo–, realizándose una descomposición que, según diferentes fuentes, puede mostrar una mayor o menor granularidad. En este caso se ha optado por referir la descomposición promulgada en el estándar técnico arquitectónico de referencia para SOA que ha sido desarrollado por el *Open Group* –véase Figura 3.4–, consorcio global que agrupa a diferentes organizaciones comprometidas con el desarrollo de estándares, buenas prácticas y tecnologías de código libre dentro del ámbito de las TI, según el cual se diferencia entre [TOG09]:

1. Capa de infraestructura –*operational systems layer*–: aporta elementos que permiten la gestión de datos y procesos, a los cuáles es posible acceder mediante servicios de la capa superior y que son imprescindibles para su ejecución.
2. Capa de componentes de los servicios –*service component layer*–: incluye los componentes software que son requeridos para la implementación de los distintos servicios que forman parte de la capa superior, tanto en términos de funcionalidad como de calidad del servicio (QoS), así como los componentes técnicos y funcionales que facilitan dicha implementación. Generalmente, un componente incluye también lógica específica de negocio, sin hacer referencia a la de integración.
3. Capa de servicios –*service layer*–: engloba al conjunto de servicios que provee SOA. Su rol principal es modelar dichos servicios, incluyendo su descripción. Por lo que respecta a esta última, provee a los clientes detalles suficientes para localizar e invocar las funciones de negocio

expuestas por el proveedor del servicio, indicando también los requisitos contractuales y las políticas de uso que permiten formalizar la relación entre ambos.

La materialización de cada uno de los servicios se realiza mediante los componentes definidos en la capa inferior, pero la instanciación propiamente dicha se realiza mediante los bloques de construcción arquitectónicos que residen en la capa de infraestructura.

4. Capa de procesos de negocio –*business process layer*–: se encarga de la externalización de los flujos propios de los procesos de negocio. Estos flujos se definirán mediante servicios concretos – presentes en la capa inferior– que pueden ser concatenados formando secuencias de procesos que den respuesta a los citados procesos; por tanto, esta capa aporta mecanismos que permiten la composición de servicios, mediante el descubrimiento y registro, la medición y negociación.
5. Capa de consumidores de servicio –*consumer layer*–: representa a los clientes del servicio, que a su vez pueden convertirse en suministradores de nuevos servicios por agregación de otros de grano más fino.

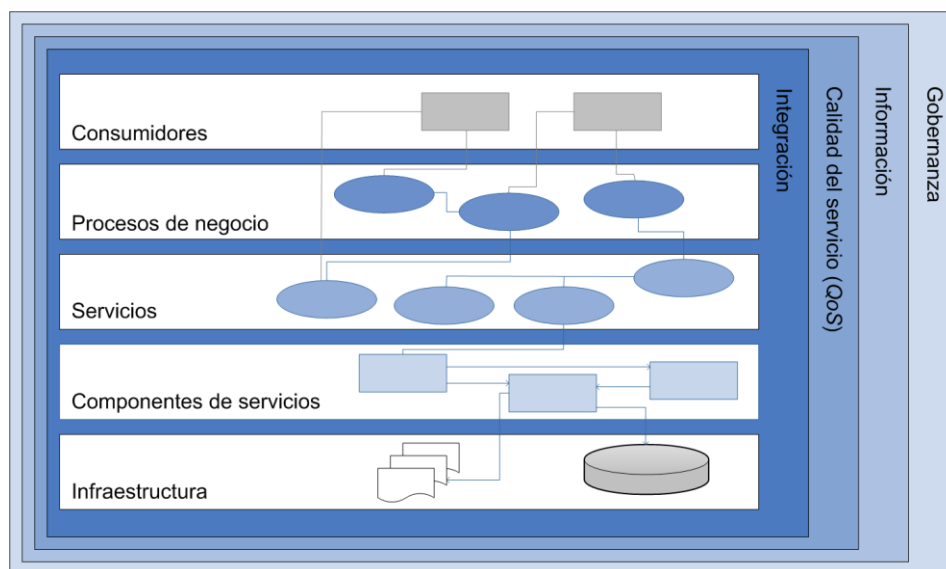


Figura 3.4 Descomposición en capas de SOA [fuente: propia].

6. Capa de integración –*integration layer*–: engloba el conjunto de características funcionales imprescindibles dentro de la arquitectura, puesto que permite la mediación entre los consumidores y los proveedores de un servicio. Funcionalmente, por tanto, permite la transformación, direccionamiento y conversión entre potenciales distintos protocolos que son necesarios para realizar ese trabajo de mediación –véase Figura 3.5–.
7. Capa de calidad del servicio –*QoS layer*–: provee mecanismos de gestión que permiten la monitorización de aspectos relacionados con la QoS como la disponibilidad, la fiabilidad o la seguridad, con la colaboración de la capa de integración. De esta forma, se garantiza el

cumplimiento de políticas de uso, contratos, requerimientos funcionales y regímenes de gobernanza que hayan sido suscritos para un determinado servicio.

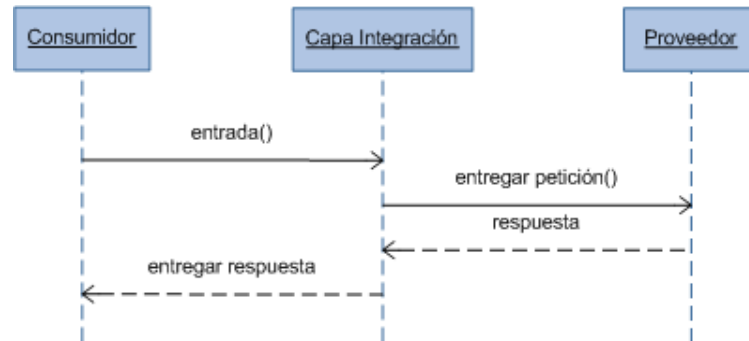


Figura 3.5 Diagrama UML de secuencia con la mediación de la capa de integración [fuente: propia].

8. Capa de información –*information layer*–: se responsabiliza de manifestar una representación unificada de cualquier información relacionada con los servicios, aplicaciones y sistemas que dan cobertura a los procesos de negocio cubiertos en la arquitectura; todo ello utilizando el lenguaje propio de la lógica de negocio de la organización que actúa como proveedora.
9. Capa de gobernanza –*governance layer*–: engloba el conjunto de herramientas que son precisas para atender todos los aspectos propios de la gobernanza SOA, asegurando la correspondencia entre los parámetros que caracterizan el funcionamiento de los servicios a través de todo su ciclo de vida y los compromisos establecidos con los consumidores –Acuerdo de Nivel de Servicio (SLA)–. Para ello, esta capa interactúa con todas las anteriores.

3.2 Arquitecturas Orientadas a Recursos

La irrupción en el ámbito de los sistemas distribuidos de las tecnologías relacionadas con Internet, ha provocado que en las últimas décadas una parte muy importante de los sistemas que se están desarrollando se articulen mediante colecciones de WS. Entendiendo la Web como un sistema distribuido para la compartición de contenidos hipermedia –aquellos que integran soportes tales como texto, imagen, video, audio, etc.– a través de Internet, han sido varias las aproximaciones arquitectónicas que se han planteado. Entre ellas la adaptación de SOA ha sido la que ha acaparado tradicionalmente más importancia, pero en los últimos años ha ido ganando adeptos una aproximación alternativa que basa su éxito en que comparte los principios nucleares de la propia Web: ROA.

3.2.1 Fundamentos de REST

De acuerdo con lo recogido en el artículo de Fielding R.T. y Taylor, R.N. [ACM02], la solución arquitectónica que permitió definir el modelo de interacciones necesario para el funcionamiento de cualquier aplicación Web se denominó Transferencia de Estado Representacional (REST). Sin embargo, Richardson y Ruby [Ric07] no consideran que REST sea propiamente una arquitectura, sino un conjunto de criterios de diseño. Más allá de esta divergencia, sí que existe una coincidencia entre los postulados

de los distintos autores, puesto que estos criterios se manifiestan en forma de restricciones arquitectónicas que tienen como objetivo minimizar el tiempo de latencia y la transferencia de información necesaria, al mismo tiempo que se incrementa la independencia y la escalabilidad de las implementaciones de los componentes. Las restricciones en cuestión son:

- División cliente-servidor: un servicio debe exponer un conjunto de métodos mientras que el agente usuario envía peticiones de acceso al servicio. Esta restricción aboga por la separación de responsabilidades, mientras que las capacidades funcionales se reservan a la vertiente servidora, la interfaz de usuario queda restringida al cliente. De esta forma, ambos componentes pueden evolucionar de forma independiente.
- Interacciones sin estado: en la vertiente servidora no se almacena ningún tipo de información sobre la sesión o el estado de la aplicación. Cada petición efectuada por un agente usuario debe contener toda la información necesaria para que el servicio entienda el objetivo de la petición sin tener que recurrir a ningún tipo de información de contexto almacenada en el servidor. En *REST*, los agentes usuario son los responsables de mantener el estado de la aplicación.
- Existencia de caché: permite reducir el ancho de banda de red necesario, puesto que se eliminan parte de las interacciones entre agentes y servidores. Cuando las respuestas son almacenadas en caché, los agentes pueden reutilizarlas si se realizan peticiones equivalentes.
- Interfaz uniforme: los recursos en la vertiente servidora exponen una interfaz genérica derivada de la semántica de los métodos del protocolo *HTTP*.
- Estructuración en capas: la funcionalidad se descompone jerárquicamente en diferentes capas lógicas. Cada capa es independiente e interactúa únicamente con las que se encuentran inmediatamente encima o debajo. Esta restricción minimiza la complejidad del sistema en su conjunto, mejora la escalabilidad –mediante la incorporación de componentes que facilitan el balanceo de la carga o proveen mecanismos de caché, etc.– y permite mejorar los mecanismos de seguridad.
- Código a demanda –opcional–: la funcionalidad del sistema puede extenderse mediante la descarga y ejecución de código en la vertiente cliente. Desde este punto de vista, las respuestas pueden ser incompletas y venir acompañadas de código ejecutable [Fre12].

Aunque cada una de estas restricciones puede considerarse de forma aislada y ha caracterizado algún otro estilo arquitectónico, la combinación de todas ellas es la que permite lograr el objetivo de la Web. Esta combinación se puede representar visualmente tal y como se muestra en el gráfico presentado en la Figura 3.6, en el cual se recogen las siguientes restricciones arquitectónicas:

- Repositorio replicado –*replicated repository*– (RR): sistemas basados en el estilo arquitectónico que realiza una replicación de los repositorios para mejorar la accesibilidad a los datos y la escalabilidad de los servicios, teniendo más de un proceso como proveedor de un servicio.

- Almacenable en memoria –*cacheable*– (\$): variante del estilo RR en la que se replican los resultados de peticiones anteriores para poder ser reutilizados en posteriores peticiones.

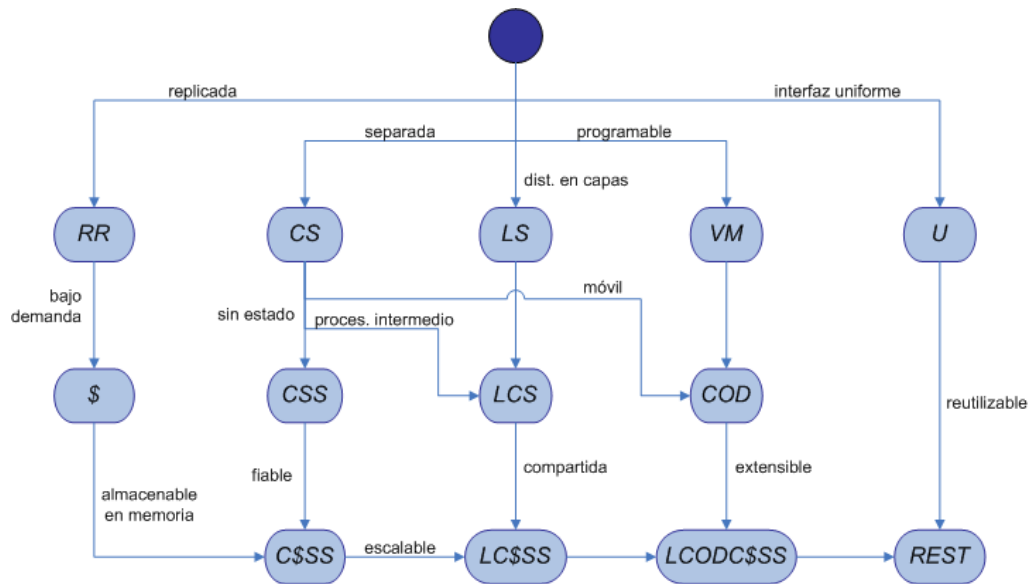


Figura 3.6 Derivación de REST a partir de las restricciones arquitectónicas [ACM02].

- Almacenable en memoria –*cacheable*– (\$): variante del estilo RR en la que se replican los resultados de peticiones anteriores para poder ser reutilizados en posteriores peticiones.
- Cliente-servidor –*client-server*– (CS): estilo que realiza una descomposición entre el componente cliente que se encarga de realizar las peticiones y el servidor que se encarga de atenderlas.
- Cliente-servidor sin estado –*client-stateless server*– (CSS): variante del estilo CS en la que no se realiza gestión alguna del estado de la sesión en el servidor.
- Cliente-servidor sin estado y con almacenamiento en memoria –*client-cache stateless server*– (C\$SS): estilo que combina las características de \$ y CSS.
- Sistema distribuido en capas –*layered system*– (LS): estilo que realiza una descomposición del sistema en diferentes capas, cada una de las cuales presta servicios a la inmediatamente superior y utiliza servicios de la inferior.
- Cliente-servidor distribuido en capas –*layered client-server*– (LCS): estilo resultante de la combinación de CS y LS.
- Cliente-servidor distribuido en capas, sin estado y con almacenamiento en memoria –*layered client-cache stateless server*– (LC\$SS): estilo que combina las características de LCS y C\$SS.
- Máquina virtual –*virtual machine*– (VM): estilo de codificación móvil en el cual el código se ejecuta en un entorno controlado, incrementando los niveles de seguridad y fiabilidad.

- Código bajo demanda –*code on demand*– (*COD*): estilo en el que un componente cliente tiene acceso a un conjunto de recursos, pero no a la forma en la que los mismos son procesados. Éste realiza una petición al servidor para obtener el fragmento de código que realiza un determinado proceso y ejecutarlo localmente.
- Sistema distribuido por capas con cliente que ejecuta código bajo demanda, servidor sin estado y con almacenamiento en memoria –*layered code-on-demand client-cache stateless server*– (*LCODC\$SS*): estilo que combina las características arquitectónicas de *COD* y *LC\$SS*.
- Interfaz uniforme –*uniform interface*– (*U*): estilo basado en la definición de una interfaz uniforme para exponer todas las funciones que provee el sistema.

De todas las restricciones reseñadas, la que ha cobrado una mayor repercusión ha sido la correspondiente a la existencia de una interfaz uniforme [Cap09]. Los conceptos utilizados en la definición de interfaces siguiendo el estilo *REST* son:

- Recurso –*resource*–: cualquier información que pueda ser identificada, como por ejemplo un documento, una imagen o una lista de incidencias. Según Fielding [Fie00], un recurso se describe formalmente como una “función $M_r(t)$ de pertenencia temporal variable en la que para un instante t se realiza una correspondencia con un conjunto de entidades o valores que son equivalentes. Los valores del conjunto pueden corresponderse con representaciones y/o identificadores de los recursos”.
- Una representación –*representation*–: secuencia de bytes, junto con unos metadatos que describen los mismos [Fie00]. De forma específica, un cliente recibe una representación cuando realiza una solicitud de un recurso, o bien envía una cuando quiere proceder a actualizar un recurso.
- Hipermedia como motor del estado de la aplicación –*HATEOS*–: esto implica que el cliente se responsabiliza del mantenimiento del estado de la aplicación, así como de la transición entre los posibles estados. La representación de los recursos debe incluir contenidos hipermedia, es decir, enlaces y formularios que permitan transitar entre el estado actual y los próximos.
- Mensaje autodescriptivos –*self-descriptiveness messages*–: al contrario que el estado de la aplicación que debe ser gestionado en el cliente, el estado de un recurso debe transmitirse en las interacciones que ocurren entre cliente y servidor, mediante las representaciones de los recursos. Los atributos, las transiciones válidas entre estados y los enlaces a otros recursos relacionados, forman parte de la representación del recurso y, por tanto, de su estado. Adicionalmente, las cabeceras *HTTP* también permiten la introducción de nuevos metadatos que ayudan a los clientes a interpretar el significado de la representación que figura en el cuerpo del mensaje *HTTP*. El hecho de que los mensajes sean autodescriptivos implica que la semántica de los mensajes intercambiados es totalmente visible e inteligible para cualquier componente que forme parte de la cadena de diálogo entre cliente y servidor.

3.2.2 Arquitectura

El estilo *REST* ha venido siendo implementado con éxito en la creación de *WS*, como alternativa a los sistemas convencionales que parten de la utilización de protocolos más complejos como *SOAP* y/o la pila de protocolos *WS*-* –*WS-Addressing*, *WS-ReliableMessaging*, *WS-Security*, etc.–. La caracterización de una arquitectura de sistemas que diera respuesta a las necesidades propias de sistemas distribuidos orientados a la publicación de *WS* siguiendo el estilo *REST* fue abordada por Richardson y Ruby [Ric07], dando lugar a la que se ha conocido como *ROA*.

Esta arquitectura observa de forma estricta las restricciones arquitectónicas de *REST* y realiza una asociación explícita entre las mismas y los protocolos Web, como el Identificador de Recurso Uniforme (*URI*), el Protocolo de Transferencia de Hipertexto (*HTTP*) y el Lenguaje de Marcado Extensible (*XML*).

Por otro lado, caracterizan *ROA* mediante la consideración de los tres conceptos nucleares enunciados en *REST* –recursos, representaciones y enlaces hipermedia– y el establecimiento de cuatro propiedades:

- **Direccionabilidad –*addressability*–**: desde el punto de vista del usuario, esta es la propiedad más importante de una aplicación o sitio Web y, por extensión, de un servicio. *ROA* asocia a cada recurso un nombre y una dirección caracterizada mediante un *URI*, permitiendo su identificación de forma unívoca, independientemente de su tipología y representación. Pero la utilización de los *URI* dentro de *ROA* no se circunscribe a lo que se acaba de recoger, distintos *URI* van a permitir caracterizar cada uno de los posibles estados de una aplicación, así como cada una de las representaciones de un recurso también vendrá caracterizada mediante un *URI* diferente.
- **Ausencia de estado –*statelessness*–**: implica que cada petición *HTTP* se realice de forma aislada, sin vinculación con ninguna otra petición anterior o posterior. Para ello, cada petición debe incluir toda la información necesaria para que el servidor pueda procesarla. Cualquier consideración que implique la necesidad de gestionar el estado de una determinada operación deberá ser gestionada en el cliente. Esta característica, además, tiene importantes repercusiones en cuanto a la escalabilidad de un servicio, puesto que éste puede ser instanciado en diferentes servidores en función de la demanda y no es preciso habilitar ningún mecanismo que gestione cuál de los servidores procesó peticiones previas de un mismo cliente.
- **Conectividad –*connectedness*–**: la presencia de enlaces a otros contenidos hipermedia dentro de la representación de un recurso facilita, en los casos en los que sea necesario, el proceso de mantenimiento del estado de la aplicación por parte del cliente, mediante el análisis de las *URI* que figuran en los citados enlaces. Independientemente, en cualquier caso estos enlaces permiten establecer relaciones entre los diferentes recursos.
- **Interfaz uniforme –*uniform interface*–**: *REST* no establece la utilización de una interfaz concreta, deja la puerta abierta a que las implementaciones que sigan sus principios realicen la elección. En el caso de *ROA* se opta por la utilización estricta de los métodos definidos en la interfaz del protocolo *HTTP*, tanto desde un punto de vista nominal como semántico. De esta forma se utiliza:
 - *GET* para la obtención de la representación de un recurso.

- *DELETE* para el borrado de un recurso.
- *POST* para la creación de un nuevo recurso.
- *PUT* para la actualización de un recurso existente o para la creación de uno nuevo. Se recomienda su utilización frente a *POST* cuando el cliente conoce el *URI* que tendrá el nuevo recurso, en caso contrario debe utilizarse el otro método.
- *HEAD* para la obtención de los metadatos sobre la representación de un recurso que se transmiten en la cabecera del protocolo *HTTP* mediante *HEAD* –el resultado es equivalente al de una petición *GET*, pero sin la obtención de la representación del recurso–.
- *OPTIONS* para obtener información sobre las opciones autorizadas para un cliente en las peticiones de acceso a la representación de un recurso o en la respuesta obtenida.

Pero, además de los métodos, *ROA* también saca partido de los códigos de respuesta utilizados en el protocolo *HTTP* para informar sobre aspectos semánticos de las operaciones.

La adopción de esta interfaz *HTTP* permite, además de facilitar el proceso de desarrollo de los *WS* y de las aplicaciones cliente consumidoras, garantizar aspectos como la seguridad y la idempotencia. La seguridad en las operaciones *GET* y *HEAD* viene dada por el hecho de que permiten acceder a información sobre un recurso –metadatos o una representación del mismo– sin que el cliente pueda realizar modificación alguna. La idempotencia, por su parte, proviene del concepto matemático que implica que la aplicación de una misma acción varias veces no altera el resultado obtenido; en este sentido, las operaciones *PUT* y *DELETE* gozan de esta propiedad. Estas dos características son muy importantes, especialmente en entornos en los que las peticiones *HTTP* se realizan en redes no fiables.

31

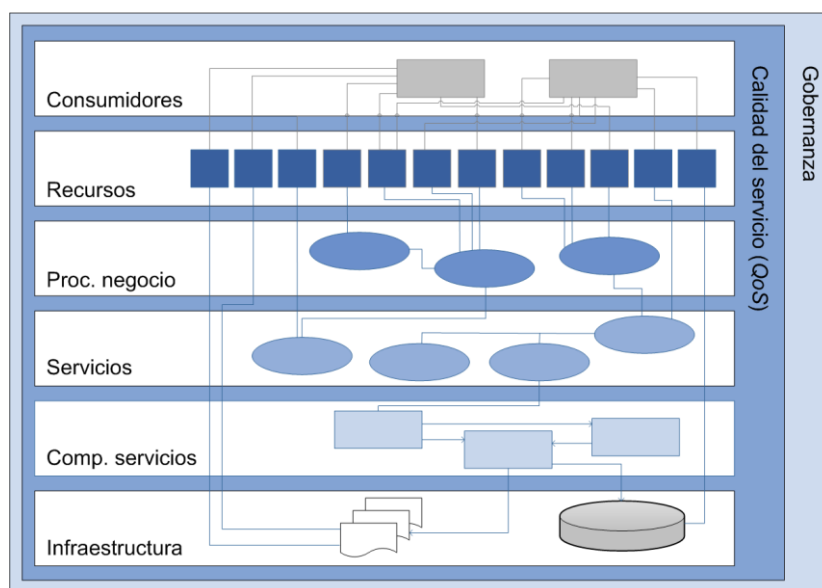


Figura 3.7 Descomposición en capas de ROA [fuente: propia].

Partiendo de estas características, la formulación de una representación basada en capas de la arquitectura *ROA* no tendría que diferir mucho del modelo presentado para *SOA*, con las siguientes salvedades –ver Figura 3.7–:

- Se introduce una capa de recursos, apareciendo cada uno de los mismos expuesto a los consumidores mediante una cobertura *WS*, la cual permite el acceso mediante la interfaz uniforme *HTTP*. De esta forma los servicios que se exponen no permitirían la invocación directa de procesos de negocio, sino el acceso a los diferentes recursos planteados en el dominio de negocio.
- Los recursos pueden ser provistos por parte de los servicios complejos de la capa de procesos de negocio, por los servicios simples de la capa de servicios, o incluso directamente apuntando a los recursos de partida que se encuentran en la capa de infraestructura –ficheros en disco y registros almacenados en tablas de bases de datos–.
- Supresión de las capas de integración e información, dada la simplicidad y uniformidad de la interfaz *REST*. En el primer caso puesto que no es necesaria la labor de esa capa de mediación entre consumidores y proveedores, puesto que se utiliza exclusivamente un único protocolo de comunicación y las aplicaciones cliente deben circunscribirse al mismo, de lo contrario las ventajas de *REST* desaparecen. En cuanto al segundo, la utilización de una interfaz sencilla, con un conjunto cerrado y conocido de operaciones, junto con la característica de autodescripción propia de *REST*, hace que ya se muestre una representación unificada de cualquier información relacionada con los servicios, las aplicaciones o los sistemas que participan en la publicación de los servicios.

Por otra parte, se ha decidido respetar las capas restantes: *QoS* y gobernanza, puesto que sus cometidos son perfectamente aplicables a una estructura basada en *WS* de acceso a recursos. Dicho esto, es necesario destacar que las restricciones propias de la interfaz *REST*, especialmente notorias en temas de seguridad –provocadas por las limitaciones de los protocolos *HTTP* y *HTTPS*, tal y como se puede ver en 4.4.8– y de *QoS* –causadas fundamentalmente por la ausencia de estado–, hacen que el grado de gestión que permiten sea inferior al existente en *SOA*.

3.3 Ontologías y Web semántica

La *SOC* y la arquitectura *SOA* sientan las bases conceptuales y arquitectónicas para la creación de sistemas que ofrezcan a los integradores de soluciones informáticas un marco de integración funcional a partir del cual poder dar respuesta a las necesidades de los clientes.

Abundando en el concepto de servicio, en el ámbito de los sistemas distribuidos cobra notoriedad una de sus especializaciones, la que permite su utilización a través de la Web. En sentido estricto, la implementación de *WS* debe seguir un conjunto de estándares de protocolos Web, entre los que cabe reseñar los siguientes:

- *WSDL*: gramática *XML* para la descripción de servicios de red en forma de puntos de acceso, a través de los que se opera mediante mensajes que contienen información sobre documentos o

procesos [WWW01]. Las descripciones que facilita este estándar son claves en el proceso de composición de servicios.

- *UDDI*: catálogo de registro de un conjunto de servicios que permite la descripción –incluida sus interfaces técnicas y las formas de acceso– y el descubrimiento de los mismos, aunque estos hayan sido creados por diferentes empresas, organizaciones u otros proveedores de *WS* [OAS04]. Este catálogo, junto con algunos otros frutos de otras aproximaciones tecnológicas, como la realizada por *Sun Microsystems* con *Jini*, actúan como intermediarios –*brokers*– en el proceso de descubrimiento de servicios.
- *SOAP*: protocolo ligero para el intercambio de información estructurada en un entorno descentralizado y distribuido [WWW07].

La creciente presencia de servicios *REST* en la Web, también ha puesto de manifiesto la necesidad de arbitrar mecanismos en el ámbito de las *ROA* para facilitar la definición y el descubrimiento de estos. Dentro de la “comunidad” *REST*, *Sun Microsystems* optó por la definición de un lenguaje específico para la definición de servicios, denominado Lenguaje Descriptivo de Aplicaciones Web (*WADL*), el cual fue propuesto en 2009 como estándar al *World Wide Consortium (W3C)*. Este organismo, por su parte, ha publicado una segunda versión del estándar *WSDL*, 2.0, que sí da cobertura a la descripción de servicios *REST*, facilitando el acceso a los catálogos de registro de esta tipología de servicios [Man08].

Los avances experimentados en la regulación e implementación de los *WS* han provocado la aparición de un gran número de servicios que intenta dar respuesta a la creciente demanda de información y de procesos de negocio. Sin embargo, el acceso a los servicios ofertados por parte de los demandantes implica dar respuesta a nuevos retos tecnológicos. Por un lado, no es razonable someter al usuario a un proceso de búsqueda y filtrado de listados de servicios ofrecidos por diferentes catálogos de proveedores. Por otro, las necesidades de los usuarios conllevan, cada vez más, la realización de tareas más complejas, las cuales contrastan con la sencillez propia de la mayoría de los servicios, caracterizados éstos por un grado importante de atomicidad; hecho que implica una necesaria combinación de los mismos.

Todo ello conduce a un escenario en la que la eficiencia en los mecanismos de descubrimiento, selección y composición de servicios adquiere una mayor relevancia. La respuesta a estas necesidades de automatización ha encontrado un aliado en las tecnologías propias de la Web semántica [Sta11], las cuales han dado respuesta a la necesidad de organizar la basta infraestructura que forma actualmente la Web.

El ingente conjunto de información que se publica a través de páginas Web, distribuidas por todo el mundo de forma desorganizada, es accedido normalmente de forma manual, mediante la ejecución de búsquedas mediante palabras clave. Para facilitar ese proceso de búsqueda e intercambio de información, de forma que pueda realizarse estrictamente entre máquinas –servidores Web y sistemas de información– en el ámbito de la Web, la Web semántica propone que se incorpore información en la Web que permita describir los contenidos de ésta de manera formal. Esta descripción formal, denominada ontología, es un lenguaje que se ocupa de describir los conceptos y sus atributos, así como las relaciones existentes en los contenidos de la Web. Sin embargo, no existe un único lenguaje para la descripción de ontologías. Aun teniendo todos estos lenguajes un mismo propósito, cada uno ofrece un

vocabulario diferente, diferentes términos y conceptos que permiten una mayor o menor flexibilidad a la hora de recoger un dominio de negocio, o incluso que estas descripciones se realicen de forma más o menos sencilla. A título meramente ilustrativo, algunos de los lenguajes más utilizados son el Marco de Descripción de Recursos (*RDF*), Lenguaje Web de Ontologías (*OWL*) o *XML*.

Llegados a este punto, dentro del ámbito de la computación ubicua, el reto consistió en aplicar los principios de la Web semántica también a la descripción de servicios. Hasta ese momento, las descripciones se realizaban utilizando parámetros estrictamente sintácticos, como por ejemplo los recogidos mediante el lenguaje *WSDL*, que ha sido el más ampliamente utilizado. *WSDL*, al igual que otras aproximaciones, carece de recursos que permitan describir aquellos aspectos de los *WS* que van más allá de su interfaz o de la implementación concreta que utiliza; los aspectos semánticos, que principalmente permitirán describir el propio servicio y las condiciones que se tienen que dar para que el servicio pueda ser prestado, quedan fuera de su alcance.

Para cubrir las deficiencias mostradas por los lenguajes habituales de descripción de servicios, se desarrollaron nuevos lenguajes estándares que permiten realizar anotaciones semánticas en las descripciones de los servicios –sirva a título ilustrativo la propuesta de anotaciones semánticas para *REST* realizada por el *W3C*, denominada *SA-REST* [Gom10]–, ayudando a que éstos puedan ser interpretados por componentes informáticos que favorezcan los procesos de descubrimiento y selección, incluso mediante algoritmos que apliquen razonamientos lógicos.

Abundando en lo reseñado anteriormente, la aproximación semántica ha de permitir la descripción de las operaciones, entradas, salidas e incluso las precondiciones de los servicios con respecto a una ontología. En este sentido, la iniciativa que ha cobrado mayor relevancia ha sido el desarrollo del Lenguaje Web Semántico de Ontologías (*OWL-S*), derivado de una aproximación anterior denominada *DAML-S*. Este estándar habilita la automatización de las siguientes tareas relacionadas con los *WS* [WWW04]:

- Descubrimiento: tarea orientada a la localización de servicios que puedan proveer una funcionalidad determinada, atendiendo a unas posibles restricciones fijadas por el cliente. Mediante *OWL-S* la información necesaria para el descubrimiento del servicio se especifica de forma que es directamente interpretable por los servidores Web y un servicio de registro o un motor de búsquedas capacitado para el uso de ontologías puede localizarlos.
- Invocación: tarea orientada a la ejecución de servicios especificando simplemente una descripción declarativa de los mismos. El lenguaje *OWL-S* provee una Interfaz de Programación de Aplicaciones (*API*) que incluye la semántica de los argumentos que es preciso utilizar cuando se invocan los servicios, así como la propia de los mensajes que se emiten cuando el servicio ejecuta la operación con éxito o cuando falla.
- Composición: tarea que implica la selección, composición y interoperación de los *WS* para la consecución de tareas complejas. *OWL-S* provee especificaciones declarativas de los prerequisites y consecuencias de la aplicación de cada uno de los servicios atómicos que participan en la composición, así como un lenguaje para describir las composiciones y las interacciones de los flujos de datos.

Para la consecución de estas tareas, este estándar se articula entorno a tres clases principales [Sta11] – ver Figura 3.8–:

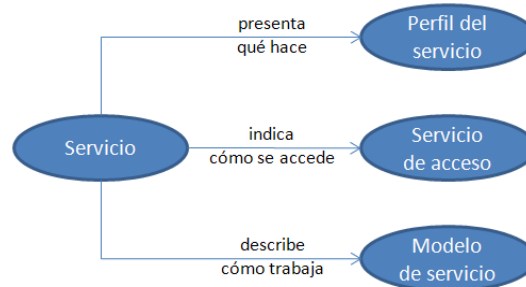


Figura 3.8 Estructura del primer nivel de OWL-S [fuente: propia].

- Perfil del servicio –*service profile*–: descripción, con mayor o menor grado de abstracción, de la función que realiza un servicio, así como de sus entradas, salidas, precondiciones y resultados o efectos. Esta cuaterna de datos se conoce como *IOPR* (*Inputs, Outputs, Preconditions and Results*) o *IOPE* (*Inputs, Outputs, Preconditions and Effects*), respectivamente.
- Modelo de procesos del servicio –*service process model*–: descripción del flujo de trabajo propio de la operativa del servicio, de forma que los clientes del mismo puedan analizarlo, monitorizar su ejecución, así como controlar su coordinación con otros servicios. Permite, además, describir la composición de servicios.
- Servicio de acceso –*grounding service*–: instancia del servicio que ofrece una fórmula concreta para su ejecución, similar a los enlaces de servicio de *WSDL* –*bindings*–. Es importante destacar que la especificación completa del mecanismo de acceso recogido en *OWL-S* indica de forma explícita que ambos lenguajes –el propio *OWL-S* y *WSDL*– son requeridos [Bar10].

35

De esta forma, con esta estructura principal, este estándar permite dar respuesta a las preguntas que se pueden observar en la figura anterior, aunque con algunas restricciones. Entre ellas puede citarse que *OWL-S* descarga en el proveedor del servicio la definición de aspectos relacionados con la calidad del sistema (QoS), aspecto que ha sido recogido dentro de otra iniciativa denominada Lenguaje Web de Ontologías para la Calidad del Servicio (*OWL-Q*).

3.4 Composición de servicios

Dejando a un lado los procesos de descubrimiento e invocación, es importante reseñar que los de composición son los que implican un mayor grado de complejidad.

Teniendo clara la necesidad de poder componer servicios e imaginando un ecosistema formado por un conjunto cerrado de servicios publicados por diferentes servidores, con interfaces publicadas mediante un mismo estándar y registrados en catálogos de servicios, y con metadatos descriptivos no sujetos a interpretación –claros y concisos–, la solución a la composición de servicios sería tan sencilla como encontrar la cadena más adecuada de procesos, cuyas entradas y salidas sean compatibles, que permita

resolver la problemática planteada. Se tardaría más o menos tiempo, pero una vez localizada la secuencia óptima, quedaría definida de forma permanente.

Sin embargo, la realidad es mucho más compleja. Pueden existir múltiples servicios que implementen la lógica necesaria para resolver una determinada tarea y es preciso seleccionar uno de ellos, puede darse el caso de que un servicio que forma parte de una composición deje de estar disponible y haya que buscar alguna alternativa, pueden publicarse con descripciones poco concisas o que puedan prestarse a interpretación; en definitiva, es recomendable disponer de mecanismos que automaticen el proceso de composición –aspecto en el que resulta vital la componente semántica– y que tutelen en todo momento la ejecución del flujo definido para resolver posibles incidencias con el mayor grado de autonomía posible.

Partiendo de esta situación, es interesante describir de forma concisa los requerimientos propios del proceso de composición [Bar10]:

- Automatización –*automation*–: el principal requerimiento es que la generación del esquema de composición se realice de la forma más automatizada posible, dado que de esta forma se minimiza –preferiblemente se elimina– la intervención del usuario. Este hecho permite reducir el tiempo necesario para realizar el proceso de composición, a la vez que se minimizan los errores humanos y, consecuentemente, se reduce el coste general del proceso.
- Dinamismo –*dynamicity*–: otro aspecto que permite realizar distinciones entre diferentes metodologías de composición es el grado de dinamismo que confieren al esquema de composición generado. Una aproximación estática implica que la selección de los servicios a componer se realice de forma previa al despliegue del servicio compuesto, por lo que no es posible su modificación mientras se esté ejecutando.

Por otro lado, una aproximación dinámica produce un esquema de composición abstracto que no se materializa hasta que se despliega el servicio. Durante el periodo de ejecución, los servicios reales se asocian a cada una de las tareas abstractas del esquema. De esta forma, se puede hacer frente a incidencias sobrevenidas en la disponibilidad de los servicios, tal y como se mencionaba al inicio de este apartado.

- Capacidades semánticas –*semantic capabilities*–: de conformidad con lo expuesto en el apartado 1.3, la descripción semántica de las capacidades de los servicios es fundamental en el proceso de composición. Las connotaciones semánticas mejoran la eficiencia del proceso de comprobación de la idoneidad de los servicios existentes a los requerimientos de los usuarios, yendo más allá de una “simple” comprobación de la adecuación de las entradas y salidas de los servicios.
- Conciencia de QoS –*QoS-Awareness*–: otro requerimiento fundamental es el que hace referencia a la observación de parámetros referentes a la calidad de los servicios durante el proceso de composición, entre otros. De esta forma, en dicho proceso no solo se tienen en consideración aspectos de carácter funcional, si no también otros relacionados con la calidad, como el tiempo de respuesta, el coste o la disponibilidad, entre otros.
- Indeterminismo –*indeterminism*–: la ausencia de esta característica viene provocada por la presencia de estructuras condicionales o bucles dentro del esquema de composición. Este hecho

puede provocar diferentes ejecuciones para distintas peticiones, aspectos que pueden incurrir en problemas que afecten a los requerimientos de QoS.

- Observación parcial –*partial observability*–: muchas de las técnicas de composición que aplican algoritmos propios del ámbito de la inteligencia artificial necesitan un conocimiento completo del estado inicial previo a la ejecución de un servicio compuesto. Sin embargo, un proceso de composición efectivo debe poder gestionar este tipo de escenarios en los que la información no es completa.
- Escalabilidad –*scalability*–: el hecho de que una técnica de composición funcione correctamente con un determinado conjunto de servicios no garantiza que lo haga con un conjunto diferente, mayor o más complejo. Por tanto, debe testarse la técnica aumentando el número de servicios y la complejidad de los mismos, para ver cómo se afecta el rendimiento.
- Corrección –*correctness*–: este requerimiento debe comprobarse cuando se quiera certificar que las propiedades de un servicio compuesto se cumplen, por ejemplo los resultados obtenidos a partir de una combinación de entrada y un conjunto de condiciones concretos.
- Independencia del dominio –*domain independence*–: una técnica de composición no debería restringirse a un dominio específico, debería poder aplicarse a diferentes entornos para solucionar un amplio espectro de problemas. Es, por tanto, un requerimiento deseable, aunque reñido en numerosas ocasiones con las connotaciones semánticas del servicio compuesto que se quiere obtener, puesto que estas suelen estar directamente vinculadas con un dominio específico.
- Adaptabilidad –*adaptativity*–: este último requerimiento surge de la necesidad de dar respuesta a una creciente corriente de investigación que busca que los servicios –atómicos o compuestos– sean capaces de adaptarse de forma autónoma a nuevos requerimientos. Como se puede entender, este requerimiento va más allá del dinamismo, puesto que este último no tiene que gestionar cambios en los requerimientos.

La complejidad que se deriva del proceso de composición ha concitado el interés de numerosos investigadores. Estas investigaciones han dado lugar a diversos sistemas que emplean tecnologías y metodologías que dan lugar a productos finales que difieren en numerosos aspectos, siendo muy necesaria una convergencia y estandarización de las mismas. En líneas generales, los modelos básicos de composición [Bar10] que dan a pie a la mayoría de las técnicas existentes, ya sea de forma exclusiva o mediante formulaciones híbridas, son los siguientes:

- Orquestación –*orchestration*–: descripción del comportamiento que manifiesta el proveedor de un servicio compuesto para lograr la implementación del mismo. En esta tipología, la lógica de control está centralizada en el proveedor y el comportamiento del mismo se describe indicando los distintos servicios que forman parte de la composición y el modo en que interactúan en relación a los mensajes intercambiados, incluyendo la lógica de negocio y el orden en el cual las interacciones deben ejecutarse; de hay el término de orquestación. La descripción de lo expuesto se realiza mediante lenguajes, entre los que cabe destacar *Web Services BPEL (WS-BPEL)* – derivado de *BPEL*–, el cual permite representar mediante *XML* la lógica de negocio, es decir, el

flujo de datos y de control de los procesos, recogiendo también los elementos partícipes –WS– de cada una de las actividades realizadas.

- Coreografía –*choreography*–: esta aproximación se basa en el principio de colaboración del conjunto de servicios que forma la composición, de manera que la lógica de control se encuentra distribuida entre todos ellos. Para ello, la descripción recoge únicamente el intercambio de mensajes, las reglas de interacción y los acuerdos que ocurren entre distintos componentes. Es, por tanto, una aproximación que no recoge la lógica de negocio de cada uno de los componentes, como pasa en la orquestación, sino que se ciñe a la interacción entre los mismos. El principal lenguaje para la definición de coreografías es el definido por el W3C, mediante una recomendación de estándar, denominado *Web Services Choreography Description Language (WS-CDL)*. Este lenguaje utiliza interacciones bilaterales de uno o dos mensajes que permiten modelar cualquier intercambio de forma incremental.
- Coordinación –*coordination*–: se orienta a la especificación de la agrupación temporal de servicios, definiendo el protocolo de coordinación necesario. Este protocolo indica la forma en la que interaccionan los servicios y también el resultado de esa interacción, resulte o no satisfactorio. Implica la existencia de un elemento adicional denominado coordinador que es el nexo de unión entre todos los servicios, canalizando él todas las interacciones de forma que los servicios no interactúan entre sí de forma directa. OASIS definió un marco de trabajo denominado *Web Services Coordination Framework (WS-CF)* pensado para este tipo de modelo de composición.
- Modelo de componentes –*component model*–: también conocido como “cableado de servicios” –*service wiring*–, describe las relaciones existentes entre las entradas y los servicios cuando estos forman parte de una composición. Es un modelo que puede utilizarse conjuntamente con el de orquestación, con este último describiendo el control y el flujo entre los servicios participantes, y el primero la conexión entre las entradas y salidas de los servicios.

4 Arquitectura para un SIG distribuido

Las ventajas de la aplicación de arquitecturas distribuidas al ámbito de los SIG han sido puestas de manifiesto en diferentes investigaciones, aunque siempre partiendo del paradigma arquitectónico SOA. Sirvan como muestra los trabajos publicados en [Tal08] y [Gum08] –los cuales abogan por la utilización de SOA como modelo arquitectónico para la publicación de WS que faciliten el desarrollo de aplicaciones SIG que permitan acceder y compartir información geoespacial a través de Internet, así como la ejecución de diferentes tareas de geoprocesamiento–, [Guo10] –el cual muestra cómo la utilización de SOA permite dar respuesta a las necesidades funcionales de las aplicaciones de Internet enriquecidas (RIA), gracias a las cuales es posible diseñar interfaces de usuario Web que recreen la amalgama funcional de las tradicionales aplicaciones SIG de escritorio– o [Ami10] –en el cual se ponen de manifiesto las ventajas de combinar SOA y los estándares OGC como mecanismo para dar respuesta a la mayoría de los problemas de las aplicaciones actuales SIG, en términos de falta de interoperabilidad, agilidad y alineación con los objetivos de negocio–.

Por otro lado, existen diferentes estudios que comparan la implementación de WS con SOA y ROA, entre los cuales se ha destacado en este trabajo el realizado en [Pau08] –ver punto 4.4.1–, aunque solo unos pocos han tenido en consideración connotaciones geoespaciales. En [Flo09] los autores resumen las aproximaciones SOA y ROA a los WS y exponen un mecanismo de encapsulación que permite proveer una interfaz SOA para aquellos servicios geoespaciales existentes que no sean compatibles con la normativa europea INSPIRE –sobre la cual se hablará en el punto 4.1.3–, pero no concretan nada sobre la adecuación de los mismos a ROA. En [Luc08] se presenta un análisis sobre la conveniencia de utilizar una interfaz REST para ofrecer los servicios de INSPIRE, el cual concluye indicando que no se obtienen beneficios reseñables, puesto que el modelo que sustenta a esta directiva es SOA y su implementación no se ajusta a ROA; no habiendo abordado las implicaciones que tendría una modificación de la normativa para dar cabida también a esta otra aproximación y los beneficios reales que se obtendrían. Por último, en [Gra12] los autores realizan una discusión técnica sobre la aplicación de los principios REST al estándar actual de geoprocesamiento publicado por el OGC –Web Processing Service (WPS), analizado en el punto 4.1.1.7–, concluyendo que la versión actual del estándar [Sch07] no se ajusta a las restricciones propias del protocolo REST, aunque no se concreta qué cambios reales se podrían realizar en éste y otros estándares del mismo organismo.

Por todo lo expuesto, se deduce que no se ha realizado un estudio pormenorizado de la aplicación de ambos modelos arquitectónicos, SOA y ROA, a todo el ámbito de servicios geoespaciales que dan cobertura a los SIG distribuidos. Dado que en el capítulo previo se han presentado las características principales de los dos modelos arquitectónicos que copan las implementaciones de Sistemas de Información (SI) distribuidos y que se conocen los retos a los que se enfrenta la comunidad geoespacial – ver punto 2.2–, el siguiente paso en este estudio será el análisis de las iniciativas que se han puesto en marcha dentro de este ámbito para la creación de SIG distribuidos que permitan hacer frente a los retos referidos.

4.1 Marco regulatorio

4.1.1 Geoservicios OGC

En línea con la evolución experimentada por los SIG que ha sido descrita con anterioridad en este documento –ver punto 2.1–, los sistemas para la captación de información primaria y el posterior procesado de la misma para la obtención de productos derivados, también han experimentado un enorme avance en las últimas décadas. No parece razonable que pudiera ser de otra forma, puesto que disponer de herramientas cada vez más versátiles y no hacerlo de información de base para realizar los análisis pertinentes, serían fenómenos contradictorios.

Sin embargo, es necesario poner de manifiesto que estos grandes volúmenes de información cartográfica no han estado al alcance de todos aquellos actores que potencialmente podían ser sus usuarios. La producción cartográfica se ha centrado fundamentalmente en organismos públicos, que en España podemos concretar en el Instituto Geográfico Nacional, la Dirección General del Catastro y, más recientemente, los organismos cartográficos de las comunidades autónomas, las diputaciones provinciales y los ayuntamientos más importantes. Por su parte, los usuarios u organizaciones que disponen de datos propios –capturados con múltiples técnicas, entre las que prevalecen los Sistemas de Posicionamiento Global (*GPS*)– necesitan superponerlos a un fondo cartográfico de referencia, propiedad de los organismos anteriores, que tradicionalmente no ha estado a su alcance o, en el mejor de los casos, lo ha estado previo pago de importantes cantidades de dinero.

40

Pese a ello, existe una importante corriente a nivel mundial que sustenta la necesidad de hacer público el acceso a la IG de base, de calidad y actualizada, contribuyendo a la dinamización de múltiples actividades que requieren, entre otras cosas, de dicha información. Como consecuencia de este fenómeno, han surgido diferentes iniciativas orientadas a la creación de estándares para la distribución de IG, con el objeto de incrementar la interoperabilidad entre los SIG.

Entre ellas, la más relevante en el ámbito geográfico es la protagonizada por el consorcio de empresas del sector de las aplicaciones SIG denominado *OGC*, sobre el cual ya se ha realizado una introducción en el apartado que analiza la historia de los SIG. Recordar que su objetivo no es otro que el de favorecer la creación de un ecosistema de sistemas de geoprocésamiento a través de las redes –corporativas e Internet– gracias a la definición de un conjunto de interfaces estándar. En [OGC14] se desarrolla más este objetivo y se referencian los siguientes objetivos estratégicos:

- Proveer la disponibilidad en el mercado de estándares abiertos y gratuitos que constituyan un valor tangible para los miembros del organismo y generen beneficios cuantificables para los usuarios.
- Liderar a nivel mundial la creación y establecimiento de estándares que hagan posible la existencia de infraestructuras para la entrega e integración de contenidos cartográficos y servicios dentro de procesos de negocio y cívicos.
- Facilitar la adopción de arquitecturas empresariales abiertas y con habilidades espaciales que sean capaces de actuar de forma global.

- Avanzar en la creación de estándares que contribuyan a la formación de novedosos mercados y aplicaciones para las tecnologías geoespaciales.
- Acelerar el proceso de asimilación por parte del mercado de la investigación en aspectos relativos a la interoperabilidad, mediante procesos de colaboración consorciados.

Categoría	Descripción	Estándares
Acceso a datos	Regulación de los mecanismos de acceso a IG.	<i>Sensor Observation Service</i> <i>Web Coverage Processing Service</i> <i>Web Coverage Service</i> <i>Web Feature Service</i> <i>Web Map Service</i> <i>Web Map Tiled Service</i> <i>Web Services Common</i>
Catalogación	Regulación de los mecanismos para la descripción de metadatos, la gestión de servicios de catálogo y la búsqueda de recursos.	<i>Catalog Service for the Web</i> <i>CSW-ebRIM Registry Service</i> <i>GeoSPARQL</i> <i>OpenSearch</i>
Estructuras de datos	Definición de formatos para la representación de IG.	<i>City Geography Markup Language</i> <i>Geography Markup Language</i> <i>Geospatial eXtensible Access Control Markup Language</i> <i>Keyhole Markup Language</i> <i>Observations and Measurements</i> <i>Sensor Model Language</i> <i>Simple Features</i> <i>Water Model Language</i>
Procesamiento	Articulación de los mecanismos de ejecución remota de procesos SIG.	<i>Coordinate Transformation Service</i> <i>Location Services</i> <i>Sensor Planning Service</i> <i>Table Joining Service</i> <i>Web Processing Service</i>
Programación	Especificación de APIs	<i>GeoAPI</i> <i>Ordering Services Framework for Earth Observation Products</i>
Transmisión de información	Mecanismos para la transmisión de IG.	<i>Filter Encoding</i> <i>GeoPackage Encoding</i> <i>Network Common Data Form</i> <i>Open Modelling Interface</i> <i>Open GeoSMS</i> <i>PUCK Protocol</i> <i>Sensor Web Enablement</i> <i>Symbology Encoding</i>
Otros	Regulaciones complementarias en la utilización de los servicios de acceso a datos.	<i>Style Layer Descriptor</i> <i>Web Map Context</i> <i>Web Services Context</i>

Tabla 4.1 Clasificación de las especificaciones de estándares OGC.

El proyecto fue iniciado en el año 1994 por un conjunto de 8 organizaciones estadounidenses –7 públicas y 1 privada–, pero su evolución imparable ha permitido alcanzar en 2014 los 478 miembros, mayoritariamente del sector privado y con alcance internacional.

La producción del OGC liberada en forma de estándares es importante, englobando un total de 42 especificaciones¹. Si se descartan aquellas que hacen referencia a variantes de alguna de las principales, estas quedan reducidas a 37, las cuales, a su vez, pueden clasificarse atendiendo al objetivo principal perseguido. La división resultante se muestra en la Tabla 4.1.

Pese a la profusión de especificaciones reseñada, el impacto que éstas han tenido en los mercados que pretenden regular ha sido muy heterogénea, siendo aquéllas que gozan de mayor aceptación –hecho que entronca perfectamente con las necesidades que tienen los usuarios, las cuales han sido identificadas a lo largo de este estudio– las que permiten acceso a datos y servicios mediante *WS*.

Dentro del ámbito de este estudio, se centra la atención en el análisis de todos aquellos estándares que son más relevantes para enfrentar los retos recogidos en el punto 2.2. Bajo esta óptica se analizarán los siguientes:

- Todos los englobados en la primera categoría –acceso a datos–.
- Los englobados dentro de la categoría de procesamiento, representados por el estándar *Web Processing Service (WPS)*, del cual podrían derivarse los restantes.

4.1.1.1. Características generales

Centrando el estudio, OGC dicta una serie de recomendaciones de carácter general que deben ser cumplidas por las diferentes implementaciones que publica este organismo. Estas recomendaciones, recogidas en [Gre10], aglutinan aspectos que regulan las características de las peticiones y de las respuestas –incluyendo sus parámetros, protocolos y codificación–. Entre estas directrices, a efectos de este estudio, cabe sintetizar información sobre los distintos protocolos y codificaciones admitidos por los métodos de estos servicios. La Tabla 4.2, Tabla 4.3 y Tabla 4.4 muestran las variantes aplicadas, de forma ilustrativa, al método *GetCapabilities* –se ha escogido éste por ser común a todas las tipologías de servicios que se analizarán en este estudio–.

Con respecto a la formulación mostrada en la Tabla 4.4, la codificación *SOAP* descrita en [Gre10] – documento que recoge aspectos comunes a todas las especificaciones OGC– hacía referencia a “*Document Literal-Wrapped*”, pero las implementaciones finalmente realizadas han optado por utilizar la opción “*Document Literal*” que permite la sobrecarga de métodos.

¹ Información extraída de [OGC14] a fecha de 24 de julio de 2014.

Además de lo expuesto, en [Per02] se recoge las características que debe considerar una implementación de servicios OGC en una arquitectura basada en el intercambio de mensajes –como es el caso de SOAP sobre protocolo HTTP– para que, además, permita el encaminamiento de servicios, en lo que OGC denomina una arquitectura simple de servicios. Son las siguientes:

Primera formulación (obligatoria)	
Protocolo	HTTP
Método	Get
Codificación	Parejas de clave y valor (KVP)
Ejemplo de petición	
http://hostname:port/path?SERVICE=WCS&REQUEST=GetCapabilities&ACCEPTVERSIONS=1.0.0&SECTIONS=Contents&UPDATESEQUENCE=XYZ123&ACCEPTFORMATS=text/xml&ACCEPTLANGUAGES=en-CA	

Tabla 4.2 Primera formulación admitida por los estándares OGC.

Segunda formulación (obligatoria)	
Protocolo	HTTP
Método	Post
Codificación	XML
Ejemplo de petición	
<pre><?xml version="1.0" encoding="UTF-8"?> <GetCapabilities xmlns="http://www.opengis.net/ows/2.0" xmlns:ows="http://www.opengis.net/ows/2.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/ows/2.0 fragmentGetCapabilitiesRequest.xsd" service="WCS" updateSequence="XYZ123" acceptLanguages="en-CA"> <AcceptVersions> <Version>1.0.0</Version> </AcceptVersions> <Sections> <Section>Contents</Section> </Sections> <AcceptFormats> <OutputFormat>text/xml</OutputFormat> </AcceptFormats> <AcceptLanguages> <Language>en-CA</Language> </AcceptLanguages> </GetCapabilities></pre>	

Tabla 4.3 Segunda formulación admitida por los estándares OGC.

- Operaciones modeladas mediante mensajes –*message-operations*–: una operación descrita de esta forma consiste en una petición y una respuesta. Ambas contendrán parámetros que son transmitidos de forma uniforme e independiente del contenido.
- Separación de la lógica de control y los datos –*separation of control and data*–: es posible que un cliente que realiza una invocación a un servicio solo esté interesado en conocer el estado de una

operación, mientras que otros requieran acceder a los datos generados como consecuencia de la operación en cuestión.

Tercera formulación (opcional)	
Protocolo	HTTP + SOAP
Método	Post
Codificación	XML (SOAP Document Literal)
Ejemplo de petición	
<pre><soap:envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <soap:body> <complexType name="RequestBaseType"> <annotation> <documentation>XML encoded operation request base</documentation> </annotation> <attribute name="service" type="string" use="required"> <annotation> <documentation>Service type identifier</documentation> </annotation> </attribute> <attribute name="version" type="string" use="required"> <annotation> <documentation>Specification version for OWS version</documentation> </annotation> </attribute> </complexType> ... </soap:body> </soap:envelope></pre>	

Tabla 4.4 Tercera formulación admitida por los estándares OGC.

- Servicio con/sin estado –*stateful vs. stateless service*–: desde el punto de vista de la simplicidad, conviene que los servicios no dispongan de estado, aunque pueden existir algunos servicios en los que se requiera cumplir ciertas precondiciones antes de su invocación. En este último caso, es conveniente modelar el diagrama de estados y las transiciones entre los mismos –siendo éstas consecuencia de la invocación de las operaciones–.
- Tipo de servicio conocido –*known service type*–: cada una de las instancias de un servicio debe corresponderse con una tipología concreta de servicio y ésta debe conocerse antes de la invocación del mismo.
- Hardware adecuado –*adequate hardware*–: se asume que cualquier aspecto relacionado con la tecnología utilizada para el hospedaje de los servicios es completamente transparente para el usuario, debiendo ser ésta adecuada para dicha finalidad.

En cuanto a la taxonomía de los geoservicios, OGC realiza la siguiente división [Per02]:

- Servicios geográficos de interacción humana –*geographic human interaction services*–: son aquellos ubicados en la vertiente cliente de la arquitectura cliente-servidor que habilitan mecanismos mediante los cuales los usuarios finales de las aplicaciones pueden buscar, visualizar,

editar, etc., contenidos cartográficos. A título ilustrativo, los siguientes ejemplos de servicios cliente forman parte de esta categoría:

- Visualizador de catálogo: permite al usuario la localización y exploración de conjuntos de datos y servicios geográficos, mediante la gestión de sus metadatos.
- Visualizador geográfico: permite al usuario la visualización de conjuntos de objetos geográficos o coberturas, así como la interrogación de los mismos para conocer sus propiedades –atributos–.
- Editor de servicios: permite al usuario la definición y gestión de sucesiones de servicios encadenados.
- Editor de IG: permite al usuario interactuar con los conjuntos de datos vectoriales pudiendo modificar sus características geométricas, alfanuméricas y, también, de visualización.
- Servicios geográficos de gestión de modelos o información –*geographic model/information management services*–: se trata de servicios ubicados en la vertiente servidora gracias a los cuales los servicios de la categoría anterior pueden acceder a las diferentes colecciones de IG. Algunos ejemplos de esta categoría son los que se muestran a continuación:
 - Servicio de acceso a objetos geográficos: permite a los clientes acceder y gestionar un repositorio de datos geográficos.
 - Servicio de acceso a mapa: permite a los clientes visualizar mapas generados a partir de la aplicación de una determinada simbología cartográfica a los datos almacenados en los repositorios.
 - Servicios de acceso a coberturas: permite a los clientes visualizar coberturas cartográficas almacenadas en los repositorios.
 - Servicio de registro: permite a los clientes acceder al repositorio de catálogo en el que se recogen las diferentes colecciones de metadatos sobre las colecciones de datos y servicios catalogados.
 - Servicio de catálogo: permite a los clientes realizar búsquedas de conjuntos de datos y servicios dentro de un repositorio de metadatos y, además, permite la gestión de los mismos.
 - Servicio de diccionario geográfico –*gazetteer*–: permite acceso a un directorio de instancias de objetos geográficos para conocer la posición en la que estos objetos se ubican.
- Servicios geográficos de gestión de tareas o flujos de trabajo –*geographic workflow/task management services*–: abarca aquellos servicios que permiten la gestión de tareas o flujos de

trabajo con datos y servicios geográficos. Los siguientes servicios son muestras de esta categoría funcional:

- Servicio de definición de cadenas de servicios: permite la definición e instanciación de cadenas de servicios, especificando las condiciones que se deben cumplir antes y después de la ejecución de cada uno de los servicios que participan en el encadenamiento.
- Servicio de promulgación de flujos de trabajo: permite la interpretación de una cadena de servicios y controla la instanciación de los servicios y la secuenciación de las actividades.
- Servicio de suscripción: permite a los clientes registrarse para la recepción de notificaciones sobre determinados eventos.
- Servicios geográficos de procesamiento –*geographic processing services*–: reúne todos aquellos servicios que permiten la ejecución de procesos sobre IG. En este sentido, se realiza una diferenciación entre aquellos que permiten procesar la componente espacial, temática, temporal y, además, los metadatos de la IG. Ejemplos de esta categoría serían los que se muestran a continuación:
 - Servicio de conversión de coordenadas: permite a los clientes modificar el sistema de coordenadas en el cual se encuentra referenciada la componente geométrica de los objetos geográficos, siempre que el sistema original y el resultante estén referenciados a un mismo datum.
 - Servicio de transformación de coordenadas: permite realizar la conversión de coordenadas entre sistemas de referencia que estén referenciados a datum diferentes.
 - Servicio de ortorectificación: permite a los clientes la rectificación de una imagen mediante la eliminación de la inclinación y el desplazamiento presente en los píxeles de la imagen como consecuencia de la orografía del terreno.
 - Servicio de conversión cobertura/vector: permite a los clientes realizar conversiones de coberturas a vectores y viceversa.
- Servicios geográficos de comunicación –*geographic communication services*–: reúnen aquellos servicios que facilitan el proceso de comunicación de IG. A título ilustrativo:
 - Servicio de codificación: permite al cliente la implementación de reglas de codificación y decodificación de la información a transmitir.
 - Servicio de transferencia: permite al cliente implementar uno o varios protocolos de transferencia para el intercambio de información entre SI distribuidos a través de una red.
- Servicios geográficos de gestión del sistema –*geographic system management service*–: dentro de esta taxonomía no han sido identificados implementaciones concretas en el ámbito geográfico. Con

carácter genérico, un servicio de gestión de un sistema permite la implementación de políticas de autorización y autenticación, pasarelas de pago, monitorización del rendimiento, etc.

Una vez caracterizadas los aspectos genéricos aplicables a los distintos geoservicios contemplados dentro del paraguas del OGC, se completa este apartado con el análisis de los estándares de servicios incluidos en la categoría de acceso a datos.

4.1.1.2. Servicio de visualización WMS

Dentro de esta tipología de servicios se concentran la inmensa mayoría de los servicios implementados a nivel mundial, siendo este estándar el que ha obtenido una mayor difusión. Para poder fundamentar esta afirmación, basta repasar los datos que figuran en la Tabla 4.5 y Tabla 4.6.

Servicio	INSPIRE	IDEE	IDEG	IDEC
WMS	12415	1227	74	453
WMTS	75	12	0	3
WFS	1446	136	0	10
WCS	15	5	0	2
WPS	7	3	0	2
SOS	0	0	0	11

Tabla 4.5 Número de geoservicios agrupados por tipología y catálogo².

Catálogo	% servicios visualización
INSPIRE	88,0%
IDEE	87,3%
IDEG	100,0%
IDEC	94,8%

Tabla 4.6 Porcentaje de geoservicios de visualización WMS por catálogo³.

La primera versión del estándar WMS –1.0.0– se remonta a Abril del año 2000 y después han surgido tres versiones adicionales –1.1.0, 1.1.1 y 1.3.0–. El estudio que se aborda a continuación se centra en la última de ellas, recogida como estándar ISO con referencia 19128 y cuya especificación de implementación se encuentra en [Del06].

² Consulta realizada el 5 de Junio de 2014.

³ Datos obtenidos a partir de la información recogida en la Tabla 4.5.

La caracterización principal de este estándar es la de proveer una interfaz para la operación de servicios que produzcan mapas a partir de información geográfica convenientemente referenciada a un sistema de referencia de coordenadas. El estándar define mapa como “un archivo digital que almacena una representación en formato imagen de la información geográfica, especialmente orientado a su visualización en la pantalla de un ordenador”.

En este sentido entiende el autor que, desde un punto de vista puramente académico, el estándar debería utilizar el concepto de documento cartográfico y no el de mapa, puesto que un mapa no es más que una tipología cartográfica –como también lo es una imagen satélite–, y cualquiera de ellas es susceptible de ser publicada mediante un servicio *WMS*.

Para tal fin, el estándar articula el siguiente conjunto de métodos obligatorios:

- *GetCapabilities*: permite a los clientes obtener metadatos sobre el servicio ofrecido. Estos metadatos, encapsulados en un documento *XML*, ofrecen información sobre el proveedor del servicio, así como una descripción de las diferentes capas que se publican.
- *GetMap*: permite a los clientes obtener la imagen correspondiente a la petición realizada, en la cual se definen, entre otros aspectos, el área geográfica, el sistema de referencia en el que se expresa dicha área, el formato de la imagen resultante, el tamaño de la misma, etc. Además, la utilización de los estándares *OGC Style Layer Descriptor (SLD)*, junto con *OGC Filter Encoding* dotan de una mayor funcionalidad a esta operación, permitiendo modificar la simbología aplicada a la información –sea esta de origen ráster o vectorial–, así como realizar filtros en la IG a representar –por ejemplo en función de un determinado valor de uno de sus atributos–.

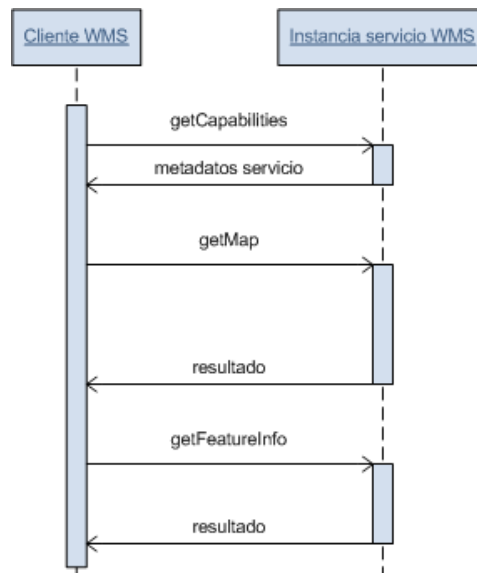


Figura 4.1 Interacciones entre un cliente y un servidor *WMS* [fuente: propia].

Así como la siguiente opcional:

- *GetFeatureInfo*: permite a los clientes obtener más información sobre los objetos geográficos representados en un mapa que pertenecen a alguna de las capas para las que se ha habilitado expresamente esta opción –parámetro *queryable="1"*–. La respuesta se obtendrá en alguna de las Extensiones Multipropósito de Correo de Internet (*MIME*), como *HTML* o *XML*.

El diagrama *UML* de secuencia que se muestra en la Figura 4.1 expone las interacciones típicas que se producen entre un cliente y un servidor *WMS*.

En el ejemplo, el cliente, después de realizar la consulta oportuna para descubrir las capacidades del servicio, acaba generando una petición de obtención de un mapa y de las propiedades de uno de los elementos geográficos en él representados. Aunque el diagrama muestre un intercambio sincronizado, este no es un requerimiento del estándar –ver punto 4.4.2.2–, lo cual puede conllevar problemas de sincronización cuando un mismo cliente realiza múltiples peticiones a un servidor y éstas no son atendidas y contestadas en el mismo orden en el que fueron enviadas –se descarga esta responsabilidad en la vertiente del servidor–.

4.1.1.3. Servicio de visualización *WMTS*

La publicación del estándar *WMS* supuso un importante avance en cuanto a la generación de servicios capaces de ofrecer contenidos cartográficos de forma estándar. Anteriormente, distintos fabricantes habían explorado la creación de *APIs* orientadas a la publicación de servicios de visualización, pero siempre buscando la fidelización de clientes a las respectivas plataformas y dejando a un lado la interoperabilidad entre las mismas. La irrupción en el mercado de las soluciones de visualización cartográfica por Internet de *Google*, supuso un importante hito en la evolución de este tipo de soluciones.

49

De esta forma, cinco años más tarde de que *WMS* se mostrara al mundo con su primera versión, en 2005 se lanza la primera versión de *Google Maps* y también de *Google Earth*. En ambos casos se consumen servicios de visualización de información publicados en Internet y, lo más importante, lo hacen haciendo gala de unos rendimientos inigualables con los sistemas que proveían los tradicionales fabricantes de tecnología servidora de geoservicios. Para ello se apoyan en una concepción que pivota sobre las siguientes características:

- Sencillez: rasgo que se aplica al modelo de datos –compuesto por escasas capas de información–, a la simbología utilizada para representar dichos datos –concebida para resultar atractiva para el común de los usuarios, el cual no posee un bagaje en el ámbito de la consulta de contenidos cartográficos– y al sistema de referencia en el cuál se almacena la información –optándose por la utilización de una esfera en vez de un elipsoide de revolución–. Estas tres aplicaciones de esta característica de diseño permiten la creación de servicios con unos rendimientos asombrosamente mejores que los existentes en ese momento.
- División: la aproximación realizada por *Google* pasa por la determinación de un conjunto de escalas de visualización –niveles de resolución– y para cada uno de ellos, a partir de un origen de coordenadas determinado, se realiza un troceado del territorio en fragmentos de tamaño regular –cuadrados– que se encapsulan en imágenes de 256 x 256 píxeles. Las aplicaciones cliente, por tanto, tendrán que ceñir sus peticiones a uno de esos niveles de resolución, al origen de coordenadas y a las teselas correspondientes al ámbito cartográfico que desean visualizar. Si el

trabajo de generación de las teselas se realiza de forma previa a la publicación, el proceso de presentar la cartografía al usuario tiene un periodo de latencia que implica simplemente el tiempo necesario para enviar las peticiones y transmitir por red las imágenes previamente generadas.

Este nuevo escenario suscita la aparición de numerosas críticas al elevado coste, en términos de rendimiento, asociado a la utilización de servicios *WMS*. Por otro lado, desde el ámbito tradicional de los SIG se entiende que la simbología ofrecida por *Google* no es de suficiente calidad, al menos para un determinado perfil de usuarios avanzados. Dentro de este contexto, la plataforma *TileCache* desarrollada bajo licencia de código libre *Berkeley Software Distribution (BSD)* por la empresa Metacarta, es la primera en exponer una interfaz que emula el funcionamiento de *Google Maps* en cuanto a la descomposición en niveles de resolución y en teselas del territorio. La primera versión de *TileCache* se distribuye en el año 2006 y su gran éxito propició que en el año 2008, la *Open Source Geospatial Foundation (OSGeo)* formulara una recomendación denominada *Tile Map Service (TMS)* [OSG12] que, basándose en las directrices de la propuesta de perfil *WMS* denominada *Web Map Service Tile Caching (WMS-C)*, formase una guía de referencia para la implementación de clientes y servidores de cartografía. Dos años más tarde, el *OGC* publica la primera y hasta la fecha única revisión del estándar *Web Map Tile Service (WMTS) –1.0.0–*, que recoge el testigo del trabajo realizado por *TMS*, aunque ciertas discrepancias entre sus redactores y los desarrolladores de *TMS*⁴ han motivado la coexistencia de ambas especificaciones.

A modo de resumen, *WMTS* es un estándar construido sobre los esfuerzos realizados anteriormente por la comunidad geoespacial para la creación de servicios escalables y con un elevado rendimiento para la distribución de mapas, gracias a la distribución de fragmentos de mapa previamente creados o bien creados en línea, con la peculiaridad de que, dado que su esquema de división geográfica está prefijado, permite habilitar mecanismos estándares en las redes que facilitan la escalabilidad –como por ejemplo el uso de sistemas de cacheado distribuidos– [Jul10].

Una vez realizada la introducción, pasa a analizarse el conjunto de métodos u operaciones que contempla con carácter obligatorio este estándar:

- *GetCapabilities*: permite a los clientes obtener metadatos sobre el servicio ofrecido. Estos metadatos, encapsulados por defecto en un documento *XML*, ofrecen información sobre el proveedor del servicio, así como una descripción de las diferentes capas que se publican.
- *GetTile*: permite a los clientes obtener el fragmento de cartografía –tesela– correspondiente a la petición realizada, en el formato especificado en la misma –tiene que corresponderse con uno de los admitidos por el servidor, según se informa en el descriptor del servicio–.

Así como la siguiente opcional:

⁴ En la definición de *WMTS* se ha optado por tomar como origen de coordenadas de la cuadrícula en la que se divide la zona cartografiada la esquina superior derecha, tal y como se produce en las estructuras de las imágenes y como estableció *Google* en sus servicios. Por el contrario, *TMS* tiene su origen en la esquina inferior izquierda, siguiendo la tradición cartográfica.

- *GetFeatureInfo*: permite a los clientes obtener más información sobre los objetos geográficos representados en una tesela –la petición debe especificar la tesela en cuestión y el píxel dentro de la misma en el que se localizan los elementos a identificar, indicando su posición (*fila,columna*)–.

La respuesta se obtendrá en el formato especificado en la petición –tiene que corresponderse con uno de los admitidos por el servidor, según se informa en el descriptor del servicio–, aunque el estándar aboga por la implementación del perfil *Simple Features* de *GML*.

El diagrama *UML* de secuencia que se muestra en la Figura 4.2 expone las interacciones típicas que se producen entre un cliente y un servidor *WMTS*.

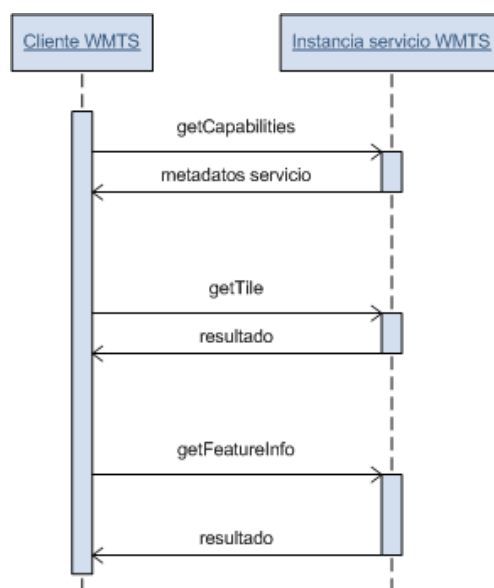


Figura 4.2 Interacciones entre un cliente y un servidor *WMTS* [fuente: propia].

En el ejemplo, el cliente, después de realizar la consulta oportuna para descubrir las capacidades del servicio, acaba generando una petición de obtención de una tesela y de las propiedades de uno de los elementos geográficos representados en una determinada localización de la misma, todo ello sin contemplarse requerimiento alguno de sincronización.

4.1.1.4. Servicio de descarga *WFS*

Los servicios de visualización cubren el principal espectro de los usuarios de IG, formado mayoritariamente por colectivos que utilizan herramientas de visualización cartográficas –en muchos casos el calificativo de aplicaciones SIG es exagerado– para un uso no profesional. Estos colectivos –según se recoge en 2.2– son ávidos consumidores de información servida por servicios de visualización.

Sin embargo, dentro de los grupos que se han caracterizado tradicionalmente por utilizar cartografía –cartógrafos, geógrafos, topógrafos, etc.–, existen necesidades que no quedan cubiertas con herramientas de simple visualización. Estos profesionales requieren trabajar con la IG en crudo, pudiendo realizar procesos de análisis y transformación que permitan derivar numerosos productos, como por ejemplo,

nuevos conjuntos de IG. Para ello, se plantean dos alternativas a la hora de obtener una copia de los datos:

- Realizar las gestiones oportunas –físicas o telemáticas– ante el productor de la información.
- Acceder a un *WS* que permita la descarga.

Lógicamente, el segundo procedimiento es razonablemente más eficaz. A priori, en ambos procesos el cliente tendrá que enfrentarse a la heterogeneidad de formatos propios del ámbito de la cartografía. Es cierto que hay formatos que se han convertido en estándares de facto para la compartición de IG –como el *shapefile* de *ESRI* para datos vectoriales o el *TIFF* para raster–, pero en algunas ocasiones las limitaciones del productor –formatos, versiones, etc.– y/o del consumidor –conocimientos, herramientas, etc.– hacen que esta tarea no sea trivial.

Para dar respuesta a esta última consideración, el *OGC* presentó en el año 2002 la primera especificación de implementación del estándar *Web Feature Service (WFS)* [Vre02] –versión 1.0.0–, la cual fue seguida de dos versiones adicionales –1.1.0 y 2.0.0–. El estudio que se aborda a continuación se centra en la última de ellas, elevada también a la categoría de estándar *ISO* con la referencia 19142 y cuya especificación de implementación se encuentra en [Vre10]. En línea con lo recogido anteriormente, *WFS* provee acceso a los clientes a través de una red a servicios que permiten la descarga de datos vectoriales almacenados en repositorios cuya tecnología de implementación es completamente transparente para el usuario, siendo el resultado expresado en *Geography Markup Language (GML)*. La selección de los datos a descargar puede realizarse mediante las operaciones disponibles para la gestión de expresiones de filtrado parametrizadas.

Sin embargo, *WFS* va más allá del concepto de sistema de descarga de IG en formato vectorial, puesto que también define operaciones para la ejecución de transacciones que permitan la actualización del repositorio de datos subyacente. Esta funcionalidad adicional también se ha denominado *Web Feature Service Transaccional (WFS-T)*.

WFS define los siguientes métodos u operaciones obligatorias:

- *GetCapabilities*: permite a los clientes obtener metadatos sobre el servicio ofrecido. Estos metadatos, encapsulados en un documento *XML*, ofrecen información sobre el proveedor del servicio, así como una descripción somera de los diferentes procesos que ofrece.
- *DescribeFeatureType*: permite a los clientes obtener una descripción del esquema de datos asociado a cada uno de los tipos de objetos geográficos –entidades– ofrecidos por la instancia del servicio. Dicha descripción es aplicable tanto a las entidades que se envíen al servidor –como consecuencia de una operación *Transaction*–, como a las que éste devuelve como respuesta –operación *Transaction* y *GetFeature*–.
- *GetPropertyValue*: permite a los clientes obtener información sobre uno de los atributos de una o varias entidades de las cuales se especifica su identificador.

- *GetFeature*: permite a los clientes obtener en formato *GML –ISO 19136:2007–* las entidades que satisfacen el filtro especificado en la petición. Dicho filtro puede contener expresiones alfanuméricas que apliquen operadores lógicos a los atributos de la capa de información en la que se encuentran las entidades y/o expresiones que utilicen operadores geográficos *–within, intersect, etc.–*. La expresión a utilizar puede residir en una consulta almacenada *–stored query–* o bien definirse en la propia petición *–abstract query expression–*. Adicionalmente, se pueden suministrar otros formatos de descarga.
- *ListStoredQueries*: permite al cliente obtener un listado de todas las consultas almacenadas. Estos filtros o expresiones pueden utilizarse en las operaciones *GetPropertyValue*, *GetValue*, *GetFeatureWithLock* y *LockFeature*; y a cada una de ellas se le asigna un identificador único.
- *DescribeStoredQueries*: permite al cliente obtener la descripción de las consultas almacenadas en el servidor del servicio, mediante el conjunto de metadatos que las definen.

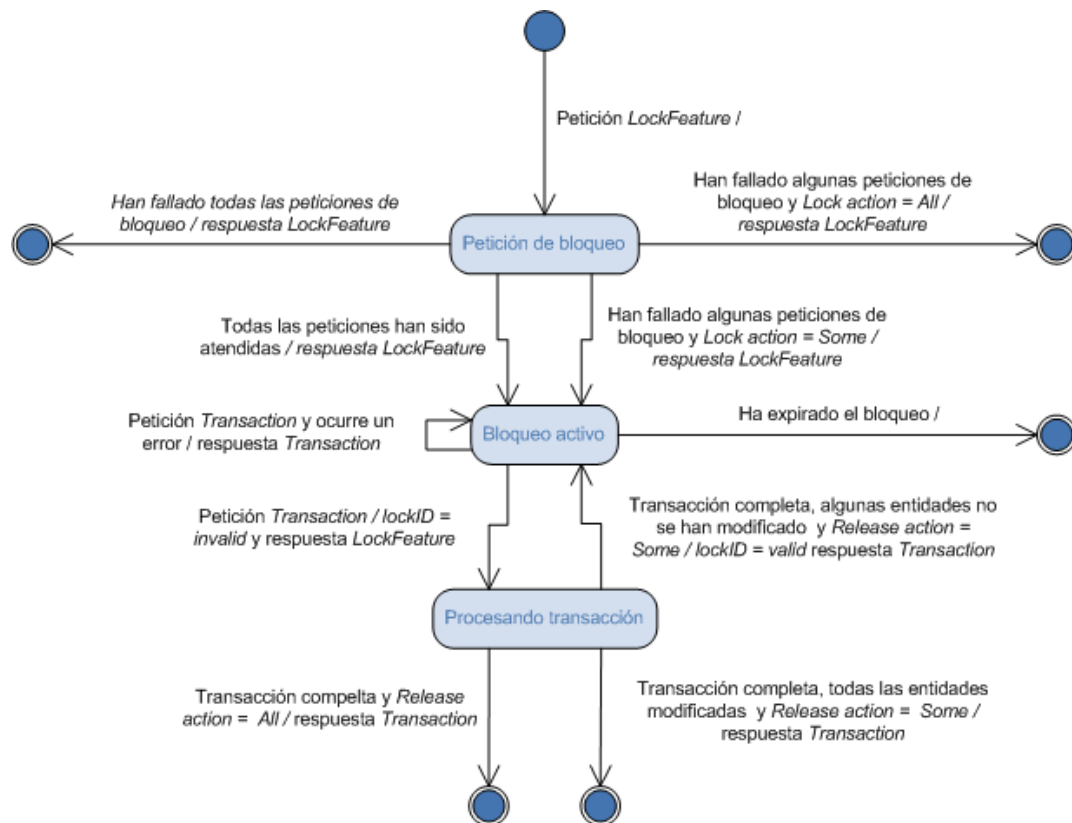


Figura 4.3 Diagrama UML de transición de estados de los bloqueos en WFS [Vre10].

Y las siguientes operaciones opcionales:

- *LockFeature*: permite a los clientes bloquear el conjunto de entidades que satisfacen el filtro especificado en la petición, de forma similar a como se realizaba en la operación *GetFeature*. Dada la falta de estado inherente a las conexiones Web, la semántica de las transacciones como las que permite este estándar no puede ser preservada; salvo que se habilite algún sistema alternativo en el cual el acceso a los recursos se realice de forma mutuamente exclusiva.

- *GetFeatureWithLock*: operación similar a *GetFeature*, aunque en esta ocasión permite a los clientes bloquear las entidades descargadas en el servidor. De esta forma, ante una hipotética edición de alguna de ellas por parte de otro operador, el bloqueo generado impedirá su modificación, preservando la integridad del sistema.

La Figura 4.3 muestra el diagrama de estados propio del mecanismo de bloqueo diseñado en el estándar *WFS*. En él se puede observar cómo el estándar permite realizar peticiones que solo se consideren exitosas si su aplicación al conjunto de entidades afectadas es satisfactoria –*Lock action = All*– o si lo es con un subconjunto –*Lock action = Some*–. Por otro lado, también queda clara la relación existente entre esta operación y la que permite ejecutar transacciones –*Transaction*–.

Cuando se solicita el bloqueo, el sistema devuelve un identificador del mismo –*lockId*– que podrá ser utilizado por el cliente que realiza la operación en un proceso posterior de desbloqueo, siempre que el periodo de expiración no haya sido excedido.

- *Transaction*: operación opcional que permite al cliente indicar la operación de transformación a aplicar a las geometrías del servicio. Esta operación permite crear, modificar, reemplazar y borrar entidades dentro del repositorio al que accede el servicio; siendo *GML* el formato canónico de representación de la entidades.
- *CreateStoredQuery*: permite al cliente crear una consulta que permite indicar las entidades afectadas por una de las otras operaciones y almacenarla en el servidor para su utilización en posteriores ocasiones.
- *DropStoredQuery*: permite al cliente eliminar del servidor una consulta.

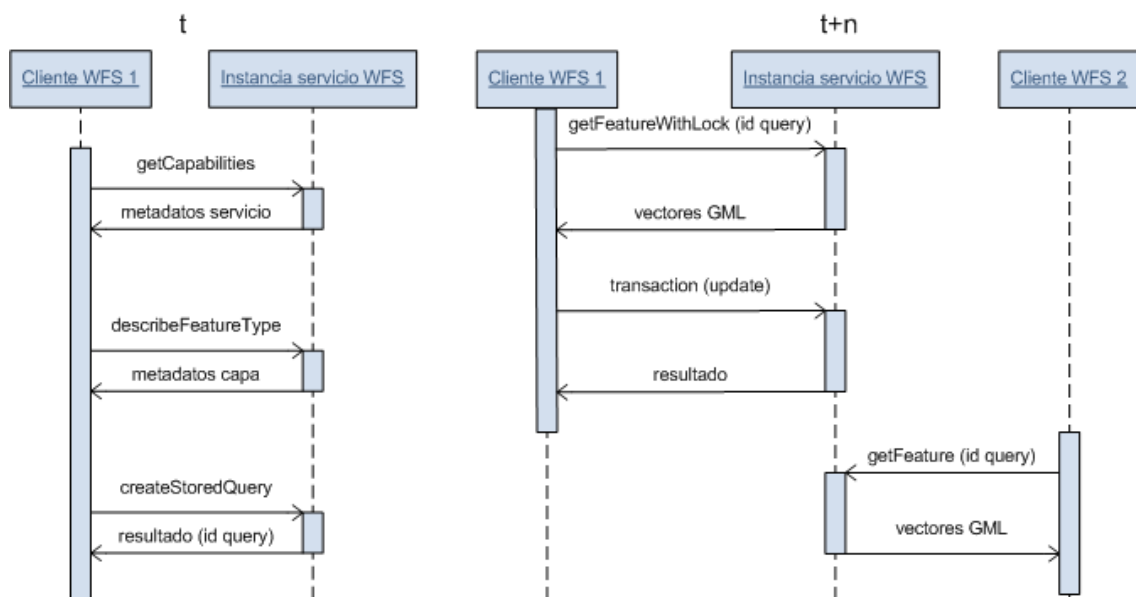


Figura 4.4 Interacciones entre un cliente y un servidor *WFS* [fuente: propia].

El diagrama *UML* de secuencia que se muestra en la Figura 4.4 –la parte derecha, $t+n$, es la continuación de la secuencia mostrada en la izquierda, t – expone las interacciones típicas que se producen entre un cliente y un servidor *WFS*, sin necesidad de sincronización.

En el ejemplo, un primer cliente inicia el intercambio con el servidor realizando una petición para descubrir las capacidades del servicio y otra para conocer la estructura de las entidades de una capa de información concreta –estas peticiones solo son necesarias cuando se desconoce el servicio o bien cuando se tenga conocimiento de que ha ocurrido algún cambio en la interfaz del mismo–. Posteriormente, a modo de ejemplo, se muestra un hipotético intercambio de solicitudes al servidor y de sus correspondientes respuestas, en el cual ese mismo cliente crea una expresión de filtro que es almacenada en el servidor y utilizada posteriormente en una petición *getFeatureWithLock* que permitirá descargar un conjunto de geometrías en formato *GML*. Una vez descargadas, el cliente realiza un proceso de edición de las mismas que concluye con la ejecución del método *Transaction*. Como resultado, las entidades en el repositorio de datos son actualizadas y es liberado el bloqueo. Un segundo cliente *WFS* realiza una petición para descargar dichas geometrías, una vez modificadas.

4.1.1.5. Servicio de descarga *WCS*

Completando la oferta de servicios orientados a la descarga de información que se inició con el estándar *WFS*, el *OGC* presentó en el año 2003 la primera especificación de implementación del estándar *WCS* [Eva03] –versión 1.0.0–, la cual fue seguida de dos versiones adicionales –1.1.0, 2.0.0–. El estudio que se aborda a continuación se centra en la modificación realizada sobre la última de ellas –2.0.1– y cuya especificación de implementación se encuentra en [Bau12]. En línea con lo recogido anteriormente, *Web Coverage Service (WCS)* provee acceso a los clientes a través de una red a servicios que permiten la descarga de coberturas multidimensionales, almacenadas en repositorios cuya tecnología de implementación es completamente transparente para el usuario.

El estudio del concepto de cobertura, tal y como se presenta en [OGC06], permite inferir que una cobertura es un documento cartográfico en el que se representa un conjunto de objetos geográficos o entidades que tienen asociados unos atributos cuyos valores vienen determinados por funciones que asocian un valor concreto para cada posición dentro del dominio espacial, temporal o espacio-temporal en el que se encuentran las entidades. Dado que las funciones mencionadas generan un número ilimitado de valores dentro del dominio, en la práctica una aproximación común es la de dividir el dominio en un conjunto finito de unidades y calcular para cada una de ellas el valor de las funciones. De esta forma, se pasa de tener una cobertura continua a una discreta –las imágenes aéreas y los modelos digitales de terreno son ejemplos ilustrativos de coberturas de carácter discreto–.

Por tanto, aunque tradicionalmente en los SIG se ha discernido entre información discreta –vectores– y continua –raster–, la diferenciación entre ambos entornos no es tan clara en todas las ocasiones y hay ciertos fenómenos que pueden representarse en cualquiera de las dos modalidades. Bajo este prisma, a efectos prácticos, la diferencia entre un servicio *WMS* y otro *WCS* es que el primero representa datos espaciales en forma de mapas estáticos –convertidos en imágenes por el servidor antes de ser enviados al cliente–, mientras que el segundo provee datos junto con toda su semántica original, a partir de los cuales es posible realizar diferentes análisis y no solo una mera representación.

Por otro lado, comparando *WFS* con *WCS*, en este caso el primero devuelve entidades espaciales discretas –puntos, polígonos, líneas, etc.– que pueden tener un conjunto de atributos asociados, mientras que *WCS* devuelve coberturas que también representan entidades espaciales, pero que en este caso abarcan la totalidad del dominio espacial y tienen asociadas un conjunto de atributos que asignan valores a cada una de las unidades espaciales en las que se divide dicho dominio.

Dentro de las coberturas se diferencian tipologías en función de los datos representados. Para conocer la lista de tipologías de coberturas soportadas dentro de *WCS*, puede consultarse [Bau10].

Entrando de lleno en el contenido del estándar, la versión 2.0.0 de *WCS* se organiza de forma modular, disponiendo de un núcleo –*core*– que debe ser implementado por cada cliente y servidor compatible, así como por un conjunto de extensiones que son opcionales y que permiten aumentar la capacidad funcional –ver Figura 4.5–.

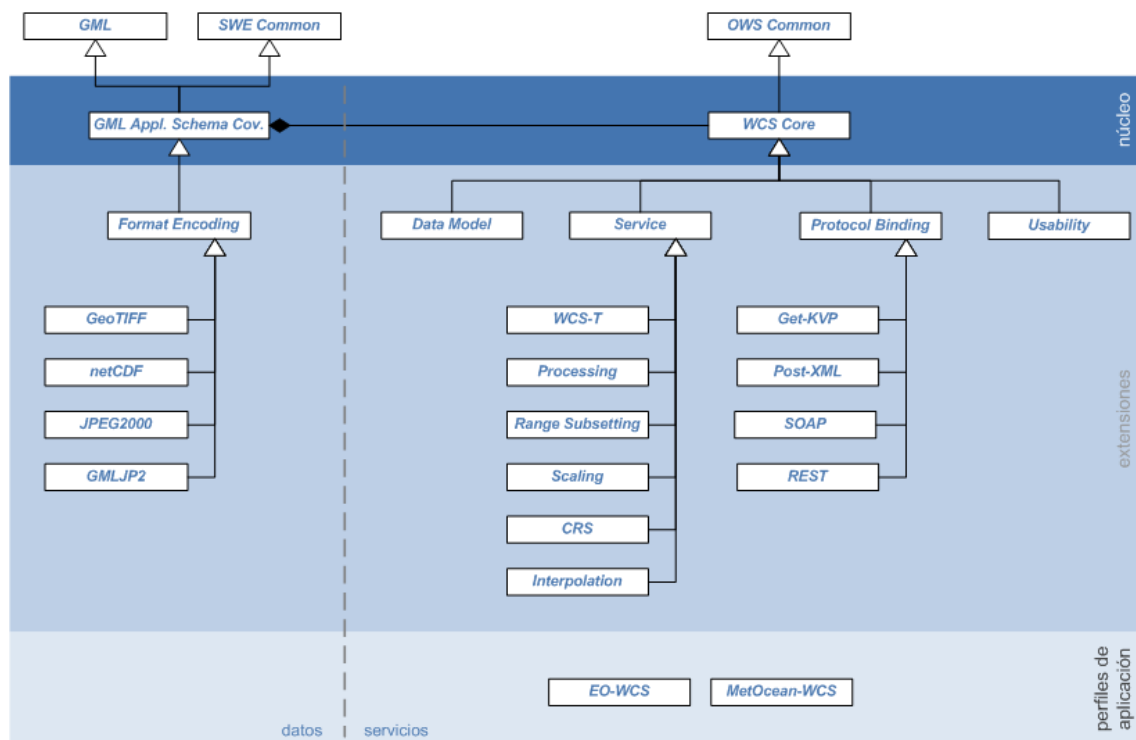


Figura 4.5 Disposición modular del estándar *WCS* [OGC14b].

En cuanto a los métodos definidos con carácter obligatorio, son los que se refieren a continuación:

- *GetCapabilities*: permite a los clientes obtener metadatos sobre el servicio ofrecido. Estos metadatos, encapsulados en un documento *XML*, ofrecen información sobre el proveedor del servicio, así como una descripción somera de las diferentes coberturas que ofrece.
- *DescribeCoverage*: permite a los clientes obtener los metadatos, en un documento *XML*, que describen de forma detallada las características de una cobertura publicada por el servicio.

- *GetCoverage*: permite a los clientes obtener una cobertura derivada de la original, siendo esta la totalidad o un fragmento de la misma comprendida en un rango de propiedades, una localización espacio-temporal y un formato de codificación determinados. La respuesta se codificará en lenguaje *GML* –según lo dispuesto en [Bau10]–, aunque se permite a los implementadores ofrecer formatos adicionales especificados mediante la codificación *MIME-type* oportuna, entre los cuáles pueden figurar los recogidos en Figura 4.5 –*GeoTIFF*, *netCDF*, *JPEG2000* y *GMLJP2*–.

Por otro lado, el estándar define dos opciones para especificar el dominio o localización de la zona de interés:

- Recortar –*trimming*–: se indica la dimensión –parámetro *dimension*– y los límites inferior y superior –parámetros *trimlow* y *trimhigh*, ambos deben estar dentro del dominio de la cobertura– a los que debe ajustarse la cobertura resultante. Dicha cobertura tiene un dominio que, en la dimensión especificada, se reduce a los límites indicados. Por tanto, la dimensión de la cobertura resultante es igual que la de la original. Se permite realizar la operación de recorte en más de una dimensión simultáneamente –incluyendo la tupla de parámetros referida para cada una de las dimensiones–. La Figura 4.6 muestra un ejemplo de petición *GetCoverage* que incluye un recorte en dos dimensiones –*x* e *y*–.

```
<?xml version="1.0" encoding="UTF-8"?>
<wcs:GetCoverage xmlns:wcs="http://www.opengis.net/wcs/2.0"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wcs/2.0 ../wgsAll.xsd"
  service="WCS" version="2.0.0">
  <wcs:CoverageId>orto_pnoa</wcs:CoverageId>
<wcs:DimensionTrim>
  <wcs:Dimension>x</wcs:Dimension>
  <wcs:TrimLow>270500</wcs:TrimLow>
  <wcs:TrimHigh>270600</wcs:TrimHigh>
</wcs:DimensionTrim>
<wcs:DimensionTrim>
  <wcs:Dimension>y</wcs:Dimension>
  <wcs:TrimLow>4630000</wcs:TrimLow>
  <wcs:TrimHigh>4630100</wcs:TrimHigh>
</wcs:DimensionTrim>
</wcs:GetCoverage>
```

57

Figura 4.6 Ejemplo de petición WCS con recorte [IGN14].

```
<?xml version="1.0" encoding="UTF-8"?>
<wcs:GetCoverage xmlns:wcs="http://www.opengis.net/wcs/2.0"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wcs/2.0 ../wgsAll.xsd"
  service="WCS" version="2.0.0">
<wcs:CoverageId>orto_pnoa</wcs:CoverageId>
<wcs:DimensionSlice>
  <wcs:Dimension>x</wcs:Dimension>
  <wcs:SlicePoint>270500</wcs:SlicePoint>
</wcs:DimensionSlice>
<wcs:DimensionSlice>
  <wcs:Dimension>y</wcs:Dimension>
  <wcs:SlicePoint>4630000</wcs:SlicePoint>
</wcs:DimensionSlice>
</wcs:GetCoverage>
```

Figura 4.7 Ejemplo de petición WCS con rebanado [IGN14].

- Rebanar o separar –*slicing*–: se indica la dimensión y una posición –parámetro *slice*, debe estar dentro del dominio de la cobertura– a los que debe ajustarse la cobertura resultante. Se permite realizar la operación de rebanado en más de una dimensión simultáneamente –incluyendo una posición para cada una de las dimensiones–. La cobertura resultante tendrá el número de dimensiones de la original, menos las que se hayan indicado en la petición de rebanado. El ejemplo recogido en la Figura 4.7 muestra una aplicación en 2 dimensiones de este concepto.

Ambas opciones pueden incorporarse en una misma petición y siempre es necesario que figure al menos una de ellas.

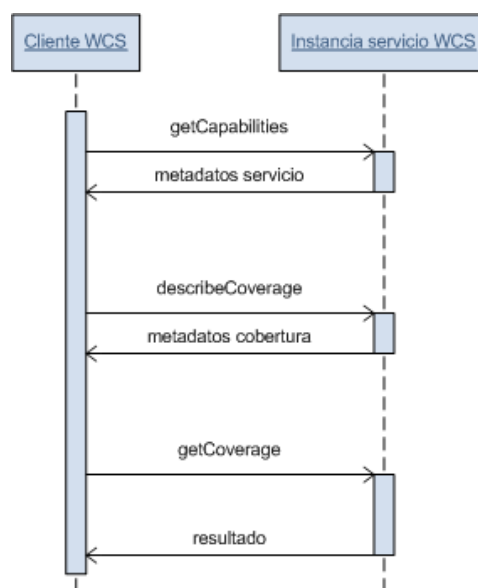


Figura 4.8 Interacciones entre un cliente y un servidor WCS [fuente: propia].

El diagrama *UML* de secuencia que se muestra en la Figura 4.8 expone las interacciones típicas que se producen entre un cliente y un servidor *WCS*. En el ejemplo, el cliente inicia el intercambio con el servidor realizando una petición para descubrir las capacidades del servicio y otra para conocer los detalles concretos de una determinada cobertura –estas peticiones solo son necesarias cuando se desconoce el servicio o bien cuando se tenga conocimiento de que ha ocurrido algún cambio en la interfaz del mismo–.

En el ejemplo, el cliente continúa su interacción generando una petición de obtención de un fragmento de cobertura, todo ello sin requerimientos de sincronización.

4.1.1.6. Servicio de descarga SOS

En las últimas décadas, se han manifestado numerosas iniciativas de carácter tecnológico que han permitido el desarrollo de sensores estáticos –ilustrados principalmente por aquellos que permiten la gestión de infraestructuras como vías de comunicación, recursos hídricos, etc., junto con los que se aglutinan entorno al paraguas de las ciudades inteligentes– y dinámicos –identificados especialmente con los sensores embarcados en satélites–.

Estos sensores permiten la obtención de extraordinarios volúmenes de información –intensidad del tráfico en un tramo concreto de una carretera, aforo de un embalse, ocupación de un contenedor de recogida de residuos urbanos, etc.– y la envían a los centros de gestión de dichos sensores, en los cuales se recolecta y se realizan los procesos oportunos con dicha información. Sin embargo, la heterogeneidad de estos sensores y de los sistemas de gestión ha provocado que los protocolos de comunicación utilizados por los distintos fabricantes fueran diferentes y esto imposibilitase la interoperabilidad tan necesaria en el ámbito tecnológico de este estudio. Por ello, el OGC decidió constituir un grupo de trabajo –*standards working group*– cuyo objetivo era y es el establecimiento de un marco de trabajo para la Habilitación de una Web de Sensores (SWE), entendiendo como tal redes de sensores accesibles vía Web y datos archivados sobre los mismos que pueden ser descubiertos y accedidos mediante la utilización de protocolos estándar y *APIs* –ver Figura 4.9– [Bot07].

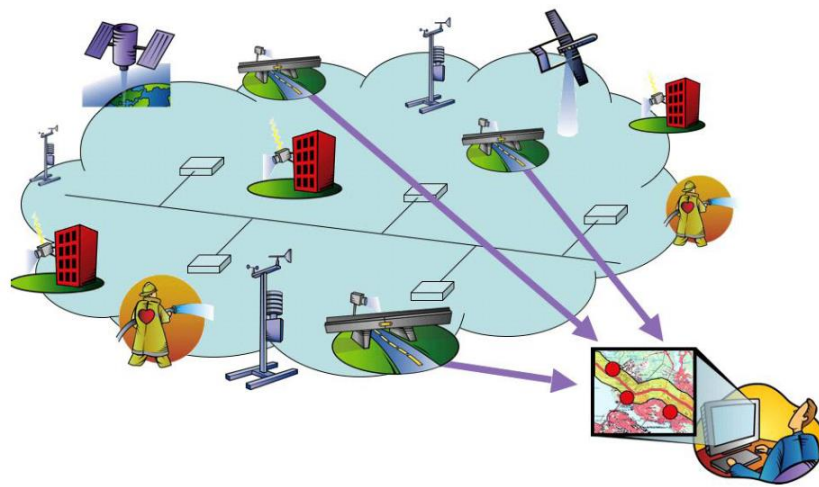


Figura 4.9 Concepto de Web de sensores [Bot07].

Dentro de este contexto se han establecido un conjunto de especificaciones para estandarizar los diferentes componentes que permiten regular toda la operativa necesaria en la gestión de una red de sensores. En la actualidad, OGC ha publicado los estándares que permiten:

- Obtener las características de los sensores⁵, así como de las observaciones por ellos capturadas, mediante la definición de la especificación *Sensor Observation Service (SOS)* [Bro12].
- Asociar el instrumento físico que captura la observación con el software embarcado en el sensor, para lo cual se especifica el protocolo *PUCK*, gracias al cual dicho software es habilitado para poder trabajar con cualquier instrumento físico que sea compatible con dicha especificación, independientemente del tipo de observación que realice [Ore11].

⁵ La especificación utiliza el término *procedure* extrapolando la aplicación de este estándar a cualquier método, algoritmo, instrumento, sensor o sistema que pueda utilizarse para realizar una observación. En este documento se asimilará por simplicidad al concepto de sensor.

- Conocer las tareas asignadas a un sensor, así como la realización de nuevos encargos determinando la viabilidad de los mismos, para lo cual se ha desarrollado la especificación *Sensor Planning Service (SPS)* [Ech11].
- Definir los ocho paquetes funcionales a utilizar en los diferentes servicios *SWE*, mediante la especificación *SWE Service Model* [Ech11b].
- Detallar la codificación de los contenidos de las observaciones y mediciones realizadas por distintas tipologías de sensores [Cox11].
- Definir procesos y componentes asociados con los mismos, encargados de la medición y posterior transformación de observaciones realizadas por sensores, mediante la especificación *Sensor Model Language (SensorML)*, permitiendo la interoperabilidad a nivel sintáctico y semántico [Bot07b].

Aparte de estos, OGC ha realizado prototipos de especificaciones para dar cobertura a la transferencia de datos en tiempo real, desde los sensores hasta el servidor, mediante la técnica de *streaming* – utilizando el lenguaje *Transducer Markup Language (TransducerTML o TML)*–, la publicación de un servicio de alertas –*Sensor Alert Service (SAS)*–, así como la definición de un servicio de notificaciones – *Web Notification Service (WNS)*– de mensajes generados por un servicio *SAS* o *SPS*.

Dentro del ámbito de este estudio, el estándar *SOS* habilita un servicio de descarga de información que, por tanto, entra dentro del conjunto de servicios a analizar. En el año 2007 el OGC publicó una primera versión de este estándar, del cual en 2012 se publicó la segunda versión –2.0–, que es la que será analizada en este documento. Esta especificación plantea la utilización de un conjunto de conceptos vinculados al de observación, los cuales es preciso detallar –ver Figura 4.10–:



Figura 4.10 Conceptos relacionados con una observación [OGC11c].

- Entidad de interés –*feature of interest*–: representación de un objeto del mundo real que dispone de una propiedad a observar. La entidad puede ser de dominio –entidad muestreada, por ejemplo un río– o de muestreo –entidad que realiza un muestreo, por ejemplo un punto de aforo en el que se estima el caudal de un río–.

- Propiedad observada –*observed property*–: descripción de la propiedad observada –salinidad del agua, velocidad del viento, etc.–. Puede hacer referencia a un concepto dentro de una ontología.
- Procedimiento –*procedure*–: instancia del proceso que ha generado la observación. Normalmente se corresponde con un sensor –o un conjunto de sensores–, pero también puede corresponderse con un cálculo, fase de un postproceso o simulación.
- Resultado –*result*–: valor resultante de una observación, pudiendo ser un valor escalar o una matriz compleja multidimensional.
- Tiempo del fenómeno –*phenomenon time*–: instante en el que se ha tomado la observación.
- Tiempo del resultado –*result time*–: instante en el que el resultado de la observación ha sido creado para su transmisión al servidor.
- Oferta de observaciones –*observation offering*–: es cada una de las capas en las que se agrupan las observaciones obtenidas de un sensor o sistema de sensores. Es interesante reseñar que un mismo sensor puede estar vinculado a varias ofertas de observaciones y, por tanto, algunas observaciones pueden estar presentes en distintas ofertas. De esta forma, se pueden establecer agrupaciones por criterios temáticos.

Entrando en el detalle de las operaciones que incorpora esta especificación, se definen las siguientes con carácter obligatorio, puesto que forman parte del núcleo del estándar:

- *GetCapabilities*: permite a los clientes obtener metadatos sobre las operaciones disponibles en el servicio SOS. Estos metadatos, encapsulados en un documento *XML*, ofrecen información sobre el proveedor del servicio, así como una descripción somera de las diferentes colecciones de observaciones agrupadas por cada uno de los sistemas de sensores que enlaza el servicio – sensor, propiedad observable, formato de descripción, tipo de observación, área observada, etc.–.
- *DescribeSensor*: permite a los clientes obtener una descripción detallada de los sensores de captación de observaciones asociados al servicio SOS para un determinado periodo de tiempo. Esta descripción, encapsulada también en un documento *XML* –se aboga la utilización del formato *SensorML*, aunque se admiten otros que aparezcan declarados en el descriptor del servicio–, recoge los aspectos regulados en [Ech11b] para *SWE*.
- *GetObservation*: permite a los clientes ejecutar una operación para recuperar información sobre una observación⁶, estructurada la misma según la especificación *OGC* sobre observaciones y mediciones [Cox11], aunque el servicio puede ofrecer otros formatos adicionalmente –tendrán que venir necesariamente referidos en el descriptor del servicio–. La determinación de la observación

⁶ Según el estándar *SOS* una observación proporciona un resultado cuyo valor es una estimación de la propiedad de una entidad de interés.

deseada puede realizarse especificando la entidad a la que pertenece la propiedad estimada en la misma, la propia propiedad y filtros de carácter temporal y espacial.

Aparte de éstas, el estándar recoge un conjunto de operaciones adicionales y optativas, agrupadas en tres extensiones diferentes:

- Operaciones avanzadas –*enhanced operations*–:
 - *GetObservationById*: provee acceso a la observación cuyo identificador coincide con el especificado en la observación. La respuesta obtenida es idéntica a la ofrecida como resultado de la invocación de la operación *GetObservation*.
 - *GetFeatureOfInterest*: provee acceso a ciertas entidades, cuyas propiedades han sido estimadas en las observaciones, mediante la indicación de ciertos filtros. Las entidades se describen según se dispone en [Ech11b].
- Transaccional –*transactional*–:
 - *InsertSensor*: permite registrar un nuevo sensor en el servidor SOS que provee el servicio. Aquellos servicios que tienen habilitada esta operación, dentro de su descriptor hacen referencia a los formatos que pueden utilizarse para dar de alta el nuevo sensor. Como respuesta, el servidor devuelve un identificador al recurso asignado al sensor en el que se almacenaran todas las colecciones de observaciones por éste enviadas –*observation offering*–.
 - *DeleteSensor*: permite eliminar un sensor del registro del servidor SOS que provee el servicio. El sensor en cuestión debe indicarse mediante su identificador y todas las observaciones vinculadas con él también serán eliminadas.
 - *InsertObservation*: permite a un cliente insertar nuevas observaciones para un sensor registrado en el sistema. Esta operación es la que, por ejemplo, utiliza el software embarcado en el sensor para transmitir información desde el mismo al servidor SOS.
- Gestión de resultados –*result handling*–:
 - *InsertResultTemplate*: en determinadas circunstancias en las que el ancho de banda disponible para la transmisión de datos entre el cliente embarcado en el sensor y el servidor SOS es limitado, es interesante disminuir el conjunto de datos a transmitir en cada envío. Para ello, SOS permite un mecanismo que evita tener que enviar todos los metadatos de una observación para cada medición que se quiera comunicar. De esta forma, se habilita la definición de una plantilla en la que se especifican los campos de información a enviar, denominándose al conjunto “resultado” –*result*–, así como el *observation offering* al que se asocia. Esta operación, por tanto, permite la inserción en el servidor SOS de una plantilla de las características referidas.

- *GetresultTemplate*: permite a un cliente obtener la plantilla utilizada para transmitir los datos desde un cliente embarcado, de conformidad con lo reseñado en la operación anterior.

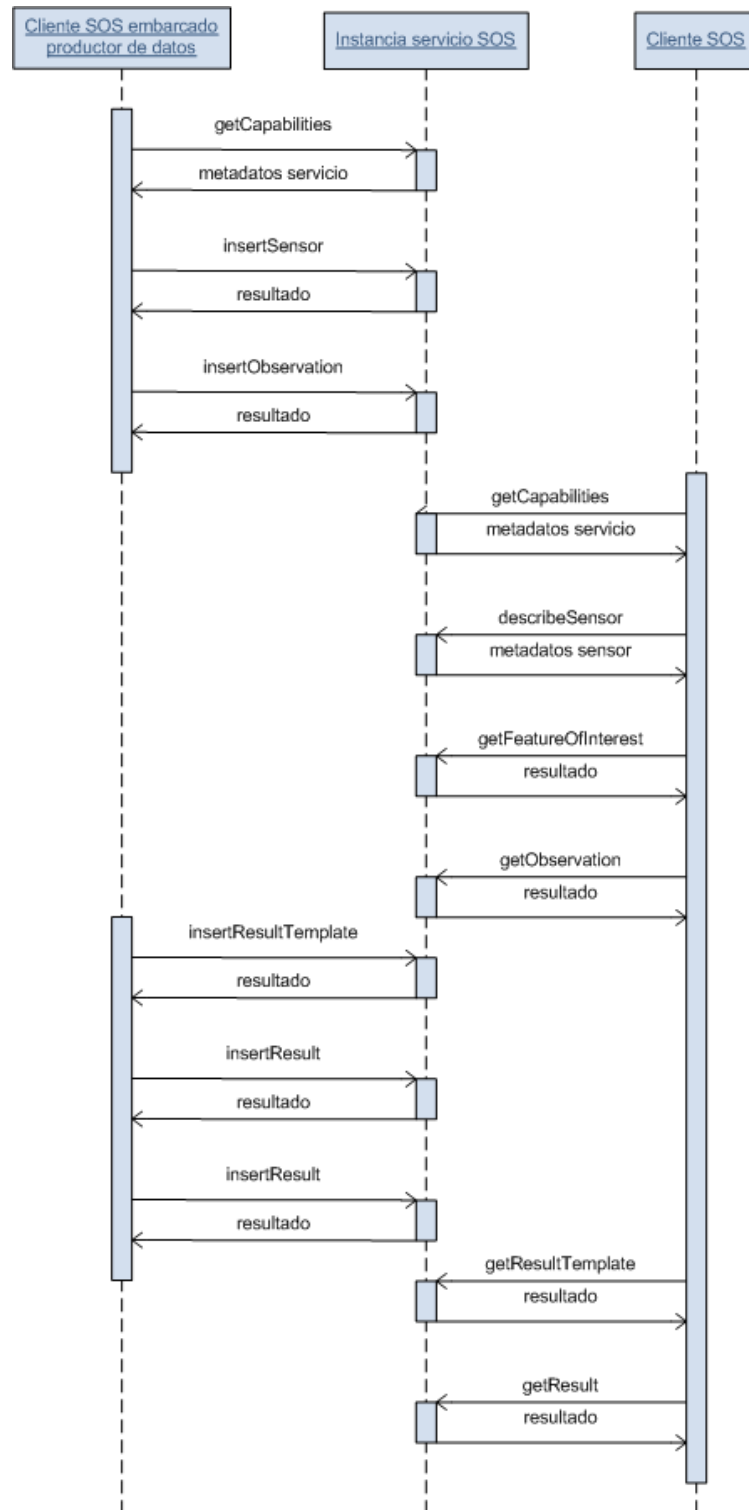


Figura 4.11 Interacciones entre clientes y un servidor SOS [fuente: propia].

- *InsertResult*: permite a un cliente embarcado comunicar a un servidor SOS cada uno de los fragmentos de información que formaran parte de la observación, siguiendo cada uno de ellos la definición mostrada en la plantilla que se adjunta a la petición.
- *GetResultTemplate*: permite a un cliente acceder a la plantilla utilizada por el sensor en cuestión para almacenar los resultados de las observaciones, de forma que éste pueda conocer la estructura de los datos para su posterior recuperación
- *GetResult*: permite a un cliente acceder al resultado de una observación, habiendo previamente recuperado la plantilla utilizada para su codificación –operación *GetResultTemplate*–, sin necesidad de acceder a todos los metadatos de la misma. En la petición debe indicarse, entre otros parámetros, la propiedad observada, la entidad de interés y el *observation offering*.

El diagrama *UML* de secuencia que se muestra en la Figura 4.11 expone las interacciones típicas que se producen entre un cliente y un servidor SOS. La secuencia muestra como en primer lugar el software cliente embarcado accede al descriptor del servicio SOS, para proceder posteriormente al registro del sensor en el que se ubica, dentro del servidor. Una vez hecho esto, el cliente embarcado ya puede comenzar a insertar observaciones en el recurso de almacenamiento que le ha sido asignado. Además, se muestra una segunda posibilidad de almacenamiento de las variables monitorizadas mediante la definición de una plantilla de resultados y el posterior almacenamiento de los mismos.

El cliente SOS consumidor de información, por su parte, procede a la descarga de información sobre el servicio, para un sensor concreto y, finalmente, sobre los datos registrados por el mismo –ya sean estas observaciones propiamente dichas o resultados–.

Advertir que las peticiones de ambas tipologías de clientes pueden intercalarse en el tiempo, aunque en el diagrama, para favorecer la legibilidad, no se hayan reflejado de tal forma.

4.1.1.7. Servicio de geoprocésamiento WPS

Esta tipología de servicios es clave dentro del planteamiento de una oferta de servicios que facilite hacer realidad los SIG distribuidos, puesto que permiten resolver cuestiones que van más allá de la mera consulta de IG o alfanumérica asociada, ofreciendo herramientas de análisis avanzado como la superposición topológica de diferentes capas de información o la geocodificación de direcciones postales.

Con la finalidad de regular el marco en el que se desarrolla la prestación de estos servicios y favorecer la interoperabilidad, OGC ha promulgado un estándar denominado WPS. Este estándar vio publicado un primer borrador de su especificación en enero de 2005 [Sch05], aunque no fue hasta junio de 2007 cuando fue finalmente presentado como estándar [Sch07], versión 1.0.0. En línea con lo recogido anteriormente, WPS provee acceso a los clientes, a través de una red, a modelos computacionales que tienen la característica de operar con datos referenciados espacialmente. Estos modelos pueden recoger cálculos de distinta complejidad, requiriendo para su funcionamiento de un conjunto de parámetros de entrada y ofreciendo un conjunto de resultados. El estándar, por tanto, no recoge qué procesos pueden realizarse, sino el marco formal en el que debe regularse la comunicación entre servidores de dichos servicios y clientes consumidores de los mismos.

WPS define los siguientes métodos u operaciones obligatorias:

- *GetCapabilities*: permite a los clientes obtener metadatos sobre el servicio ofrecido. Estos metadatos, encapsulados en un documento *XML*, ofrecen información sobre el proveedor del servicio, así como una descripción somera de los diferentes procesos que ofrece.
- *DescribeProcess*: permite a los clientes obtener una descripción detallada de uno o varios de los procesos ofrecidos por un servicio. Esta descripción, encapsulada también en un documento *XML*, incluye los parámetros de entrada y los resultados ofrecidos.
- *Execute*: permite a los clientes ejecutar un proceso concreto de los ofrecidos por un servidor a través del servicio, utilizando los valores de los parámetros de entrada que haya especificado el cliente y devolviendo el o los resultados derivados de la ejecución. Los valores de entrada pueden hacer referencia a datos que se encuentren ubicados localmente, o bien accesibles a través de la red utilizando algún geoservicio que permita la descarga de información –*WMS*, *WFS*, *WCS*, etc.–.

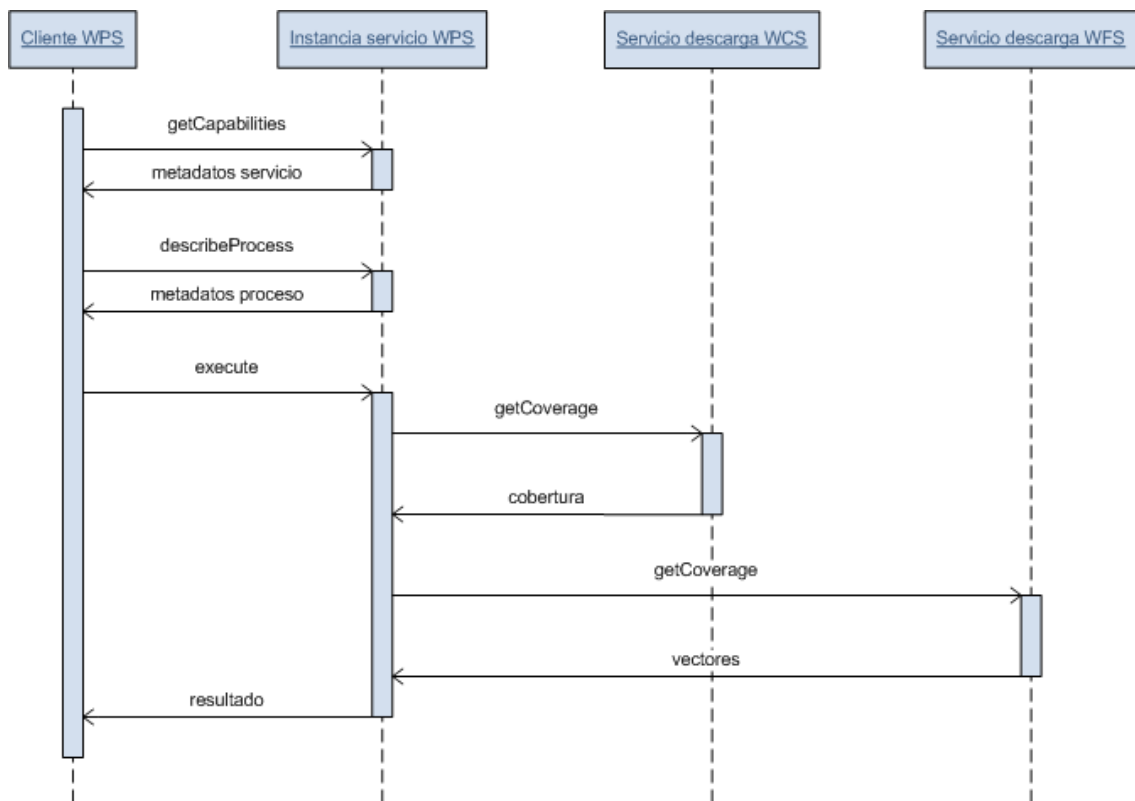


Figura 4.12 Interacciones entre un cliente y un servidor WPS [fuente: propia].

El diagrama *UML* de secuencia que se muestra en la Figura 4.12 expone las interacciones típicas que se producen entre un cliente y un servidor WPS. En el ejemplo, el cliente operando en modo síncrono, después de realizar las consultas oportunas para descubrir las capacidades del servicio y del proceso en cuestión –estas peticiones solo son necesarias cuando se desconoce el servicio y el proceso, o bien cuando se tenga conocimiento de que ha ocurrido algún cambio en la interfaz del servicio–, acaba

generando una petición de ejecución de un proceso, para el cual es preciso acceder a dos geoservicios adicionales de descarga –WFS y WCS– que permiten obtener los datos que debe procesar el servicio.

4.1.2 Infraestructuras de datos espaciales

El intenso trabajo realizado por el OGC a lo largo de toda su trayectoria ha conllevado la publicación de un número muy importante –y en continuo aumento– de WS de IG que han puesto de manifiesto un nuevo problema: cómo se puede tener conocimiento de la existencia de los mismos y cómo pueden utilizarse. Es decir, la problemática ya esbozada en este mismo documento cuando se hablaba de la creación de WS y de la necesidad de establecer mecanismos que faciliten el descubrimiento y la utilización de los mismos, hecho al que, entre otras opciones, intenta dar respuesta la Web semántica. Sin embargo, dentro del propio ámbito de la IG y en línea con el concepto de infraestructura que figura en el segundo objetivo estratégico del OGC, se acuña el término de Infraestructura de Datos Espaciales (IDE).

Según [Min14], se entiende por IDE “un sistema informático integrado por un conjunto de recursos – catálogos, servidores, programas, datos, aplicaciones, páginas Web, etc.– dedicados a gestionar información geográfica –mapas, ortofotoimágenes, imágenes de satélite, etc.– disponibles en Internet, que cumplen una serie de condiciones de interoperabilidad –normas, especificaciones, protocolos, interfaces, etc.– que permiten que un usuario, utilizando un simple navegador, pueda utilizarlos y combinarlos según sus necesidades”.

El establecimiento de una IDE, a nivel local, regional, estatal o global, requiere del acuerdo de los productores, integradores y usuarios de datos espaciales del ámbito territorial en el que se establece. Este acuerdo debe considerar también las IDE definidas, o en definición, en otros ámbitos territoriales superiores, con las cuales deberá converger.

En España la primera IDE fue desarrollada por el gobierno autonómico de Cataluña mediante el proyecto Infraestructura de Datos Espaciales de Cataluña (IDEC), que inició su andadura en 2002. Actualmente han proliferado las IDE correspondientes a la práctica totalidad de los niveles jerárquicos administrativos españoles –siendo la más representativa la Infraestructura de Datos Espaciales de España (IDEE)–, extendiéndose su uso a otros ámbitos, como ocurre en el proyecto hospedado por IDEC referente a la IDE sectorial del litoral catalán –IDE-Litoral–. Por supuesto, a nivel europeo, este movimiento también ha tenido frutos importantes, aunque muy dispares en función de cada país.

4.1.3 INSPIRE

El movimiento expuesto en el apartado anterior ha tenido, a partir del año 2004, un claro motor de expansión en el gobierno de la Unión Europea (UE) mediante la definición del proyecto *INSPIRE*, en otras palabras, la IDE de la UE.

4.1.3.1. Marco legislativo

INSPIRE se materializa en forma de directiva propuesta por la Comisión Europea en julio de 2004, con la finalidad de regular el marco legal que permita el establecimiento y la operación de una IDE en el ámbito

de los países miembros de la UE, tal y como aparece recogido en [NSD08]. El propósito de dicha infraestructura es el de dar cobertura a la definición, implementación, monitorización y evaluación de políticas comunitarias relacionadas con el medio ambiente a todos los niveles, europeo, nacional y local, además de ayudar a la publicación de información.

Para la consecución de este propósito, *INSPIRE* –nodo principal– debe basarse en las IDE de cada uno de los países miembro de la UE, actuando éstas como nodos secundarios. A su vez, dentro de cada estado, se dispondrán niveles jerárquicos adicionales de IDE. Sin embargo, para que cualquier IDE forme parte del paraguas *INSPIRE*, los elementos por ella gestionados han de ser: metadatos, capas temáticas de datos espaciales –tal y como se recogen en los anexos I, II y III de la directiva–, servicios de datos espaciales, servicios de red y tecnologías, términos de uso, mecanismos de coordinación y monitorización, procesos y procedimientos.

Los principios que guían *INSPIRE* son:

- Las IDE de los estados miembro deben estar diseñadas para asegurar que la información espacial se almacena, se publica y se mantiene en el nivel más apropiado. Esta regla persigue que el organismo que genera la información sea el encargado de administrarla.
- Es posible la combinación de datos espaciales, provenientes de diferentes fuentes en todo el ámbito de la UE, de forma consistente y compartida entre diferentes usuarios y aplicaciones.
- Es posible que la información espacial confeccionada por una autoridad pública sea compartida con todos los distintos niveles de autoridades públicas.
- Los datos espaciales deben estar disponibles bajo condiciones que no deben ser excesivamente restrictivas para su uso extensivo.
- Es fácil descubrir la información espacial disponible, evaluar su adecuación para la finalidad que se persiga, así como disponer de las condiciones aplicables a su utilización.

El proceso de construcción de *INSPIRE* comienza su andadura el 15 de mayo de 2007 con la aprobación de la directiva por parte del Parlamento Europeo. A partir de ese momento, los países miembro dispusieron de 2 años para trasponer los requerimientos impuestos por la directiva a la legislación de cada uno de los países. Por otra parte, para asegurar que las distintas IDE son compatibles y utilizables en todo el ámbito comunitario, la directiva obliga a la adopción de un conjunto de reglas de implementación –*implementing rules*– que son fruto de la Comisión Europea.

4.1.3.2. Arquitectura de servicios en red

Desde un punto de vista arquitectónico, *INSPIRE* se materializa en forma de la arquitectura de servicios en red –*network services architecture*– presentada en la Figura 4.13. En el modelo planteado pueden diferenciarse los siguientes componentes principales:

- Conjuntos de datos espaciales –DT DS–: componente nuclear en el que reside toda la información geográfica –y alfanumérica– accesible mediante la infraestructura. Estos conjuntos de datos pueden no ser compatibles con el modelo de datos *INSPIRE* –recogidos, como se ha citado anteriormente, en los anexos I, II y III de la directiva–, siendo en este segundo caso necesario el acoplamiento de un componente adicional de armonización. Los restantes componentes serán los encargados de facilitar el acceso a esta información, según los principios definidos en *INSPIRE*.

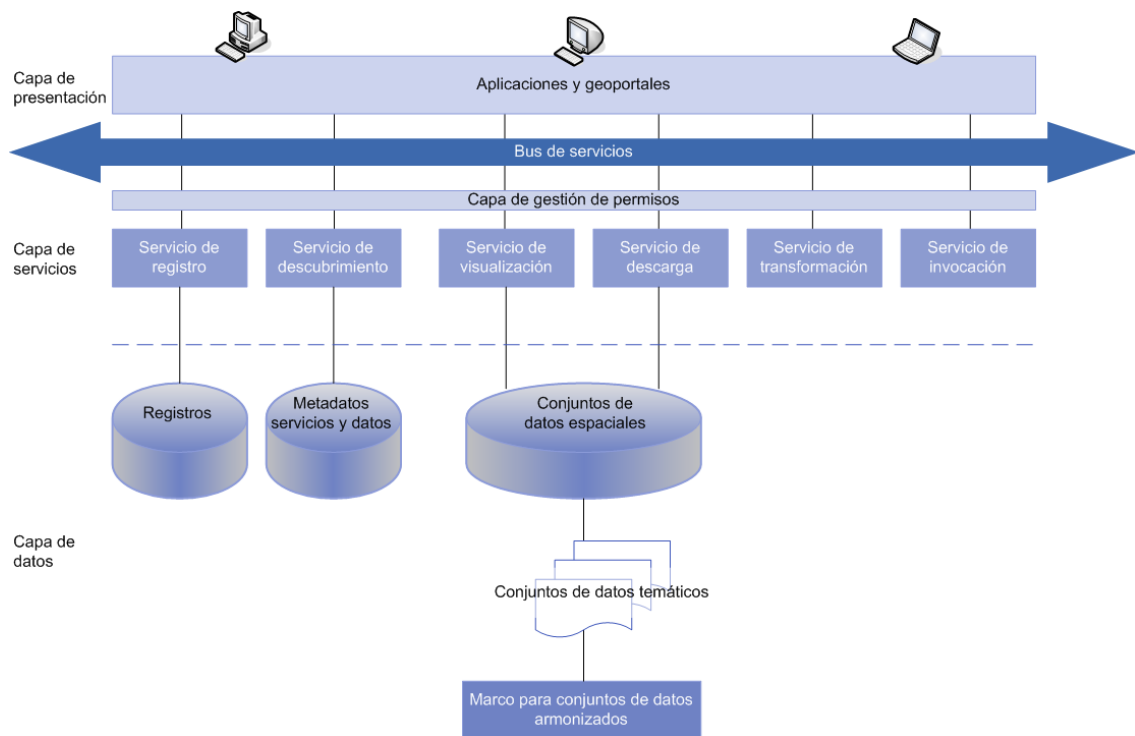


Figura 4.13 Arquitectura técnica de *INSPIRE* [NSD08].

- Conjuntos de metadatos –DT MD–: componente que aglutina los conjuntos de metadatos que permiten describir los diferentes conjuntos de datos espaciales, información clave a la hora de habilitar mecanismos de publicación y búsqueda de información. Se entiende por metadato a la información y documentación que describe un recurso –datos, servicios, etc.–, facilitando su descubrimiento y utilización.
- Conjuntos de servicios: agrupados en función de las distintas tipologías de servicios que define *INSPIRE* y que serán analizadas en el próximo subapartado. Estos servicios son accesibles a través del bus de servicios, mediante la gestión realizada por la capa de administración de derechos –*rights management layer*–. Estos tres elementos forman el componente *DT NS*.

Cabe reseñar que el proceso de registro de los servicios, siguiendo los protocolos definidos en SOA, se lleva a cabo mediante el componente denominado servicio de registro, el cual se apoya a su vez en una base de datos de registros.

- Aplicaciones y geoportales: interfaces gráficas de usuario que permiten a los usuarios acceder a los distintos tipos de servicios a través del bus de servicios. Reseñar que en este ámbito de estudio, un geoportal no es más que un portal Web en el que se administran contenidos geográficos.

4.1.3.3. Tipologías de servicios

Centrando el estudio en las tipologías de servicios, *INSPIRE* recoge las siguientes agrupaciones [NSD08]:

- Servicios de descubrimiento –*discovery services*–: son aquéllos que posibilitan a los clientes la búsqueda de conjuntos de datos y servicios espaciales en función del contenido de los metadatos registrados de cada uno de ellos.

Dentro de la comunidad geográfica, históricamente, se han otorgado diferentes nombres a los instrumentos que realizan las búsquedas citadas: *Catalogue Services*, *Spatial Data Directory*, *Clearinghouse*, *Geographic Catalogue* y *Geodata Discovery Service*.

- Servicios de visualización –*view services*–: son aquéllos que, como mínimo, facilitan la visualización, navegación, *zoom* de aproximación y alejamiento, desplazamiento y superposición de conjuntos de datos espaciales, la visualización de una leyenda descriptiva del contenido servido y la consulta de cualquier metadato relevante. Cuando el acceso a uno de estos servicios implique el pago previo de una cantidad económica, el propietario de la IDE debe encargarse de habilitar la plataforma de comercio electrónico pertinente.

El proceso de implementación de un servicio de visualización implica considerar aspectos como los siguientes: naturaleza de los metadatos, utilización de un sistema de coordenadas común, información sobre la dimensión temporal, simbología aplicada a la representación, correspondencia entre las capas del servicio y los temas *INSPIRE*, localización idiomática, etc.

Los conjuntos de datos sujetos a opciones de visualización deben englobarse en alguna de las categorías recogidas en los anexos I, II y III de la directiva. Para la materialización de este tipo de servicios, *INSPIRE* toma como referencia las especificaciones realizadas en la versión 1.3.0 del estándar *WMS* –también recogido por la *ISO* con el estándar 19128– y en la 1.0.0 del estándar *WMTS*, ambos promulgados por el *OGC* [IOC11].

- Servicios de descarga –*download services*–: son aquéllos que facilitan la realización de copias, totales o parciales, de los datos publicados por el servicio. Estas copias deben poder ser descargadas por el cliente y, en caso de implicar un pago previo, al igual que en el caso de los servicios de visualización, es preciso disponer de una plataforma de comercio electrónico.

Los conjuntos de datos sujetos a opciones de descarga deben englobarse en alguna de las categorías recogidas en los anexos I, II y III de la directiva. Además es preciso que dispongan de metadatos descriptivos, deben actualizarse con una periodicidad acorde con el artículo 5, y deben

ser interoperables y armonizados según lo dispuesto en los artículos 7 al 10. En este sentido, el servicio de descarga puede implicar la ejecución de algún proceso intermedio que transforme los datos originales para cumplir los requerimientos indicados.

En la actualidad, *INSPIRE* solo ha desarrollado directrices para la implementación de servicios de descarga referentes a conjuntos temáticos englobados en el anexo I, dejando para el futuro las implementaciones correspondientes a los restantes anexos. Según se recoge en [IOC13], *INSPIRE* determina dos tipologías de servicios de descarga:

- Conjunto de datos predefinido *–pre-defined dataset–*: servicios que no permiten especificar filtros que determinen la parte del conjunto que es requerida, de forma que solo pueda ser descargable en su totalidad o en partes previamente definidas. En este caso, la directiva contempla el acceso mediante el formato de sindicación *Atom* o gracias a la extensión del estándar *WFS* –también recogido por la *ISO* con el estándar 19142–, combinado con el que regula la especificación de filtros, *Filter Encoding*, también promulgado por el *OGC* –estándar *ISO* 19143–.
- Acceso directo *–direct access–*: servicios que, en contraposición a lo referido anteriormente, permiten a los clientes especificar mediante filtros el subconjunto de datos a descargar, de forma completamente dinámica. En este sentido, la directiva contempla la utilización de la extensión del estándar *WFS* y *Filter Encoding*, referidos anteriormente.

70

Actualmente, al no haberse regulado la implementación de los otros dos anexos, no se aplica el estándar *OGC WCS* como servicio de descarga, aplicable exclusivamente a conjuntos de datos ráster.

- Servicios de transformación *–transformation services–*: son aquéllos que habilitan transformaciones a los conjuntos de datos con la finalidad de conseguir la interoperabilidad. Se trata, por tanto, de servicios auxiliares, utilizados por otros para la consecución de las especificaciones reguladas en *INSPIRE*, aunque también pueden ser invocados directamente –por ejemplo, se podría citar un servicio que permita la transformación entre diferentes sistemas de coordenadas–.
- Servicios de invocación de datos espaciales *–invoke spatial data services–*: son aquéllos que permiten la invocación de servicios espaciales que no son directamente invocables por no cumplir con las características exigidas por *INSPIRE* –debe disponer de una interfaz bien definida y documentada de acuerdo a los estándares *ISO* y *OGC*–. Para estos servicios, un servicio de invocación *INSPIRE* actúa como fachada, definiendo los datos de entrada y de salida que se esperan de un servicio concreto, facilitando además el encadenamiento de diferentes servicios. En este último caso, la descripción del encadenamiento ha de realizarse en lenguaje *XML*.

La directiva *INSPIRE* aconseja la implementación de las diferentes tipologías de servicios mediante la utilización de los estándares existentes, siendo los publicados por *OGC* los más representativos. Aunque los estándares *OGC* soportan una mezcla de protocolos y codificaciones, solo la

correspondiente a la utilización del protocolo *SOAP* coincide con la recomendación realizada por el *W3C* para el protocolo de envío de mensajes en el ámbito de los *WS*. En este sentido, *INSPIRE* determina que *SOAP* ha de ser el protocolo utilizado en la mensajería propia de los *WS*, eliminando de esta forma la problemática propia de la mezcla de diferentes tecnologías.

La decisión de adoptar *SOAP* como protocolo de comunicación también cuenta con otros argumentos favorables:

- Es el estándar dentro de *SOA* para la comunicación entre *WS*.
- Favorece la integración completa de los diferentes entornos de desarrollo –independientemente de la plataforma y la tecnología empleadas–.
- Es extensible, permitiendo la incorporación de funcionalidades propias de ámbitos de gestión, como por ejemplo los derechos de acceso.
- Permite el encadenamiento de servicios y la automatización de los flujos de trabajo correspondientes a los procesos de negocio.

Dadas las circunstancias, aquellos servicios *OGC* que se deseen incorporar a *INSPIRE* deberán disponer de una pasarela o fachada –*facade*– *SOAP*, tal y como se muestra en la Figura 4.14.

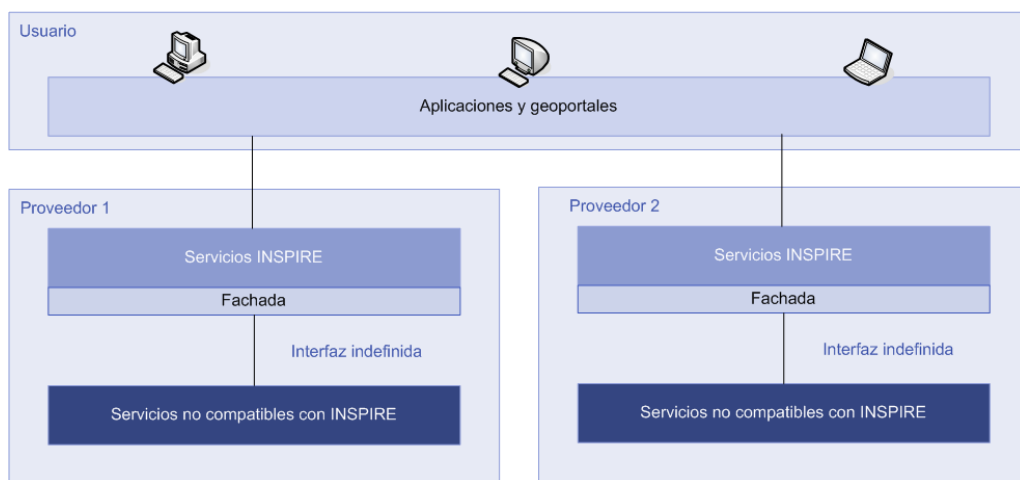


Figura 4.14 Habilidad de servicios no compatibles con *INSPIRE* [NSD08].

Es importante reseñar que a partir de 2006, todas las especificaciones de estándares reguladas por *OGC* –correspondientes a nuevos estándares o nuevas versiones de los ya existentes– deben permitir la interacción vía *SOAP*.

4.1.3.4. Gestión de servicios

Se analizan en este punto dos aspectos relativos a la gestión de los servicios que resultan de vital importancia para la consecución de los fines planteados en *INSPIRE*: metodologías empleadas para la gestión de la publicación, búsqueda y conexión, así como para la gestión de los derechos de acceso.

En lo tocante al primer punto, el planteamiento adoptado no es otro que el marco básico de funcionamiento para la gestión de servicios en *SOA*, analizado en [Gar08] e ilustrado previamente en este documento en la Figura 3.2. Este marco, tal y como ya se puso de manifiesto, se basa en la presencia de tres módulos –cliente, suministrador y registro– y en la ejecución de tres fases –publicación y suscripción, búsqueda y enlace–.

Sin embargo, enlazando con el segundo aspecto que abarca el ámbito de la gestión de los derechos de acceso, el modelo de tres fases planteado se presenta insuficiente, siendo preciso añadir un cuarto paso situado entre la publicación y el enlace: aceptación. La gestión de los derechos de acceso será una de las funciones a desempeñar dentro de la arquitectura del sistema; en los ejemplos mostrados de arquitecturas *SOA* y *ROA* esta problemática se aborda dentro de la capa de gestión de QoS.

4.1.3.5. Calidad de los servicios

INSPIRE propone un conjunto de requerimientos para delimitar los aspectos que permiten cuantificar la calidad de los servicios publicados [NSD08]. Son los siguientes:

- Rendimiento –*performance*–: hace referencia al tiempo de respuesta, el cuál se contabiliza como el intervalo transcurrido entre la recepción de la petición por parte del servidor y el envío del primer byte de la respuesta por parte del mismo. Dicho tiempo debe garantizarse para un volumen de peticiones que no supere la capacidad máxima del sistema.
- Capacidad –*capacity*–: se determina mediante el número de peticiones que son enviadas a un servicio en un lapso de tiempo acordado.
- Disponibilidad –*availability*–: determina la probabilidad de que un sistema se mantenga operativo.
- Fiabilidad –*reliability*–: representa la habilidad del servicio para ejecutar sus funciones en un marco e intervalo de tiempo determinados.
- Seguridad –*security*–: hace referencia al aspecto de calidad que recoge la confidencialidad y el no repudio mediante autenticación, la encriptación de mensajes y la habilitación del control de accesos.
- Cumplimiento –*compliance*–: nuevo aspecto de calidad referente a la adecuación del servicio a las reglas, leyes, estándares y acuerdos *SLA*.

La propia implementación de la directiva aporta documentación concreta sobre los valores mínimos aceptables para los tres primeros parámetros de calidad reseñados y para cada una de las tipologías de

servicios. Así, a modo de ejemplo, la guía de implementación de servicios de visualización *INSPIRE* [IOC11] marca las siguientes cifras:

- Disponibilidad: el servicio debe estar operativo como mínimo el 99% del tiempo. La Tabla 4.7 muestra los tiempos en los que un servicio no estará operativo para diferentes disponibilidades y periodos, aunque los autores del estudio recomiendan utilizar la referencia anual.
- Rendimiento: para una imagen de 470 Kb –equivalente a 800x600 píxeles con una profundidad de color de 8 bits–, el tiempo de respuesta para iniciar la contestación a una petición *WMS GetMap*⁷ ha de ser de 5 segundos como máximo en condiciones normales –equivalentes a periodos no coincidentes con picos de carga del servidor que aloja el servicio, indicándose como referencia un 90% del tiempo–.
- Capacidad: el mínimo número de peticiones al servicio atendidas concurrentemente debe ser de 20 por segundo durante un periodo de 1 minuto, siendo cada una de ellas equivalente al tamaño de referencia especificado para el parámetro de rendimiento.

El cumplimiento de estos parámetros definitorios de la QoS es un aspecto innegociable, debiendo dotarse a las plataformas que integren los servicios de recursos suficientes, como mínimo, garantizar estos resultados.

% Disponibilidad	Máximo tiempo de interrupción		
	Semanal	Mensual	Anual
98	3.4 h	14.55 h	7.27 d
98.6	2.4 h	10.19 h	5.09 d
99	1.7 h	7.27 h	3.63 d
99.5	0.8 h	3.64 h	1.82 d
99.9	10 min	0.73 h	8.73 h
99.99	1 min	4 min	52 min
99.999	6 s	26 s	5 min

Tabla 4.7 Tiempo de inoperatividad para diferentes disponibilidades y periodos [IOC11b].

⁷ El estándar *OGC WMS* es el adoptado, junto con *WMTS*, por *INSPIRE* para la publicación de servicios de visualización. Este estándar dispone de varios métodos u operaciones, siendo la petición *GetMap* la utilizada para solicitar un fragmento del mapa de interés en forma de imagen.

4.1.3.6. Convergencia tecnológica

La propia definición de IDE recogida en [Min14], pone de manifiesto que conceptualmente una infraestructura de estas características debe basarse en una arquitectura tecnológica que ofrezca los recursos necesarios para la gestión de la publicación de servicios, la búsqueda y descubrimiento, así como su utilización.

Globalmente, la arquitectura que ha ido ganando más aceptación en los entornos de empresariales orientados a la definición de procesos de negocio basados en la utilización de servicios ha sido SOA, tal y como aparece recogido en [Gar08]. Por tanto, parece lógico pensar que esta arquitectura es la idónea para la implementación de una IDE, hecho que se corrobora en numerosos artículos, entre los que pueden destacarse [Gha10] [Dia10]. Cabe aceptar como un hecho irrefutable que las IDE se basan en la utilización de WS y está es la forma mayoritariamente adoptada para forjar una SOA en la actualidad.

Una vez se acuerda la relación entre IDE y SOA y dado que *INSPIRE* implica el intento de establecer una IDE de los países miembro de la UE, es fácil entender porqué dentro de *INSPIRE* se ha optado por la implantación de este modelo arquitectónico.

Sin embargo, pese a lo que se acaba de acreditar, también es necesario destacar que algunos autores, como los de [Flo09] son partidarios de adoptar ROA como principal alternativa a SOA en cuanto a la definición de patrones metodológicos que favorecen la creación de sistemas distribuidos, puesto que su definición se sustenta en REST, una interfaz web simple basada en el protocolo HTTP, sin las abstracciones adicionales propias de los protocolos basados en el intercambio de mensajes como SOAP. Los WS basados en REST, denominados con el adjetivo “RESTful”, son una alternativa a los WS propios de SOA que también han sido acuñados con el apelativo de “Big Web Services”.

Por tanto, el paso que se acomete en el próximo apartado es el de analizar los estándares en los que se basan las distintas tipologías de servicios geográficos.

4.1.3.7. Grado de implantación

En línea con lo recogido en 4.1.3.6, el ámbito de las IDE ejemplifica la adopción de la arquitectura SOA para facilitar el acceso a datos y servicios heterogéneos y distribuidos, mediante la observación de un conjunto de especificaciones estándares. Este marco tecnológico, combinado con la existencia de una jerarquía de nodos interconectados, plantea un escenario teórico en el que se favorece la integración e interoperabilidad entre clientes y servicios. De esta forma, siguiendo los principios establecidos por *INSPIRE* y recogidos en 4.1.3.1, un cliente que desea descubrir qué información o servicios existen sobre un fenómeno concreto en un ámbito geográfico determinado, podría realizar la búsqueda en cualquiera de los nodos de la red y esta consulta se propagaría por todos los niveles jerárquicos restantes.

Sin embargo, la realidad muestra un escenario bastante distante de la concepción teórica. Existen numerosos nodos IDE implementados en todo el mundo, pero la mayoría de ellos adolecen de ciertas deficiencias en lo que respecta a su acoplamiento dentro de la red que los conecta.

A título ilustrativo, dentro de los niveles jerárquicos circunscritos al ámbito español, parecería lógico que la búsqueda en el catálogo de la IDEE de un servicio WMS que publique cartografía topográfica sobre

una zona concreta de Ourense, permitiera localizar cualquier recurso de esas características independientemente de que el mismo sea publicado por la propia IDEE o por la Infraestructura de Datos Espaciales de Galicia (IDEG). Veamos unos ejemplos. Se procederá a buscar seis topónimos diferentes en el catálogo de IDEG y en el de IDEE, especificando en este último que se quieren filtrar los resultados y considerar únicamente los registros que provienen de IDEG. La Tabla 4.8 muestra los resultados. Como puede observarse, la casuística es muy variada, aunque queda claro que se está produciendo una de las siguientes situaciones:

Topónimo	IDEE	IDEG
Castro Caldelas	1	1
Ferrol	2	2
Lugo	2	14
Ourense	40	43
Santiago de Compostela	1	265
Vigo	3	3

Tabla 4.8 Resultados de búsqueda de conjuntos de datos en IDEE e IDEG⁸.

- Ambos catálogos realizan búsquedas con filtros heterogéneos, es decir, no realizan la búsqueda del patrón especificado en los mismos conjuntos de metadatos.
- El catálogo de IDEE tiene una réplica del catálogo de IDEG y no existe una sincronización entre ambos catálogos.
- El catálogo de IDEE lanza la consulta al de IDEG y, debido a anomalías en la catalogación de los conjuntos de datos mediante los metadatos correspondientes, el catálogo de IDEG no devuelve los mismos resultados que cuando la búsqueda se efectúa directamente en el mismo.

En cualquier de los casos estamos ante una realidad que choca frontalmente con lo promulgado conceptualmente por las IDE y legalmente por INSPIRE. Por otro lado, si realizamos en ambos catálogos una búsqueda de diferentes tipologías de geoservicios OGC –en el caso de la IDEE se especifica que la búsqueda se haga contra el catálogo de Galicia, correspondiente a la IDEG–, obtendríamos el resultado mostrado en la Tabla 4.9.

Tipología de servicio	IDEE	IDEG
WMS	0	74
WFS	0	0
WCS	0	0

Tabla 4.9 Resultados de búsqueda de servicios en IDEE e IDEG⁹.

⁸ Consulta realizada el 16 de mayo de 2014.

Es fácil corroborar que sigue observándose el mismo problema, con el agravante de que en este caso, ni tan siquiera IDEG publica a través del catálogo todos los servicios que posee y publica.

Dentro de la Web de la IDEG, en uno de sus apartados se publica una relación de los 4 servicios de tipología WFS gestionados desde el ámbito de la Xunta de Galicia, tal y como se puede observar en la Figura 4.15 –<http://sitga.xunta.es/sitganet/catalogo/inicio.html?lang=es>–. En cambio, en el catálogo de la propia IDE, si se realiza una búsqueda de servicios WFS, no se obtiene ningún resultado –<http://xeocatalogo.xunta.es/geonetwork/srv/gl/main.home>–.

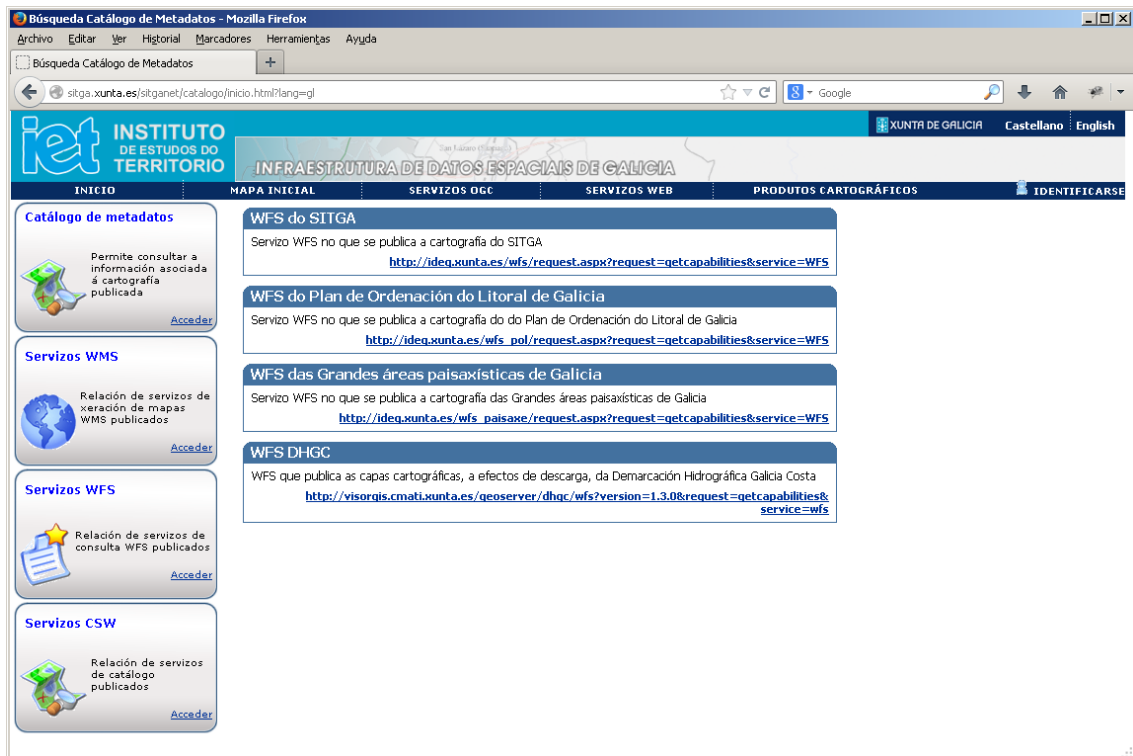


Figura 4.15 Servicios WFS publicados por IDEG [IET14]¹⁰.

Topónimo	INSPIRE	IDEE
Barcelona	4752	3255
Madrid	602	62
Ourense	329	47 ¹¹

Tabla 4.10 Resultados de búsqueda de conjuntos de datos en INSPIRE e IDEE¹².

⁹ Consulta realizada el 16 de mayo de 2014.

¹⁰ Consulta realizada el 16 de mayo de 2014.

¹¹ En este caso no se ha restringido la búsqueda en IDEE al catálogo de IDEG, de ahí que aparezcan cuatro resultados más de los mostrados en la Tabla 4.8. Consulta realizada el 16 de mayo de 2014.

Pero véase ahora en la Tabla 4.10 lo que ocurre si se sube un eslabón en el árbol jerárquico de las IDE europeas y se comparan los resultados obtenidos consultando el catálogo de *INSPIRE* y el de IDEE. Como se puede colegir, las discrepancias identificadas en la comparativa entre IDEE e IDEG son extrapolables también a este nivel. Por tanto, estamos ante una problemática que afecta de forma transversal a toda la estructura jerárquica y que acarrea una merma en la calidad de la conectividad entre los distintos catálogos.

Entre las causas principales de esta situación, cabe destacar las dos siguientes:

- Generación de metadatos eminentemente manual: muchos organismos productores de cartografía o publicadores de servicios no contemplan en sus cadenas de producción y/o publicación la generación de los metadatos que permitirán identificar estos recursos en los catálogos de las IDE. Este hecho suele venir justificado por el coste que supone la introducción de estos mecanismos en dichos procesos, aunque este argumento cae por su propio peso si se tiene en cuenta el impacto económico que supone la creación manual de los metadatos. Esto último, en el supuesto de que llegue a realizarse –en muchos casos, simplemente no se acaban generando–, implica además una falta de homogeneidad en la información incorporada en los metadatos y una mayor presencia de errores.
- Complejidad de los estándares: la principal virtud de un estándar es precisamente la determinación de un modelo de comunicación universal dentro de un dominio concreto. Sin embargo, esta propia universalidad puede dar pie, como efecto secundario, a un grado de complejidad que puede resultar contraproducente. En el ámbito de las IDE cabe diferenciar entre los estándares utilizados para la definición de los metadatos de los recursos –*ISO 19115* para la descripción de información geográfica e *ISO 19119* para la descripción de *WS* geográficos, así como los diferentes perfiles derivados de ambos– y el utilizado para la consulta y actualización de contenidos de los catálogos, así como para la conectividad entre los mismos, que es el *Catalogue Service for the Web (CSW)*, promulgado por el OGC. En referencia explícita a la complejidad, es especialmente recomendable el estudio realizado en [Gra12b] sobre este atributo en los servicios OGC, partiendo del análisis de sus esquemas.

Por otro lado, otra de las consecuencias motivadas por esta situación es la escasa participación en la publicación de recursos en los catálogos de organismos que no sean productores oficiales, en lo que se conoce como una aproximación de arriba abajo –*top-down*–. En estas circunstancias, los restantes actores que también generan contenidos o servicios de carácter geográfico, desempeñan un papel secundario o incluso irrelevante a efectos de las IDE. Para facilitar la participación de estos otros actores, han surgido iniciativas como la recogida en [Dia11] que definen un conjunto de herramientas marco.

¹² Consulta realizada el 16 de mayo de 2014. En la búsqueda en el catálogo *INSPIRE* se ha indicado que solo se recuperen aquellos recursos que provienen de catálogos en España.

4.2 Escenario de implantación

Se ha optado por escoger una agencia cartográfica oficial como entorno en el que circunscribir el estudio. El rol de estos organismos ha ido cambiando con el paso del tiempo, en un inicio su cometido era la publicación impresa de las diferentes series cartográficas que cubrían el ámbito geográfico bajo su responsabilidad. Pero posteriormente, la necesidad de disponer de acceso en tiempo real a la última información disponible hizo que las opciones tradicionales –compra de los mapas impresos y, más recientemente, descarga de la IG en formato digital– hayan ido perdiendo adeptos frente a los servicios de acceso en línea, acuñados como “geoservicios”.

Además, en los últimos tiempos, se pone de manifiesto la necesidad de acceder no solo a los datos, sino también a procesos sobre los mismos cuya ejecución en la vertiente cliente implicaría disponer de herramientas específicas y conocimientos adecuados para utilizarlas.

Centrada la problemática y antes de pasar a la definición de una arquitectura para la implementación de SIG distribuidos, es necesario plantear un escenario de aplicación concreto en el que se va a centrar la investigación. En este sentido, se ha optado por el Instituto Cartográfico y Geológico de Cataluña (ICGC). La elección de esta institución no es casual. Aunque se podría haber encontrado algún otro ejemplo, el ICGC es una institución que ha mostrado un interés muy grande por la difusión gratuita¹³ de la mayoría de sus productos cartográficos. Este hecho se ha acompañado de una política activa de creación de geoservicios que permiten el acceso a esos mismos productos desde plataformas compatibles con los estándares OGC WMS y WCS, aparte de otros que no tienen interfaces estándar. Además, a través del Centro de Soporte a la IDEC (CS IDEC), el ICGC ha realizado una tarea pedagógica muy importante de difusión de los estándares de interoperabilidad, reconocida por las instancias europeas que gestionan INSPIRE.

En cualquier caso, los objetivos planteados para este organismo son extrapolables a otras organizaciones homólogas. Éstas deben ser capaces de convertirse en motores de desarrollo de la sociedad en términos geoespaciales, proveyendo los recursos necesarios en esta materia.

4.2.1 Geoservicios publicados

La cartera de productos del ICGC está formada por un conjunto de series cartográficas y geológicas que cubren el territorio catalán –total o parcialmente– de forma sistemática, lo cual implica que están sometidas a un proceso de actualización continuado –cuya frecuencia varía en función del producto–, aparte de otros productos que no están sujetos a esta sistematicidad y que obedecen a proyectos específicos que no tienen porqué tener una continuidad en el tiempo.

El anexo que se incluye al final de esta memoria muestra un listado de los diferentes geoservicios publicados por el ICGC –ver Tabla 11.1–, gracias a los cuales los usuarios pueden acceder en línea a los

¹³ Esta distribución está sujeta a la observación de unas condiciones que, *grosso modo*, impiden la comercialización de los productos por terceros.

- Presencia de una interfaz homogeneizadora que intenta soslayar algunos de los aspectos operados por el ausente bus de servicios, concretamente el de la realización de transformaciones y un mínimo desacoplamiento entre las interfaces expuestas a los usuarios y las propias de los servicios subyacentes. Sin embargo, esta interfaz no actúa como frontal en todos los casos, puesto que existen ciertos servicios que son accedidos de forma directa.

El escenario planteado pone de manifiesto limitaciones en cuanto a la accesibilidad de los servicios, teniendo en cuenta las interfaces publicadas –algunas de ellas propietarias como las de los servicios *REST* publicados mediante *ESRI ArcGIS Server* o las de los servicios ICGC publicados mediante componentes desarrollados internamente– y la inexistencia de componentes orientados a la composición u orquestación de servicios.

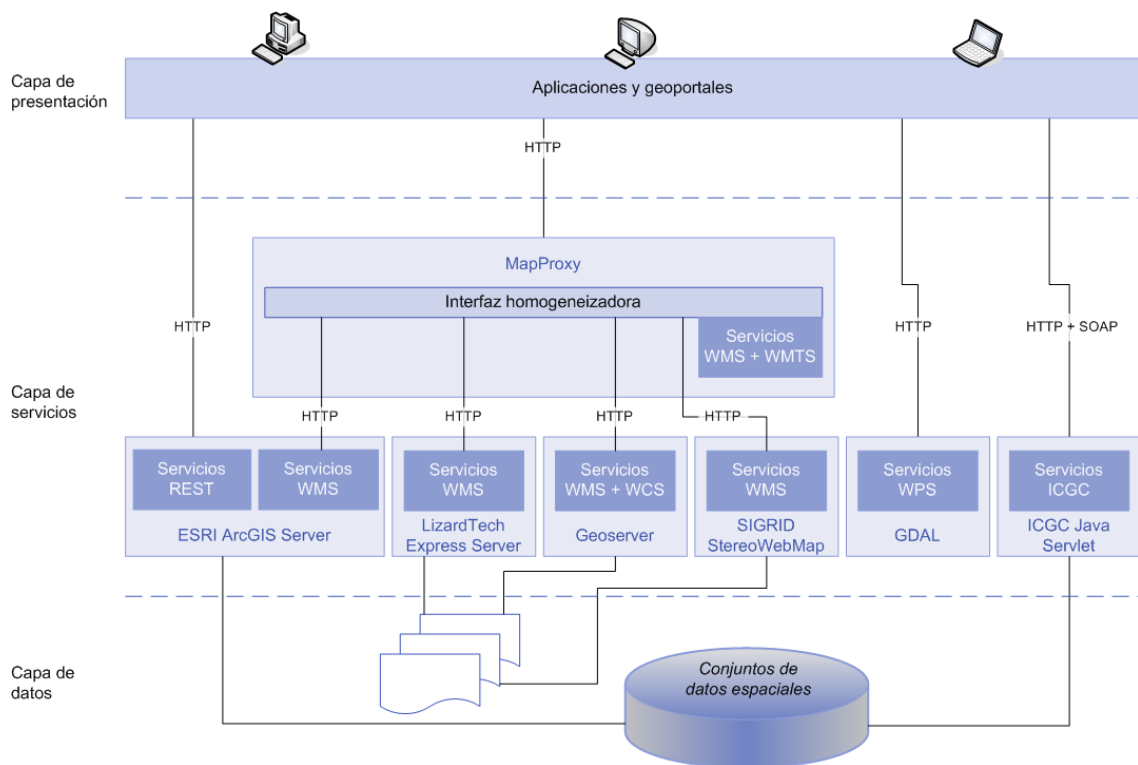


Figura 4.18 Arquitectura de los geoservicios del ICGC [fuente: propia].

Más allá de lo que se acaba de explicar, indicar que el ICGC, mediante la plataforma IDEC, dispone de un servidor de catálogo compatible con el estándar *OGC CSW*, gracias al cual se puede realizar una búsqueda de los geoservicios publicados por este organismo, entre otros muchos de otros proveedores.

4.2.3 Retos planteados

En términos comparativos con respecto a otras organizaciones homólogas, es incontestable que el ICGC ha realizado y continua realizando un importante esfuerzo para ofrecer una amplia gama de servicios que ofrezcan una alternativa a los tradicionales medios de acceso a la información cartográfica, los cuáles implicaban la utilización de soportes analógicos y, en el mejor de los casos, la distribución en formato digital en los centros de atención al usuario.

Dicho lo cual y sin ánimo de desmerecer este trabajo, más bien todo lo contrario, la descripción de los servicios ofrecidos y de la arquitectura que los sustenta muestra ciertas deficiencias que es preciso resolver en aras a proporcionar un escenario objetivo de excelencia, al cual debe aspirar toda organización. En línea con lo apuntado en el punto 2.2, el aumento exponencial de las aplicaciones que emplean IG de base ha puesto de manifiesto la necesidad de disponer de:

- Servicios de descarga de información: son numerosos los ámbitos técnicos en los que es preciso poder trabajar con los datos localmente y, en algunos, poder realizar modificaciones de los mismos que posteriormente es preciso trasladar a los servidores de origen –entornos transaccionales–. A modo de ejemplo, se pueden citar aquellos técnicos encargados del mantenimiento en campo de componentes de redes de servicios –*utilities*–, los cuales deben poder descargar en sus dispositivos móviles las geometrías y los atributos asociados a los elementos –en formato vectorial– que representan en el modelo de datos los citados componentes, para poder consultarlos y realizar las modificaciones –en cuanto a su emplazamiento y/o a sus características alfanuméricas–. Pero también podemos pensar en técnicos de planificación territorial que precisan descargar los datos para poder realizar sus labores técnicas interactuando con las geometrías de dichos datos –algo tan sencillo como dibujar una nueva línea que se apoye en vértices de una geometría de referencia, implica disponer de esta última–.

En este sentido, es reseñable que el ICGC ofrece la práctica mayoría de sus productos de forma gratuita, tal y como ya se ha citado anteriormente. Sin embargo, el proceso de descarga viene condicionado por los siguientes aspectos:

- **Ámbito restringido:** la información se distribuye según una división espacial –normalmente siguiendo unas cuadrículas regulares– que no puede ser modificada por el usuario. Si éste precisa descargar una zona que, por ejemplo, esté a caballo entre tres de esas divisiones, tendrá que proceder a la descarga de los ficheros correspondientes a cada una de ellas individualmente. Es decir, el usuario no puede precisar el ámbito geográfico a descargar.
- **Fragmentación de los elementos:** fruto de la división del territorio que se acaba de reseñar, aquella IG que abarca más de una de esas divisiones se encuentra fragmentada. Este hecho es especialmente problemático en el caso de la información vectorial puesto que el proceso de unificación –disolución– debe tener en cuenta aspectos semánticos –relacionados con sus atributos– que pueden llegar a complicar este proceso. Este hecho es especialmente reseñable si se tiene en cuenta que esta información se encuentra en bases continuas que son con las que se trabaja en realidad internamente.

Es, por tanto, una necesidad imperiosa que se ofrezca a los usuarios servicios de acceso en línea a esas bases cartográficas continuas, de forma que sean dichos usuarios los que establezcan sus ámbitos de interés y descarguen exclusivamente aquello que realmente necesiten.

- **Servicios de geoprocesamiento:** la necesaria economía de recursos propia de muchos dispositivos –véase el caso de los terminales móviles– y la simplicidad de las interfaces propia de las aplicaciones que utilizan IG y cuyo público objetivo es gente profana en el ámbito de los SIG y de la cartografía en general –sirva como ilustración el caso de los usuarios de redes sociales en las que

se proporciona la visualización de ciertas variables sobre componentes cartográficos–, provoca que haya una mayor demanda de servicios de análisis y procesado de la información de forma remota.

El ICGC publica algunos servicios en este ámbito, concretamente uno de transformación de coordenadas y otro de geocodificación. Sin embargo, este último no sigue el estándar *WPS* –tiene una interfaz propietaria del ICGC– y tampoco se ofrece de forma pública –puesto que es fruto de los acuerdos de licencia que tiene firmado este organismo con los proveedores de datos municipales–. Es, por tanto, una oferta heterogénea y con mucho margen de mejora para poder ofrecer a los usuarios muchas más herramientas de geoprocesamiento remoto.

- Entornos de distribución de estos datos en línea, fiables y con tiempos de respuesta adecuados: el éxito o fracaso de cualquier aplicación está condicionado, entre otros aspectos, a un acceso a la información a gestionar acorde con los requerimientos de la aplicación. En el ámbito de la IG los grandes volúmenes de información necesarios para cubrir los territorios implican disponer de sistemas con elevados requerimientos de disponibilidad –365 días al año y 24 horas al día–, a la vez que con tiempos de respuesta extremadamente bajos.

Dado que las organizaciones que disponen de una información más fiable –precisa– del territorio son las agencias cartográficas oficiales y suponiendo que éstas sean capaces de garantizar unos periodos de actualización de la IG suficientemente cortos –aspecto que suele estar comprometido por las frecuentes restricciones presupuestarias–, parece lógico que éstas jueguen un papel primordial como suministradores de las organizaciones que publican dichas aplicaciones.

En lo que respecta al caso de estudio evaluado, los diferentes entornos servidores se implementan físicamente mediante un mínimo de dos líneas de producción –incluyendo las diferentes capas arquitectónicas–, lo cual permite realizar un reparto de la carga en circunstancias normales y, en caso de caída de alguna de las líneas o de precisarse un mantenimiento de las mismas, garantizar el servicio, aunque sea con unos tiempos de respuesta mayores.

El entramado incluye también un sistema de monitorización que se encarga de realizar un conjunto de peticiones representativas a las diferentes líneas. Cuando se produce una incidencia, se activan unas alertas que informan a un operador de las circunstancias que acaecen y éste, manualmente, inicia el procedimiento correspondiente; aunque cabe reseñar que no se trata de un servicio con monitorización 24 horas al día los 7 días de la semana (24x7).

Por otro lado, más allá de la disponibilidad, es fundamental observar el rendimiento que muestra el servicio. En este sentido, el ICGC debería garantizar un tiempo de respuesta homologable a lo dispuesto en la normativa *INSPIRE* –ver punto 4.1.3.5– para los servicios de visualización, todo ello sin perjuicio de que se puedan establecer acuerdos *SLA* con determinados clientes para ofrecer servicios de mayores prestaciones.

Para corroborar este extremo, se ha escogido un servicio *WMS* correspondiente a cada una de las tecnologías servidoras que publican cartografía topográfica u ortofotos y se ha realizado una

batería de 30 minutos en la que se han realizado peticiones *GetMap* aleatorias¹⁵ ¹⁶. Dicha batería se descompone en:

- 2 series de 14 y 15 minutos, respectivamente, simulando una carga sostenida con un único usuario concurrente.
- 1 serie de 1 minuto, intercalada entre las dos anteriores, con un total de 20 usuarios concurrentes, emulando las especificaciones *INSPIRE* referidas anteriormente.

Los resultados obtenidos –ver Tabla 4.11¹⁷– son contundentes:

- En los periodos de carga en los que solo hay un usuario realizando peticiones, tan solo el primero de los servicios ha funcionado de forma completamente satisfactoria. Ninguna de las peticiones ha excedido el tiempo máximo de 5 segundos, *conditio sine qua non* dado que las pruebas se han realizado en un periodo de tiempo normal, según la terminología *INSPIRE*. En el segundo, el porcentaje de peticiones que ha excedido el tiempo máximo es inaceptable, aunque es necesario indicar que las necesidades computacionales derivadas del cálculo en tiempo real de una ortofoto expedita –proceso de rectificación de la fotografía original para corregir las distorsiones originadas por la perspectiva cónica propia del proceso de captura– son elevadas. En el tercer servicio, por su parte, se ha producido un porcentaje prácticamente residual de peticiones que han excedido el tiempo máximo, pero hay que tenerlo en cuenta.
- En los periodos de carga en los que se produce una concurrencia de veinte usuarios, ninguno de los servicios es capaz de atender, procesar y responder un mínimo de 20 peticiones por segundo. Todo ello sin entrar a valorar los tiempos medios originados que solo en un caso están por debajo de los 5 segundos.

Además de todo lo expuesto, la limitación establecida por *INSPIRE* se orienta a servicios de visualización que publican datos compatibles con la estructuración establecida dentro de esa misma normativa.

¹⁵ Las pruebas han sido realizadas la madrugada del 15 de octubre de 2014 con la herramienta *Apache JMeter*, versión 2.11 y revisión 1554548, trabajando contra el entorno de producción del ICGC.

¹⁶ Se han escogido dimensiones de imagen –expresadas en píxeles–, que permitan obtener un tamaño de imagen –expresada en KB– próximo al determinado como referencia en *INSPIRE*. La heterogeneidad de la información publicada en los diferentes servicios hace imposible la determinación de unas dimensiones de imagen iguales en todas las pruebas.

¹⁷ Conc. = número de usuarios concurrentes, # Pet. = número de peticiones atendidas, T. media = tiempo medio, T. mín. = tiempo mínimo, T. máx. = tiempo máximo, Pet. exc. = porcentaje de peticiones que han superado el tiempo límite según *INSPIRE*, Desv. est. = desviación estándar, Rend. = rendimiento. Todos los tiempos aparecen expresados en milisegundos y el rendimiento en peticiones por segundo.

icc_mapesbase								
Contenido	Ortofoto 1:5.000 (capa orto5m)							
Tamaño imagen	1600 x 1200 píxeles							
Formato	JPEG							
Ejemplo de petición								
http://geoserveis.icc.cat/icc_mapesbase/wms/service?REQUEST=GetMap&VERSION=1.1.0&SERVICE=WMS&SRS=EPSG:23031&WIDTH=1350&HEIGHT=1350&LAYERS=orto5m&STYLES=&FORMAT=JPEG&BGCOLOR=0xFF FFFF&TRANSPARENT=TRUE&EXCEPTION=INIMAGE&BBOX=404710,4634651,405162,4635103								
Conc.	# Pet.	T. media	T. mín.	T. máx.	Pet. exc.	Desv. est.	Rend.	Tamaño
1	1011	1722 ms	1026 ms	4193 ms	0 %	458,2	0,6 pet/s	465 KB
20	314	4067 ms	978 ms	15058 ms	33,8 %	2901,9	4,6 pet/s	
icc_ortoxpres								
Contenido	Ortofoto 1:25.000 expedita 2013 (capa Catalunya 25cm. 2013)							
Tamaño imagen	1500 x 1500 píxeles							
Formato	JPEG							
Ejemplo de petición								
http://www.ortoxpres.cat/server/sgdwms.dll?REQUEST=GetMap&VERSION=1.1.0&SERVICE=WMS&SRS=EPSG:23031&WIDTH=1500&HEIGHT=1500&LAYERS=Catalunya%2025cm.%202013&STYLES=&FORMAT=JPEG&BGCOLOR=0xFFFFFFFF&TRANSPARENT=TRUE&EXCEPTION=INIMAGE&BBOX=336242,4610850,336694,4611302								
Conc.	# Pet.	T. media	T. mín.	T. máx.	Pet. exc.	Desv. est.	Rend.	Tamaño
1	193	9081 ms	3675 ms	56559 ms	80,4 %	6850.0	0,1 pet/s	468 KB
20	24	33892 ms	6096 ms	56269 ms	100 %	13911,9	0,2 pet/s	
icc_bt50m								
Contenido	Base topográfica 1:50000 (todas las capas)							
Tamaño imagen	750 x 562 píxeles							
Formato	JPEG							
Ejemplo de petición								
http://geoserveis.icc.cat/icc_bt50m/wms/service?REQUEST=GetMap&VERSION=1.1.0&SERVICE=WMS&SRS=EPSG:23031&WIDTH=750&HEIGHT=562&LAYERS=01_VEGE_PA,02_VEGE_LN,03_ALTI_LN,04_ALTI_LN,05_POBL_PA,06_POBL_PA,07_POBL_PA,08_HIDR_PA,09_HIDR_PA,10_HIDR_LN,11_HIDR_LN,12_POBL_PA,13_POBL_PA,14_HIDR_LN,15_HIDR_LN,16_POBL_LN,17_POBL_LN,18_POBL_LN,19_POBL_LN,20_VIES_LN,21_VIES_LN,22_VIES_LN,23_VIES_LN,24_VIES_LN,25_VIES_LN,26_POBL_LN,27_POBL_LN,28_POBL_LN,29_ALTI_PNTX,30_ALTI_TX,31_VIES_PNTX,32_VIES_PNTX,33_VIES_TX,34_TOPO_TX&STYLES=&FORMAT=image/jpeg&BGCOLOR=0xFFFFFFFF&TRANSPARENT=TRUE&EXCEPTION=INIMAGE&BBOX=377824,4640972,392824,465222								
Conc.	# Pet.	T. media	T. mín.	T. máx.	Pet. exc.	Desv. est.	Rend.	Tamaño
1	945	1810 ms	953 ms	11265 ms	0,6 %	735,3	0,6 pet/s	461 KB
20	70	13050 ms	1960 ms	29076 ms	78,6 %	7135,9	1,1 pet/s	

Tabla 4.11 Resultados pruebas de rendimiento de geoservicios del ICGC.

Este segundo aspecto no se cumple en ninguno de los geoservicios del ICGC y, aparte, cabe remarcar que el estándar *OGC WMS* no es compatible con lo dispuesto por *INSPIRE* para un servicio de visualización –aunque sí que es necesario precisar que este último se basa en lo dispuesto por *OGC*–. Dicho lo cual, se ha escogido el valor de *INSPIRE* como referencia para realizar una aproximación a la problemática del rendimiento y disponibilidad de las plataformas del caso de estudio.

4.3 Especificación funcional

A lo largo del presente capítulo se han analizado los estándares y normativas que regulan el panorama actual de los SIG distribuidos. Se ha comenzado describiendo los estándares más significativos en el ámbito de la difusión de IG a través de Internet, pasando posteriormente a estudiar las IDE, estructuras tecnológicas que permiten facilitar el descubrimiento y el acceso a los geoservicios existentes y promulgados según los citados estándares. Abundando en este hecho, se ha realizado una incursión en la directiva europea *INSPIRE* que regula el marco de colaboración en la difusión de IG de los estados miembros de la UE.

Atendiendo a las tipologías de servicios que define *INSPIRE* –ver punto 4.1.3.3–, se podría extrapolar que una plataforma SIG distribuida debe ofrecer un catálogo de geoservicios que agrupe servicios de: descubrimiento, visualización, descarga y transformación –ampliados en este estudio a servicios genéricos de procesamiento y descartando los de invocación, puesto que todos los servicios a definir han de ser accesibles directamente–. Sin embargo, la concepción de los mismos debe ser revisada, aspecto al cual se procede en los puntos siguientes.

4.3.1 Servicios de descubrimiento

Es lógico pensar que si se desconoce la existencia de un servicio es imposible utilizarlo, por lo que resulta imprescindible habilitar mecanismos que publiciten la existencia de los mismos. En el ámbito de la IG, y más concretamente en el de las IDE, el estándar *OGC CSW* realiza esta tarea [Neb07], definiendo un protocolo que permite la interrogación de un catálogo para conocer los geoservicios y datos que tiene referenciados, gracias a los metadatos que sobre los mismos son almacenados en el mismo.

Podría pensarse, por tanto, que el mero hecho de incorporar en la arquitectura que se está definiendo un componente que actúe como servidor *CSW* es suficiente. Sin embargo, aunque el estándar *CSW* facilita enormemente el descubrimiento de datos y servicios, el proceso de búsqueda se basa exclusivamente en la utilización de palabras clave –*keywords*– y, por tanto, no utiliza las posibilidades que ofrece una búsqueda semántica. En este sentido, sería determinante poder contar con la incorporación dentro de *CSW* de alguno de los lenguajes que permiten la definición de ontologías –ver punto 3.3–, como *OWL* –estándar desarrollado por el *W3C* para la definición de ontologías en la Web que provee la representación de aspectos semánticos de forma explícita– o su especialización para *WS*: *OWL-S*. Mediante la incorporación de las consideraciones semánticas que permite *OWL-S* en *CSW* sería posible realizar búsquedas semánticas de servicios y para conseguir esto el *OGC* ha desarrollado el perfil *ebRIM* –versión 3.0– del estándar *CSW* [Mar09]. Aparentemente puede parecer extraño que *OGC* no haya optado por la creación de un perfil *UDDI*, dado que es el estándar más utilizado para el registro de información en catálogos; la justificación parece estar en el hecho de que éste se centra principalmente

en la catalogación de servicios, mientras que *ebRIM* es mucho más flexible a la hora de registrar también datos, hecho indispensable en los catálogos CSW [Gen06].

Existen aproximaciones menos exigentes en cuanto a la funcionalidad del catálogo, como la representada por la introducción de anotaciones semánticas en tres niveles diferentes: metadatos del servicio –nivel 1–, modelos de datos y procesos geoespaciales –nivel 2– y entidades geográficas –nivel 3–. Esta aproximación aparece perfectamente documentada en [Mau09] y en su aplicación a los metadatos del servicio –nivel 1–, mediante la utilización de elementos ya presentes en los esquemas de los ficheros descriptivos de capacidades –el elemento *type* asociado al elemento *keywords*–, permite especificar la ontología a la que hacen referencia los conceptos que aparecen como palabras clave y que permiten realizar búsquedas más específicas y semánticamente adecuadas.

4.3.2 Servicios de visualización

En este ámbito los dos estándares promulgados por el OGC, *WMS* y *WMTS*, cubren funcionalmente todas las necesidades planteadas. Sin embargo, es necesario realizar una serie de consideraciones –ver puntos 4.1.1.2 y 0–:

- Los servicios *WMS* que dan acceso a IG, permiten a los clientes un grado de personalización muy elevado en cuanto a la determinación de las capas que se desean visualizar, el rango espacial necesario y el tamaño de la imagen resultante –aspectos que condicionan la resolución espacial–.
- Es necesario dar cobertura a los estándares *OGC SLD* y *Filter Encoding*, en aras a ampliar la capacidad funcional del estándar *WMS*. Tal y como se apuntó en 4.1.1.2, *SLD* permite al cliente especificar la simbología a aplicar a la capa o capas –en el caso de que tengan un origen raster sería solo una y en las de origen vectorial pueden ser varias– que solicita y/o los filtros que se desean utilizar para restringir la IG que se quiere recuperar, extremo este último que se realiza mediante las indicaciones del estándar *Filter Encoding*.
- Dadas las características actuales del mercado de aplicaciones consumidoras de IG, existe una demanda mucho más importante de servicios que sirvan de forma muy rápida los contenidos cartográficos, aunque esto suponga renunciar a la especificación de ciertas variables como las dimensiones de las imágenes o el área geográfica solicitada, que de aquellos que ofrezcan mayor versatilidad –*WMS*–. Esta estructuración mucho más rígida viene regulada por el estándar *WMTS*, gracias al cual el espectro de peticiones admitido en *WMS* se reduce a un conjunto finito determinado por un conjunto de cuadrículas –una por nivel de *zoom*–, ofreciendo unos rendimientos y una escalabilidad operacional mucho mayor.
- Con carácter general, los servicios *WMS* y *WMTS* deben ser compatibles con las últimas versiones de los estándares –1.3.0 y 1.0.0, respectivamente– y, en el ámbito de la UE, debe considerarse el cumplimiento de las especificaciones del perfil *INSPIRE* para los servicios de visualización *ISO 19128* –tal y como se ha referido anteriormente en este documento, es equivalente al *OGC WMS*– y *WMTS 1.0.0*.

- Contenidos estandarizados: independientemente de las virtudes que ofrece la Web semántica en términos de interoperabilidad –ver punto 3.3–, otro factor determinante lo constituye la utilización de modelos de datos consensuados dentro de la comunidad geoespacial.

En este sentido, a nivel mundial se han producido diferentes iniciativas, todas ellas vinculadas al mundo de las IDE, siendo las más importantes –por la importancia de los organismos que las promulgan en relación a la cobertura geográfica que representan– la desarrollada por el *Federal Geographic Data Committee (FGDC)* en Estados Unidos de América [FGD14] y, en el ámbito europeo, la regulada por *INSPIRE* [Eur14].

Catálogo	Conjuntos temáticos
Anexo 1	Catastro Direcciones Hidrografía Lugares protegidos Redes de transporte Topónimos Unidades administrativas
Anexo 2	Cubierta terrestre Geología Ortoimágenes Relieve
Anexo 3	Condiciones atmosféricas y fenómenos meteorológicos Distribución de especies Distribución de la población Edificios Fenómenos oceanográficos Hábitats y biotopos Instalaciones agrícolas y de acuicultura Instalaciones de monitorización medioambiental Instalaciones industriales Recursos energéticos Recursos minerales Regiones biogeográficas Regiones marinas Salud y seguridad Servicios públicos y del gobierno Suelos Unidades estadísticas Usos del suelo Zonas de riesgo natural Zonas sujetas a ordenación, restricciones o reglamentaciones y unidades de notificación

Tabla 4.12. Listado de contenidos temáticos de *INSPIRE* [Eur14].

Dado el contexto geográfico en el que se ubica el caso de estudio, se opta por la publicación de los datos siguiendo las especificaciones de la normativa *INSPIRE*. Esta normativa se articula en tres anexos que publican el conjunto de guías técnicas –ver Tabla 4.12–, los cuales son de obligado

cumplimiento para todos los estados miembros de la UE. Esta estructuración temática supone un reto importante para las instituciones productoras de información, puesto que todas ellas disponen de modelos de datos propios, cuya correspondencia con las indicaciones de la normativa no resultan triviales; pero las ventajas que ofrece disponer de servicios de visualización homogéneos en cuanto a su estructuración en capas, son realmente irrenunciables.

Pero aparte de las consideraciones sobre los modelos de datos que se tratarán en el punto 4.3.3, es necesario recoger las indicaciones que realiza *INSPIRE* en el capítulo de la representación de los datos.

Cada uno de los conjuntos temáticos aglutina diferentes capas y para cada una de ellas se define una representación concreta. Esta representación debe ser la disponible por defecto –parámetro *STYLES*–, aunque el organismo publicador pueda decidir incluir representaciones adicionales. Si el servicio, aparte de las capas definidas en la normativa, publica alguna adicional, debe implementar de forma obligatoria una simbología por defecto que siga estas directrices:

- Entidades puntuales: cuadrado gris con un tamaño de 6 píxeles.
- Entidades lineales: línea negra sólida con un grosor de 1 píxel.
- Entidades poligonales: contorno con simbología idéntica a la de las entidades lineales y relleno de color gris.

Por tanto, se recomienda la publicación de geoservicios de contenidos estandarizados, aunque esto no es óbice para que los organismos publicadores también opten por ofrecer geoservicios adicionales que muestren contenidos según modelos de datos y representaciones propietarias.

- Visualización homogénea: enlazando con la problemática anterior, tradicionalmente la IG se ha simbolizado de acuerdo a la escala de representación. A diferentes escalas, la misma información puede mostrarse de forma diferente –más allá de simplificar o no sus geometrías– debido a:
 - Convenciones cartográficas que pueden variar de una a otra escala.
 - Elementos que pueden aparecer o no representados y que pueden agregarse o no a otros en función de que su tamaño sea o no representable a una escala determinada.
 - Elementos cuya geometría puede aparecer modificada –simplificada o no– en función de la resolución asociada a una escala determinada.

A título ilustrativo y retomando lo que se refirió en el punto 4.2.1, la Figura 4.19 muestra un claro ejemplo extraído del caso de estudio, en el que se observa las importantes diferencias existentes en la simbología aplicada a las entidades geográficas que forman los diferentes mapas topográficos, a escalas 1:50.000, 1:25.000, 1:5.000 y 1:1.000 respectivamente, en orden descendente. Por ejemplo, las agrupaciones de edificios aparecen representadas a escala 1:50.000 mediante polígonos con una línea perimetral negra y un relleno sólido de color gris oscuro, a escala 1:25.000 se ha cambiado el color del relleno sólido –optándose por un gris más claro–, a escala 1:5.000 el relleno sólido se muestra en un color beige y, por último, a escala

1:1.000 el polígono se representan exclusivamente mediante una línea perimetral de color rojo –el nivel de detalle mayor permite incluso representar las divisiones existentes como consecuencia de los distintos volúmenes de las construcciones–.

Desde un punto de vista tradicional y contemplando cada producto de forma aislada, esta diversidad en la representación puede estar plenamente justificada y perdurar en los entornos y soportes cartográficos tradicionales.

Sin embargo, desde un punto de vista Web, la publicación de geoservicios que muestren en función de la resolución demandada en las peticiones de los clientes, diferentes representaciones de la misma IG, resulta extraordinariamente confuso para los usuarios.

Es, por tanto, necesario definir unas representaciones visualmente homogéneas a través de las diferentes escalas, aplicando filtros en los elementos a representar –asignación de rangos de escala– que permitan mantener la densidad de información mostrada de forma constante. En este sentido, la aproximación realizada por *Google* en sus productos *Google Maps* y *Google Earth*, replicada posteriormente por *Microsoft* en su producto *BingMaps* y por otros proyectos o iniciativas como *OSM*, persigue facilitar la legibilidad cartográfica empleando simbologías muy sencillas y una cuidada selección de las capas de información a mostrar en función de la escala de visualización.

- Densidad de información constante: este principio, enunciado en la ley Radical de Töpfer [Fra94], indica que el nivel de detalle –*level of detail*– observable en una representación debe ser constante. A priori, se presupone que una misma organización mantiene una coherencia, en cuanto a dicho principio, entre sus diferentes productos cartográficos. Sin embargo, cuando se trabaja con cartografía digital, es preciso matizar ciertos aspectos que condicionan la calidad del resultado:
 - En el ámbito *raster* es habitual disponer de plataformas de publicación de geoservicios que analizan las imágenes originales y generan estructuras piramidales con diferentes niveles de resolución, los cuales han de ajustarse a los requerimientos de las aplicaciones que los consumirán. Los algoritmos precisos para el remuestreo de las imágenes están perfectamente documentados.
 - En el ámbito vectorial, por el contrario, las bases de datos espaciales que actúan como repositorios de la IG no disponen de capacidades para la publicación de diferentes versiones de la geometría de una entidad, en función de la petición recibida a través de un geoservicio. Lo único que se gestiona es si, en función de la resolución espacial asociada a una petición, una capa de información ha de ser publicada o no –restricciones de escala permitidas en los entornos SIG tradicionales–. Este hecho conlleva que no se cumpla el principio enunciado anteriormente, dando lugar a la aparición de trabajos de investigación que han permitido acuñar los conceptos de bases de datos espaciales multiresolución y multirepresentación, de las cuales se hablará con mayor detalle en el punto 4.3.3.

Recapitulando, es importante que los geoservicios de visualización respeten este principio, estableciendo los mecanismos adecuados.



Figura 4.19 Representaciones cartográficas de los diferentes productos del ICGC [ICG14].

- Visualización 3D: las capacidades de representación gráfica de las actuales plataformas de computación, que tienen su mayor exponente en el ámbito de la animación y de los juegos, permiten presentar visualizaciones 3D con gran agilidad. En el mundo de la IG, la capacidad de representación 3D ha sido tradicionalmente deficitaria. La mayoría de los SIG permitían almacenar IG con sus tres componentes espaciales, pero a la hora de representarla lo hacían de forma muy rudimentaria.

Se han desarrollado diferentes iniciativas para la creación de motores gráficos –librerías de programación con capacidades gráficas– para plataformas pesadas –aplicaciones de escritorio– y ligeras –navegadores Web–. En el ámbito Web, la que ha tenido una mayor difusión ha sido *Web Graphics Library (WebGL)*, basado en *OpenGL ES* –ambos desarrollados por la fundación sin ánimo de lucro Khronos– [Khr14]. Se trata de una librería javascript para la representación de gráficos interactivos bidimensionales (2D) y 3D en cualquier navegador compatible y sin la necesidad de instalar complementos adicionales –*plug-ins*–. La Figura 4.20 muestra un ejemplo de utilización de esta tecnología para la cual se proyectan –en el cliente– las teselas servidas por un servicio *WMTS* sobre el Modelo Digital de Terreno (MDT) publicado por un servicio *WCS*.

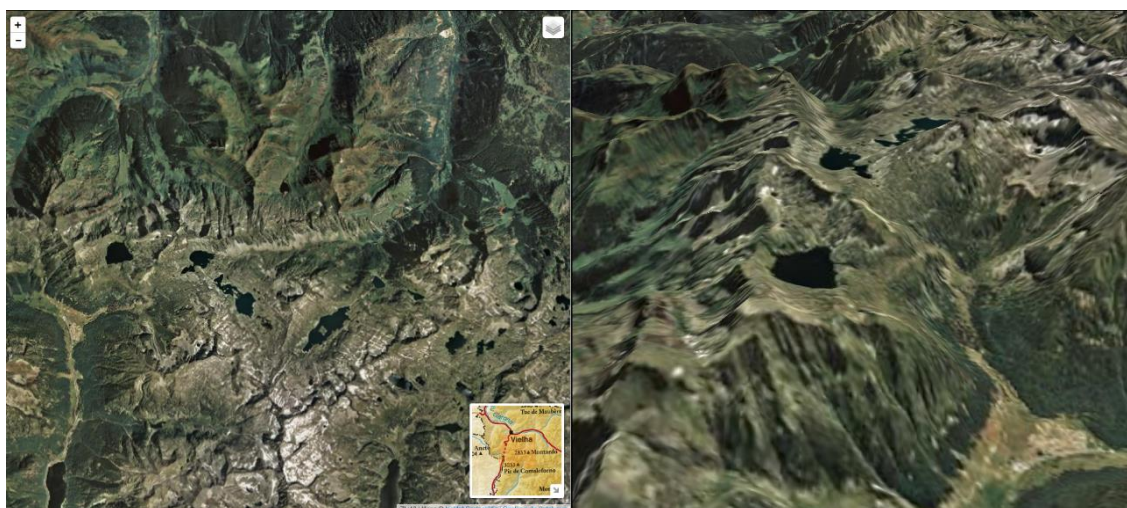


Figura 4.20 Visualización 3D –dcha.– del Lac Redon en el Pirineo catalán –izda.–¹⁸.

Pese a que existan aplicaciones que permitan realizar procesos de composición como el que se acaba de referir, también se pueden contemplar otros escenarios. Uno de ellos consistiría en la publicación de servicios de descarga de información que permitan obtener datos en formatos directamente consumibles por aplicaciones con motores gráficos 3D, tal podría ser el caso del formato *Keyhole Markup Language (KML)* –utilizado en la aplicación *Google Earth*–; para lo cual pueden utilizarse servicios *WFS* que ofrezcan este formato.

¹⁸ Representación obtenida a partir de la aplicación “Generador de escenas 3D”, lanzada con carácter experimental por el ICGC (<http://betaserver.icgc.cat/wxs/generador/escenas3d.html>) en octubre de 2014.

Un segundo escenario podría ser la publicación de un servicio que permita obtener directamente representaciones de modelos 3D expresadas, por ejemplo, en el lenguaje XML denominado *CityGML* –estándar desarrollado por el OGC a partir de la versión 3.1.1 del estándar *GML*– [Gro12]. Aunque todavía no exista un estándar en el ámbito de la comunidad geoespacial, el OGC desarrolló un experimento de interoperabilidad denominado *3D Portrayal IE* [OGC11], el cual concluyó en un informe de ingeniería [Coo12]. Éste contiene detalles sobre el procesado de información 3D en el ámbito de los servicios OGC, así como las mejores prácticas para la representación de grandes volúmenes de datos en escenarios de planificación urbana; tratando de forma especial los dos borradores de estándar siguientes:

- *Web 3D Service (W3DS)*: servicio de representación de IG 3D –modelos urbanos, paisajísticos, etc.– en el cual se suministra la información en forma de escenas en las que aparecen representados los elementos, optimizadas para la visualización en tiempo real con altas tasas de transferencia. Estas escenas 3D pueden ser visualizadas en los navegadores que dispongan de *plug-ins* o en aplicaciones que muestren globos virtuales.
- *Web View Service (WVS)*: servicio de representación de IG 3D –modelos urbanos, paisajísticos, etc.– en el cual se suministran imágenes 2D que representan vistas 3D de escenas construidas a partir de la IG de base. Complementariamente, un servidor de estas características también publica capas de imágenes que incluyan información temática o geométrica.

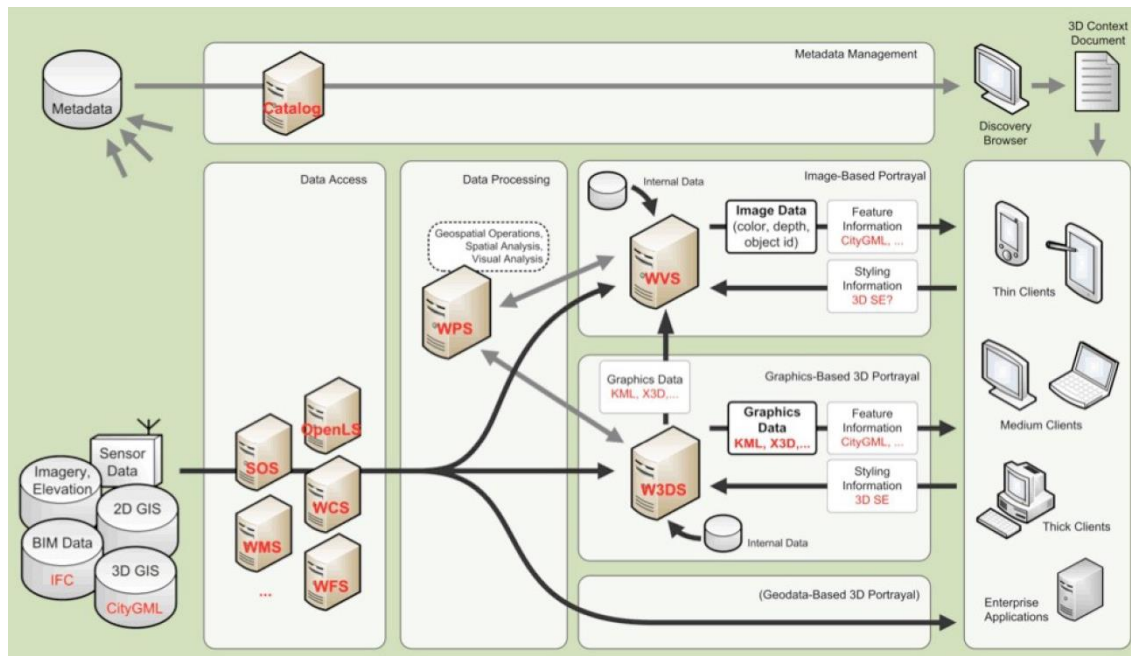


Figura 4.21 Aproximaciones y contexto de representaciones 3D basadas en WS [Coo12].

La Figura 4.21 muestra cual sería el encaje de estas dos nuevas propuestas de estándares dentro de una arquitectura en la que también entran en juego servicios que implementan otros estándares analizados en este documento. Dado que estas dos nuevas propuestas todavía no han fructificado, a efectos de la plataforma distribuida que se está definiendo en este estudio no se tendrán en

consideración, aunque serán incorporados en un futuro cuando sean aprobados, así como otras iniciativas que puedan surgir relacionadas con esta temática.

4.3.3 Servicios de descarga

Sin duda, este es un aspecto decisivo en las circunstancias actuales, puesto que la capacidad de poder acceder directamente a la IG en origen, sin procesamientos intermedios y sin necesidad de utilizar paquetes de elementos predefinidos –como sería la situación mostrada en el caso de estudio–, es clave en la realización de muchas de las funciones que plantean las tecnologías actuales –por ejemplo para la actualización remota de datos–, favoreciendo movimientos de tipo *crowdsourcing* –consistente en la externalización de ciertas tareas, dejándolas en manos de un grupo numeroso de personas o una comunidad, a través de una convocatoria abierta– que son propios de la denominada *Web 2.0*. Aparte, la disponibilidad de este tipo de enlaces a datos es igualmente vital como punto de entrada para los procesos de geoprocésamiento, de los cuales se hablará en el punto siguiente.

En este estudio se han descrito las características funcionales de los estándares *OGC WFS*, *WCS* y *SOS*. Al igual que en el caso de los servicios de visualización, en estos los estándares citados recogen perfectamente las necesidades planteadas, permitiendo el acceso a datos vectoriales –*WFS*–, coberturas –*WCS*– y provenientes de sensores –*SOS*–; pero existen igualmente aspectos que es preciso considerar en aras al diseño de la arquitectura objetivo:

- Aplicación de caracterizaciones de tipo semántico que faciliten la interpretación de los datos –vectoriales–, favoreciendo su correcta utilización y su vinculación con otros provenientes de otras fuentes de información. Se han contemplado y contemplan, puesto que este es un campo de investigación en clara expansión, diferentes posibilidades. Entre las más representativas se pueden citar:
 - Anotaciones semánticas: esta es la aproximación ya avanzada en 4.3.1, siendo de aplicación en este contexto los niveles 2 y 3. El procedimiento implicado es poco invasivo, en el sentido de que no precisa cambios en los esquemas de datos –por ejemplo *GML*, *KML* o *SensorML*–.
 - Descripción mediante tripletas: este enfoque se basa en la descripción de los recursos mediante construcciones “sujeto-predicado-objeto” –en este contexto un recurso es cada una de las entidades geográficas–. El sujeto indica el recurso, el predicado denota rasgos o aspectos del recurso y expresa una relación entre el sujeto y el objeto, siendo éste último el valor asignado al predicado para un sujeto concreto. Por otro lado, cada tripleta se describe gráficamente mediante un grafo etiquetado y dirigido.

El *W3C* ha definido el estándar *RDF* –mediante la publicación del esquema *XML* para *RDF (RDFS)*– para la descripción de los modelos de datos mediante estas tripletas y otro estándar para la definición de un lenguaje estructurado de consultas en grafos *RDF*, denominado *SPARQL Protocol and RDF Query Language (SPARQL)*, que han tenido una gran repercusión en el ámbito de la Web semántica.

Dada la creciente necesidad y las distintas iniciativas existentes para la definición de geometrías en *RDF* y la consulta consiguiente, el consorcio *OGC* decidió publicar un estándar que cubriera esos dos ámbitos; el resultado se denominó *GeoSPARQL* y fue publicado en el año 2012 [Her12]. El nuevo estándar definía los siguientes componentes:

- Nuclear *–core–* que definía las clases *RDF* necesarias para los objetos espaciales.
- Vocabulario topológico *–topology vocabulary–* definido mediante propiedades *RDF* que permiten establecer y consultar relaciones topológicas entre los objetos.
- Geometría *–geometry–* encargado de definir tipos de datos *RDFS* para la serialización de las geometrías asociadas a los objetos.
- Topología de la geométrica *–geometry topology–* encargado de definir las funciones de consulta topológica.
- Vinculación *RDFS –RDFS entailment–* encargado de definir el mecanismo que permite gestionar la coincidencia de las tripletas con las consultas planteadas.
- Reescritura de consultas *–query rewrite–* encargado de definir reglas de transformación de una simple tripleta en una derivada.

95

Llegados a este punto y aceptando la conveniencia de arbitrar mecanismos de razonamiento que saquen partido de los aspectos semánticos, la cuestión a resolver es cómo implementar este modelo de datos basado en *RDF*. Una primera aproximación pasaría por utilizar alguna de los motores de bases de datos que permiten el almacenamiento de información en formato *RDF* –a título ilustrativo en [WWW14] se muestra una lista, no exhaustiva, de implementaciones en este ámbito–. Sin embargo, esta solución conlleva una serie de inconvenientes:

- Implica la transformación de todos los datos desde su modelo relacional original al modelo de ontología estructurado mediante *RDF*. Teniendo en cuenta que los entornos de producción deben continuar radicando en las estructuras relacionales, puesto que las cadenas de trabajo *–workflows–* disponen de herramientas y metodologías muy eficientes y de costosa implantación y que, además, no existen soluciones basadas en ontologías que se planteen como alternativas reales en entornos de producción geoespaciales; la única alternativa pasa por realizar réplicas de los datos en ambos entornos y mantenerlos sincronizados desde el punto de vista de su actualización.

Esta situación, consecuentemente, plantea un incremento de los costes de explotación y una potencial fuente de introducción de errores en los mecanismos de conversión –lo cual implicará arbitrar mecanismos de verificación y validación–. En definitiva, el entorno resultante, pese a ser funcionalmente más

completo –puesto que ofrece posibilidades de razonamiento basadas en la componente semántica–, será más ineficiente.

- El enorme volumen de información que se gestiona en el ámbito de la IG implica un hándicap importante desde el punto de vista de las herramientas basadas en ontologías, puesto que éstas no pueden manipular de forma eficiente grandes cantidades de instancias ontológicas debido al alto consumo de memoria [Zha08].

Por todas estas circunstancias, entiende el autor que la solución, arquitectónicamente hablando más eficiente, pasa por la disposición de una capa intermedia que permita:

- Disponer de las estructuras ontológicas que permitan la ejecución de algoritmos de razonamiento propios de la Web semántica.
- Realizar una equivalencia entre las estructuras ontológicas y las propias del modelo relacional.

Este planteamiento ha sido el abordado por los autores de [Zha08]. En su estudio, se realiza un trabajo de diseño e implantación de una interfaz que provee una capa ontológica para IG accesible a partir de bases de datos relacionales y servicios *WFS*. Por motivos de rendimiento, los autores optan por implementar la recuperación de las geometrías de las entidades a partir de petición *GetFeature* a servicios *WFS*, mientras que abordan la recuperación de los atributos asociados a dichas entidades accediendo a las bases de datos espaciales –las mismas a las que acaban accediendo los servidores *WFS* para recuperar información–, puesto que de esta forma el volumen que ocupa el resultado es muy inferior al propio del *GML* resultante de una respuesta *WFS*. La arquitectura resultante se recoge en la Figura 4.22 y contiene los siguientes componentes:

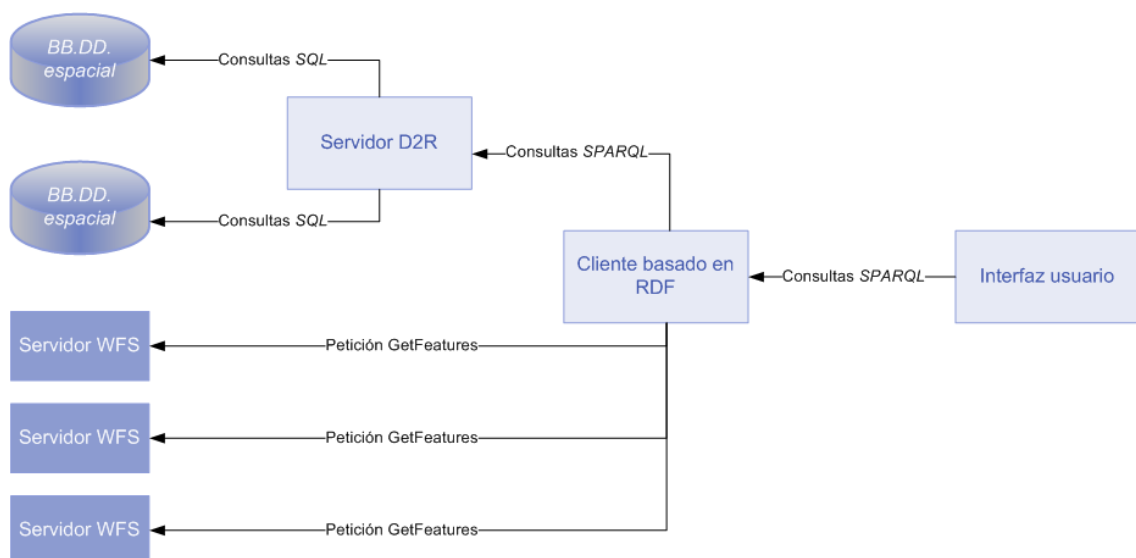


Figura 4.22 Componentes de la interfaz *RDF* para *WFS* diseñada en [Zha08].

- Interfaz de usuario: elemento que recoge la herramienta de interacción con el usuario y que le permite especificar condiciones de búsqueda de carácter semántico. Estas condiciones serán traducidas a consultas *SPARQL* que procesará el siguiente componente.
- Cliente basado en *RDF*: recibe las consultas *SPARQL* de la interfaz de usuario y las descompone en dos conjuntos. Por un lado las que atañen a las geometrías de las entidades, generando las peticiones *WFS* correspondientes hacia los servidores correspondientes. Por el otro, el resto de información atributiva – alfanumérica– que es enviada en forma de consultas *SPARQL* a un componente específico de interacción con las bases de datos relacionales.
- Servidor *D2R*: componente desarrollado por la organización *D2RQ* [DRQ14] que permite a exploradores *RDF* y *HTML* navegar por el contenido de las bases de datos relacionales, permitiendo realizar consultas utilizando el lenguaje de consultas *SPARQL*.

Esta arquitectura permite, consiguientemente, dar respuesta a las necesidades planteadas en el presente estudio, a la vez que también soluciona el problema de heterogeneidad semántica existente entre los diferentes repositorios de IG, aspecto que dificulta enormemente la compartición e interoperabilidad de este tipo de información y de los servicios que la publican.

Por último, mencionar la importancia que tienen la publicación de recursos mediante *RDF* en lo que respecta a lo que se conoce como “datos enlazados” –*linked data*–. Esta es una iniciativa del *W3C* que describe una metodología para la publicación de datos de forma estructurada, de forma que puedan enlazarse entre sí –independientemente de su origen– y resultar más útiles. Utilizando tecnologías propias de la Web –como *HTTP*, *URI* y *RDF*– se persigue que el descubrimiento de estos recursos se pueda realizar de forma automatizada, sin necesidad de intervención humana. En este marco, el proyecto Enlazando Datos Abiertos –*Linking Open Data*–, liderado por el grupo del *W3C* encargado de divulgar y explicar la Web semántica –*Semantic Web Education and Outreach*–, persigue ampliar la Web con la creación de una base de datos común de servidores de datos en *RDF* –los cuales referencian datos publicados de forma libre para todo el mundo, sin restricciones de derechos de autor, patentes u otros mecanismos de control, partiendo de la base de que en la mayoría de los casos han sido generados u obtenidos con fondos públicos– y el establecimiento de enlaces *RDF* entre datos de distintos orígenes.

Estas dos alternativas no son mutuamente excluyentes, bien al contrario, la combinación de ambas técnicas, anotaciones semánticas y una interfaz ontológica *RDF*, permite ofrecer una solución para todo el espectro de aplicaciones, desde aquellas que utilizan el contexto semántico para ayudar a describir la información, hasta las que utilizan algoritmos de razonamiento más o menos avanzados.

- Modelos de datos estandarizados: en línea con lo recogido en 4.3.2 con respecto a la publicación de geoservicios con contenidos estandarizados, resulta imprescindible disponer de geoservicios

que permitan la descarga de datos estructurados siguiendo las especificaciones estándar que, en el ámbito de estudio, las marca la normativa *INSPIRE*. Este hecho implica la necesidad de incluir mecanismos en los procesos de publicación que permitan la convergencia de los modelos propietarios con el modelo *INSPIRE*. Estas equivalencias no son siempre triviales, de hecho, en el caso de la Base Topográfica de Cataluña (BTC) a escala 1:5.000 que publica el ICGC, no es posible encontrar una relación directa entre ambos modelos –la cardinalidad de las relaciones entre las diferentes capas es *n:m*–, tanto a nivel sintáctico como semántico.

Un ejemplo que permite ilustrar perfectamente este extremo lo supone el estudio realizado por la UE dentro del proyecto *GIS for Europe (GIS4EU)*, en el cual se analiza el cumplimiento de las especificaciones de *INSPIRE* con respecto al tema de redes de carreteras [Lle08] dentro del modelo de datos de la BTC a escala 1:5.000 indicada anteriormente. Dado que los objetivos con los que fue definida la citada base –captura un conjunto de datos de referencia básicos, desde un punto de vista topográfico, para la producción de mapas y su consulta dentro de aplicaciones espaciales– difieren de los de la normativa *INSPIRE*, las conclusiones del estudio indican que los modelos de datos no encajan muy bien. De las seis capas que se contemplan en el tema analizado –*RoadNode*, *BasicRoadLink*, *RoadLink*, *RoadArea*, *VehicleTrafficArea* y *RoadServiceArea*–, solo una encaja con el modelo de datos de la base del ICGC: *RoadLink*. Aun así, los atributos asociados a esta capa en ambos modelos no son compatibles en su mayoría.

Siguiendo con la recomendación realizada en el punto 4.3.2 en referencia a los contenidos estandarizados, en este caso también se plantea la creación de servicios de descarga que sigan las directrices de *INSPIRE*, puesto que este aspecto favorecerá las posibilidades de interoperabilidad con datos provenientes de otros servicios que también sigan esas indicaciones. Dadas las connotaciones que este hecho tiene para organismos como el que se cita en el caso de estudio planteado en este trabajo, resulta interesante simultanear servicios que expongan datos estructurados en función de un estándar con otros que publiquen modelos propietarios, los cuales podrían estar más indicados para ciertas aplicaciones. Imagínese, a título ilustrativo, un servicio de descarga orientado a la actualización remota de los datos almacenados en el repositorio subyacente, que se corresponde con el modelo de datos del entorno de producción de una determinada organización.

- Servicios transaccionales: enlazando con el ejemplo que acaba de recogerse, la existencia de tecnologías móviles con capacidades de gestión cada día más avanzadas ha permitido disponer en la actualidad de herramientas que abordan procesos de mantenimiento de la información de base de una organización de forma remota. Típicamente, un usuario de una de estas brigadas de trabajo en campo descarga en su dispositivo unos contenidos que necesita consultar y, en caso de que precise actualizar la información, procede a realizar los cambios oportunos y al final de la jornada, una vez ha regresado a su centro de trabajo, procede a volcar en el repositorio de datos los cambios aplicados en su versión local. Este tipo de soluciones tecnológicas han venido aplicándose de forma generalizada y con gran éxito en las empresas que gestionan activos de redes de servicios –*utilities*–, empezando inicialmente por procesos de actualización que solo atañían a la información atributiva de las entidades, para pasar posteriormente a permitir la edición de la componente geométrica de las mismas.

Hoy en día, salvo en casos muy concretos en los que no se disponga de cobertura de telecomunicaciones, estos procesos de actualización es factible realizarlos de forma conectada, sin precisar tener que realizar una exportación e importación de todo un bloque de datos para trabajar de forma desconectada. Para ello, resulta imprescindible disponer de servicios de descarga que permitan acceder a los datos remotamente y en este ámbito también ha realizado un trabajo reseñable el OGC con la incorporación en la especificación del estándar WFS de métodos para la gestión de transacciones [Vre10] –inserción, actualización, sustitución y borrado–. Estos mismos principios también han sido aplicados a las coberturas de información, habiendo en este caso desarrollado el citado organismo una extensión del estándar WCS que da cobertura a la ejecución de transacciones [Whi09].

Sin embargo, más allá de los beneficios que suponga para una organización la existencia de servicios de descarga transaccionales, una aplicación colateral, pero no por ello menos importante socialmente hablando, la constituye la aplicación de estas capacidades en el ámbito de lo que se denomina Web 2.0. Tal y como se enunció en 2.2, una de las características más importantes de este paradigma del diseño de páginas Web es la de fomentar la colaboración en la red para la creación –en este caso actualización– de contenidos. Existen proyectos exitosos de colaboración para la creación de contenidos cartográficos, siendo el más importante de ellos, por la dimensión que ha alcanzado, el anteriormente referido OSM [OSM14]. Este proyecto aglutina a miles de ciudadanos de todo el mundo en la captura y mantenimiento de una base cartográfica que permita disponer de datos actualizados y accesibles de forma gratuita para todo el mundo.

Esta capacidad de actualización colaborativa ha permitido dar respuesta a las necesidades cartográficas que tienen las organizaciones de ayuda internacional cuando necesitan conocer el estado real de las infraestructuras en un determinado lugar, con el fin de montar la logística necesaria para ayudar a la población, siendo paradigmática la situación vivida en Haití como consecuencia del terremoto que padeció ese país en marzo de 2010 y que supuso la destrucción de la práctica totalidad de sus infraestructuras. En esas circunstancias, la cartografía existente hasta la fecha carecía de validez y era preciso acometer su actualización con carácter de urgencia. Partiendo de imágenes satélite obtenidas en las 48 horas siguientes al suceso, en un mes un equipo formado por unos 600 voluntarios de la plataforma OSM consiguió crear una base cartográfica que sirvió de referencia para dichas organización [OSM14b].

- Modelos de datos multiresolución: tradicionalmente los organismos productores de cartografía han abordado la elaboración de modelos de datos diferentes para cada una de las escalas de trabajo, de forma que para cada base cartográfica se define un modelo de datos específico. Este hecho también está presente en el caso de estudio planteado en este trabajo, existiendo modelos diferenciados para la BTC 1:5.000 y para la 1:50.000, por ejemplo. A parte de las implicaciones que este hecho tiene a efectos de visualización –planteadas en el punto 4.3.2–, también hay que valorar las implicaciones que tiene desde el punto de vista de los modelos de datos y los repositorios de almacenamiento:
 - Requerimientos de almacenamiento elevados: lógicamente, el hecho de almacenar para diferentes escalas todas las entidades, implica disponer de unos recursos de almacenamiento apropiados.

- Inconsistencia de datos: disponer de flujos de producción –y por tanto de los procedimientos de actualización– independientes origina problemas debidos a la falta de sincronización entre las diferentes series cartográficas.

Para ilustrar esta circunstancia, la Figura 4.23 muestra la discrepancia existente entre los mapas topográficos de Cataluña publicados por el ICGC a escalas 1:50.000 y 1:25.000 – el rectángulo naranja muestra sobre el mapa a escala 1:50.000 la zona correspondiente al fragmento capturado a escala 1:25.000–. La carretera C-51 figura con su trazado actual en el mapa de escala menor y con su trazado antiguo en el de escala mayor –apareciendo el actual en estado de construcción–.

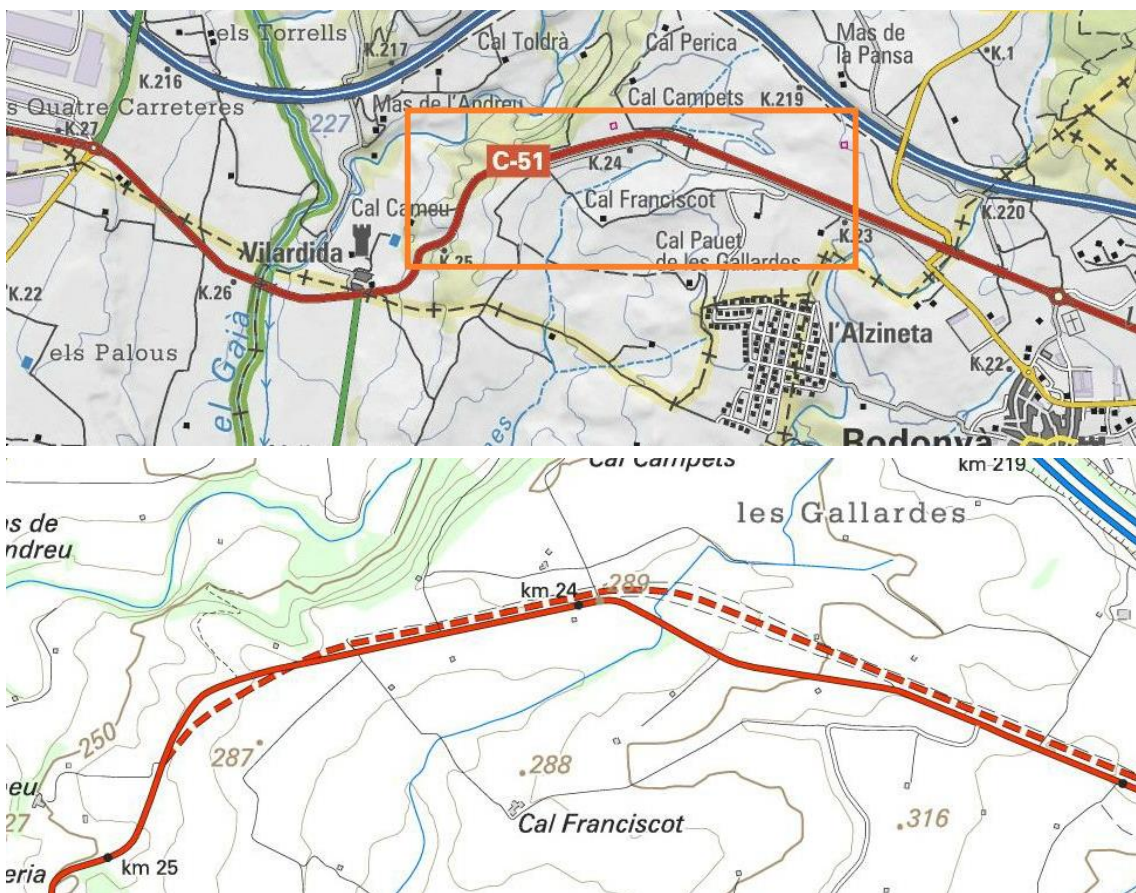


Figura 4.23 Problemas de sincronización –superior BTC 1:50.000 e inferior BTC 1:25.000– [ICG14]¹⁹.

- Densidad de información variable: el nivel de detalle –*level of detail*– observable debe cumplir con el principio de “densidad de información constante” enunciado en la ley Radical de Töpfer [Fra94]. Para vencer las limitaciones existentes en las bases de datos espaciales tradicionales –según se ha introducido en el punto 4.3.2–, se puede optar por la aproximación multirepresentación, caracterizada por almacenar para cada entidad una

¹⁹ Capturas obtenidas el 16 de mayo de 2014.

geometría diferente ajustada a un nivel de resolución concreto, o por la variante multiresolución, caracterizada por disponer de una única geometría con el nivel máximo de detalle que se pueda requerir y proceder a su simplificación y generalización de forma dinámica [Zho04].

Ambas aproximaciones tienen ventajas e inconvenientes. El modelo multirepresentación ofrece un mayor rendimiento al tener todas las geometrías generadas de antemano, a costa de una menor flexibilidad –un cambio en los niveles de resolución implica regenerar la estructura– y unos mayores requerimientos de almacenamiento. Pese a ello, el principal hándicap de esta técnica lo constituye las dificultades que implica mantener una densidad de información constante. Esta última circunstancia viene dada porque la densidad de información está condicionada a aspectos relativos a las consultas que se produzcan – área de interés y capas de información solicitadas–, así como a la resolución que tenga el dispositivo cliente que muestra los resultados de las peticiones.

Por su parte, el modelo de multiresolución requiere menor espacio de almacenamiento y garantiza la constancia en el nivel de detalle, aunque esto sea a cambio de un mayor consumo de recursos de procesamiento y un rendimiento inferior; todo ello teniendo en consideración que la tarea de generalización automática necesaria para generar las distintas geometrías adecuadas para cada una de las resoluciones solicitadas, no es trivial y ha sido foco de múltiples investigaciones que de momento no han podido fructificar en ningún sistema completamente autónomo y de amplio espectro [Zho04] [Par00].

Aparte de todas las consideraciones realizadas sobre los servicios de descarga y centrándose el estudio en el ámbito europeo, es necesario atender a las consideraciones que realiza la normativa *INSPIRE* con respecto a distintos parámetros que permiten evaluar la calidad de un servicio de estas características. Según la misma:

- Rendimiento: el tiempo de respuesta máximo para conocer los metadatos de un servicio no debe superar los 10 segundos en una situación normal. En las mismas condiciones, ante una petición de descarga de entidades, el tiempo que transcurre hasta que se empieza a recibir datos no debe superar los 30 segundos y después debe mantenerse una tasa de descarga mayor de 0,5 Mb o de 500 entidades por segundo. Un tiempo máximo de espera de 10 segundos y las mismas tasas de descarga serán los valores exigibles en la descarga de información atributiva de las entidades.
- Capacidad: el mínimo número de peticiones de descarga simultáneas, de acuerdo con los parámetros de rendimiento anteriormente recogidos, debe ser de 10 peticiones por segundo, pudiendo limitarse el máximo número de peticiones ejecutables en paralelo a 50.

Complementariamente a todo lo recogido, reseñar también la conveniencia de añadir al formato de descarga soportado actualmente por este estándar los que se recogen en 4.3.4.1.

4.3.4 Servicios de geoprocésamiento

La oferta de servicios geoespaciales viene condicionada, como en otros muchos ámbitos, por la demanda existente. En este sentido, la gran mayoría de las aplicaciones son consumidoras de representaciones cartográficas en forma de mapa u ortofoto, de ahí que exista una gran demanda de servicios de visualización.

En mucha menor medida, hay usuarios que precisan disponer de información en local para poder realizar variopintos procesos, siendo necesaria en este caso la habilitación de servicios que permitan la descarga en línea de información y, en el mejor de los casos, su actualización en tiempo real.

Sin embargo, no es esta la única utilidad que se puede encontrar a este tipo de servicios. La existencia de comunidades de usuarios no expertos en materias de cartografía o SIG, que requieren la utilización de aplicaciones en las que es preciso realizar ciertas labores de análisis, plantea la necesidad de disponer de plataformas servidoras de herramientas de procesamiento espacial, y es aquí donde entran en juego los servicios de geoprocésamiento.

Las posibilidades tecnológicas actuales permiten dibujar un escenario todo lo ambicioso que se desee. Existen librerías de programación que disponen de *APIs* compatibles con los lenguajes de creación de *WS*. El caso de las librerías de código libre *GDAL* y *Geotools* –basada a su vez en la librería *Java Topology Suite (JTS)*– son ejemplos contrastados de esta circunstancia, sentando las bases para la creación de entornos servidores de procesos geoespaciales. Por otro lado, tal y como se apuntó en 2.1.3, las bases de datos espaciales también ofrecen la posibilidad de ejecutar funciones espaciales sobre las geometrías almacenadas en las propias bases, utilizando el Lenguaje de Consultas Estructurado (*SQL*) que también puede ser invocado desde *WS*.

Dicho lo cual y en aras a favorecer la interoperabilidad de los servicios, es recomendable que estas iniciativas de publicación converjan en la utilización de protocolos estándar, entre los que cabe reseñar el propio de la comunidad geoespacial representada por el *OGC*: el estándar *WPS*. Esta interoperabilidad, además, ha de permitir la constitución de servicios complejos de geoprocésamiento, los cuales se basen en la composición de servicios más sencillos.

4.3.4.1. Análisis espacial vectorial

Para determinar el abanico de servicios a implementar por la plataforma SIG distribuida que se está definiendo, es interesante comenzar revisando el modelo de objetos de geometría estándar definido por el *OGC* en el documento [Her11], cuya jerarquía de clases se recoge en la Figura 4.24.

Como se puede observar, la clase raíz se denomina *Geometry* y para ella el mismo estándar define un conjunto de operaciones –ver Figura 4.25–, que agrupa en tres categorías:

- Básicas –*basic*–: reúne todas aquellas funciones que permiten conocer alguna propiedad del elemento geométrico –área, perímetro, dimensión, sistema de referencia, geometría de su contorno, mínimo rectángulo envolvente, etc.–.

- Consulta *–query–*: recoge todas aquellas funciones que permiten conocer relaciones espaciales topológicas existentes con otras geometrías (intersección, igualdad, superposición, etc.)²⁰.
- Análisis *–analysis–*: engloba todas aquellas funciones que permiten realizar análisis geométricos – distancia mínima, zona afectada, cierre convexo, intersección, etc. – entre distintas geometrías.

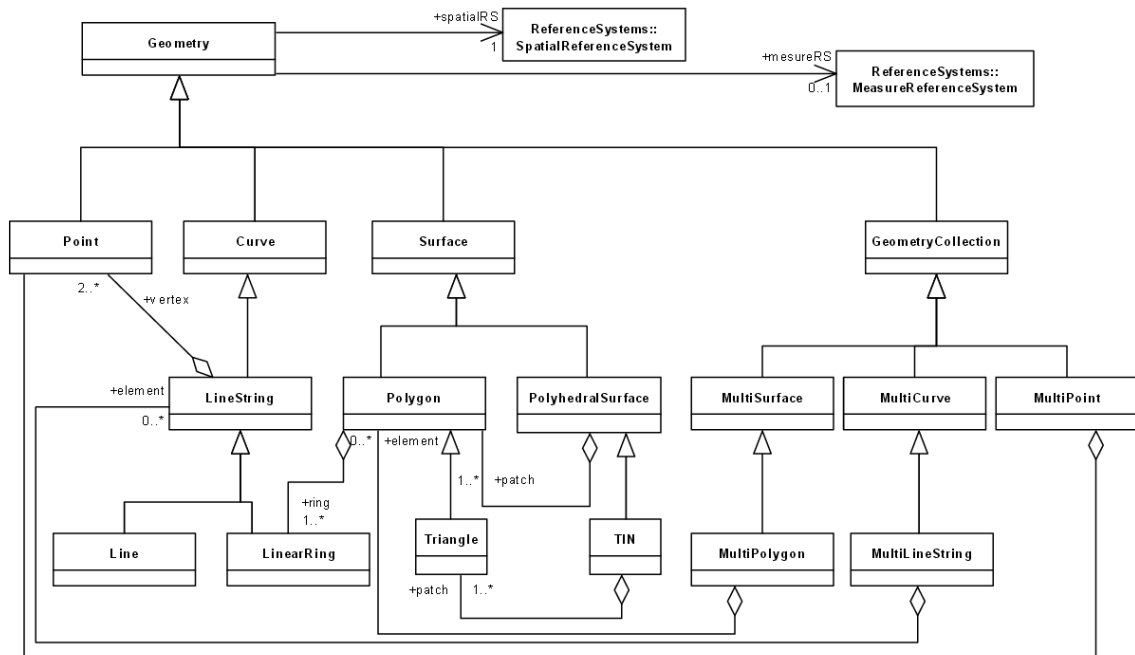


Figura 4.24 Jerarquía de clases de geometría según OGC [Her11].

Estos tres conjuntos van a constituir el núcleo funcional de los geoservicios de procesamiento propuestos para la nueva arquitectura distribuida. Dentro de las incluidas en la primera categoría, básicas, se seleccionan las operaciones recogidas en la Tabla 4.13, a las cuales se incluyen dos adicionales que permiten conocer el área y el perímetro de la geometría, incluyendo la denominación de la función según el estándar *ISO 13249-3* [ISO11].

El citado estándar define tipos de datos específicos para IG, rutinas y esquemas para la gestión de dicha información, así como para su recuperación en función de aspectos espaciales como la geometría, la localización y la topología, tomado como referencia por el OGC en su estándar de implementación sobre el acceso a entidades [Her10] utilizando el lenguaje *SQL* y por los fabricantes de las principales bases de datos espaciales.

²⁰ En esta categoría, a juicio del autor, no tiene sentido incorporar las operaciones *locateAlong* y *locateBetween*, puesto que no son coherentes con la definición dada, se trata de funciones que permiten derivar nuevas geometrías – ubicadas en posiciones concretas – a partir de una dada y tendría más sentido agruparlas en el epígrafe de básicas, como la función *boundary* –contorno– que sí forma parte de la misma.

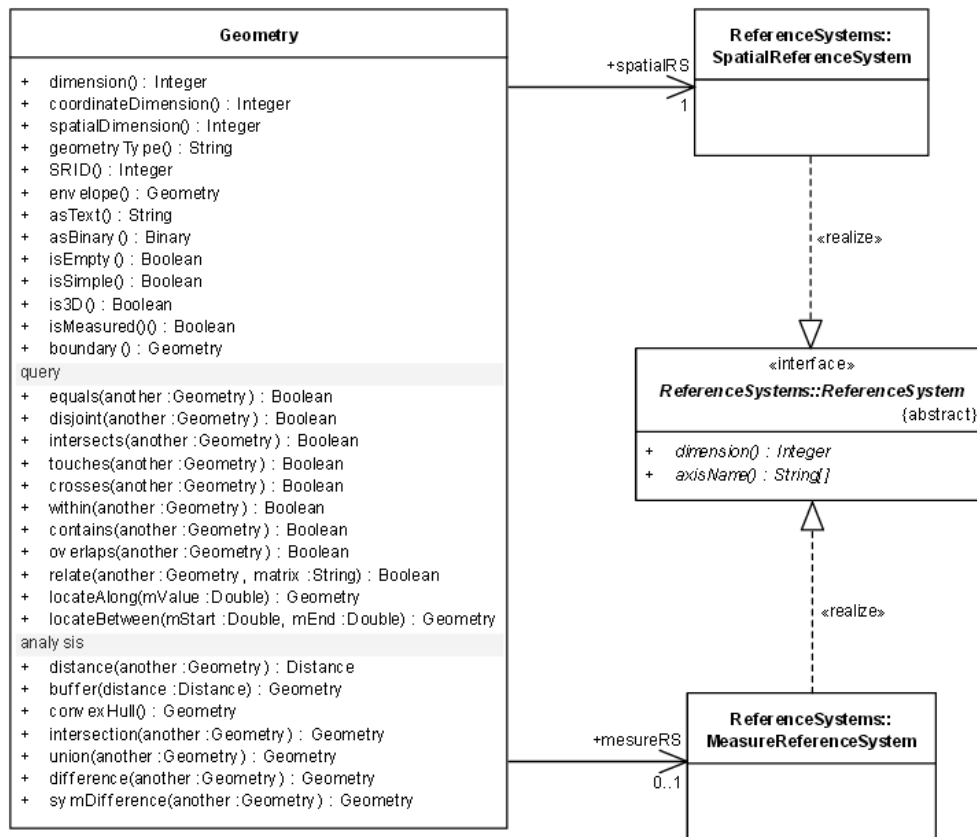


Figura 4.25 Operaciones de la clase geometría según el OGC [Her11].

Función	ISO 13249-3	Descripción
Área	<i>ST_Area(a)</i>	Devuelve el área de la superficie correspondiente a la geometría <i>a</i> .
Envoltura	<i>ST_Envelope(a)</i>	Devuelve la geometría correspondiente al mínimo rectángulo envolvente de la geometría <i>a</i> .
Límite	<i>ST_Boundary(a)</i>	Devuelve la geometría correspondiente al límite de la geometría <i>a</i> .
Perímetro	<i>ST_Perimeter(a)</i>	Devuelve la longitud de la geometría lineal o la correspondiente al perímetro de una geometría superficial <i>a</i> .

Tabla 4.13 Relación de funciones básicas seleccionadas.

El citado estándar define tipos de datos específicos para IG, rutinas y esquemas para la gestión de dicha información, así como para su recuperación en función de aspectos espaciales como la geometría, la localización y la topología, tomado como referencia por el OGC en su estándar de implementación sobre el acceso a entidades [Her10] utilizando el lenguaje SQL y por los fabricantes de las principales bases de datos espaciales.

Centrando el estudio en las relaciones espaciales de carácter topológico –segundo conjunto–, el mismo estándar define un conjunto de operadores booleanos que permiten conocer el cumplimiento o no de una

determinada relación entre dos geometrías dadas. Para ello se basa en un modelo matemático utilizado comúnmente para expresar dichas relaciones, conocido como *Dimensionally Extended nine-Intersection Model (DE-9IM)* [Ege89] [Cle93]. Este modelo diferencia tres partes dentro de una geometría: zona interior, exterior y contorno *–boundary–*. Aparte, determina también la dimensión de la geometría resultante, teniendo en cuenta que las geometrías de carácter puntual *–derivadas de la clase Point–* tienen una dimensión igual a 0, las lineales 1 *–derivadas de la clase Curve–* y las superficiales 2 *–derivadas de la clase Surface–*. Por otro lado, desde un punto de vista gráfico, el modelo se formaliza mediante una matriz de 3x3 elementos, correspondientes a todas las posibles variantes de intersección entre las tres partes referidas de dos geometrías *–Figura 4.26, donde I representa interior, C contorno y E exterior–*. Las nueve combinaciones resultantes permiten deducir un total de ocho relaciones espaciales *–contiene, cruza, dentro de, disjunto, igual, interseca, superpone y toca–*, a las cuales se añade una de carácter genérico *–relaciona–* que permite deducir cualquiera de las anteriores y dos adicionales para facilitar la respuesta a necesidades de análisis topológico comunes *–cubierta por y cubre–*.

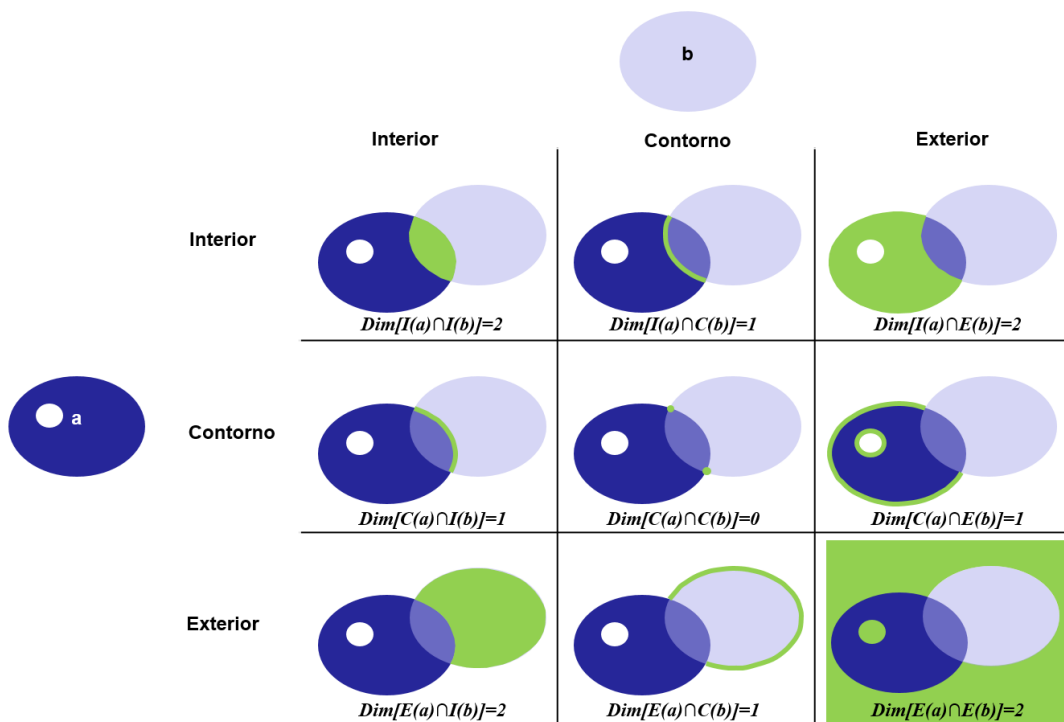


Figura 4.26 Matriz de combinaciones del modelo *DE-9IM* [fuente: propia].

Este conjunto resultante, formado por un total de 11 funciones, constituye el segundo bloque funcional a ser proporcionado por los geoservicios de procesamiento, apareciendo recogidos en la Tabla 4.14.

Por su parte, el último bloque viene representado por las funciones de análisis espacial. En este caso concreto se respetan las identificadas para la clase *Geometry*, aunque se introducen las siguientes mejoras:

- Tratamiento de los atributos alfanuméricos asociados a las geometrías de entrada, para poder conocer los atributos que tendrá la resultante *–suponiendo que ambas puedan tener una colección de atributos diferente–*. Dado que el estándar se fija exclusivamente en la componente geométrica

de las entidades, no se realiza ninguna consideración al respecto y este es un aspecto fundamental desde el punto de vista funcional de un SIG.

- Incorporación de funciones con una gran demanda en el ámbito SIG y que permiten dar respuesta a problemáticas muy habituales. Estas funciones no vienen recogidas en el estándar, pero suelen formar parte del abanico funcional de los SIG tradicionales.

Función	ISO 13249-3	Descripción
Contiene	<i>ST_Contains(a,b)</i>	Se cumple si y solo si ningún punto de la geometría <i>b</i> está en el exterior de <i>a</i> y al menos un punto del interior de <i>b</i> está en el interior de <i>a</i> . Es la inversa de <i>ST_Within</i> .
Cruza	<i>ST_Crosses(a,b)</i>	Se cumple si las geometrías <i>a</i> y <i>b</i> tienen algún punto interior en común, pero no todos.
Cubierta por	<i>ST_CoveredBy(a,b)</i>	Se cumple cuando todos los puntos de la geometría <i>a</i> son puntos de <i>b</i> y el interior de ambas geometrías tiene al menos un punto en común. Es la inversa de <i>ST_Covers</i> .
Cubre	<i>ST_Covers(a,b)</i>	Se cumple cuando todos los puntos de la geometría <i>b</i> están contenidos en el interior de <i>a</i> . Es la inversa de <i>ST_CoveredBy</i> .
Dentro de	<i>ST_Within(a,b)</i>	Se cumple cuando el interior de la geometría <i>B</i> está contenido en el interior de <i>a</i> . Es la inversa de <i>ST_Contains</i> .
Disjunto	<i>ST_Disjoint(a,b)</i>	Se cumple si las geometrías <i>a</i> y <i>b</i> no intersectan, es decir, si no comparten ningún espacio. Es la inversa de <i>ST_Intersects</i> .
Igual	<i>ST_Equals(a,b)</i>	Se cumple si ambas geometrías son idénticas.
Interseca	<i>ST_Intersects(a,b)</i>	Se cumple si las geometrías <i>a</i> y <i>b</i> intersecan, es decir, si comparten algún espacio. Es la inversa de <i>ST_Disjoint</i> .
Relaciona	<i>ST_Relate(a,b)</i>	Determina si las geometrías <i>a</i> y <i>b</i> están topológicamente relacionadas, comprobando las intersecciones entre sus interiores, perímetros y exteriores, de acuerdo con lo dispuesto en la matriz de intersecciones.
Superpone	<i>ST_Overlaps(a,b)</i>	Determina si las geometrías <i>a</i> y <i>b</i> tienen algún punto en común, pero no todos y si ambas tienen la misma dimensión.
Toca	<i>ST_Touches(a,b)</i>	Se cumple si ambas geometrías, <i>a</i> y <i>b</i> , tienen al menos un punto de su contorno en común, pero no ningún punto interior.

Tabla 4.14 Relación de funciones de carácter topológico.

La relación de funciones resultante se muestra en la Tabla 4.15. Aunque en todas ellas los parámetros que representan a geometrías se han abordado considerando que representan a un único objeto, los geoservicios deben permitir que estos parámetros hagan referencia a colecciones de entidades. Estas colecciones –geometrías y atributos asociados– han de poder expresarse en formatos estándares y/o de amplia utilización por la comunidad de desarrolladores de aplicaciones SIG en el ámbito Web. En este sentido, los formatos contemplados para la especificación de las entidades son:

- *GML*: formato *XML* estándar del *OGC* [Bau10] para la descripción de entidades geográficas. Ha de poder pasarse explícitamente código en este formato como parámetro de las funciones e implícitamente mediante la especificación del *URI* correspondiente a una petición *getFeatures*, de

un servicio compatible con el estándar *WFS* –este estándar determina que el formato por defecto es *GML*, según se recoge en 4.1.1.4–.

Función	ISO 13249-3	Descripción
Cierre convexo	<i>ST_ConvexHull(a)</i>	Devuelve la geometría correspondiente al cierre convexo de la geometría <i>a</i> .
Diferencia	<i>ST_Difference(a,b)</i>	Devuelve la geometría <i>c</i> que representa el conjunto de puntos de la geometría <i>a</i> que no están contenidos en una segunda geometría <i>b</i> . La geometría <i>c</i> conserva los atributos de <i>a</i> .
Diferencia simétrica	<i>ST_SymDifference(a,b)</i>	Devuelve la geometría <i>c</i> que representa el conjunto de puntos de las geometrías <i>a</i> y <i>b</i> que no están contenidos en ambas a la vez. La geometría <i>c</i> contiene la suma de los atributos presentes en <i>a</i> y <i>b</i> , aunque según la procedencia de cada uno de los puntos de <i>c</i> , estarán informados los atributos heredados de <i>a</i> o de <i>b</i> .
Distancia	<i>ST_Distance(a,b)</i>	Determina la distancia mínima existente entre cualquier combinación de puntos de las geometrías <i>a</i> y <i>b</i> .
Extrae	<i>Clip(a,b)</i> ²¹	Variante de <i>ST_Intersection</i> que realiza la misma operación, pero la geometría <i>c</i> resultante solo contiene los atributos presentes en <i>a</i> .
Identidad	<i>Identity(a,b)</i>	Devuelve la geometría <i>c</i> que resulta de unir todos los puntos de <i>a</i> , estén o no contenidos en <i>b</i> . La geometría <i>c</i> contiene la suma de los atributos presentes en <i>a</i> y <i>b</i> , aunque según la procedencia de cada uno de los puntos de <i>c</i> , estarán informados los atributos heredados solo de <i>a</i> o de <i>a</i> y <i>b</i> –estos último ocurrirá en los puntos que se corresponden con la intersección de las geometrías originales–.
Intersección	<i>ST_Intersection(a,b)</i>	Devuelve la geometría <i>c</i> que representa el conjunto de puntos que definen la intersección entre las geometrías <i>a</i> y <i>b</i> . La geometría <i>c</i> contiene la suma de los atributos presentes en <i>a</i> y <i>b</i> .
Unión	<i>ST_Union(a,b)</i>	Devuelve la geometría <i>c</i> que representa el conjunto de puntos que definen la unión entre las geometrías <i>a</i> y <i>b</i> . La geometría <i>c</i> contiene la suma de los atributos presentes en <i>a</i> y <i>b</i> , aunque según la procedencia de cada uno de los puntos de <i>c</i> , estarán informados los atributos heredados de <i>a</i> y/o de <i>b</i> .
Zona de influencia	<i>ST_Buffer(a,d)</i>	Devuelve la geometría <i>c</i> que resulta de unir todos los puntos cuya distancia a cualquier punto de la geometría <i>a</i> es menor o igual a la distancia <i>d</i> . La geometría <i>c</i> conserva los atributos de <i>a</i> .

Tabla 4.15 Relación de funciones de análisis espacial.

- *GeoJSON*: especificación geoespacial del lenguaje de Notación JavaScript de Objetos (*JSON*) que es especialmente adecuada para la transmisión de información debido a su ligereza [But08].
- *GeoRSS*: estándar emergente para la codificación y difusión de información de carácter geográfico –geometrías y propiedades asociadas– en un canal o fuente *Web* –*Web feed*–. Es compatible con

²¹ Función no disponible en los estándares *ISO 13249-3* y *OGC* [Her11].

tres especificaciones de canales Web: *RSS*²² –versiones 1.0 y 2.0– y *Atom*²³. Por otro lado dispone de dos variantes de codificación: *GeoRSS Simple* y *GeoRSS GML* –este último es un perfil de aplicación del estándar *OGC GML* que permite la descripción de un mayor rango de tipologías de geometrías que la versión *Simple* y permite además la expresión de sus coordenadas en sistemas de referencia de coordenadas (*CRS*) diferentes de coordenadas geográficas en datum *WGS84*–.

- *GPX* –*GPS Exchange Format*–: lenguaje de marcado basado en *XML* diseñado para el intercambio de datos *GPS* entre diferentes aplicaciones. Permite describir puntos de paso –*waypoints*–, posiciones del dispositivo –*tracks*– y rutas –*routes*–.
- *KML*: lenguaje de marcado basado en *XML* para representar datos geográficos que fue desarrollado inicialmente como formato de datos de la herramienta de visualización 3D *Keyhole LT*, propiedad de la empresa *Keyhole Corporation* que fue adquirida en 2004 por *Google Incorporated*. Actualmente es el formato para el intercambio de datos de *Google Maps* y *Google Earth*, siendo por ello un estándar de facto que en el año 2008 fue presentado como estándar oficial del *OGC* [Wil08].
- *KMZ* –*Keyhole Markup Language Zipped*–: variante comprimida del formato *KML*.
- *SHP* –*ESRI Shapefile*–: formato binario de almacenamiento de IG vectorial desarrollado por *ESRI* y presentado con la aplicación *ESRI ArcView*. Es, sin género de duda, el estándar *de facto* en el ámbito de los SIG por antonomasia.
- *WKT* –*Well Known Text*–: codificación en formato *ASCII* estandarizada por el *OGC* [Her11] –y posteriormente por *ISO* mediante la norma *ISO 19125-1:2004*– para la descripción de objetos espaciales de tipo vectorial, siendo su sintaxis muy fácil de utilizar, de forma que es muy generalizado su uso en la industria, especialmente en el ámbito de las bases de datos espaciales.
- *WKB* –*Well Known Binary*–: variante codificada en binario del formato *WKT*.

²² La Sindicación Realmente Simple (*RSS*), en sus dos versiones, constituye el estándar industrial –o *de facto*– más extendido para sindicarse o compartir contenido en la web. Su finalidad es la de establecer un formato de codificación de contenidos utilizado por los proveedores de contenidos para la difusión de los mismos al conjunto de usuarios suscriptores. Existen programas, denominados “agregadores”, que recapitulan información de diferentes canales *RSS* y los integran en una interfaz Web para la consulta de los usuarios lectores de esos “agregadores” –*Bloglines*, *MyYahoo*, son dos ejemplos representativos–.

²³ *Atom* se refiere a un par de estándares desarrollados por la organización de estándares denominada *Internet Engineering Task Force (IETF)*. El primero de ellos, *Atom Syndication Format* es un formato expresado en lenguaje *XML* para la sindicación de contenidos. El segundo, *Atom Publishing Protocol*, es un protocolo basado en *HTTP* para la creación y actualización de recursos.

4.3.4.2. Análisis espacial raster

La necesidad de disponer de herramientas de análisis sobre datos *raster* es cada vez mayor, dada la elevada profusión de información de este tipo –motivada especialmente por el creciente número de satélites operativos y por los avances en las técnicas de captura aplicadas a los vuelos fotogramétricos–.

Las funciones de análisis *raster* pueden agruparse atendiendo a diferentes criterios, aunque el más habitual es por la similitud del método de examen y tratamiento de los datos que realizan. En este sentido, pueden diferenciarse las siguientes categorías [Coc08]:

- Locales: funciones en las que se realizan cálculos que tienen que ver con el contenido específico de cada píxel, sin tener en cuenta los que le rodean. A su vez se diferencian dos subcategorías:
 - Reclasificación: permiten obtener una nueva imagen a partir de una dada. Para ello actúan:
 - Recodificando los valores o categorías existentes.
 - Concentrando las categorías en un número más reducido –utilizando variables cualitativas–.
 - Agrupando los valores existentes por intervalos –utilizando variables cuantitativas–.
 - Modificando los valores de las celdas mediante una operación algebraica de una constante o la aplicación de una función determinada –trigonométrica, etc.–.
 - Superposición: permiten obtener una nueva imagen a partir de dos o más dadas. Esta tarea puede abordarse realizando una:
 - Superposición lógica: se parte de unas imágenes iniciales de tipo binario –cada píxel debe tener un valor asignado igual a 0 o a 1–. Las celdas de valor igual a 1 indican que se cumple una determinada condición, al contrario que las restantes. Este tipo de superposición –equivalente al *and* lógico– permite localizar las áreas de un territorio en las que se cumplen dos condiciones a la vez –aquellos píxeles que en ambas imágenes tienen valor igual a 1–.
 - Superposición algebraica: permite obtener una nueva imagen o capa de información mediante operaciones matemáticas –suma, resta, multiplicación, división y otras–.
 - Intersección o cruce de mapas nominales u ordinales: permite conseguir una nueva imagen a partir de dos imágenes cuyos valores temáticos pertenezcan a una escala nominal u ordinal, permitiendo encontrar los posibles cruces de categoría de las capas de origen. Son muy recomendables para la realización de estudios sobre la dinámica del territorio.

- Enmascarado de mapas: tipología en la que una de las imágenes originales se coloca encima de otra, haciendo desaparecer los valores existentes en la anterior –para todos los píxeles de la capa superior que tienen un valor no nulo, para los restantes se mostraría el valor asociado en la capa inferior–.
- Vecindad: funciones en las que se realizan cálculos para cada píxel teniendo en cuenta el valor de los píxeles próximos. Se agrupan a su vez en funciones de vecindad:
 - Inmediata: calculan el valor de cada celda en función de los valores que tienen las más próximas. Entre las funciones más habituales en el ámbito geoespacial se pueden citar:
 - Filtrado: determinan el valor de cada celda mediante una función –media ponderada o no, moda, mediana, etc.– del valor asociado a cada una de las ocho que la rodean.
 - Mapa de pendientes: permiten obtener la pendiente –en una imagen correspondiente a un MDT o a un mapa de pendientes–, obteniendo el valor del desnivel relativo existente en cada celda analizado el valor en la misma junto con el de las ocho que la rodean, o bien con las cuatro situadas en los puntos cardinales.
 - Orientación de la pendiente: permiten el valor del ángulo de inclinación del plano de máxima pendiente, respecto a un punto cardinal de referencia.
 - Extendida: calculan el valor de cada celda en función de celdas próximas, pero que no tienen porqué ser contiguas. A título ilustrativo, entre las funciones más habituales de esta categoría encontramos:
 - Mapa de distancias: muestran para cada celda la distancia a una escogida como referencia. La distancia se calcula midiendo la dimensión existente en unidades de celdas –determinadas éstas siguiendo la dirección horizontal, vertical y/o diagonal, en función de la geometría del *raster*– multiplicada por la resolución de cada celda.
 - Polígonos Thiessen: consiste en la asignación por proximidad de las celdas de un determinado territorio a una serie de puntos –celdas aisladas– previamente identificados. De esta forma se determina el área de influencia de cada uno de los puntos.
 - Análisis de intervisibilidad: permite diferenciar aquellas celdas que son visibles desde un punto –celda– determinado.
- Zonales: funciones en las que los cálculos se relacionan con la geometría de los objetos geográficos –zonas– a diferenciar, mediante el reconocimiento de los mismos y el cálculo de sus características de forma y tamaño. Así, la identificación de zonas supone la delimitación de áreas homogéneas, formadas por celdas contiguas, atendiendo a una componente temática.

Partiendo de la categorización mostrada, es posible desarrollar una extensa panoplia de funciones de análisis *raster*, de las cuales en este espacio se han recogido algunas de ellas que se añadirán al catálogo funcional básico. Aparte, también es necesario indicar que es funcionalmente muy interesante disponer de funciones de análisis espacial como las recogidas en la Tabla 4.15 que puedan ejecutarse combinando información vectorial y *raster*, pudiendo asignar, por ejemplo, a cada punto de una geometría resultante *c* proveniente de *a* el valor de la celda de una capa *raster* superpuesta.

4.3.4.3. Transformación de coordenadas

Aunque el catálogo final de funciones de un SIG distribuido concreto debe ajustarse a las necesidades del negocio concreto de la organización que lo quiera poner en marcha, en este estudio se completa la colección de servicios básicos de geoprocetamiento con tres servicios adicionales. El primero viene recogido en la especificación de servicios de red de *INSPIRE* [NSD08], la cual contempla la creación de servicios de transformación de coordenadas [NSD10], basándose en el borrador publicado por el OGC [Dal01].

Como ya se ha referido anteriormente, *INSPIRE* aborda la creación de una IDE europea, superando la problemática existente a la hora de intentar gestionar IG proveniente de los diferentes países miembro de la UE. Entre los problemas detectados en aras a garantizar la interoperabilidad, la heterogeneidad de los modelos de datos es uno de los problemas que se enfrentaron con mayor celeridad, fructificando en los diferentes anexos de la normativa que fijaban conjuntos temáticos –ver Tabla 4.12–. En el primer anexo, figuraban los *CRS* y los sistemas de cuadrículas geográficas, puesto que se consideraban como datos de referencia –datos que constituyen el marco de referencia espacial al que deben enlazarse los restantes conjuntos de información temáticos–, aunque a diferencia de los otros conjuntos de dicho anexo, estos no pueden ser visualizados o descargados –por eso no fueron incluidos en la citada tabla–. En este aspecto, el equipo de trabajo de *INSPIRE* –*thematic working group*– encargado de estos dos conjuntos relacionados con la geodesia, acabó dictaminando que [INS08]:

- Para la componente horizontal se dispone la utilización del Sistema de Referencia Terrestre Europeo de 1989 (*ETRS89*) en aquellas zonas en las que sea aplicable. En las áreas restantes se utilizará el Sistema de Referencia Terrestre Internacional (*ITRS*) u otro *CRS* geodésico compatible con el anterior.
- Para la componente vertical se dispone el uso del Sistema de Referencia Vertical Europeo (*EVRS*) en aquellas zonas en las que sea aplicable, quedando para las restantes la posibilidad de utilizar otros sistemas de referencia verticales (*VRS*).
- En cuanto a las proyecciones cartográficas se dispone la utilización de las siguientes:
 - Azimutal equivalente de Lambert a efectos de análisis espacial y visualización.
 - Cónica conforme de Lambert para cartografía a escalas iguales o inferiores a 1:500.000.
 - Transversal de Mercator para cartografía conforme a escalas mayores de 1:500.000.

Sentadas las bases que favorecen la interoperabilidad es preciso que los publicadores de información en Internet dispongan de servicios de procesamiento que permitan realizar conversiones de datos entre distintos CRS, incluyendo entre ellos los que se acaban de reseñar. En España esta problemática se ha evidenciado especialmente al tener que realizar una transición del antiguo CRS utilizado, basado en el dátum europeo del año 1950 (*ED50*), al nuevo *ETRS89*. Por todo lo expuesto, es preciso que la plataforma SIG distribuida incluya este servicio, pudiendo realizar las conversiones en cualquiera de los formatos vectoriales indicados para las operaciones de análisis espacial –*GML*, *GeoJSON*, *GeoRSS*, *GPX*, *KML*, *KMZ*, *SHP*, *WKB* y *WKT*–. En el caso de estudio analizado, el ICGC dispone de un servicio *WPS* que realiza este proceso de transformación, aunque solo se permite introducir un par de coordenadas x e y en cada petición –*IcgcTransfCoord*, tal y como se recoge en el Anexo A–.

4.3.4.4. Conversión de formato

El segundo de los geoservicios adicionales debe afrontar uno de los problemas tradicionales de los SIG, la multitud de formatos existentes y la conversión de datos entre los mismos. Dadas las restricciones existentes en los formatos de entrada válidos para los parámetros de las funciones de procesamiento –en este trabajo se han escogido seis–, la posibilidad de encadenar la salida de un servicio de conversión de datos a la entrada de uno de análisis topológico, por ejemplo, amplía el abanico de usuarios/aplicaciones que pueden favorecerse incrementando sus posibilidades de procesamiento remoto.

A título ilustrativo, el ICGC publica una aplicación Web denominada *ConvertBox*²⁴ que, utilizando la librería *GDAL*, permite realizar conversiones de coordenadas –de forma similar a como se realiza en el servicio *IcgcTransfCoord*, aunque admitiendo varios pares de coordenadas en una única petición– y conversión de formatos –*ESRI Shapefile*, *Downers Grove North (DGN)*, *Drawing Exchange Format (DXF)*, *Mapinfo*, *KML*, *GPX*, *GML*, *GeoJSON*, *GeoRSS*, *AVCBin* y *EOO* de *Arc/INFO*–, con la posibilidad de realizar simultáneamente un cambio de CRS.

4.3.4.5. Geocodificación

El último servicio de geoprocésamiento que se incorpora es el de geocodificación, con sus variantes directa e inversa. Esta tipología de servicio tiene una demanda creciente, puesto que son innumerables las bases de datos que registran información vinculada a direcciones postales –abordando el ámbito urbano con las calles y el interurbano con la red viaria de carreteras– y la necesidad de representar dicha información de forma espacial es prioritaria, puesto que permite abordar problemas de planificación muy reseñables –distribución de servicios públicos, *geomarketing*, etc.–. Los principales problemas de este tipo de funciones vienen dados por la:

- Diversidad de formatos en los que se pueden recoger los datos sobre las direcciones postales, incluyendo la definición de diccionarios de tipos de vías heterogéneos –en cada modelo se ha definido un diccionario de tipos de vía válidos–.

²⁴ <http://betaportal.icgc.cat/convertbox/>. Consulta realizada el 15 de septiembre de 2014.

- Peculiaridades propias del idioma: es muy importante que un geocodificador permita realizar búsquedas de vías de forma no literal, admitiendo variantes fonéticas e incluso la introducción de errores ortográficos. Los motores de geocodificación suelen estar perfectamente calibrados para el idioma inglés y es imprescindible introducir variantes que lo hagan aplicable a otros contextos lingüísticos. A título ilustrativo, el motor de geocodificación de *ESRI ArcGIS Desktop* actuando sobre la base de vías del ICGC ofrece una puntuación mayor –mayor grado de aproximación– al texto “Josp Tarradellas” que a “José Tarradellas” cuando se hace referencia a la vía “Josep Tarradellas”. La primera expresión es incorrecta, la segunda recoge el nombre en español, mientras que en la base de datos está almacenado en catalán.

Por tanto, resulta imprescindible que este servicio se fundamente en un motor suficientemente versátil como para poder ser personalizado para las características lingüísticas y de formato propias del modelo de datos al que realice las consultas.

4.4 Diseño arquitectónico

Establecida desde un punto de vista funcional la plataforma, en este apartado se aborda la definición del modelo arquitectónico que ha de sustentarla. Dicha definición debe incorporar mecanismos que permitan la escalabilidad de la misma desde un punto de vista funcional y operativo, para lo cual es imprescindible que se investiguen los siguientes aspectos:

- Revisión de los protocolos de publicación de los servicios, tal y como se ha referido anteriormente en este estudio, los estándares son imprescindibles para entender el avance que ha experimentado la gran mayoría de los diferentes ámbitos empresariales de implantación de SI. Su utilización en un SIG distribuido es, consecuentemente, irrenunciable. Sin embargo, también es necesario tener presente el coste en términos de complejidad que viene asociado con la utilización de los mismos, hecho especialmente relevante en el ámbito de la IG. Por tanto, es preciso realizar una labor de adaptación de los estándares promulgados por el OGC a las corrientes actuales de desarrollo.
- Inclusión de mecanismos de seguridad que permitan el establecimiento de canales de comunicación seguros entre clientes y servidores basados, en la medida de lo posible, en las opciones que ofrecen de forma nativa los propios protocolos. De esta forma, se habilita que den cabida a servicios que lleven implícitas restricciones de acceso.
- Incorporación de componentes que faciliten el descubrimiento y la composición u orquestación de los servicios, facilitando de esta forma la visibilidad de los servicios más allá de las opciones que ofrecen las IDE y favoreciendo la creación de servicios de procesamiento complejos –basados en el encadenamiento de los procesos de grano fino, mayoritarios en los catálogos de geoservicios–.

Los siguientes apartados abordan el análisis de estos tres factores, apoyándose el estudio en las introducciones teóricas realizadas en el capítulo 3, el marco regulatorio establecido en 4.1 y el catálogo funcional definido en 4.3.

4.4.1 Modelo de referencia

El primer aspecto que es preciso definir es el modelo arquitectónico de referencia. En los apartados 3.1 y 3.2 se plantearon las características principales de los dos modelos que tienen mayor implantación: *SOA* y *ROA*. La selección de uno de ellos conlleva un análisis comparativo que tiene que ir más allá de los principios en los que se sustentan, abordando las características conceptuales y las consideraciones de carácter tecnológico.

En este sentido, la amplia comunidad de investigadores que ha estudiado ambos enfoques arquitectónicos pone de manifiesto posiciones enfrentadas, fundamentadas en argumentos poco precisos o que, si más no, tienen una importante componente subjetiva.

Abundando en lo reseñado, es digno de mención el trabajo presentado por los autores de [Pau08], puesto que expone una amplia y detallada aproximación a la problemática expuesta, realizando una comparación entre los protocolos que dan cobertura a *SOA* –personalizados en *SOAP* y la pila de protocolos del *WS-I* y *ROA* –*REST*–. Dicha comparativa se articula en torno a cuatro niveles de abstracción, para los cuales se muestran de forma resumida en la Tabla 4.16 las conclusiones obtenidas.

Principios arquitectónicos

Descripción

Es en este nivel en el que se centran la mayoría de los estudios existentes, comparando los fundamentos y requerimientos arquitectónicos de ambos modelos. Su estudio centra la investigación en este ámbito en la descomposición en capas, la gestión de la heterogeneidad y el bajo nivel de acoplamiento.

Conclusiones

- Las dos aproximaciones tienen características cuantitativas similares.
- *REST* utiliza la Web como un medio para poner los servicios al alcance de los clientes, considerándose *HTTP* como un protocolo de aplicación. Por su parte, *SOAP+WS-I* contemplan la Web como un medio de transporte universal que permite realizar túneles entre proveedores y consumidores, atravesando los diferentes componentes de la red.
- Los esfuerzos de estandarización concretados en la pila de protocolos *WS-I* van más allá de la mera sincronización que se gestiona en *REST*, englobando aspectos que permiten la integración de aplicaciones técnicamente muy complejas y heterogéneas.
- Debido a la sincronización requerida en *REST*, los clientes no pueden continuar utilizando un servicio cuando este no se encuentra temporalmente disponible y es el cliente el que tiene que gestionar este contratiempo. En la aproximación *SOA*, en cambio, el bus de servicios permite realizar una gestión de los mensajes que habilita su transmisión a través de canales persistentes y fiables.
- La característica de uniformidad de la interfaz *REST* garantiza un mínimo grado de acoplamiento entre cliente y proveedor, puesto que las operaciones disponibles son las que recoge el protocolo *HTTP*. En este sentido, esta aproximación es más ventajosa que la de *SOAP+WS-I*, aunque también cabe tener en cuenta que, aparte de la interfaz, es necesario observar el modelo de datos de los parámetros de los mensajes, puesto que éste es el que define el formato del contenido del mensaje en cada una de las operaciones; aspecto que no queda garantizado en ninguna de las aproximaciones.

Aspectos conceptuales

Descripción

Se compara cómo se realiza la gestión de los servicios desde un punto de vista contractual y se discuten las similitudes y diferencias en cuanto a las metodologías que siguen ambos modelos para la publicación de servicios.

Conclusiones

- La aproximación basada en *SOAP* y la pila de protocolos *WS-I* implica tener que tomar un menor número de decisiones con respecto a la arquitectura del sistema, aunque en cada una de ellas existen más alternativas.
- *REST* muestra una interfaz más sencilla que *SOAP+WS-I*, puesto que contiene una lista cerrada de operaciones, aunque sigue siendo preciso especificar qué acciones están disponibles para cada uno de los recursos y cuál es la semántica aplicable a cada una de ellas.
- En *REST* el diseño de un *WS* implica establecer la norma para identificar los recursos, establecer el diseño de las *URI*, decidir las connotaciones semánticas de la interacción con los recursos, así como la definición de las relaciones existentes entre los mismos. Por el contrario, el diseño en *SOAP+WS-I* se centra en el diseño del contenido de los mensajes, los patrones de intercambio de dichos mensajes y la determinación de las operaciones disponibles en los servicios.

Aspectos tecnológicos

Descripción

Se analiza la implementación desde un punto de vista técnico de ambos modelos. Para ello se analiza el protocolo de transporte, el formato empleado para enviar la información, la identificación, descripción, fiabilidad, seguridad y composición de los servicios; así como la posibilidad de realizar transacciones.

Conclusiones

- Ambas aproximaciones implican tomar el mismo número de decisiones, pero la opción *REST* dispone de menos alternativas.
- La aproximación *SOAP+WS-I* conlleva independencia con respecto al protocolo de transporte –aunque tratándose de *WS* esta decisión queda automáticamente restringida al protocolo *HTTP*–, el cual debe ser identificado junto con otros aspectos relacionados con la calidad del servicio, la seguridad y la política de transacciones. En cambio, la aproximación basada en *REST* ofrece muchas menos posibilidades.
- En la aproximación *SOA* solo existe un formato de mensajes estandarizado: *SOAP*. En cambio, en el ámbito *REST* se ofrece la posibilidad de utilizar cualquier tipo de formato *MIME*. Este hecho puede complicar la comunicación entre proveedor y cliente, puesto que el cliente tiene que adaptarse a alguno de los formatos publicados por el proveedor. Además, el hecho de poder disponer de varias representaciones de un mismo recurso implica un esfuerzo de mantenimiento mayor.
- La identificación de los servicios es mucho más sencilla en *REST*, puesto que los propios *URI* encapsulan toda la información necesaria para su determinación unívoca en la Web. En *WS-I* se ha realizado un esfuerzo para diseñar un estándar para la identificación de los recursos, denominado *WS-Addressing*, pero no es tan conciso y legible como el sistema de *REST*.
- La descripción de los servicios en *WS-I* se basa en el estándar *WSDL*, gracias al cual los desarrolladores pueden generar de forma automática el código de aplicaciones cliente. En el ámbito *REST* la descripción se hace mediante la documentación de la *API* de los servicios, de forma más orientada a humanos. Se ha intentado abordar esta problemática con la especificación *WADL* y con la versión 2.0 de *WSDL* que también

permite describir servicios *REST*.

- Los mecanismos de seguridad, fiabilidad y gestión de transacciones que facilita *SOAP+WS-I* son mucho más versátiles y potentes que los existentes para *REST*, basados en las opciones que ofrece *HTTP* y *HTTPS*.
- La falta de descripción formal de las interfaces en *REST* y la posibilidad de no utilizar siempre mensajes *XML*, provoca que sea mucho más compleja la utilización de técnicas de composición basadas en *WSDL* como la propuesta por el estándar *WS-BPEL*, en contraposición a los servicios basados en *SOAP+WS-I*.

Herramientas disponibles

Descripción

Se centra en la evaluación de las herramientas existentes en el mercado para implementar los aspectos analizados en los niveles anteriores.

Conclusiones

- Existen numerosas plataformas de *WS* para ambas aproximaciones, así como librerías para el desarrollo de clientes de *WS*.

Tabla 4.16 Comparativa de los protocolos arquitectónicos de *SOA* y *ROA*.

Como consecuencia, se puede inferir que la simplicidad asociada a *REST* debe ser entendida desde un punto de vista cuantitativo, puesto que el mayor número de capas que componen la opción *SOAP+WS-I* implica tomar un mayor número de decisiones desde un punto de vista arquitectónico. Sin embargo, si se precisa la funcionalidad ofrecida por los protocolos *WS-I*, la implementación de mecanismos similares en *REST* puede resultar extremadamente compleja si se quiere garantizar la interoperabilidad. Por todo ello, parece razonable pensar que en la actualidad sigue siendo más “sencillo” construir soluciones empresariales basadas en el modelo *SOA*, tanto más cuanto más complejos sean los procesos de negocio a modelar. En cambio, para aquellos escenarios menos exigentes desde un punto de vista funcional, *ROA* es una alternativa con un nivel de madurez y sencillez que la hace más asequible.

Centrando el análisis en el mundo de las IDE, tal y como se ha recogido en el punto 4.1.3.6, se impone el modelo *SOA* como arquitectura para su materialización. Esta tipología arquitectónica es la que goza de más adeptos dentro de las soluciones empresariales en las que se precisa exponer una lógica de negocio compleja a la que es posible dar respuesta mediante la articulación ordenada –y preferiblemente dinámica– de distintos servicios de grano fino. Sin embargo, la complejidad inherente a la implantación de una arquitectura *SOA* puede resultar una barrera difícilmente franqueable para determinadas organizaciones, salvo que suavicen los requerimientos y se acabe realizando una implementación que no saque especial partido a la potencialidad que ofrece este modelo. Este es el caso de la mayoría de las implantaciones *SOA* existentes en el ámbito geográfico, las cuales son utilizadas simplemente para la publicación de servicios que habilitan funciones básicas como la consulta de IG y su descarga, sin plantearse la adopción de herramientas para la gestión de la calidad del servicio, la seguridad, las transacciones o la composición de servicios, por citar algunos ejemplos. En este contexto, la adopción de la alternativa arquitectónica *ROA* debe ser considerada.

A efectos prácticos, por tanto, no resultaría apropiado establecer un diseño arquitectónico que apostará exclusivamente por una de las variantes, *SOA* o *ROA*, aunque son modelos que, por si solos, bastarían

para cimentar una arquitectura SIG distribuida. Como ya se ha reseñado, es numerosa la literatura existente en favor de uno y otro modelo arquitectónico y dado que se pretende ofrecer una plataforma con un abanico de geoservicios notablemente heterogéneo, es preciso abordar un estudio de los diferentes estándares a utilizar para su publicación, desde el punto de vista de sus interfaces. El objetivo es caracterizar en cada caso las distintas interfaces y los cambios que implican en la especificación vigente. Por ello, en los siguientes puntos se analizarán las características de las interfaces previstas por los distintos geoservicios descritos en el punto 4.1, en relación a las dos arquitecturas para SI distribuidos que se han estudiado: *SOA* y *ROA*.

4.4.2 Servicio de visualización *WMS*

La especificación *WMS* definió desde su origen una interfaz que utiliza el protocolo *HTTP* en dos modalidades diferenciadas, cada una con una codificación específica –ver Tabla 4.17–.

Método	Protocolos	Codificación
GetCapabilities	<i>HTTP (Get)</i>	<i>KVP</i>
	<i>HTTP (Post)</i>	<i>XML</i>
GetMap	<i>HTTP (Get)</i>	<i>KVP</i>
	<i>HTTP (Post)</i>	<i>XML</i>
GetFeatureInfo	<i>HTTP (Get)</i>	<i>KVP</i>
	<i>HTTP (Post)</i>	<i>XML</i>

Tabla 4.17 Protocolos y codificaciones admitidas en las operaciones *WMS*.

Como se puede observar, no se recoge una interfaz para el protocolo *SOAP*, sin embargo la estructura –tal y como se observará en el análisis recogido en 4.4.2.4.1– permite la utilización de este protocolo sobre *HTTP* como protocolo de transporte. Asimismo, tampoco se recoge una interfaz *REST* y para calibrar el grado de convergencia de los métodos definidos con esta interfaz, en los puntos siguientes se analiza la adecuación de esta especificación a las restricciones que esta impone, según se recoge en 3.2.2.

4.4.2.1. División cliente-servidor

Esta restricción la cumple el estándar *WMS*, al igual que el resto de estándares *OGC* analizados en este estudio, puesto que los clientes y los servidores compatibles interactúan mediante el paradigma de *WS* [Pap08], el cual promueve una clara distinción entre consumidores y proveedores en términos de roles y funcionalidades. La adición de una nueva capa a una instancia de un servicio *WMS* no implica realizar cambios en la funcionalidad de la interfaz de usuario, ya que ésta permanece inalterada.

4.4.2.2. Interacciones sin estado

Según se recoge en [Per02], la comunicación entre clientes y servidores en cualquier servicio *OGC* es por naturaleza sin estado. En general, la interacción con el servicio sigue el patrón petición-respuesta en el que no existe dependencia alguna con interacciones previas. Una de las consecuencias de esta falta de estado es la ausencia de sincronización entre peticiones y respuestas, tal y como se ha referido anteriormente en el análisis del diagrama de secuencia.

4.4.2.3. Existencia de caché

La capacidad de activar de forma selectiva el almacenamiento de las respuestas, de forma que los clientes puedan reutilizarlas posteriormente ante la necesidad de plantear las mismas peticiones, no se contempla en la especificación del estándar *WMS*.

Para evaluar las consecuencias de esta limitación es importante entender cuando es interesante disponer de opciones de cacheado de las respuestas. En este sentido, el cacheado permite aumentar el rendimiento de una arquitectura Web cuando se trabaja con información estática, cosa que es altamente improbable para las peticiones de tipo *GetMap* y *GetFeatureInfo*, aunque no cuando se trata del método *GetCapabilities* –ver punto 4.4.2.4.1–.

En el caso de las peticiones *GetMap*, dado el conjunto de parámetros que determinan las características específicas del mapa solicitado –ver la Tabla 4.18–, es francamente improbable que se produzcan dos peticiones exactamente iguales y, de producirse, cabría analizar si el tiempo que ha transcurrido entre ambas es razonable en referencia al periodo de almacenamiento en caché de los mapas producidos.

Parámetro	Obligatorio	Descripción
VERSION	✓	Versión del estándar en la que se realiza la petición.
REQUEST	✓	Método utilizado.
LAYERS	✓	Capas del servicio solicitadas.
STYLES	✓	Estilos aplicados a la representación de las capas.
CRS	✓	Sistema de referencia en el que se expresan las coordenadas.
BBOX	✓	Área geográfica solicitada.
WIDTH	✓	Ancho en píxeles de la imagen correspondiente al mapa solicitado.
HEIGHT	✓	Alto en píxeles de la imagen correspondiente al mapa solicitado.
FORMAT	✓	Formato de imagen en el que se solicita el mapa.
TRANSPARENT	✗	Aplicación de transparencia a los píxeles de la imagen que se correspondan con zonas en las que no hay información en el mapa.
BGCOLOR	✗	Color a aplicar a aquellos píxeles de la imagen que se correspondan con zonas en las que no hay información en el mapa.
EXCEPTIONS	✗	Formato en el que se solicita la recepción de posibles excepciones en la gestión de la solicitud realizada.
TIME	✗	Valor temporal de la capa solicitada.
ELEVATION	✗	Elevación de la capa solicitada.

Tabla 4.18 Relación de parámetros asociados a una petición *GetMap* *WMS*.

Esta situación solo sería razonable plantearla en un escenario en el cual los clientes utilicen aplicaciones en las que el desplazamiento por el territorio se realice siguiendo un patrón fijo –una cuadrícula espacial– que impida la infinita variabilidad que permite la simple combinación de coordenadas mínimas y máximas asociadas al parámetro *BBOX*. En este caso, cabe tener en cuenta que el protocolo *HTTP* dispone de mecanismos para el control de las cachés, basados en la utilización de diferentes cabeceras –*headers*– tanto en las peticiones como en las respuestas. En lo que respecta a las peticiones, las cabeceras disponibles son:

- *If-Modified-Since*: el servidor solo devolverá el recurso solicitado si ha sido modificado desde la fecha y hora indicadas por el cliente.
- *If-None-Match*: el servidor solo devolverá el recurso solicitado si su etiqueta unívoca –*Etag*– actual es diferente a la suministrada por el cliente.
- *Cache-control*: permite especificar al cliente cuando solicita una copia del recurso, qué comportamiento espera en relación a la gestión de la caché.

Por otro lado, en la vertiente servidora, las opciones son:

- *Cache-control*: idéntico comportamiento que el referido anteriormente, pero por parte del servidor.
- *Expires*: facilita una fecha en la que el documento o recurso pasa a considerarse como obsoleto.
- *If-Modified-Since*: permite especificar el uso de la caché siempre que el recurso cacheado no haya sido modificado desde la fecha indicada en la petición.
- *Last-Modified*: permite al servidor indicar la fecha y hora en que el recurso fue modificado por última vez. Gracias a esta cabecera se pueden realizar peticiones condicionales con la cabecera *If-Modified-Since*.
- *Etag*: su funcionamiento es similar a *Last-Modified*, pero en este caso no se envía la fecha de modificación del recurso, sino una etiqueta única asociada a esa versión concreta del mismo.

Claro está, esta alternativa de gestión de caché implica ajustar el comportamiento de la aplicación servidora y/o de los clientes, aunque cabría analizar en cada caso si admite tal configuración.

4.4.2.4. Interfaz uniforme

La interfaz uniforme implica que el servidor expone una interfaz derivada de la semántica de los métodos del protocolo *HTTP*, actuando éste como protocolo de aplicación. En este caso, en cualquiera de las opciones de protocolo, método y codificación ofrecidas por *WMS*, no se cumple con este requerimiento en ninguno de los métodos del estándar –ver Figura 4.27–.

El análisis de los métodos presentes en el estándar *WMS* también permite referir ciertas consideraciones con respecto a las características de seguridad e idempotencia:

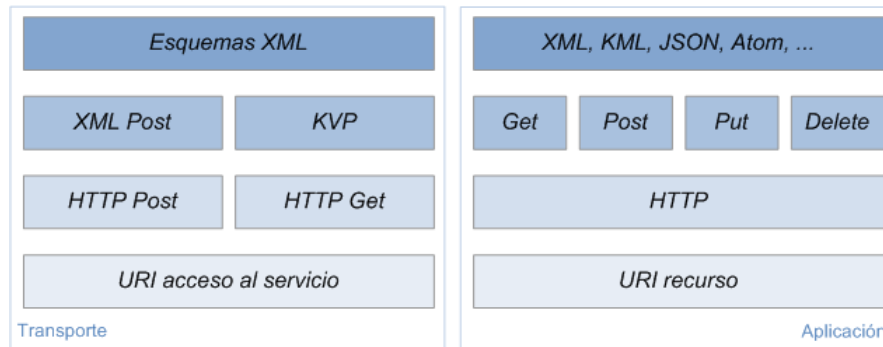


Figura 4.27 *HTTP* como protocolo de transporte –sin opción *SOAP*– / aplicación [fuente: propia].

- Seguridad: esta propiedad implícita a la utilización que hace *REST* del protocolo *HTTP* no se cumple en la especificación *WMS*. El incumplimiento se produce aunque se utilice un método como *GET*, que es por definición seguro en *REST*. El hecho de que la operación *GetMap* –ver Tabla 4.19– acceda a recursos que no existen físicamente y que serán fruto de operaciones que pueden implicar cambios en el estado del servidor, así lo pone de manifiesto –todo ello sin entrar en la infracción que supone, desde el punto de vista de *REST*, la utilización del método *POST* para el acceso a recursos–.
- Idempotencia: esta propiedad de la utilización que hace *REST* del protocolo *HTTP* sí se cumple en la especificación *WMS*, puesto que se puede afirmar que, aunque dicho estándar solo utiliza uno de los métodos que tienen esta propiedad según *REST* –*GET*–, tanto su utilización como la del método *POST* de forma reiterada en las distintas operaciones no conllevan resultados diferentes.

Entrando en los conceptos implícitos dentro de la interfaz, tal y como aparecen recogidos en 3.2.1, es interesante realizar el análisis que se acomete en los siguientes puntos.

4.4.2.4.1. Recursos

La especificación técnica del estándar *WMS*, debido a su concepción, realiza una utilización de las *URI* que es meridianamente diferente de la promulgada en *REST*. A título ilustrativo, la utilización del método *POST* del protocolo *HTTP* impide diferenciar entre distintas peticiones, de forma que no se garantiza que el acceso a cada hipotético recurso se realice mediante un *URI* unívoco, puesto que la identificación del recurso concreto asociado a cada petición se realiza a través del documento *XML* que se envía asociado a la petición –*payload*– y no en el *URI*. Este hecho no ocurre cuando se utiliza la implementación *GET*, aunque esto implique considerar la sucesión de *KVP* –*query string*– como parte del *URI*.

Por otro lado, si se analizan los diferentes métodos, podríamos identificar los recursos mostrados en la Tabla 4.19. En lo que respecta al método *GetCapabilities*, está perfectamente definido el recurso que subyace: el descriptor del servicio. En algunas aplicaciones servidoras –como el caso de *ESRI ArcGIS for Server*– se permite especificar la ubicación de dicho recurso, pudiendo diferenciar entre un descriptor externo –por tanto gestionado por el administrador del sistema– o interno –en este último caso el recurso es generado por el servidor cada vez que se recibe una petición asociada a este método–. El contenido de dicho descriptor está estrechamente vinculado con el servicio en cuestión y solo sería preciso

actualizarlo cuando se produjese alguna modificación en los contenidos publicados por el servicio. Esta actualización puede afectar a parámetros operativos del propio servicio o bien a alguna de las capas que lo componen. Pero, en cualquier caso, estamos hablando de intervenciones de carácter esporádico y asociadas a labores de mantenimiento, de ahí que sea una información candidata a ser almacenada en la caché del cliente, enlazando con lo expuesto en 4.4.2.3.

Método	Recurso	Representación
GetCapabilities	Descriptor del servicio	XML
GetMap	Mapa o documento cartográfico	A especificar por el servidor –parámetro <i>FORMAT</i> –
GetFeatureInfo	Entidad o píxel	A especificar por el servidor –parámetro <i>INFO_FORMAT</i> –

Tabla 4.19 Relación entre recursos, representaciones y métodos WMS.

El segundo método, *GetMap*, apunta también a la existencia de un recurso: el mapa. Sin embargo, analizando la idiosincrasia de los mapas publicados por los servicios WMS se puede observar la siguiente diferenciación, en cuanto a la tipología de las fuentes de información:

- Datos vectoriales: en este caso siempre se encuentra separada la geometría de los objetos geográficos –y sus correspondientes atributos alfanuméricos– de la representación. La totalidad de los principales formatos asociados tradicionalmente con los SIG –lógicamente incluidos los más extendidos como *shapefile* y *geodatabase* del fabricante *ESRI*– cumplen con esta característica. Más recientemente encontramos una excepción en el formato *KML* que sí reúne ambas componentes.

Las aplicaciones servidoras WMS disponen de herramientas específicas para formular la asociación entre ambos aspectos –en el caso del sistema de *ESRI*, *ArcGIS for Server*, este proceso se realiza con *ArcGIS for Desktop*; por lo que respecta al sistema de *Integrgraph*, *Geomedia Web Map*, el homólogo es *Geomedia Professional*–. Estas herramientas permiten crear un documento –en la jerga de *ESRI* se conoce como “proyecto” y es un fichero de extensión “*mxd*”– en el cual se establecen, entre otras cosas, las distintas capas que formaran parte del documento cartográfico y sus correspondientes simbologías. Este fichero se utilizará durante el proceso de creación del servicio para indicar aspectos como la información que publicará el servicio en cuestión y la forma en la que será representada la misma. De esta forma, cuando el servidor reciba una petición, se generará un proceso interno que:

- Filtrará la información afectada según la zona geográfica y las capas especificadas en la petición.
- Aplicará la simbología convenida en el proceso de creación del servicio.
- Generará el resultado final en forma de imagen, cuyo formato se corresponderá con el que se haya especificado en la petición –debiendo ser uno de los que soporta el servidor–.

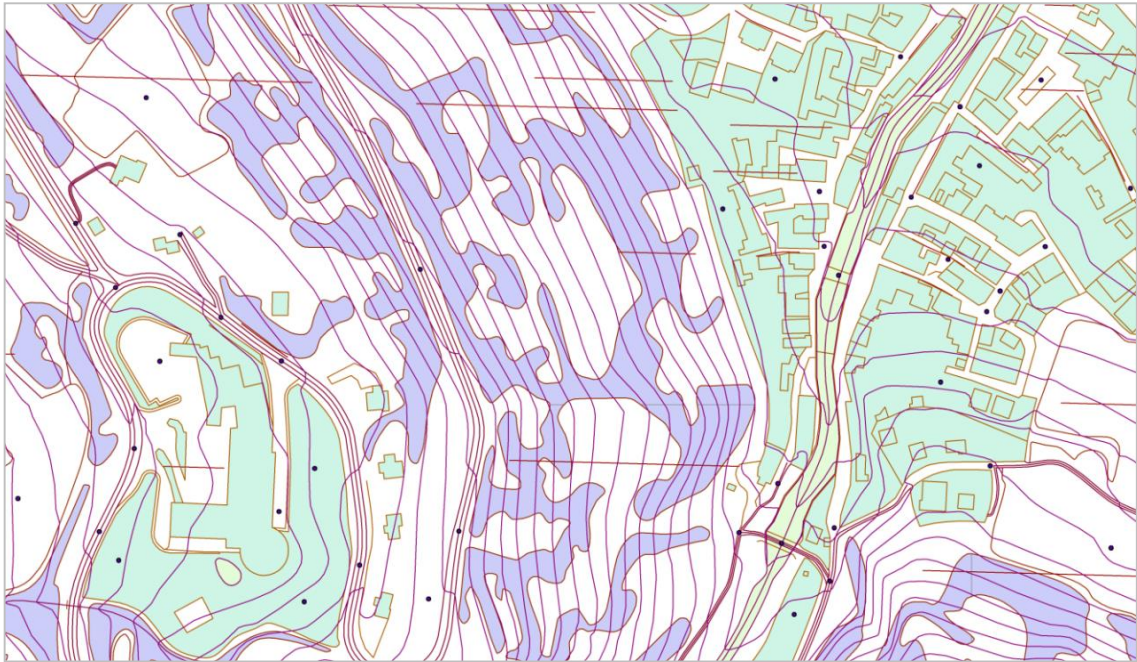


Figura 4.28 BTC 1:5.000 –simbología aleatoria–25 [fuente: propia].

La Figura 4.28 muestra una representación de los objetos geográficos que forman parte de la BTC a escala 1:5.000, almacenados en formato *geodatabase*. Los vectores correspondientes a dichos objetos son representados siguiendo una simbología que ha sido determinada de forma aleatoria por la aplicación con la que se han visualizado los datos –*ArcMap* versión 10.2 del paquete *ArcGIS for Desktop*–. Esta simbología es modificable y en ningún caso, tal y como se ha comentado, se almacenará con los datos vectoriales.

- Datos raster: en esta otra variante, por el contrario, las imágenes ya representan propiamente mapas o imágenes georreferenciadas. Éstos pueden ser: datos vectoriales convenientemente simbolizados para formar mapas, a los cuáles se les ha aplicado un proceso posterior de conversión a algún formato imagen, o imágenes desde un inicio –por ejemplo, procedentes de vuelos fotogramétricos o de satélites–. En el primero de los casos, aunque los datos en un origen fueran vectoriales, al tratarse directamente de formatos raster, en la petición no es posible especificar que se quieren ver solamente ciertas capas de información.

Imagínese, a modo de ejemplo, un mapa topográfico compuesto por 43 capas de información; de las cuáles 11 hacen referencia a aspectos altimétricos y toponímicos. Si se crean dos servicios, uno publicando directamente los datos en formato vectorial y aplicándoles una determinada simbología y otro partiendo del resultado anterior y convirtiendo *off-line* el mapa resultante a formato raster; en el primero, a título ilustrativo, en una petición se podría indicar que se quieren representar todas las capas, excepto las correspondientes a la altimetría y la toponimia –ver

²⁵ Información propiedad del ICGC.

Figura 4.29–; en el segundo, en cambio, no existiría esta opción y el mapa se serviría con todas sus capas de información –ver Figura 4.30–, tal y como fue creado en el proceso de conversión referido.

Por tanto, en esta segunda tipología de datos, ante una petición, el servidor simplemente tiene que encargarse de realizar una extracción de parte de la imagen original, teniendo en cuenta el ámbito geográfico solicitado y aplicarle técnicas de muestreo que permitan obtener la resolución implícita en la petición recibida²⁶. Sin embargo, en ambos casos se puede afirmar que el recurso como tal no existe físicamente hasta que se ha procesado la petición *GetMap* y es imposible, atendiendo a lo expuesto en 4.4.2.3 y dejando a un lado posibles actualizaciones de los datos recogidos en los repositorios de base de los que se pueda nutrir el servicio, generar todos los recursos de tipo mapa posibles.

Por lo que respecta al método *GetFeatureInfo*, la información a la que se accede puede ser:

- Un conjunto de datos alfanuméricos vinculados con las geometrías vectoriales ubicadas en la localización geográfica que se especifica en la petición. Los datos asociados a las geometrías se obtienen accediendo al repositorio en el que se encuentran almacenadas las mismas. En este caso, el recurso sería la entidad geográfica almacenada en el repositorio de datos y éste aglutinaría información sobre la geometría y los datos alfanuméricos que la describen.
- Valor asociado a la celda del fichero raster coincidente con la localización geográfica que se especifica en la petición. La información se obtiene accediendo al fichero raster y extrayendo directamente el valor de la celda –o píxel– en cuestión, siendo ella el recurso.

123

En este último método si podemos considerar que se accede a recursos que existen físicamente – vectores o celdas– y que, por tanto, pueden disponer de un identificador único. Cuando una petición de tipo *GetFeatureInfo* implique a varios recursos, por el ámbito geográfico especificado, se podría considerar la respuesta como un puntero al listado de *URI* correspondientes a los recursos en cuestión. Consecuentemente, después de analizar los distintos métodos, el único en el que no se puede asimilar un *URI* a un recurso es el denominado *GetMap*, ya que dicho identificador hace referencia a un punto de acceso a un servicio –tal y como puede observarse en la Figura 4.27– y en ningún caso apunta a un recurso físico, concluyéndose que se utiliza *HTTP* como mero protocolo de transporte sobre el cual se encapsulan peticiones en las que la semántica de las mismas reside en el *URI* –método *Get*– o en el fichero *XML* asociado –método *Post*–, respondiendo conceptualmente a una *RPC*.

Aparte de lo reseñado, cabe considerar las siguientes cuestiones adicionales:

- Gestión de los recursos: el estándar *WMS* no define un marco que permita la actualización, creación o eliminación de los recursos subyacentes. Por consiguiente, no es posible equiparar operaciones existentes en la actual especificación a los correspondientes métodos *HTTP POST*, *PUT* y *DELETE* para la gestión de los recursos –modificaciones, altas y bajas, respectivamente–.

²⁶ La resolución –expresada en unidades del mapa por píxel– en el eje de abscisas y en el de ordenadas se determina, respectivamente, mediante las fórmulas $resx = \frac{x_{max}-x_{min}}{width}$ y $resy = \frac{y_{max}-y_{min}}{height}$.

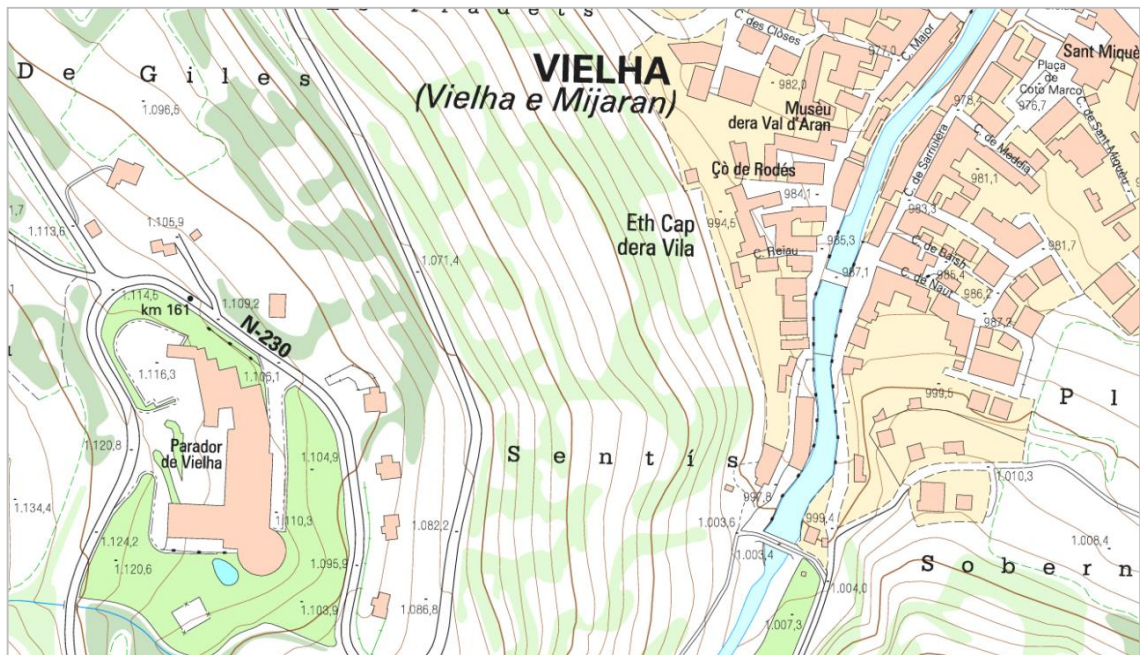
Figura 4.29 BTC 1:5.000 –filtrada–²⁷ [fuente: propia].

Figura 4.30 Mapa Topográfico de Cataluña (MTC) 1:5.000 [ICG14].

²⁷ Cartografía oficial propiedad del ICGC. La figura muestra una representación diferente a la de la figura anterior de la misma información en origen. La versión que se sirve atacando directamente los datos vectoriales corresponde con la base topográfica; la que se sirve previamente convertida a raster, hace referencia al mapa topográfico.

- Función desempeñada por el servicio: cuando se utiliza la variante *Post*, el servicio actúa como un intermediario –*proxy*– que recibe las peticiones y las redirige a los procesos internos adecuados. En cambio, en la variante *Get*, hay algunos autores [Gra12] que entienden que el *URI* de base atiende a un recurso. A juicio del que escribe, este *URI* tiene la misma consideración que el anterior, con la diferencia que es complementado con los parámetros *KVP* –estos últimos forman lo que se conoce como *query string*– del estándar *WMS*, dando lugar a un nuevo *URI*. Este último, tal y como se ha analizado anteriormente, se puede asimilar a un puntero a un recurso en los métodos *GetCapabilities* y *GetFeatureInfo*.
- Formato de los *URI*: existen interesantes reflexiones sobre la sintaxis que deben tener estos identificadores en el contexto *REST*. Es mucha la literatura existente al respecto – [Ric07], [Fre12] y [Fie00] son algunos ejemplos–, aunque a juicio del autor un *URI* que concite la esencia *REST* no debe utilizar el componente *query string*, de forma que el recurso sea accesible a través de un árbol de directorios único para cada uno de ellos, en lo que se denomina un *URI* autodescriptivo. En caso contrario, no se está hablando propiamente del *URI* de un recurso, sino del *URI* del punto de acceso a un servicio que procesará la petición y permitirá el acceso al recurso, si es que éste existe y no es generado sobre la marcha. En caso de que éste no exista, queda en evidencia que lo que realmente se está produciendo es la ocultación de una interfaz *RPC* bajo una falsa cobertura *REST*.

4.4.2.4.2. Representaciones

La existencia de éstas viene condicionada completamente a la existencia de los propios recursos. Por tanto, lo que aquí se refiera cabe contextualizarlo dentro de lo que se ha analizado en el ámbito de los recursos y el estándar *WMS*.

125

La Tabla 4.19 muestra los formatos en los que el estándar especifica que deben ser codificadas las respuestas a las tres tipologías de peticiones que éste contempla, es decir, las representaciones habilitadas. Como se puede observar, los métodos *GetMap* y *GetFeatureInfo* permiten al servidor especificar las representaciones que puede publicar y al cliente indicar cuál de ellas es la que necesita. Sin embargo, al igual que ocurre con el formato de los *URI* en el estándar *WMS*, también puede realizarse una reflexión similar en cuanto a las representaciones.

Existe un consenso [Fre12] entorno a la utilización preferente del formato *JSON* como respuesta a peticiones *REST*, por su sencillez, optando opcionalmente por la incorporación de un formato *XML* simplificado –sin esquemas y espacios de nombres, tan solo enlaces y datos–. Esta buena práctica no es observada en el estándar *WMS* para los siguientes métodos:

- *GetCapabilities*: la respuesta no incorpora la opción de publicar el recurso mediante una representación *JSON* y la versión *XML* no responde a los criterios de sencillez expuestos.
- *GetFeatureInfo*: no se especifica el formato, así que queda a criterio de los desarrolladores de los servidores decidir qué representaciones soportan, mientras que los clientes pueden especificar en las peticiones que realizan el formato deseado –parámetro *INFO_FORMAT* que debe corresponderse con alguno de los admitidos por el servidor en el fichero descriptor del servicio–.

Con respecto al método *GetMap*, dado que el recurso es un mapa, por tanto una imagen, no tiene sentido aplicar la práctica en cuestión, puesto que tanto *JSON* como *XML* no son formatos válidos para la distribución de imágenes.

4.4.2.4.3. Hipermedia

Las respuestas *XML* definidas en el estándar *WMS* incorporan en algunos casos enlaces a otros recursos. Sin embargo, en una implementación *REST* dichos enlaces han de ser la vía que permita al cliente conocer las transiciones válidas desde un estado determinado, lo cual supone una dimensión que no ha sido abordada dentro del estándar analizado.

De esta forma, a título ilustrativo, la respuesta a una petición *GetMap* podría llevar asociada, además de una imagen con el documento cartográfico, un documento *XML* con enlaces a los recursos correspondientes a las entidades mostradas en el mapa –equivalentes a las respuestas obtenidas a peticiones *GetFeatureInfo*–.

4.4.2.4.4. Mensajes autodescriptivos

Las consideraciones realizadas de los conceptos anteriormente analizados dentro de la restricción de interfaz uniforme, permiten anticipar que este estándar no contempla la utilización de mensajes autodescriptivos dentro de las respuestas diseñadas para las distintas operaciones. Al no tratarse de un estándar concebido para que sus mensajes puedan ser interpretados directamente por los posibles intermediarios, los cuales conocen el sentido habitual de cada uno de los métodos del protocolo *HTTP* y el uso de sus cabeceras, éstos no pueden habilitarse para la hipotética ejecución de ciertos procesos –a título ilustrativo la gestión de cachés intermedias–.

4.4.2.5. Estructuración en capas

Las arquitecturas distribuidas, como las que dan lugar a los entornos de publicación de geoservicios, muestran una distribución en capas. Dentro de estas capas, estos servicios se ubican en una intermedia, como mediadores entre los clientes y los repositorios de datos. En el ámbito *REST*, los mediadores pueden transformar de forma activa el contenido de los mensajes, ya que éstos son autodescriptivos –por ejemplo para su filtrado o cacheado–.

Desde la perspectiva del estándar *WMS* se podría plantear la existencia de nodos que realizaran ciertas tareas de balanceo de carga, transformación de los recursos –existentes previamente o generados a partir de llamadas *RPC* camuflados bajo un falso encapsulamiento *REST*–, etc. Este escenario sería especialmente interesante si se considera uno de los aspectos más problemáticos en la utilización de cartografía: la disparidad de sistemas geodésicos de referencia. Si los datos en el servidor están almacenados en un sistema concreto y el servidor no es capaz de procesar la transformación implícita al sistema solicitado por el cliente en la petición, esta labor podría quedar supeditada a la intervención de un mediador situado en otra capa que sí disponga de esta capacidad. Sin embargo, el estándar *WMS* no hace referencia alguna a este tipo de montaje y presupone que toda la lógica queda a merced del cliente. Si éste no puede representar información en los sistemas permitidos por el servidor –recogidos en el descriptor del servicio–, tendrá que optar bien por no hacer petición alguna, bien por hacerla en uno de

los admitidos y, posteriormente, derivar el resultado a algún servicio que realice la transformación. En este sentido, OGC ha publicado un estándar específico para la transformación de coordenadas – *Coordinate Transformation Service (CTS)* [Fie99]–, tal y como ya se ha referido anteriormente.

4.4.2.6. Código a demanda

La especificación *WMS* no recoge consideración alguna con respecto a la posibilidad de descargar código en el cliente para la ejecución del mismo en un nivel diferente del servidor, es algo que contraviene la propia idiosincrasia de este tipo de servicios en los que es la capacidad de cálculo del servidor la que es utilizada por los clientes.

4.4.2.7. Conclusiones

Recapitulando las consideraciones recogidas en el análisis del grado de cumplimiento de los principios de *REST* por parte de la actual especificación del protocolo *WMS*, se obtiene la Tabla 4.20 que permite colegir, sin género de dudas, que dicha interfaz no es compatible.

Principio	Cumplimiento	Aspectos clave
División cliente-servidor	✓	
Ausencia de estado	✓	
Existencia de caché	✗	Las características de las peticiones hacen que resulte imposible sacar partido de los mecanismos de cacheado presentes en el protocolo <i>HTTP</i> , salvo en el caso de las peticiones referentes al fichero descriptor del servicio.
Interfaz uniforme	✗	Se utiliza el protocolo <i>HTTP</i> como mero protocolo de transporte, no aprovechando las posibilidades que presentan los métodos que éste ofrece para la gestión de recursos.
Separación en capas	✓	
Código a demanda	✗	No se contempla esta posibilidad.

Tabla 4.20 Cumplimiento de los principios *REST* en el estándar *WMS*.

Pero además, un análisis más pormenorizado también pone de manifiesto la complejidad de intentar crear una nueva interfaz *REST*, adicional a la existente, para dicho estándar. Esta última afirmación se sustenta en los siguientes aspectos:

- Recursos heterogéneos: no todas las tipologías de recursos son accesibles directamente. El mapa es un recurso virtual que solo se puede generar una vez conocidos los valores de los parámetros recogidos en las peticiones.
- Inutilidad de los mecanismos de cacheado: salvo contadas aplicaciones cliente que dispongan de un diseño que realice peticiones siguiendo una determinada cuadrícula, para el resto las técnicas de cacheado son estériles dada la imposibilidad práctica de que se repitan peticiones anteriores

que pudieran ser atendidas por la caché. Solo el descriptor del servicio o los datos asociados a entidades o celdas podrían beneficiarse de estas técnicas, suponiendo que puedan repetirse consultas en un plazo razonable como para considerar que pudieran seguir existiendo en la cache.

Descriptor del servicio	
<a href="http://<server>:<port>/<service>/capabilities">http://<server>:<port>/<service>/capabilities	
GET	Permite obtener el fichero descriptor del servicio, recogiendo las distintas capas de información – <i>layers</i> –, así como otros metadatos –autoría del servicio, condiciones de uso, etc.–.
POST	Permite actualizar el contenido del fichero descriptor, especificando el contenido a modificar. Operación disponible para perfiles de usuario con privilegios de administración.
PUT	Permite crear el fichero descriptor, especificando el contenido del mismo en la misma acción. Operación disponible para perfiles de usuario con privilegios de administración.
DELETE	Permite eliminar el fichero descriptor. Operación disponible para perfiles de usuario con privilegios de administración.
Mapa	
<a href="http://<server>:<port>/<service>/maps/<CRSIdentifier>/subset=lon(<minValue>:<maxValue>)/subset=lat(<minValue>:<maxValue>)/subset=layers(<layerList>)">http://<server>:<port>/<service>/maps/<CRSIdentifier>/subset=lon(<minValue>:<maxValue>)/subset=lat(<minValue>:<maxValue>)/subset=layers(<layerList>)	
GET	Permite obtener el mapa correspondiente a un ámbito geográfico concreto definido, referidas al CRS indicado, e incluyendo las capas referidas.
POST	No se contempla esta operación, puesto que el mapa no es más que la representación de una información residente en unos repositorios que no son accesibles por la propia naturaleza de este estándar. Este tipo de operación sería propia del estándar <i>WFS</i> o <i>WCS</i> , según la IG mostrada.
PUT	No se contempla esta operación por las mismas consideraciones del método <i>POST</i> .
DELETE	No se contempla esta operación por las mismas consideraciones del método <i>POST</i> .
Entidad o píxel	
<a href="http://<server>:<port>/<service>/maps/<CRSIdentifier>/subset=lon(<minValue>:<maxValue>)/subset=lat(<minValue>:<maxValue>)/subset=layers(<layerList>)/features/subset=queryLayers(<queryableLayerList>)/pixelRow/<pixelCol>">http://<server>:<port>/<service>/maps/<CRSIdentifier>/subset=lon(<minValue>:<maxValue>)/subset=lat(<minValue>:<maxValue>)/subset=layers(<layerList>)/features/subset=queryLayers(<queryableLayerList>)/pixelRow/<pixelCol>	
GET	Permite obtener la información correspondiente a las entidades o celdas que, estando dentro del ámbito de mapa definido –replicando la estructura propia del recurso mapa–, se ubican en la posición asociada a un píxel concreto y pertenecen a alguna de las capas consultables.
POST	No se contempla esta operación, puesto que la información residente en los repositorios que nutren al servicio no es accesible por la propia naturaleza de este estándar. Este tipo de operación sería propia del estándar <i>WFS</i> o <i>WCS</i> , según la naturaleza de la IG representada.
PUT	No se contempla esta operación por las mismas consideraciones del método <i>POST</i> .
DELETE	No se contempla esta operación por las mismas consideraciones del método <i>POST</i> .

Tabla 4.21 Propuesta de recursos y operaciones asociadas a la interfaz *REST* del estándar *WMS*.

Existen, por tanto, cuestiones conceptuales que hacen incompatible la actual formulación del estándar *WMS* con una hipotética interfaz *REST*. Dicho lo cual, se podría plantear una modificación de la especificación que permitiera diseñar una interfaz *REST* para *WMS*. Para ello sería preciso contemplar una relajación en la observación de las propiedades de *REST* que pasaría por la asimilación de los recursos virtuales de tipo mapa a recursos físicos. Este hecho implicaría que:

- El proceso implícito de generación del recurso se realice exclusivamente con el método *GET*, garantizando que en ningún caso dicho proceso pueda implicar que el servidor alcance un estado de inestabilidad –lo cual violaría la propiedad de seguridad del método *GET*–.
- Inclusión de los actuales parámetros de la operación *GetMap* que determinan el ámbito geográfico y el contenido como parte del *URI*.

Todo ello podría completarse con una extensión de la funcionalidad actual que permita una gestión integral de los recursos, utilizando de forma apropiada todos los métodos *HTTP* contemplados en *REST*, allí donde resulte plausible. Para ejemplificar esta aproximación, la Tabla 4.21 reúne los *URI* correspondientes a los diferentes recursos y la semántica de los métodos aplicables a cada uno de ellos²⁸.

Esta formulación de *URI* –en la que los típicos parámetros de una petición *HTTP* han pasado a incorporarse dentro de la estructura de directorios– aunque pueda aceptarse, dado que no existe ninguna forma estandarizada de expresar *URI* en *REST*, no es recomendable dado que no provee contexto para formular la petición y este hecho hace más difícil que los clientes puedan entender la *API* del servicio. Los recursos deben poder ser contemplados de forma jerárquica a través de sus *URI*, ofreciendo a los consumidores una forma sencilla de entender las relaciones existentes entre los mismos y esto resulta muy complicado de acuerdo con el diseño de *URI* anterior [Fre12].

Pese a estas objeciones, la aproximación realizada es, desde un punto de vista estructural, equivalente a la recogida por *OGC* en su propuesta de interfaz *REST* para el estándar *WCS* –ver punto 4.4.5.4.1–.

Por otro lado, también resulta interesante explorar la implementación realizada por *ESRI* para la creación de una interfaz *REST* para el conjunto de servicios que permite publicar su plataforma de servicios *ArcGIS Server*. Cuando se opta por crear un nuevo servicio basado en un proyecto de *ArcGIS* – documento *mxr* al que se ha hecho referencia anteriormente–, el entorno de administración de éste permite especificar qué capacidades funcionales quieren habilitarse. Como puede observarse en la Figura 4.31, hay una de las opciones que aparece seleccionada por defecto –indicada mediante un rectángulo rojo–, habiéndose seleccionado también que desea publicarse la interfaz *WMS*. Con respecto a la primera, denominada “*mapping*”, agrupa un conjunto de funciones para la obtención de documentos cartográficos [Dal01b]:

²⁸ Las partes del *URI* que son variables se identifican entre los símbolos “<” y “>”.

- *ExportMap*: permite generar la imagen correspondiente a un mapa generado dinámicamente a partir de un servicio. No se permite la modificación de la simbología asignada por defecto –no existe una especificación equivalente a *SLD*–. Es la operación homóloga a *GetMap* en *WMS*.

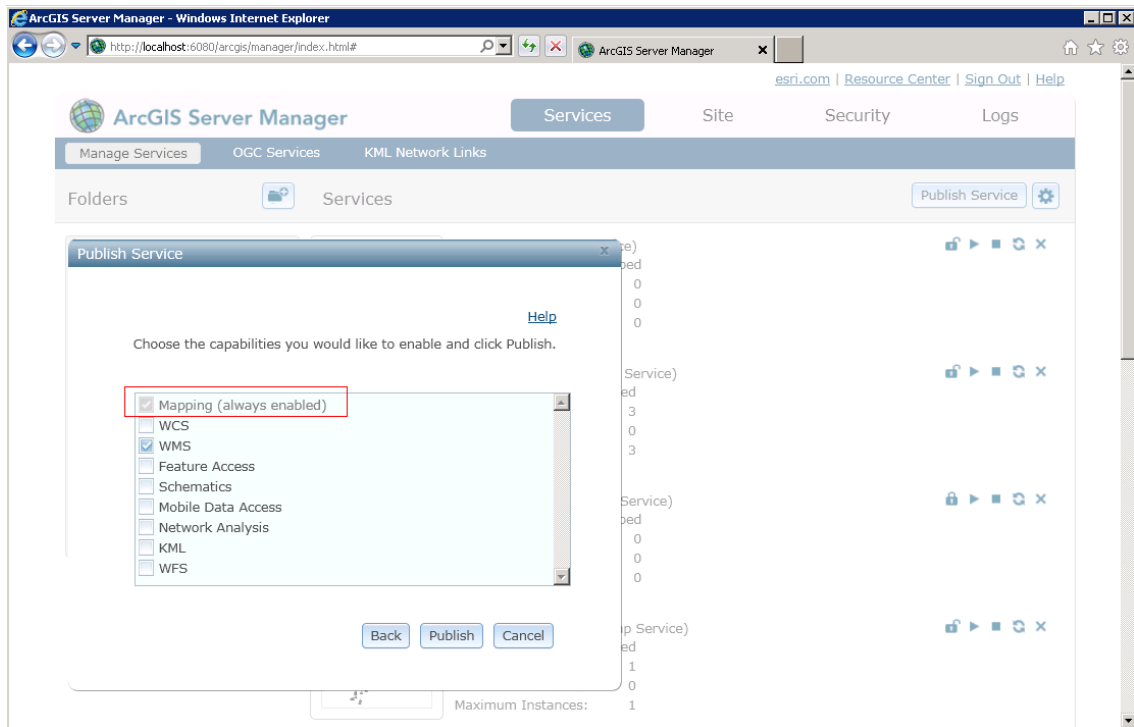


Figura 4.31 Interfaz de administración de ArcGIS Server [fuente: propia].

- *Identify*: devuelve información acerca de las entidades existentes en una o varias capas que se encuentran en la ubicación seleccionada por el usuario sobre un mapa. Es la operación homóloga a *GetFeatureInfo* en el estándar *WMS*.
- *Find*: devuelve información sobre las entidades de una o varias capas que tienen en uno o varios de sus atributos un valor equivalente a la palabra clave –*keyword*– especificada por el usuario. Es una operación con un resultado equivalente al anterior, aunque especificándose en este caso un filtro de carácter alfanumérico, en vez de geográfico.
- *GenerateKML*: genera un documento *KML*, comprimido en formato *KMZ*. Dicho documento contiene un *URI* que apunta al servicio que proporciona el *KML* con las propiedades y parámetros especificados.
- *Query on a Layer/Table*: devuelve el conjunto de entidades de una capa o registros de una tabla que cumplen con unos determinados criterios de consulta.

Las dos primeras funciones emulan el comportamiento de sendas operaciones del protocolo *WMS*, las restantes muestran herramientas que no se encuentran en dicho estándar. Pero además, en la citada herramienta se aprecia la posibilidad de habilitar las interfaces *WFS* o *Feature Access* –ambas orientadas a la descarga de información–, entre otras.

Estas funciones, tal y como se ha dicho, son expuestas mediante una interfaz que, según su fabricante²⁹, sigue el protocolo *REST*. Si se amplía el estudio a todas las funciones agrupadas en la *API REST*, *ESRI* ofrece la posibilidad de publicar servicios de: catalogación, descarga de entidades y geoprocetamiento – incluyendo operaciones de análisis espacial, geocodificación y, en general, cualquiera de las herramientas de geoprocetamiento existentes en la plataforma de escritorio, las cuales son accesibles como servicios mediante una interfaz obligatoria *REST* y opcional *WPS*–.

Como puede observarse, *ESRI* ha apostado por dotar a su plataforma de una interfaz *REST* completa y hacerla pública para que sea utilizada por otras organizaciones desarrolladoras de plataformas servidoras y clientes SIG [ESR14]. Siguiendo en esta línea, acaba realizando una petición al *OGC* para que eleve dicha *API* a la categoría de estándar. En julio de 2012, *OGC* anuncia la revisión pública de la propuesta de estándar. Los comentarios recibidos durante la fase de exposición muestran unas críticas feroces a esta iniciativa [ESR10], desde un punto de vista conceptual y práctico. Sin entrar en aspectos relacionados con la conveniencia o no de un estándar *API REST* de *OGC*, sí que es cierto que la interfaz publicada por *ESRI* es claramente contraria a los principios promulgados por *REST*. Sin ánimo de ser exhaustivo, obsérvese un par de aspectos reveladores:

- No se publican los *URI* de los recursos, sino operaciones como “exportar mapa”, “identificar” o “encontrar”.
- Los métodos soportados del protocolo *HTTP* son *GET* y *POST*, aunque este segundo no es utilizado, desde un punto de vista semántico, como es definido por el *W3C*; se utiliza para realizar consultas de información, como método alternativo a *GET*.
- Los *URI* de la operación *export* –equivalente a la petición *getMap* de *WMS*– muestra una complejidad impropia de la simplicidad que aboga *REST*, abusando de la utilización de parámetros, no apreciándose una diferencia sustancial con la especificación de *WMS*. Como muestra, pueden compararse estas dos peticiones equivalentes. Más allá de la forma de expresar las capas, las *URI* no muestran diferencias conceptuales.

Petición *REST*:

http://geoserveis.icc.cat/arcgis/rest/services/icc_bt5m_v_r/MapServer/export?imageSR=EPSG:25831&size=1000,1000&LAYERS=show:1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27&format=jpg&transparent=true&f=image&BBOX=418161,4618592,419161,4619592

Petición *WMS*:

http://geoserveis.icgc.cat/arcgis/services/icc_bt5m_v_r/MapServer/WMServer?REQUEST=GetMap&VERSION=1.1.1&SRS=EPSG:25831&Service=WMS&WIDTH=1000&HEIGHT=1000&LAYERS=

²⁹ Se hace esta matización puesto que son muchas las objeciones que pueden plantearse sobre el grado de cumplimiento que esta formulación hace de las propiedades de *REST*, en línea con aspectos que ya se han apuntado en este estudio y otros que se verán posteriormente.

[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27&STYLES=&FORMAT=image/png&BGCOLOR=0xFFFFFFFF&TRANSPARENT=TRUE&EXCEPTION=INIMAGE&BBOX=418161,4618592,419161,4619592](#)

Este hecho apunta a que no se debe esperar un mejor rendimiento de la interfaz *REST* con respecto a la *WMS*. No obstante, con la finalidad de validar esta hipótesis, se ha realizado una prueba de rendimiento aprovechando el geoservicio de acceso público del ICGC que publica con ambas interfaces la cartografía correspondiente a la BTC a escala 1:5.000. Los resultados obtenidos³⁰ se muestran mediante diferentes técnicas de representación:

- Tabla resumen que agrupa diferentes parámetros estadísticos –Tabla 4.22–: número de muestras procesadas, tiempo medio, mínimo y máximo –expresados en milisegundos–, desviación estándar, peticiones atendidas por segundo, *Kilobytes* de datos transferidos por segundo y tamaño medio –expresado en *bytes*– de las respuestas recibidas.

	Muestras proc.	Media	Mínimo	Máximo	Desv. Est.	Media bytes
WMS	33246	1102	111	6110	1446,16	8146,7
REST	34032	1077	111	6076	1379,47	4961,9

Tabla 4.22 Resultados de la comparativa de interfaces *WMS* y *REST* de *ArcGIS for Server*.

Los resultados obtenidos corroboran la predicción realizada. Tanto el rendimiento como los tiempos observados –estos últimos teniendo en cuenta la desviación estándar observada– son equiparables. La única variación, que además es muy significativa, afecta al tamaño medio de las imágenes resultantes de las peticiones al geoservicio. Como se puede observar, la interfaz *WMS* genera respuestas con un tamaño medio que es aproximadamente un 65% superior al de la interfaz *REST*. Este hecho tiene una incidencia directa en el ancho de banda requerido, aspecto que puede resultar determinante en algunos escenarios –ancho de banda disponible inferior al necesario en función del volumen de peticiones atendidas concurrentemente–. Cabría entrar en

³⁰ Las pruebas han sido realizadas con la herramienta *Apache JMeter*, versión 2.11 y revisión 1554548, trabajando contra el entorno de producción público del ICGC. Para ello se han planificado dos baterías de pruebas –una por cada una de las interfaces– que han sido lanzadas de forma consecutiva durante la madrugada del día 28 de octubre de 2014. De esta forma, se ha evitado la contaminación de los resultados por las interrelaciones que se crearían entre las peticiones de ambas baterías, puesto que acceden a los mismos procesos y repositorios físicos del servidor; así como minimizado el impacto que tendría la concurrencia con otros usuarios del geoservicio ajenos a esta prueba.

La duración de las citadas baterías ha sido de 25 minutos y se ha modelado la evolución del tiempo de respuesta –expresado en milisegundos– y del rendimiento –expresado en peticiones por segundo– en función del número de usuarios concurrentes que efectuaban peticiones aleatorias. Cada batería se ha inició con un único usuario y se fueron incorporando los 39 restantes con una frecuencia de 1 cada 30 segundos.

contacto con el servicio de soporte técnico de *ESRI* para determinar el porqué de esta diferencia, puesto que ambos casos, se ha solicitado la respuesta en formato *PNG*. En cualquier caso, cabe anticipar que no es un problema achacable al estándar *WMS*, si no a la implementación del mismo realizada por *ESRI*.

- Gráficos de tiempo de respuesta –Figura 4.32– y peticiones atendidas por segundo – Figura 4.33–, en función del número de usuarios concurrentes.

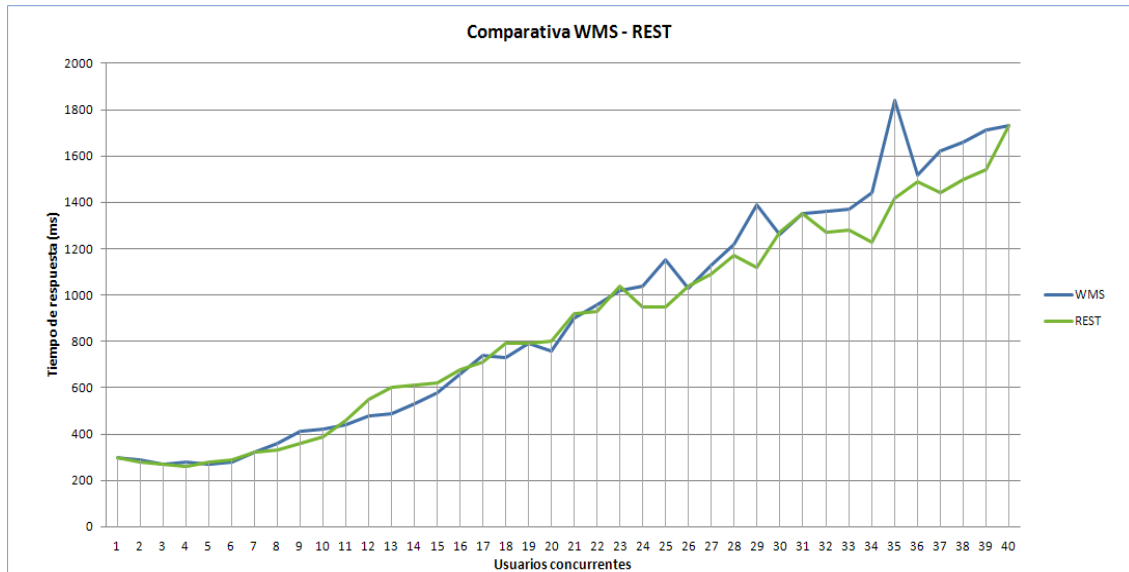


Figura 4.32 Tiempo de respuesta en función del número de usuarios concurrentes –WMS/REST–.

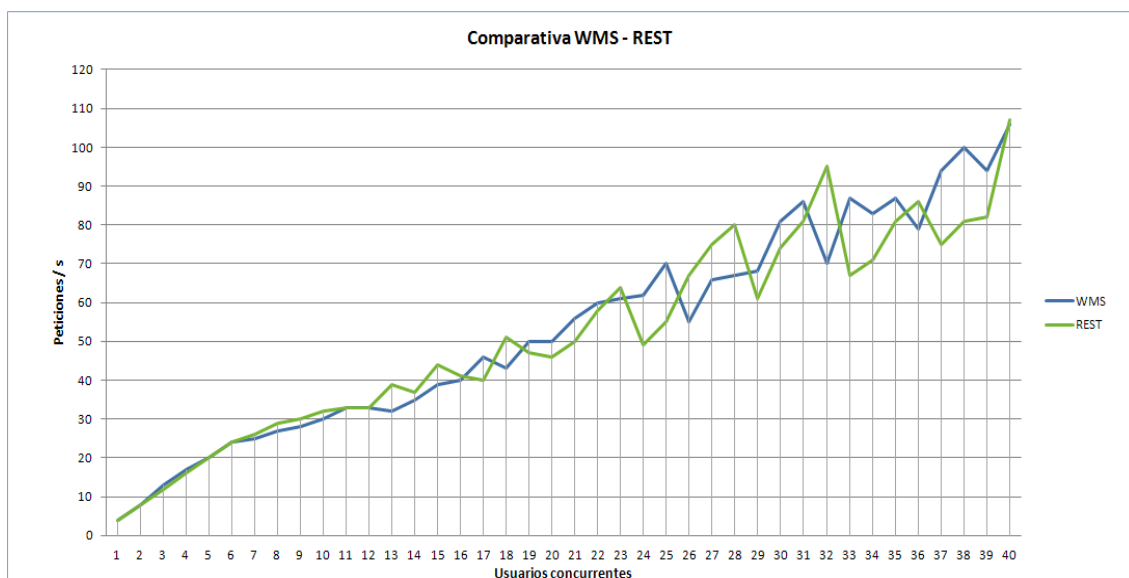


Figura 4.33 Rendimiento en función del número de usuarios concurrentes –WMS/REST–.

La evolución mostrada por el tiempo de respuesta verifica también que ambas interfaces muestran un comportamiento equiparable, más allá de que la interfaz *WMS* ofrezca un

comportamiento más irregular con volúmenes de concurrencia superiores a 24 usuarios. Por otro lado, el tiempo de respuesta se mantiene prácticamente estable hasta que el número de usuarios concurrentes supera los 6. Esta circunstancia se fundamenta en el hecho de que el geoservicio se apoya en un total de tres servidores, cada uno con dos instancias –*threads*– de procesamiento paralelo.

En cuanto a la evolución del rendimiento observado en ambas interfaces, se muestra la lógica correspondencia con lo reseñado para el tiempo de respuesta. El rendimiento tiene un incremento acusado, lineal y equivalente en ambas interfaces hasta llegar a la máxima concurrencia que el servicio es capaz de procesar en paralelo: 6. A partir de ese momento, el rendimiento sigue creciendo aunque en una proporción inferior –de haber seguido la progresión inicial, para 40 usuarios concurrentes se obtendría un rendimiento próximo a las 170 peticiones por segundo– y mostrando una mayor irregularidad.

Esta irregularidad, que al igual que el tiempo de respuesta se acrecienta a partir de una concurrencia superior a 24 usuarios, tiene su origen en los bloqueos internos que se producen tanto en el acceso al procesador como a los restantes recursos. Por tanto, no se observa ninguna de las mejoras mostradas en el punto 3.2.1: no se minimiza el tiempo de latencia ni la transferencia de información necesaria, así como tampoco se incrementa la independencia o la escalabilidad.

A modo de resumen, la plataforma que se está diseñando debe mantener la interfaz *KVP* del estándar *OGC WMS* e incorporar el protocolo *SOAP*, para favorecer la utilización de estos servicios de visualización en las plataformas empresariales *SOA*, así como *REST* para hacer lo propio en el ámbito *ROA*, pese a las limitaciones reflejadas a lo largo de este estudio.

En este sentido, en 2005 se publicó una propuesta de cambio del estándar *WMS* para la inclusión de los protocolos *SOAP* y *WSDL* [Por13]. Durante los meses de julio de 2007 y abril de 2008, el *OGC* condujo un conjunto de pruebas encaminadas, entre otras cosas, al desarrollo de una interfaz *SOAP* y *WSDL* para los cuatro servicios fundacionales: *WMS*, *WFS* (incluida la extensión transaccional *WFS-T*), *WCS* y *WPS*, permitiendo de esta forma su integración dentro de las herramientas de encadenamiento de servicios que siguen estándares industriales [Dus05]. Un par de años más tarde, el *OGC* publicó una nueva versión del documento que define aspectos comunes a las interfaces de los estándares de implementación de los múltiples *WS OGC* [Gre10], entre los cuáles se incluye la interfaz *SOAP* para las distintas operaciones de los estándares *WMS*, *WFS* y *WCS*.

4.4.3 Servicio de visualización *WMTS*

La especificación *WMTS* define una serie de interfaces para cada uno de estos métodos, las cuales aparecen recogidas en la Tabla 4.23. Este estándar, además de ofrecer una interfaz *SOAP*, es el primero que presenta desde su concepción una interfaz *REST*, aparte de dar cobertura a la formulación tradicional *KVP* sobre *HTTP*.

En este sentido, los puntos siguientes analizan la adecuación de la interfaz *REST* de este estándar a los planteamientos conceptuales establecidos para este protocolo y, por extensión, para la arquitectura *ROA*, tratando de determinar el grado de cumplimiento de los mismos.

Método ³¹	Protocolos	Codificación
GetCapabilities	<i>HTTP (Get)</i>	<i>KVP</i>
	<i>HTTP (Post)</i>	<i>XML</i>
	<i>HTTP (Post) + SOAP</i>	<i>XML</i>
	<i>REST</i>	-
GetTile	<i>HTTP (Get)</i>	<i>KVP</i>
	<i>HTTP (Post)</i>	<i>XML</i>
	<i>HTTP (Post) + SOAP</i>	<i>XML</i>
	<i>REST</i>	-
GetFeatureInfo	<i>HTTP (Get)</i>	<i>KVP</i>
	<i>HTTP (Post)</i>	<i>XML</i>
	<i>HTTP (Post) + SOAP</i>	<i>XML</i>
	<i>REST</i>	-

Tabla 4.23 Protocolos y codificaciones admitidas en las operaciones *WMTS*.

4.4.3.1. División cliente-servidor

Esta restricción la cumple el estándar *WMTS* puesto que los clientes y los servidores compatibles interactúan mediante el paradigma de *WS*, tal y como se ha referido en 4.4.2.1. La modificación de una instancia de un servicio *WCS* no implica realizar cambios en la a funcionalidad de la interfaz de usuario, porque ésta permanece inalterada.

135

4.4.3.2. Interacciones sin estado

Según se indicó en 4.4.2.2, la comunicación entre clientes y servidores en cualquier servicio *OGC* es por naturaleza sin estado. En general, la interacción con el servicio sigue el patrón petición-respuesta en el que no existe dependencia alguna con interacciones previas.

4.4.3.3. Existencia de caché

La especificación del estándar *WMTS* [Jul10] recoge la necesidad de habilitar mecanismos de cacheado para mejorar el rendimiento de los servidores, abogando por la incorporación de información sobre la fecha de expiración en las respuestas a peticiones de los clientes, utilizando las cabeceras *HTTP* adecuadas: *Expires* para la versión 1.0 y, además de la anterior, *Cache-control* para la 1.1 de dicho protocolo. Por otro lado, en sentido amplio, las opciones recogidas en 4.4.2.3 sobre el protocolo *HTTP* y los mecanismos que éste habilita para el control de las cachés, serían perfectamente aplicables.

³¹ En la interfaz *REST* no existen los métodos tal y como se conciben en las interfaces restantes. Estos métodos se transforman en *URI* de acceso a los recursos correspondientes.

4.4.3.4. Interfaz uniforme

Como ya se ha reseñado anteriormente, la interfaz uniforme implica que el servidor expone una interfaz derivada de la semántica de los métodos del protocolo *HTTP*, actuando éste como protocolo de aplicación. En este caso, utilizando la interfaz *REST* presentada en la especificación, se cumple este requerimiento y se puede considerar que se utiliza el protocolo *HTTP* como protocolo de aplicación.

Dado que las operaciones definidas en el estándar no permiten realizar modificaciones de los repositorios publicados mediante un servicio *WMTS*, la interfaz solo contempla la utilización del método *Get* del protocolo *HTTP* –ver Figura 4.34–.

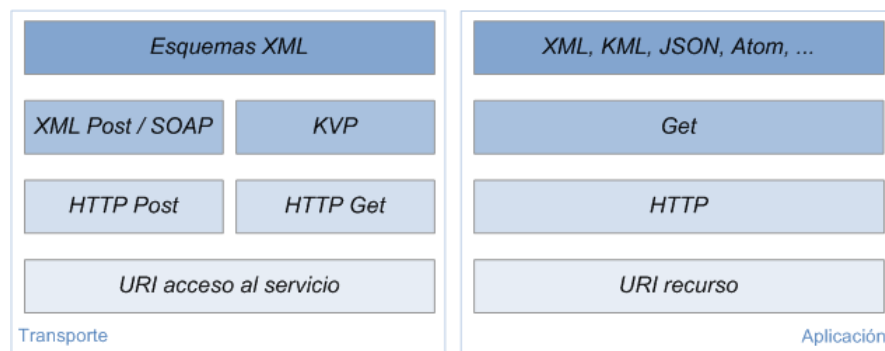


Figura 4.34 *HTTP* como protocolo de transporte –con opción *SOAP*– / aplicación [fuente: propia].

Por último, en lo que respecta a los conceptos implícitos dentro de la interfaz, tal y como aparecen recogidos en 3.2.1, es interesante realizar el análisis que se aborda en los siguientes puntos.

4.4.3.4.1. Recursos

La especificación técnica del estándar *WMTS* identifica tres tipologías de recursos –ver Tabla 4.24–:

Método	Recurso	Representación
GetCapabilities	Descriptor del servicio	<i>XML</i> ³²
GetTile	Tesela	<i>MIME-Type</i>
GetFeatureInfo	Objeto geográfico	<i>GML</i> o <i>MIME-Type</i> convenido

Tabla 4.24 Relación entre recursos, representaciones y métodos *WMTS*.

- Descriptor del servicio –*service metadata*–, accesible mediante una petición de acceso al recurso –equivalente en las otras interfaces a *GetCapabilities*– que se corresponde con un *URI*, el cual

³² La especificación recomienda esta representación, pero no la plantea con carácter obligatorio.

devuelve el descriptor del mismo. No obstante, no se extrae todo el partido posible de la aplicación de *HTTP* como protocolo de aplicación ya que no se utilizan los métodos *POST*, *PUT* y *DELETE* para crear, actualizar y eliminar, respectivamente, el recurso. Según el estándar, el acceso a este fichero descriptor se realizará utilizando la siguiente nomenclatura³³:

<http://<server>:<port>/1.0.0/WMTSCapabilities>

- Tesela o fragmento del mapa en el que se representa un conjunto de capas de información, accesibles mediante el *URI* –equivalente a la operación *GetTile* en las restantes interfaces–. Al igual que en el caso anterior, se cumple con el requerimiento *REST* de utilización del protocolo *HTTP* como protocolo de aplicación, con las mismas limitaciones anteriormente referidas. Según la especificación, el acceso a esta unidad de mapa se realiza utilizando la siguiente nomenclatura:

<http://<server>:<port>/<style>/<dimension>/<tileMatrixSet>/<tileMatrix>/<tileRow>/<tileCol>>

En el *URI* se advierte la existencia de determinados parámetros:

- *Style*: las capas de información de base pueden representarse de formas o estilos diferentes a la hora de generar un mapa, referenciados mediante un identificador. Este aspecto también aparece recogido en el estándar *WMS* con la especificación del parámetro *STYLES*. De esta forma, la misma información de base puede mostrarse mediante simbologías distintas, favoreciendo su utilización en distintos ámbitos.

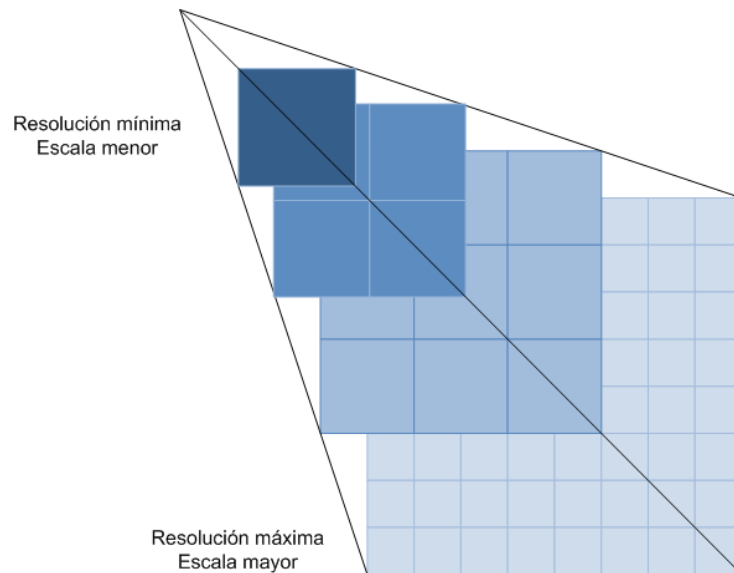


Figura 4.35 Conjunto de matrices [fuente: propia].

³³ Las partes del *URI* que son variables se identifican entre los símbolos “<” y “>”.

- *Tile matrix set*: cada una de las capas o mapas publicados en un servicio tiene asociada un conjunto de matrices de teselas, a las cuales se hace referencia mediante un identificador. Cada una de estas matrices se corresponde con un nivel de resolución concreto –ver Figura 4.35–.
- *Tile matrix*: identificador de cada una de las matrices que componen un *tile matrix set* –ver Figura 4.36–.

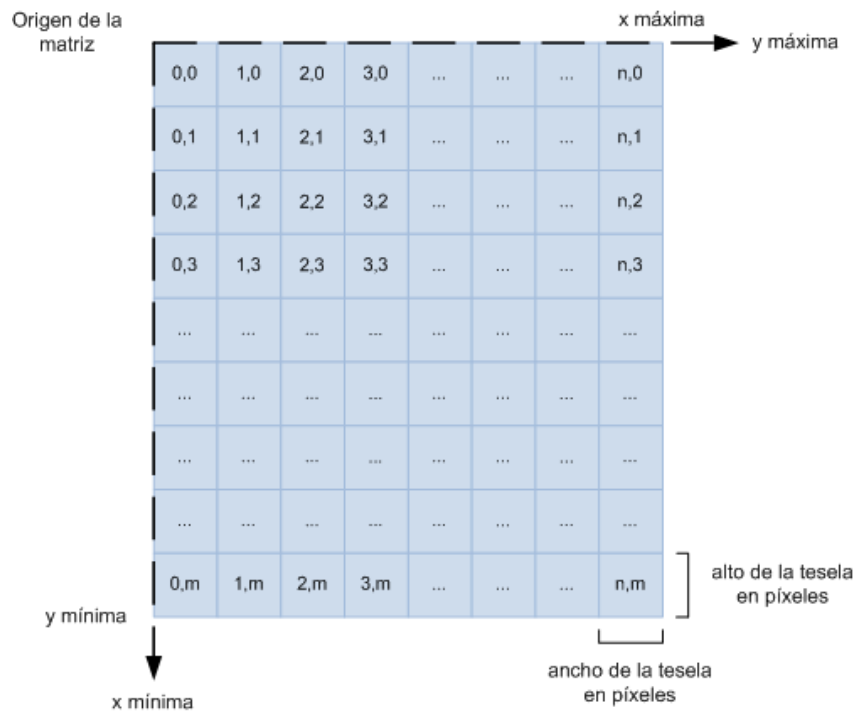


Figura 4.36 Conjunto de teselas que forman una matriz (n,m) [fuente: propia].

- *Tile row*: índice de la fila dentro de la matriz en la que se encuentra la tesela a recuperar.
- *Tile col*: índice de la columna dentro de la matriz en la que se encuentra la tesela a recuperar.
- Objeto geográfico o píxel, accesible mediante el *URI* correspondiente al recurso –equivalente a la operación *GetFeatureInfo* en las restantes interfaces–. Pese a que la especificación habla de un tipo de recursos denominado *FeatureInfo*, realmente la información a la que se accede es la asociada con el objeto geográfico –en el caso de capas de origen vectorial– o con el píxel o celda –cuando se trata de información de origen raster–, tal y como se ha reflejado en el estudio del método homónimo del estándar *WMS*. Por tanto, el objeto geográfico o el píxel son realmente los recursos y su representación el resultado obtenido al acceder al *URI* del píxel de la tesela.

Según la especificación del estándar, el acceso a esta unidad de mapa se realiza utilizando la siguiente nomenclatura:

<http://<server>:<port>/<style>/<dimension>/<tileMatrixSet>/<tileMatrix>/<tileRow>/<tileCol>/<pixelRow>/<pixelCol>>

El *URI* es el resultado de concatenar al *URI* del recurso asociado a la tesela, la fila y columna en la que se ubica el píxel en el que hay que buscar la información.

4.4.3.4.2. Representaciones

El estándar *WMTS* no plantea obligatoriamente la utilización de un formato de representación concreto para ninguna de las operaciones. Sin embargo, sí que establece un conjunto de recomendaciones para los siguientes recursos –ver Tabla 4.24–:

- Servicio: se indica que el formato más común para la descripción de los metadatos asociados al servicio es *XML* –*service metadata*–. El siguiente *URI* muestra el acceso a un hipotético recurso de estas características.

<http://<server>:<port>/1.0.0/WMTSCapabilities.xml>

- Objeto geográfico o píxel: se recomienda la implementación del perfil *Simple Features* de *GML*, el cual define tres niveles de contenidos para la descripción de las geometrías y sus atributos asociados; siendo el recomendado el nivel básico –0–, el que provee un documento *GML* más sencillo. Dado que este perfil incluye en el documento la geometría de los elementos y que esta información, aparte de ser muy voluminosa, no es requerida en múltiples escenarios, se recoge la posibilidad de generar un esquema *XML* para los documentos de respuesta que no incluya esta información.

Asimismo, la especificación también recomienda, en aras a facilitar la presentación de la información, dar cobertura al formato *HTML* –*MIME-type* “*text/html*”–.

El siguiente *URI* muestra el acceso a un hipotético recurso de estas características:

<http://<server>:<port>/<style>/<dimension>/<tileMatrixSet>/<tileMatrix>/<tileRow>/<tileCol>/<pixelRow>/<pixelCol>.xml>

No obstante, para los recursos anteriores se permite a los implementadores ofrecer los formatos que crean oportunos, especificados mediante la codificación *MIME-type* adecuada y declarados en el recurso descriptor; de forma que los clientes, haciendo uso de la cabecera de aceptación –*Accept header*– puedan especificar el formato deseado entre los declarados por el servidor. El siguiente *URI* muestra el acceso a un recurso de tipo tesela en formato de representación *JPEG*:

<http://<server>:<port>/<style>/<dimension>/<tileMatrixSet>/<tileMatrix>/<tileRow>/<tileCol>.jpg>

En cualquier caso, las reflexiones planteadas en 4.4.2.4.2 para el estándar *WMS* entorno a la utilización de *JSON* y *XML*, son enteramente aplicables.

4.4.3.4.3. Hipermedia

El estándar *WMTS* muestra, en referencia a este apartado, el mismo comportamiento recogido en 4.4.2.4.3 para el estándar *WMS*. A título ilustrativo, se podría haber completado la respuesta a una petición de una tesela con un documento *XML* que recoja los identificadores de las teselas vecinas en el mismo nivel, las teselas comprendidas en el mismo ámbito en un nivel inferior, así como la tesela que comprende a la actual en un nivel superior.

4.4.3.4.4. Mensajes autodescriptivos

Este estándar especifica la posibilidad de utilizar cabeceras del protocolo *HTTP* para poder habilitar mecanismos de cacheado de las respuestas. En este sentido, se entiende que las respuestas contienen información suficiente para su interpretación completa sin precisar información adicional, cumpliéndose con esta característica.

4.4.3.5. Estructuración en capas

En línea con lo recogido en 4.4.2.5, desde la perspectiva del estándar *WMTS* se podría contemplar la existencia de nodos que realicen ciertas tareas, especialmente en la gestión de sistemas de cacheado distribuidos, aunque el estándar no realiza ninguna consideración al respecto.

4.4.3.6. Código a demanda

La especificación *WMTS* no recoge consideración alguna con respecto a la posibilidad de descargar código en el cliente para la ejecución del mismo, aunque dada la naturaleza de este estándar tampoco parece ser esta una necesidad.

4.4.3.7. Conclusiones

Sin duda, el hecho de que se esté hablando de una especificación de implementación tan reciente, en comparación con la de los demás servicios tradicionales del *OGC –WMS, WFS, WCS y WPS–* ha permitido al organismo regulador abordar directamente las demandas actuales del mercado. En este sentido, la incorporación “de serie” de una interfaz *SOAP* y *REST*, es un aspecto más que reseñable y que, sin duda, ha de marcar el devenir de los demás estándares.

Recapitulando las consideraciones recogidas en el análisis realizado sobre la interfaz *REST* que ofrece el protocolo *WMTS*, se obtiene la Tabla 4.25 que permite colegir, sin género de dudas, que dicha interfaz es completamente compatible.

Como se puede observar, la interfaz *REST* se adapta perfectamente a los principios teóricos en los que se sustenta este protocolo, con la única salvedad de no contemplar la posibilidad de descargar código en el cliente para enriquecerlo. Esta última característica, aparte de corresponderse con el único principio de carácter opcional en *REST*, carece de sentido en esta especificación, puesto que lo único que precisa este estándar es el envío de recursos completos que no precisan procesado alguno en la vertiente cliente.

Principio	Cumplimiento	Aspectos clave
División cliente-servidor	✓	
Ausencia de estado	✓	
Existencia de caché	✓	
Interfaz uniforme	✓	La interfaz solo considera la utilización del método <i>Get</i> para la consulta de los diferentes recursos y podría mejorarse gestionando los mismos con los otros métodos del protocolo <i>HTTP</i> .
Separación en capas	✓	
Código a demanda	✗	No se contempla esta posibilidad.

Tabla 4.25 Cumplimiento de los principios *REST* en el estándar *WMTS*.

Por tanto, de cara a la implementación de *WMTS* en la plataforma SIG distribuida que se está configurando, tan solo sería preciso realizar alguna modificación o adaptación del estándar orientada a la inclusión de los métodos *POST*, *PUT* y *DELETE* para las gestión completa de los recursos, así como la incorporación de vínculos hipermedia para la conexión entre diferentes recursos.

De cara a concluir el análisis de las interfaces *WMTS*, se ha realizado una prueba de rendimiento que permita establecer el impacto que tiene la utilización de una interfaz *WMS* frente a una *WMTS* para un servicio en el que la información se ha fragmentado en teselas. Para esta prueba se ha contado con la capacidad de la plataforma *Mapproxy* instalada en el entorno de producción de acceso público del ICGC, la cual habilita ambas interfaces. Esta plataforma, cuando recibe peticiones *WMS*, calcula la tesela equivalente al ámbito geográfico solicitado. Si este ámbito no se corresponde exactamente con una tesela concreta, el servidor calcula la o las teselas afectadas y genera una imagen nueva, con el coste que esto supone. En este caso, para evitar este problema, se han calculado ámbitos geográficos coincidentes con el origen de coordenadas, la escala y el tamaño de las teselas existentes, tal y como se puede observar en los dos ejemplos de peticiones siguientes:

Petición *WMTS*:

http://geoserveis.icc.cat/icc_mapesmultibase/utm/wmts/orto/UTM25831/10/430/568.jpeg

Petición *WMS*:

http://geoserveis.icc.cat/icc_mapesmultibase/utm/wms/service?LAYERS=orto&FORMAT=image%2Fjpeg&SERVICE=WMS&VERSION=1.1.1&REQUEST=GetMap&STYLES=&SRS=EPSG%3A25831&WIDTH=256&HEIGHT=256&BBOX=313040,4693768,313168,4693896

En las dos baterías de pruebas realizadas, una por interfaz, se han hecho peticiones al nivel 10 de la pirámide de teselas correspondiente a la ortofoto de Cataluña, el cual se ha generado a partir del

producto correspondiente a escala 1:5.000. Los resultados obtenidos³⁴ se muestran mediante las técnicas de representación utilizadas anteriormente –ver 4.4.2.7–:

- Tabla resumen que agrupa diferentes parámetros estadísticos –Tabla 4.26–: número de muestras procesadas, tiempo medio, mínimo y máximo –expresados en milisegundos–, desviación estándar, peticiones atendidas por segundo, *Kilobytes* de datos transferidos por segundo y tamaño medio –expresado en *bytes*– de las respuestas obtenidas. Los resultados reflejan que el tiempo medio dedicado por el servidor para generar una respuesta *WMS* es el doble del necesario para *WMTS*, aunque el tiempo de respuesta mínimo y máximo sean prácticamente equivalentes. Este hecho se explica atendiendo a los siguientes gráficos.

Interfaz	Muestras proc.	Media	Mínimo	Máximo	Desv. Est.	Media bytes
WMS	181204	301	8	61285	650,49	31670
WMTS	229109	149	5	61299	535,64	22783,9

Tabla 4.26 Resultados de la comparativa de interfaces *WMS* y *WMTS* de *Mapproxy*.

- Gráficos de tiempo de respuesta — y peticiones atendidas por segundo –Figura 4.38–, en función del número de usuarios concurrentes.

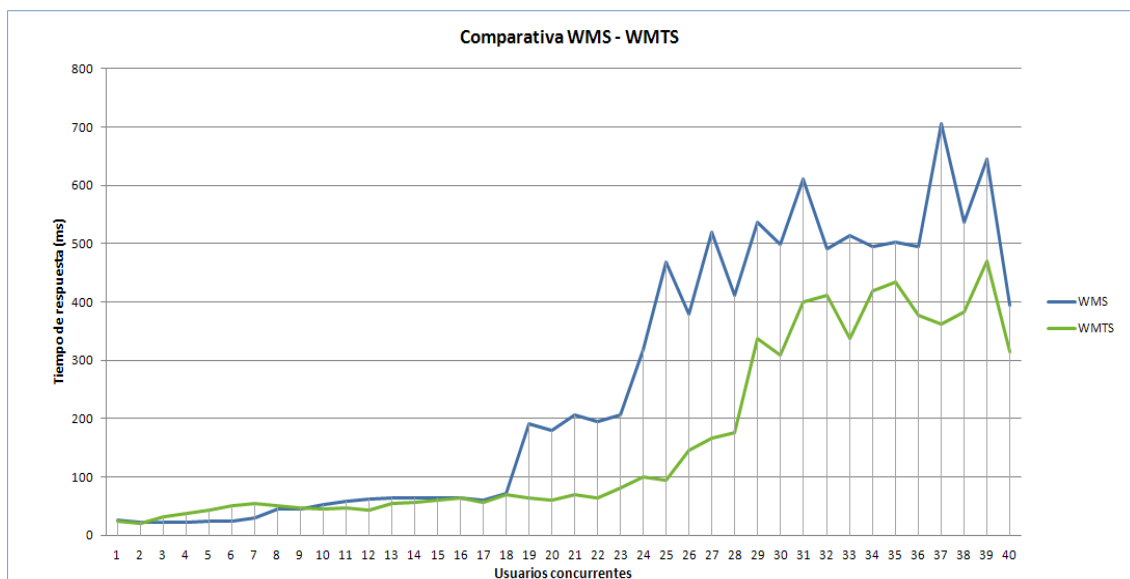


Figura 4.37 Tiempo de respuesta en función del número de usuarios concurrentes –*WMS/WMTS*–.

³⁴ Las pruebas han sido realizadas con la misma herramienta reseñada en las pruebas comparativas de las interfaces *REST* y *WMS* de la plataforma tecnológica *ArcGIS for Server* de *ESRI* –ver cita 3030, página 131–, utilizando además una configuración idéntica en cuanto a la fecha seleccionada y a la duración de las pruebas.

La evolución mostrada en el primero de los gráficos refleja cómo, hasta llegar a los 18 usuarios concurrentes, ambas interfaces muestran un tiempo de respuesta similar, incluso superando en algunos casos la interfaz *WMTS* el de la interfaz *WMS*. A partir de ese momento, divergen las gráficas, produciéndose un extraordinario incremento de los tiempos de respuesta en la interfaz *WMS* que no se observa en la *WMTS* hasta que se alcanza una concurrencia de 25 usuarios. Posteriormente, los incrementos relativos en ambas interfaces son equivalentes, mostrándose ambos comportamientos muy irregulares.

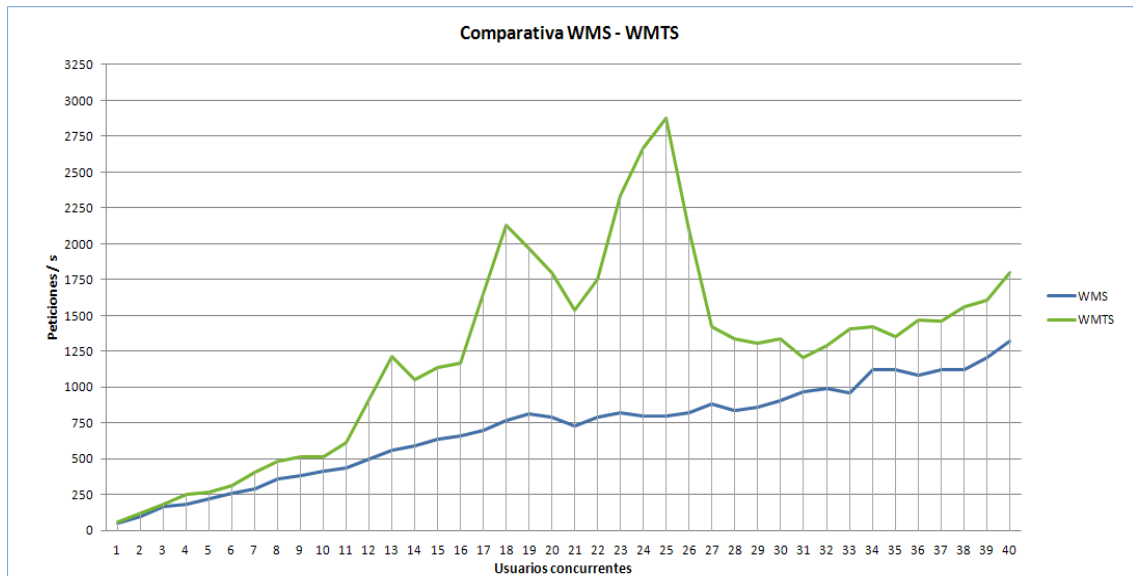


Figura 4.38 Rendimiento en función del número de usuarios concurrentes –*WMS/WMTS*–.

En cambio, en el segundo de los gráficos se observa que, aunque la evolución del rendimiento es siempre positiva en ambas interfaces, se manifiesta un incremento notablemente diferente desde estadios de concurrencia mucho más tempranos. A partir de una concurrencia de 6 usuarios el rendimiento de la interfaz *WMTS* es sensiblemente mayor que el de la interfaz *WMS* y se produce un despegue a partir del momento en que se alcanza un nivel de concurrencia de 11 usuarios. Posteriormente, el comportamiento de la interfaz *WMS* sigue siendo muy estable, con un crecimiento sostenido, mientras que el de la interfaz *WMTS* es mucho más voluble, con incrementos de 250 peticiones por segundo con tan solo un incremento de una unidad en la concurrencia y caídas equivalentes en otros momentos. Cuando se alcanzan los 25 usuarios simultáneos, el rendimiento de la interfaz *WMTS* cae en picado hasta que se estabiliza entorno a los 27. Desde ese momento, ambas interfaces muestran un comportamiento similar.

Del análisis de la evolución del tiempo de respuesta y del rendimiento puede deducirse que el sobrecoste que conlleva para la plataforma servidora analizar los parámetros de la petición y determinar la tesela correspondiente, va ganando importancia a medida que va aumentando la concurrencia. Hay momentos en los que el rendimiento de la interfaz *WMTS* llega a triplicar el de la *WMS*. Tampoco hay que olvidar otro aspecto que condiciona los resultados obtenidos, el hecho de acabar sirviendo imágenes previamente generadas implica que el tiempo de procesado es prácticamente insignificante con respecto al tiempo necesario para la transmisión, aspecto que puede provocar importantes oscilaciones en función del tráfico existente.

4.4.4 Servicio de descarga WFS

La especificación *WFS* define una serie de interfaces para cada uno de estos métodos, las cuales aparecen recogidas en la Tabla 4.27. La primera es la tradicional formulación dentro de *OGC* en la que se utiliza el protocolo *HTTP* con variantes para sus métodos *GET* y *POST*. La segunda muestra una interfaz para el protocolo *SOAP* sobre *HTTP*, utilizando el método *POST* de este último, la cual pone de manifiesto que entre los presupuestos arquitectónicos considerados por los diseñadores de este estándar figuraba la arquitectura *SOA*.

Operación	Protocolos	Codificación
GetCapabilities	<i>HTTP (Get)</i>	<i>KVP</i>
	<i>HTTP (Post)</i>	<i>XML</i>
	<i>HTTP (Post) + SOAP</i>	<i>XML</i>
DescribeFeatureType	<i>HTTP (Get)</i>	<i>KVP</i>
	<i>HTTP (Post)</i>	<i>XML</i>
	<i>HTTP (Post) + SOAP</i>	<i>XML</i>
GetPropertyValue	<i>HTTP (Get)</i>	<i>KVP</i>
	<i>HTTP (Post)</i>	<i>XML</i>
	<i>HTTP (Post) + SOAP</i>	<i>XML</i>
GetFeature	<i>HTTP (Get)</i>	<i>KVP</i>
	<i>HTTP (Post)</i>	<i>XML</i>
	<i>HTTP (Post) + SOAP</i>	<i>XML</i>
GetFeatureWithLock	<i>HTTP (Get)</i>	<i>KVP</i>
	<i>HTTP (Post)</i>	<i>XML</i>
	<i>HTTP (Post) + SOAP</i>	<i>XML</i>
LockFeature	<i>HTTP (Get)</i>	<i>KVP</i>
	<i>HTTP (Post)</i>	<i>XML</i>
	<i>HTTP (Post) + SOAP</i>	<i>XML</i>
Transaction	<i>HTTP (Post)</i>	<i>XML</i>
	<i>HTTP (Post) + SOAP</i>	<i>XML</i>
CreateStoredQuery	<i>HTTP (Post)</i>	<i>XML</i>
	<i>HTTP (Post) + SOAP</i>	<i>XML</i>
DropStoredQuery	<i>HTTP (Get)</i>	<i>KVP</i>
	<i>HTTP (Post)</i>	<i>XML</i>
	<i>HTTP (Post) + SOAP</i>	<i>XML</i>
ListStoredQueries	<i>HTTP (Get)</i>	<i>KVP</i>
	<i>HTTP (Post)</i>	<i>XML</i>
	<i>HTTP (Post) + SOAP</i>	<i>XML</i>
DescribeStoredQueries	<i>HTTP (Get)</i>	<i>KVP</i>
	<i>HTTP (Post)</i>	<i>XML</i>
	<i>HTTP (Post) + SOAP</i>	<i>XML</i>

Tabla 4.27 Protocolos y codificaciones admitidas en las operaciones *WFS*.

Por otro lado, no se recoge una interfaz *REST* y para calibrar la adecuación de los métodos definidos a esta interfaz, en los puntos siguientes se analiza la convergencia de esta especificación con las restricciones que *REST* impone, según se recoge en 3.2.2.

4.4.4.1. División cliente-servidor

Esta restricción la cumple el estándar *WFS* puesto que los clientes y los servidores compatibles interactúan mediante el paradigma de *WS*, tal y como se ha referido en 4.4.2.1. La adición de una nueva capa *–feature type–* a una instancia de un servicio *WFS* no implica realizar cambios en la funcionalidad de la interfaz de usuario, porque ésta permanece inalterada.

4.4.4.2. Interacciones sin estado

Según se indicó en 4.4.2.2, la comunicación entre clientes y servidores en cualquier servicio *OGC* es por naturaleza sin estado. En general, la interacción con el servicio sigue el patrón petición-respuesta en el que no existe dependencia alguna con interacciones previas.

4.4.4.3. Existencia de caché

La capacidad para activar selectivamente el almacenamiento de las respuestas, de forma que los clientes puedan reutilizarlas posteriormente ante la necesidad de plantear las mismas peticiones, no se contempla en la especificación del estándar *WFS*. En líneas generales, el razonamiento aplicado en el punto 4.4.2.3 para el estándar *WMS* sigue siendo válido, aunque es interesante reseñar ciertos aspectos diferenciales sobre la conveniencia de implementar este mecanismo.

Por un lado, nos encontramos con dos métodos, *GetCapabilities* y *DescribeProcess*, que son claramente almacenables, puesto que se trata de ficheros *XML* con contenido estático –mientras no se realice un proceso de actualización del contenido del servicio–. Por el otro, los métodos restantes muestran una casuística que tiene un encaje difícil de cara a sacar partido a un hipotético almacenamiento de las respuestas:

- *GetPropertyValue*, *GetFeature*: siempre que se publiquen repositorios con frecuencias de actualización muy bajas, podría optarse por utilizar una caché para las respuestas; aunque la posibilidad de que se descargue dos veces exactamente la misma información es improbable. Si por el contrario, se trata de información sujeta a actualizaciones, un escenario de almacenamiento de las respuestas, carece igualmente de sentido.
- *GetFeatureWithLock*: se podría indicar lo mismo que en los métodos anteriores, pero en este caso, la propia definición del servicio implica que esta operación es un paso previo para una posterior edición de los datos, por lo que es preciso bloquearlos y así evitar problemas de sincronización entre usuarios que desean realizar tareas de edición sobre los mismos datos. Por tanto, no parece realista imaginar dos peticiones pidiendo recuperar y bloquear las mismas geometrías para una posterior edición.

- *Transaction* y *LockFeature*: al tratarse de métodos orientados exclusivamente a la edición, en los cuales la respuesta es meramente informativa del resultado de la operación, carece de sentido plantear un almacenamiento de las mismas.
- *CreateStoredQuery*, *DropStoredQuery*: son métodos en los cuales la respuesta simplemente informa sobre el resultado de la operación, por tanto no tiene sentido plantear un almacenamiento de las mismas.
- *ListStoredQueries*, *DescribeStoredQueries*: si la frecuencia de consulta de la información suministrada por ambos métodos es importante, se podría plantear el cacheado de la respuesta, siempre y cuando la frecuencia con la que se actualizan las consultas en el servidor sea baja. En todo caso, la latencia de este tipo de operaciones es muy baja y las ventajas del almacenamiento prácticamente desaparecen.

Pese a ello, cabe tener en cuenta las opciones recogidas en 4.4.2.3 sobre el protocolo *HTTP* y los mecanismos que éste habilita para el control de las cachés.

4.4.4.4. Interfaz uniforme

Como ya se ha reseñado anteriormente, la interfaz uniforme implica que el servidor expone una interfaz derivada de la semántica de los métodos del protocolo *HTTP*, actuando éste como protocolo de aplicación. En este caso, en cualquiera de las opciones de protocolo, método y codificación ofrecidas por *WFS*, no se cumple con este requerimiento en todos los métodos del estándar –ver Figura 4.34–.

La reflexión realizada en el punto 4.4.2.4 es igualmente aplicable, aunque se haya introducido la posibilidad de utilizar el protocolo *SOAP* sobre el método *Post* de *HTTP*. La cuestión es que existen métodos que utilizan *HTTP* como mero protocolo de transporte –de forma muy especial *Transaction*, *FeatureLock* y *GetFeatureLock*–, en el cual se encapsulan peticiones en las que la semántica de las mismas reside en el *URI* –método *Get*– o en el fichero *XML* asociado –método *Post*–, respondiendo conceptualmente al modelo *RPC*. Los métodos restantes también siguen este modelo, aunque podrían adaptarse a las recomendaciones de *REST* con cambios en su sintaxis y nomenclatura.

También es interesante referir ciertas consideraciones con respecto a las características de seguridad e idempotencia. En lo que respecta a la primera, el hecho de que se utilice el método *GET* para ejecutar operaciones que pueden implicar un cambio en el estado del servidor contraviene lo dispuesto para este método en *REST*, de ahí que no se cumpla con esta característica. En cuanto a la idempotencia, *WFS* no contraviene dicha característica, tal y como se puede deducir del análisis de aquellos métodos que modifican el contenido del servidor:

- *CreateStoredQuery*: la ejecución reiterada de este método no tiene consecuencias. Si el identificador indicado para la consulta o filtro ya existe en el servidor, implica la actualización de la misma en el servidor por la recogida en la petición.
- *DropStoredQuery*: la ejecución reiterada de este método no tiene consecuencias, ya que las peticiones de eliminación posteriores intentarán eliminar un recurso que no existe y el servidor devolverá un código de error sin mayores consecuencias.

- **Transaction:** la ejecución reiterada de este método podría tener consecuencias en entornos donde se pudiera simultanear la edición de geometrías desde varios clientes. En este caso, la existencia de un mecanismo de bloqueo que actúa como semáforo de acceso a recursos compartidos –ver Figura 4.3–, evita que se produzcan situaciones problemáticas en el servidor.

Por último, en lo que respecta a los conceptos implícitos dentro de la interfaz, tal y como aparecen recogidos en 3.2.1, es interesante realizar el análisis que se aborda en los siguientes puntos.

4.4.4.4.1. Recursos

Al igual que ocurría con la especificación del estándar *WMS*, en *WFS* también se realiza una utilización de las *URI* que es meridianamente diferente de la promulgada en *REST*, siendo completamente aplicables las conclusiones mostradas para la primera en el punto 4.4.2.4.1. Por otro lado, la Tabla 4.28 muestra las tipologías de recursos que pueden diferenciarse en este caso:

Método	Recurso	Representación
GetCapabilities	Descriptor del servicio	XML
DescribeFeatureType	Descriptor de la capa	XML
GetPropertyValue	Objeto geográfico	XML
GetFeature	Objeto geográfico	GML
GetFeatureWithLock	Objeto geográfico	GML
LockFeature	Objeto geográfico	XML
Transaction	Transacción	XML
CreateStoredQuery	Consulta	XML
DropStoredQuery	Consulta	XML
ListStoredQueries	Consulta	XML
DescribeStoredQueries	Consulta	XML

Tabla 4.28 Relación entre recursos, representaciones y métodos *WFS*.

- Descriptor del servicio, accesible mediante la operación *GetCapabilities*. En este caso el *URI* corresponde con la ruta de acceso al recurso y, por tanto, se cumple con el requerimiento *REST* de utilización del protocolo *HTTP* como protocolo de aplicación, aunque no se saque partido de toda su potencialidad, puesto que no se utilizan los métodos *POST*, *PUT* y *DELETE* para crear, actualizar y eliminar, respectivamente, el recurso.
- Descriptor de la capa, accesible mediante la operación *DescribeFeatureType*. Al igual que en el caso anterior, el *URI* corresponde con la ruta de acceso al recurso y, por tanto, se cumple con el requerimiento *REST* de utilización del protocolo *HTTP* como protocolo de aplicación, con las mismas limitaciones anteriormente referidas.

- Geometrías vectoriales que forman las capas de información, accesibles mediante las operaciones *GetFeature* y *GetFeatureWithLock*. Se cumple con el requerimiento *REST*, dado que el *URI* se corresponde directamente con el acceso a cada uno de los recursos cuando se utiliza un filtro que apunte de forma unívoca al identificador del recurso en cuestión. En caso de que el filtro no sea tan restrictivo, el *URI* actúa como enlace a un listado de recursos.

La recuperación de subconjuntos dentro de un listado, en un proceso de paginación, se puede acometer utilizando los parámetros *STARTINDEX* y *COUNT* en sucesivas peticiones, permitiendo de esta forma hacer un recorrido por el mismo.

Para este tipo de recursos, el estándar tampoco aprovecha las opciones de creación, modificación o borrado que contempla *REST* a través de los métodos *PUT*, *POST* y *DELETE* del protocolo *HTTP*. Todo lo contrario, habilita la operación *Transaction* a tal efecto, siendo este un claro ejemplo de utilización de *HTTP* como protocolo de transporte y el *URI* un punto de acceso al servicio. Respecto a la operación *FeatureLock*, también se podría extraer la misma conclusión, puesto que se puede asimilar a una operación de actualización –por tanto susceptible de utilizar el método *PUT*– sobre el recurso, siendo la propiedad de estar bloqueado un atributo más del recurso.

- Transacción para la edición del contenido –entidades– de una determinada capa.
- Consultas, accesibles mediante la operación *DescribeStoredQueries*. En este caso el *URI* corresponde con la ruta de acceso al recurso y, por tanto, se cumple con el requerimiento *REST* de utilización del protocolo *HTTP* como protocolo de aplicación.

148

Capítulo aparte merece el análisis de los dos métodos diseñados en el estándar para la creación y eliminación de recursos de tipo filtro: *CreateStoredQuery* y *DropStoredQuery*, respectivamente. Aparentemente pudiera parecer que esta implementación es compatible con los principios de *REST*, pero un análisis más detallado nos permite ver que no utiliza un *URI* autodescriptivo en ninguna de las operaciones. Además, *DropStoredQuery* permite operar con el método *GET*, lo cual contraviene lo dispuesto en *REST* –que conmina a la utilización de *DELETE*– y lo recogido en materia de seguridad por la especificación del protocolo *HTTP*.

4.4.4.4.2. Representaciones

El estándar *WFS* obliga a la implementación de una única representación para cada una de las tipologías de recursos comentadas anteriormente –ver Tabla 4.28–. En todos los casos la representación exigida es *XML* –y su derivado *GML* para representar las geometrías resultantes de peticiones *GetFeature*–. No obstante, el estándar deja abierta la opción a que los proveedores de aplicaciones servidoras y clientes puedan manejar otras tipologías de representaciones, informando obligatoriamente de las mismas en el descriptor del servicio.

En cualquier caso, las reflexiones planteadas en 4.4.2.4.2 para el estándar *WMS* entorno a la utilización de *JSON* y *XML*, son enteramente aplicables.

4.4.4.4.3. Hipermedia

El estándar *WFS* muestra, en referencia a este apartado, el mismo comportamiento recogido en 4.4.2.4.3 para el estándar *WMS*. Como ejemplo, en la respuesta a una petición de acceso a una entidad concreta se podrían incluir los *URI* correspondientes a aquellas otras entidades con las que está conectada o que se encuentran a una cierta distancia de ella.

4.4.4.4.4. Mensajes autodescriptivos

El estándar *WFS* muestra, en referencia a este apartado, el mismo comportamiento recogido en 4.4.2.4.4 para el estándar *WMS*.

4.4.4.5. Estructuración en capas

En línea con lo recogido en 4.4.2.5, desde la perspectiva del estándar *WFS* se podría contemplar la existencia de nodos que realicen ciertas tareas. Tal y como se apuntó en el estándar *WMS*, a título ilustrativo, se podría transferir la capacidad de realizar transformaciones entre el sistema de referencia de los datos y el solicitado en la petición –indicado mediante el elemento *SRSNAME* incorporado en el filtro asociado a la petición–.

4.4.4.6. Código a demanda

La especificación *WFS* no recoge consideración alguna con respecto a la posibilidad de descargar código en el cliente para la ejecución del mismo. Aunque esta restricción tenga su origen en la inexistencia de una motivación clara para establecer los mecanismos que permitieran abordarla, es cierto que el acceso directo a los repositorios en los que se alojan las geometrías podría tener implicaciones importantes en el ámbito de la seguridad del sistema.

4.4.4.7. Conclusiones

Partiendo de la base de la existencia de unas interfaces *KVP* y *SOAP*, la ausencia de una interfaz *REST* es un aspecto que llama la atención puesto que, tratándose de un servicio de descarga y, por tanto, de acceso a recursos que existen físicamente, parecería conveniente disponer de un conjunto de operaciones de acceso a los mismos.

Recapitulando las consideraciones recogidas en el análisis del grado de cumplimiento de los principios de *REST* por parte de la actual especificación del protocolo *WFS*, se obtiene la Tabla 4.29.

Se puede colegir que la actual especificación no cumple con los requerimientos de *REST*. Sin embargo, a diferencia del estándar *WMS* –ver punto 4.4.2.7– en el cuál la adaptación de una interfaz *REST* implica la relajación de aspectos como la nomenclatura de los *URI*, en este caso sí resulta factible la creación de dicha interfaz de forma ortodoxa, atendiendo a los siguientes aspectos:

- Recursos homogéneos: todos los recursos tienen una característica común y es que no son virtuales, existen físicamente en diferentes repositorios.

Principio	Cumplimiento	Aspectos clave
División cliente-servidor	✓	
Ausencia de estado	✓	
Existencia de caché	✗	Las operaciones que implican recuperar información sobre las características del servicio, de las capas que publica o de las consultas almacenadas, pueden sacar partido de los mecanismos de cacheado presentes en el protocolo <i>HTTP</i> , aunque el estándar no hace referencia a ello. Las que permiten recuperar entidades – <i>GetFeature</i> –, en cambio, no es razonable que saquen provecho de la existencia de caché, al menos de forma general, puesto que difícilmente dos peticiones diferentes harán referencia a una misma entidad.
Interfaz uniforme	✗	Se utiliza el protocolo <i>HTTP</i> como mero protocolo de transporte en la mayoría de las operaciones, no aprovechando las posibilidades que presentan los métodos que éste ofrece para la gestión de recursos. Sin embargo, las características de los recursos implícitos sí permitirían aprovechar esta característica de <i>REST</i> .
Separación en capas	✓	
Código a demanda	✗	No se contempla esta posibilidad.

Tabla 4.29 Cumplimiento de los principios *REST* en el estándar *WFS*.

Por tanto, es perfectamente factible crear los *URI* de acceso a los mismos, disponiendo de los métodos *GET*, *POST*, *PUT* y *DELETE* para gestionar la consulta, actualización, creación y eliminación, respectivamente, de los recursos. La Tabla 4.30 muestra las correspondencias entre los métodos del protocolo *HTTP* y la interfaz *KVP* para la propuesta de interfaz *REST* de *WFS*.

Posteriormente, se presenta la propuesta de *UR*³⁵ para cada una de los recursos –Tabla 4.31–. Se considera que un mismo servidor *WFS* puede publicar más de un servicio.

- Representaciones según estándares oficiales y *de facto*: con carácter general, se ofrecerán en formato *XML* y *JSON*, aunque en el caso de las entidades, estas se describirán en *GML*, *GeoJSON* y cualquiera de los formatos vectoriales restantes recogidos en 4.3.4.1.
- Inclusión de contenidos hipermedia e implementación de mensajes autodescriptivos: facilitando el cumplimiento de estas características de *REST*.

³⁵ Las partes del *URI* que son variables se identifican entre los símbolos “<” y “>”.

Recurso	Método HTTP	Método WFS KVP
Descriptor del servicio	GET	GetCapabilities
	POST	-
	PUT	-
	DELETE	-
Descriptor de la capa	GET	DescribeFeatureType
	POST	-
	PUT	-
	DELETE	-
Objeto geográfico	GET	GetFeature / GetPropertyValue
	POST	LockFeature / Transaction
	PUT	Transaction
	DELETE	Transaction
Transacción	GET	Transaction
	POST	Transaction
	PUT	Transaction
	DELETE	Transaction
Consulta	GET	DescribeStoredQueries / ListStoredQueries
	POST	CreateStoredQuery
	PUT	CreateStoredQuery
	DELETE	DropStoredQuery

Tabla 4.30 Equivalencias entre operaciones HTTP y KVP para WFS.

Descriptor del servicio	
<a href="http://<server>:<port>/<service>/capabilities">http://<server>:<port>/<service>/capabilities	
GET	Permite obtener el fichero descriptor del servicio, recogiendo las distintas capas de información – <i>feature types</i> –, así como otros metadatos –autoría del servicio, condiciones de uso, etc.–.
POST	Permite actualizar el contenido del fichero descriptor, especificando el contenido a modificar. Operación disponible para perfiles de usuario con privilegios de administración.
PUT	Permite crear el fichero descriptor, especificando el contenido del mismo en la misma acción. Operación disponible para perfiles de usuario con privilegios de administración.
DELETE	Permite eliminar el fichero descriptor. Operación disponible para perfiles de usuario con privilegios de administración.
Descriptor de la capa	
<a href="http://<server>:<port>/<service>/featureTypes/<featureType>">http://<server>:<port>/<service>/featureTypes/<featureType>	
GET	Permite obtener el fichero descriptor de una capa – <i>feature type</i> – que muestra sus metadatos descriptivos. Si no se especifica el fragmento <i><featureType></i> el servicio debe devolver un fichero con enlaces a la representación por defecto de cada uno de los descriptores de capa.
POST	Permite actualizar el contenido del fichero descriptor, especificando el contenido a modificar. Operación disponible para perfiles de usuario con privilegios de administración.

PUT Permite crear el fichero descriptor, especificando el contenido del mismo en la misma acción. Operación disponible para perfiles de usuario con privilegios de administración.

DELETE Permite eliminar el fichero descriptor. Operación disponible para perfiles de usuario con privilegios de administración.

Objeto geográfico

<http://<server>:<port>/<service>/featureTypes/<featureType>/<featureID>?lock=<yes|no>>

<http://<server>:<port>/<service>/featureTypes/<featureType>/<spatialGrid1>/.../<spatialGridn>?lock=<yes|no>>

Permite obtener la entidad especificada mediante:

- Identificador concreto *-featureID-*. Se permite especificar una colección de identificadores separados por comas.

Identificadores de las celdas correspondientes a los diferentes niveles de indexación espacial: la mayoría de los clientes desconocen el identificador de la entidad a recuperar, salvo que se hayan practicado consultas anteriormente que hayan permitido conocerlo. Sin embargo, emulando el sistema empleado por el estándar *WMTS* y por los formatos de datos vectoriales *SIG*, se opta por la definición de un conjunto de niveles de indexación espacial –equivalentes a las diferentes matrices de *WMTS* que se pueden observar en la Figura 4.35– y en cada nivel se delimita una retícula de celdas –equivalentes a las teselas de *WMTS-*, según se muestra en la Figura 4.39, cuyas dimensiones –expresadas en unidades terreno según el *CRS-* varían en función del nivel. A cada una de las entidades vectoriales del repositorio al que apunta el servicio *WFS*, en el momento de su creación o actualización, se le asigna la celda o celdas en las que se encuentra ubicada, en función su geometría.

GET

Cuando el cliente realiza una búsqueda de entidades, especificará la codificación *<spatialGrid>* –en formato “fila,columna”– correspondiente a cada uno de los niveles de indexación. El servicio devolverá todas aquellas entidades que se ubiquen en las celdas correspondientes. A mayor número de niveles especificados, mayor restricción en el número de entidades encontradas. Este mecanismo se plantea como respuesta a las peticiones de búsqueda de entidades a partir de un ámbito espacial establecido. Para que el cliente pueda conocer cómo puede traducir un área de búsqueda a las correspondientes celdas, en el fichero descriptor de la capa se indicará el origen de coordenadas, así como el ancho y alto de cada una de las celdas, para cada uno de los niveles, expresados ambos en unidades del sistema de coordenadas en el que se encuentran almacenados los datos –siendo esta información otro de los metadatos recogidos en el citado fichero descriptor–.

Se puede incluir un parámetro “*lock*” de bloqueo de la entidad o las entidades recuperadas.

POST

Permite actualizar una entidad o un conjunto de entidades que formen parte de una determinada transacción –cuyo identificador se especifique en el fichero que acompaña a la petición–, en el repositorio asociado a una capa. La petición se realizará especificando el fragmento de *URI* que llega hasta la definición de la capa afectada *-featureType-*. En caso de que se realice una modificación de la geometría, se actualizará la codificación *<spatialGrid>* que permitirá su posterior recuperación.

PUT

Permite crear una entidad o un conjunto de entidades que formen parte de una determinada transacción –cuyo identificador se especifique en el fichero que acompaña a la petición–, en el repositorio asociado a una capa. La petición se realizará especificando el fragmento de *URI* que llega hasta la definición de la capa afectada *-featureType-*. Además, como parte del proceso de introducción de la entidad en el repositorio, se creará la codificación *<spatialGrid>* que permitirá su posterior recuperación.

DELETE	Permite eliminar una entidad o un conjunto de entidades que formen parte de una determinada transacción –cuyo identificador se especifique en el fichero que acompaña a la petición–, del repositorio asociado a una capa. La petición se realizará especificando el fragmento de <i>URI</i> que llega hasta la definición de la capa afectada – <i>featureType</i> –.
Transacción	
<a href="http://<server>:<port>/<service>/transactions/<transactionID>">http://<server>:<port>/<service>/transactions/<transactionID>	
GET	Permite obtener los metadatos que describen la transacción especificada mediante su identificador. Esta información referirá, entre otras cosas, el identificador del usuario que ha creado la transacción, la fecha o los identificadores de las entidades afectadas. Si no se especifica el fragmento <i><transactionID></i> , el servicio debe devolver un fichero con enlaces a la representación por defecto de cada una de las transacciones existentes.
POST	Permite actualizar la información asociada a la transacción asociada al identificador indicado.
PUT	Permite crear una nueva transacción, indicando toda la información asociada. Esta operación se realiza eliminando del <i>URI</i> el fragmento <i><transactionID></i> , dado que se el cliente desconoce cuál será el identificador que asignará el sistema a la nueva transacción.
DELETE	Permite eliminar la transacción asociada al identificador indicado.
Consulta	
<a href="http://<server>:<port>/<service>/queries/<queryID>">http://<server>:<port>/<service>/queries/<queryID>	
GET	Permite obtener los metadatos asociados a la consulta especificada mediante su identificador. Si no se especifica el fragmento <i><queryID></i> el servicio debe devolver un fichero con enlaces a la representación de cada una de las consultas almacenadas en el servidor.
POST	Permite actualizar la información asociada a la consulta asociada al identificador indicado.
PUT	Permite crear una nueva consulta, indicando toda la información asociada. Esta operación se realiza eliminando del <i>URI</i> el fragmento <i><queryID></i> , dado que se el cliente desconoce cuál será el identificador que asignará el sistema a la nueva consulta.
DELETE	Permite eliminar la consulta asociada al identificador indicado.

Tabla 4.31 Propuesta de recursos y operaciones asociadas a la interfaz *REST* del estándar *WFS*.

- Aprovechamiento de los mecanismos de cacheado: la interfaz debe recoger la posibilidad de utilizar las técnicas de cacheado que ofrece el protocolo *HTTP*, más allá del rendimiento que se pueda sacar a la existencia de una caché para las entidades geográficas. En todo caso, será una opción disponible en caso de que el escenario concreto de utilización resulte apropiado –usuarios que accedan de forma reiterada a un conjunto de entidades–.

A modo de resumen, la plataforma que se está diseñando debe mantener la interfaz *KVP* del estándar *OGC WMS*, junto con la interfaz *SOAP* definida en [Gre10] e incorporar el protocolo *REST*. En este último aspecto, el *OGC* ya registró una petición de cambio del estándar *WFS* para la creación de una interfaz *REST* [Coo08], aunque todavía no se ha visto materializada en el estándar. Por otro lado, la importancia de poder habilitar interfaces *REST* para acceder a repositorios de datos y realizar las típicas operaciones

de creación, consulta, modificación y borrado (*CRUD*), ha sido abordada por diferentes iniciativas de organizaciones dedicadas al desarrollo de software –sirva como ejemplo /db [Vre11]–.

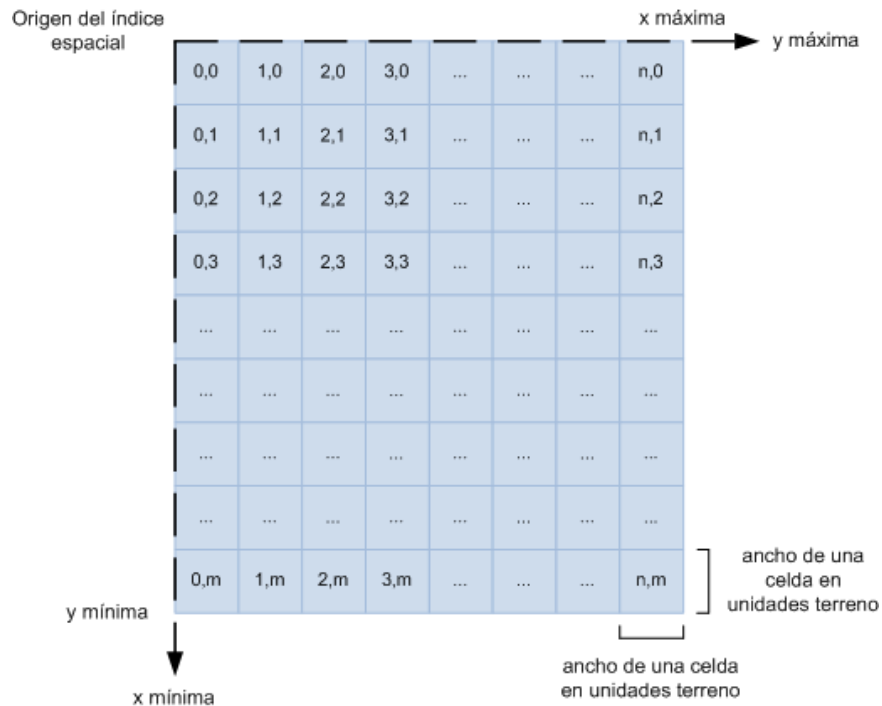


Figura 4.39 Conjunto de celdas que forman un índice espacial de resolución (n,m) [fuente: propia].

4.4.5 Servicio de descarga WCS

Para cada uno de estos métodos se definen una serie de interfaces, las cuales aparecen recogidas en la Tabla 4.32. La primera versión del estándar *WCS* permitía la utilización del protocolo *HTTP* sobre operaciones de tipo *GET* y *POST*. Con la publicación de la versión 1.1.0 ya se introdujo la variante *SOAP* y más recientemente, como resultado de los trabajos que está realizando el grupo de trabajo de *OGC* denominado *RESTful Service Policy Standard Working Group* [Sla14], se ha publicado con carácter de borrador, el documento de propuesta de una interfaz *REST* para el estándar *WCS* [OGC11b].

Es especialmente interesante como se aborda, desde la perspectiva del *OGC* esta nueva propuesta de interfaz, para lo se pasa a analizar el grado de alineación de esta propuesta con respecto a los principios de la arquitectura *REST*.

4.4.5.1. División cliente-servidor

Esta restricción la cumple el estándar *WCS* puesto que los clientes y los servidores compatibles interactúan mediante el paradigma de *WS*, tal y como se ha referido en 4.4.2.1. La adición de una nueva cobertura a una instancia de un servicio *WCS* no implica realizar cambios en la funcionalidad de la interfaz de usuario, porque ésta permanece inalterada.

Operación	Protocolos	Codificación
GetCapabilities	HTTP (Get)	KVP
	HTTP (Post)	XML
	HTTP (Post) + SOAP	XML
	REST ³⁶	-
DescribeCoverage	HTTP (Get)	KVP
	HTTP (Post)	XML
	HTTP (Post) + SOAP	XML
	REST	-
GetCoverage	HTTP (Get)	KVP
	HTTP (Post)	XML
	HTTP (Post) + SOAP	XML
	REST	-

Tabla 4.32 Protocolos y codificaciones admitidas en las operaciones WCS.

4.4.5.2. Interacciones sin estado

Según se indicó en 4.4.2.2, la comunicación entre clientes y servidores en cualquier servicio OGC es por naturaleza sin estado. En general, la interacción con el servicio sigue el patrón petición-respuesta en el que no existe dependencia alguna con interacciones previas.

4.4.5.3. Existencia de caché

La capacidad para activar de forma selectiva el almacenamiento de las respuestas, de forma que los clientes puedan reutilizarlas posteriormente, no se contempla en la especificación del estándar WCS y tampoco en el borrador de interfaz REST. En líneas generales, el razonamiento aplicado en el punto 4.4.2.3 para el estándar WMS sigue siendo válido, aunque es interesante reseñar ciertos aspectos diferenciales sobre la conveniencia de implementar este mecanismo.

Por un lado, nos encontramos con dos métodos, *GetCapabilities* y *DescribeCoverage*, que son claramente almacenables, puesto que se trata de ficheros XML con contenido estático –mientras no se realice un proceso de actualización del contenido del servicio–. Por otro lado, con respecto al método *GetCoverage*, dispone del parámetro *SUBSET* gracias al cual se puede especificar el subconjunto de la cobertura original que se desea obtener –Figura 4.40–. La probabilidad de que se reciban dos peticiones que coincidan exactamente en los requisitos de la cobertura resultante parece altamente improbable. Además, el almacenamiento en caché solo tendría sentido en datos que tienen tasas de actualización muy bajas.

³⁶ La interfaz REST del protocolo WCS no ha sido adoptada como estándar. Esta información se ha extraído del borrador elaborado por OGC [OGC11b].

```

SubsetSpec:      subset = axisName ( intervalOrPoint )
axisName:       NCName
intervalOrPoint: interval | point
interval:       low : high
low:            point | *
high:           point | *
point:          number | "token" // " = ASCII code 0x42

```

Figura 4.40 Sintaxis del parámetro *subset* en notación *EBNF* [OGC11b].

Pese a ello, cabe tener en cuenta las opciones recogidas en 4.4.2.3 sobre el protocolo *HTTP* y los mecanismos que éste habilita para el control de las cachés.

4.4.5.4. Interfaz uniforme

Como ya se ha reseñado anteriormente, la interfaz uniforme implica que el servidor expone una interfaz derivada de la semántica de los métodos del protocolo *HTTP*, actuando éste como protocolo de aplicación. En este caso, utilizando la interfaz *REST* presentada en el borrador, se cumple este requerimiento y se puede considerar que se utiliza el protocolo *HTTP* como protocolo de aplicación.

Dado que las operaciones definidas en el estándar no permiten realizar modificaciones de los repositorios publicados mediante un servicio *WCS*, la interfaz solo contempla la utilización del método *GET* del protocolo *HTTP* –ver Figura 4.34–. Dicho lo cual, es reseñable que las características de esta especificación muestran un grado de incumplimiento de la característica de seguridad y de cumplimiento de la de idempotencia equiparable al del estándar *WMS* –ver punto 4.4.2.4–.

156

Por último, en lo que respecta a los conceptos implícitos dentro de la interfaz, tal y como aparecen recogidos en 3.2.1, es interesante realizar el análisis que se aborda en los siguientes puntos.

4.4.5.4.1. Recursos

La especificación del estándar *WCS* permite identificar tres tipologías de recursos –ver Tabla 4.33–:

Método	Recurso	Representación
<i>GetCapabilities</i>	Descriptor del servicio	<i>XML</i>
<i>DescribeCoverage</i>	Descriptor de la cobertura	<i>XML</i>
<i>GetCoverage</i>	Cobertura	<i>GML o MIME-Type convenido</i>

Tabla 4.33 Relación entre recursos, representaciones y métodos en *WCS*.

- Descriptor del servicio, accesible mediante una petición de acceso al recurso –equivalente en las otras interfaces a *GetCapabilities*– que se corresponde con un *URI*. Por tanto, se cumple con el requerimiento *REST* de utilización del protocolo *HTTP* como protocolo de aplicación, aunque no se saque partido de toda su potencialidad, puesto que no se utilizan los métodos *POST*, *PUT* y

DELETE para crear, actualizar y eliminar, respectivamente, el recurso. Según el borrador del estándar, el acceso a este fichero descriptor se realizará utilizando la siguiente nomenclatura³⁷:

<http://<server>:<port>/capabilities>

- Descriptor de la cobertura, accesible mediante una petición de acceso al recurso –equivalente en las otras interfaces a *DescribeCoverage*– que se corresponde con un *URI*, cumpliéndose, por tanto, el requerimiento *REST* de utilización del protocolo *HTTP* como protocolo de aplicación, con las mismas limitaciones anteriormente referidas. Según el borrador del estándar, el acceso a este fichero descriptor se realizará utilizando la siguiente nomenclatura:

<http://<server>:<port>/coverage/<coverageIdentifier>/description>

- Cobertura, accesible mediante una petición de acceso al recurso –equivalente en las otras interfaces a *GetCoverage*– que se corresponde con un *URI*. Nuevamente se cumple con el requerimiento *REST* de utilización del protocolo *HTTP* como protocolo de aplicación, con las mismas limitaciones anteriormente referidas, sin las cuales podría plantearse una extensión funcional del estándar que permitiera la gestión de transacciones de actualización de las capas publicadas. Según el borrador del estándar, el acceso a este recurso se realizará utilizando la siguiente nomenclatura, aplicada a una hipotética cobertura espacio-temporal:

[http://<server>:<port>/coverage/<coverageIdentifier>/subset=lon\(<minValue>:<maxValue>\)/subset=lat\(<minValue>:<maxValue>\)/subset=t\(<date&time>\)](http://<server>:<port>/coverage/<coverageIdentifier>/subset=lon(<minValue>:<maxValue>)/subset=lat(<minValue>:<maxValue>)/subset=t(<date&time>))

4.4.5.4.2. Representaciones

El estándar *WCS* obliga a la implementación de una única representación para cada una de las tipologías de recursos comentadas anteriormente –ver Tabla 4.33–. En todos los casos, salvo para las geometrías en las que se trata de una variante *XML* específica, *GML*, la representación exigida es *XML*. No obstante, tal y como se ha comentado con anterioridad, se permite a los implementadores ofrecer formatos adicionales especificados mediante la codificación *MIME-type* oportuna; de forma que los clientes, haciendo uso de la cabecera de aceptación –*Accept header*– pueden especificar el formato deseado, siempre que sea uno de los admisibles por el servidor.

En cualquier caso, las reflexiones planteadas en 4.4.2.4.2 para el estándar *WMS* entorno a la utilización de *JSON* y *XML*, son enteramente aplicables. Además, también resultaría interesante que, de conformidad con las buenas prácticas recogidas en [Fre12], se incluyera en el *URI* de acceso a la representación de un recurso el formato del mismo mediante su extensión. De esta forma, a título ilustrativo, el acceso a la representación *XML* del descriptor del servicio utilizaría el siguiente *URI*:

<http://<server>:<port>/capabilities.xml>

³⁷ Las partes del *URI* que son variables se identifican entre los símbolos “<” y “>”.

4.4.5.4.3. Hipermedia

El estándar *WFS* muestra, en referencia a este apartado, el mismo comportamiento recogido en 4.4.2.4.3 para el estándar *WMS*. En este contexto, a modo de ejemplo, podría contemplarse la posibilidad de obtener los *URI* que identifican cada una de las celdas que componen una determinada cobertura.

4.4.5.4.4. Mensajes autodescriptivos

Este estándar especifica la posibilidad de utilizar cabeceras del protocolo *HTTP* para indicar el formato de representación deseado en el acceso a los recursos. En este sentido, se entiende que las respuestas contienen información suficiente para su interpretación completa sin precisar información adicional, cumpliéndose con esta característica.

4.4.5.5. Estructuración en capas

En línea con lo recogido en 4.4.2.5, desde la perspectiva del estándar *WCS* se podría contemplar la existencia de nodos que realicen ciertas tareas. Tal y como se apuntó en el estándar *WMS*, a título ilustrativo, se podría transferir la capacidad de realizar transformaciones entre el sistema de referencia de los datos y el solicitado en la petición –indicado mediante el parámetro *SUBSET*–.

4.4.5.6. Código a demanda

La especificación *WCS* no recoge consideración alguna con respecto a la posibilidad de descargar código en el cliente para la ejecución del mismo, pudiéndose plantear la misma reflexión recogida en 4.4.4.6 para el estándar *WFS*.

4.4.5.7. Conclusiones

Este estándar, tal y como ocurre con *WFS*, se caracteriza porque todas las tipologías de recursos existen físicamente; tanto los descriptores de servicio y cobertura, como la propia cobertura, son recursos identificables. Por ello, parece razonable plantear la especificación de una interfaz *REST* como la que se ha analizado. Sin embargo, un análisis más profundo pone de manifiesto ciertos puntos que no se corresponden con los principios de *REST*. Pártase primero del análisis del cumplimiento de los principios *REST* por parte de la actual especificación, según se muestra en la Tabla 4.34.

Como se puede observar, la situación planteada es equivalente a la que se presentó con motivo del estándar *WMS*. En las conclusiones recogidas en el punto 4.4.2.7 se ponía de manifiesto que el recurso mapa, como tal, realmente no existía y su generación en tiempo de ejecución no permitía sacar rendimiento de las características de cacheado, así como tampoco se permitía la definición de unos *URI* que mostraran claramente ventajas con respecto a las existentes para la interfaz *KVP*.

En este caso, la cobertura sí que existe –más allá de que esta esté formada por uno o varios ficheros–, pero las peticiones *GetCoverage* apuntan a fragmentos de la misma que constituyen en sí mismos

nuevos recursos, pudiendo ser estos además de una resolución diferente a la original³⁸, cuya generación adolece de las mismas consideraciones planteadas en el caso de la interfaz *KVP* del estándar *WMS*. Dicho esto y dado que en este caso desde el *OGC* sí se ha presentado un borrador de interfaz *REST* para *WCS*, cabría preguntarse el grado de cumplimiento de la misma con respecto a los principios *REST*.

Principio	Cumplimiento	Aspectos clave
División cliente-servidor	✓	
Ausencia de estado	✓	
Existencia de caché	✗	Las características de las peticiones hacen que resulte imposible sacar partido de los mecanismos de cacheado presentes en el protocolo <i>HTTP</i> , salvo en el caso de las peticiones referentes a los ficheros descriptores del servicio y de la cobertura.
Interfaz uniforme	✗	Se utiliza el protocolo <i>HTTP</i> como mero protocolo de transporte, no aprovechando las posibilidades que presentan los métodos que éste ofrece para la gestión de recursos.
Separación en capas	✓	
Código a demanda	✗	No se contempla esta posibilidad.

Tabla 4.34 Cumplimiento de los principios *REST* en el estándar *WCS*.

De los dos principios obligatorios –existencia de caché e interfaz uniforme–, el primero se resuelve mediante la utilización de las técnicas de cacheado propias del protocolo *HTTP*, aunque resulte dudosa su utilidad dada la alta improbabilidad, en términos generales, de que se repitan peticiones. Con respecto al segundo, hay que indicar que el *URI* correspondiente a la cobertura muestra una complejidad que excede los criterios de simplicidad abogados por *REST*. Como puede observarse –ver punto 4.4.5.4.1–, se ha añadido un fragmento –equivalente a un directorio virtual– al *URI* por cada parámetro que delimita la cobertura, emulando a lo que realiza la interfaz *KVP* con los parámetros de la *query string* –las objeciones que se pueden plantear son las mismas que las recogidas en el punto 4.4.2.7 para la propuesta de interfaz *REST* para el estándar *WMS*–. Además, la funcionalidad ofrecida es sensiblemente

³⁸ Dada una petición *WCS* en la que se especifica un área geográfica determinada –parámetro *BBOX*, el cuál recoge las coordenadas mínimas y máximas del área en cuestión– y un número de píxeles de anchura y altura de la cobertura resultante –parámetros *WIDTH* y *HEIGHT*–, se puede determinar las resoluciones en los dos ejes aplicando la siguiente fórmulas $res_x = \frac{x_{max}-x_{min}}{width}$ y $res_y = \frac{y_{max}-y_{min}}{height}$.

La especificación *WCS* también permite indicar directamente la resolución deseada mediante los parámetros *RESX* y *RESY*. En este caso, dado un *BBOX* determinado, al fijarse las resoluciones en los dos ejes, lo que se modifica es la medida en píxeles de la cobertura resultante, mediante las fórmulas: $width = \frac{x_{max}-x_{min}}{res_x}$ y $height = \frac{y_{max}-y_{min}}{res_y}$

inferior a la de la operación *GetCoverage* –esta última permite descargar coberturas con resoluciones y *CRS* diferentes a los de la imagen original–.

Por tanto, más allá de la nada desdeñable posibilidad de facilitar el acceso a este tipo de servicios desde aplicaciones basadas en arquitecturas *ROA*, son cuestionables las ventajas que pueda ofrecer esta nueva interfaz, especialmente si se tiene en cuenta la ausencia de mecanismos transaccionales que impide sacar partido, para ese mismo recurso, de las operaciones *POST*, *PUT* y *DELETE*. Cabrá analizar si finalmente el *OGC* oficializará esta u otra propuesta de interfaz y estudiar, llegado el caso, la especificación final y su posible extrapolación a otros estándares.

A modo de conclusión, en aras de favorecer al máximo la interoperabilidad de la plataforma distribuida propuesta, se adopta la disposición de una interfaz *REST* para el estándar *WCS*.

4.4.6 Servicio de descarga SOS

La especificación *SOS* define una serie de interfaces para cada uno de estos métodos, las cuales aparecen recogidas en la Tabla 4.35. La existencia de la opción de utilizar el protocolo *SOAP* sobre *HTTP*, utilizando el método *POST* de este último, pone de manifiesto que, entre los presupuestos arquitectónicos considerados por los diseñadores de este estándar, figuraba la arquitectura *SOA*.

Pese a ello, es interesante analizar el grado de alineación de este estándar con respecto a los principios de la arquitectura *ROA* y el protocolo *REST*, según se recoge en 3.2.2 y al igual que se ha hecho con los demás estándares analizados en este documento.

4.4.6.1. División cliente-servidor

Esta restricción la cumple el estándar *WFS* puesto que los clientes y los servidores compatibles interactúan mediante el paradigma de *WS*, tal y como se ha referido en 4.4.2.1.

La adición de un nuevo sensor a una instancia de un servicio *SOS* no implica realizar cambios en la funcionalidad de la interfaz de usuario, porque ésta permanece inalterada.

4.4.6.2. Interacciones sin estado

Según se indicó en 4.4.2.2, la comunicación entre clientes y servidores en cualquier servicio *OGC* es por naturaleza sin estado. En general, la interacción con el servicio sigue el patrón petición-respuesta en el que no existe dependencia alguna con interacciones previas.

4.4.6.3. Existencia de caché

La capacidad para activar de forma selectiva el almacenamiento de las respuestas, de forma que los clientes puedan reutilizarlas posteriormente ante la necesidad de plantear las mismas peticiones, no se contempla en la especificación del estándar *SOS*. En líneas generales, el razonamiento aplicado en el punto 4.4.2.3 para el estándar *WMS* sigue siendo válido, aunque es interesante reseñar ciertos aspectos diferenciales sobre la conveniencia de implementar este mecanismo.

Operación	Protocolos	Codificación
GetCapabilities	HTTP (Get)	KVP
	HTTP (Post)	XML
	HTTP (Post) + SOAP	XML
DescribeSensor	HTTP (Get)	KVP
	HTTP (Post)	XML
	HTTP (Post) + SOAP	XML
GetObservation	HTTP (Get)	KVP
	HTTP (Post)	XML
	HTTP (Post) + SOAP	XML
GetObservationById	HTTP (Get)	KVP
	HTTP (Post)	XML
	HTTP (Post) + SOAP	XML
InsertSensor	HTTP (Get)	KVP
	HTTP (Post)	XML
	HTTP (Post) + SOAP	XML
DeleteSensor	HTTP (Get)	KVP
	HTTP (Post)	XML
	HTTP (Post) + SOAP	XML
InsertObservation	HTTP (Get)	KVP
	HTTP (Post)	XML
	HTTP (Post) + SOAP	XML
GetFeatureOfInterest	HTTP (Get)	KVP
	HTTP (Post)	XML
	HTTP (Post) + SOAP	XML
GetResult	HTTP (Get)	KVP
	HTTP (Post)	XML
	HTTP (Post) + SOAP	XML
GetResultTemplate	HTTP (Get)	KVP
	HTTP (Post)	XML
	HTTP (Post) + SOAP	XML
InsertResult	HTTP (Get)	KVP
	HTTP (Post)	XML
	HTTP (Post) + SOAP	XML
InsertResultTemplate	HTTP (Get)	KVP
	HTTP (Post)	XML
	HTTP (Post) + SOAP	XML

Tabla 4.35 Protocolos y codificaciones admitidas en las operaciones SOS.

Entre las operaciones consideradas como nucleares y, por tanto obligatorias, *GetCapabilities* y *DescribeSensor* proporcionan un tipo de información que es claramente almacenable, al tratarse de ficheros *XML* con contenido estático –al igual que en otros estándares, mientras no se realice un proceso de actualización del contenido del servicio–. Esta misma consideración podría realizarse con la operación *GetResultTemplate*, entendiéndose que la plantilla por la que se interroga no se modifica de forma habitual y, por tanto, puede tratarse como contenido estático.

Por su parte, las operaciones *GetObservation*, *GetObservationById*, *GetResult* y *GetFeatureOfInterest* apuntan a la recuperación de una información estática que, una vez captada por los sensores y enviada al servidor *SOS*, no sufre actualizaciones o transformaciones. Por tanto, si el análisis de un entorno concreto pone de manifiesto que los clientes precisarán acceder con una frecuencia superior al tiempo de rotación de los repositorios de cacheado, en todas esas operaciones será interesante habilitar la gestión de memoria cache.

Las operaciones restantes, en cambio, tienen una idiosincrasia difícilmente compatible con la obtención de un rendimiento por el hecho de almacenar en cache las respuestas. Las que hacen referencia a la inserción de sensores, observaciones o resultados en el servidor, son operaciones de almacenamiento, nunca de recuperación; por tanto carece de sentido arbitrar el mecanismo de cacheado. Por su parte, la de borrado del sensor, tampoco es una operación de consulta, de ahí que se pueda practicar la misma consideración. Pese a ello, cabe tener en cuenta las opciones recogidas en 4.4.2.3 sobre el protocolo *HTTP* y los mecanismos que éste habilita para el control de las cachés.

4.4.6.4. Interfaz uniforme

Como ya se ha reseñado anteriormente, la interfaz uniforme implica que el servidor expone una interfaz derivada de la semántica de los métodos del protocolo *HTTP*, actuando éste como protocolo de aplicación. En este caso, en cualquiera de las opciones de protocolo, método y codificación ofrecidas por *SOS*, no se cumple con este requerimiento en todos los métodos del estándar –ver Figura 4.34–. Este aspecto es evidente si se tiene en cuenta el significado de las operaciones que implican una modificación de la información almacenada en el servidor, como *InsertSensor*, *DeleteSensor*, *InsertObservation*, *InsertResult* e *InsertResultTemplate*. En ellas, el significado de la petición –insertar o eliminar recursos, según los casos– es parte del *URI*, respondiendo conceptualmente al modelo *RPC*. Los métodos restantes también siguen este modelo. Sin embargo, podría realizarse una adaptación de todos ellos siguiendo las recomendaciones de *REST* en cuanto a los métodos *HTTP* a utilizar y su nomenclatura, tal y como se analizará posteriormente.

Abundando en el análisis, también se produce una utilización incorrecta de los métodos del protocolo *HTTP*, desde el punto de vista de *REST*, puesto que todas las operaciones contemplan exclusivamente la utilización de los métodos *GET* y *POST*, independientemente de si se tratan de operaciones de consulta, actualización, inserción o eliminación de información. De este hecho también se derivan dos consecuencias adicionales. Por un lado, la circunstancia recogida, junto con la posibilidad de utilizar el método *POST*, que por definición no es idempotente, deriva el incumplimiento de la característica de idempotencia. Baste analizar la semántica de las diferentes operaciones que pueden implicar diferentes respuestas a una misma petición, para llegar a esta conclusión:

- *InsertSensor*: la ejecución reiterada de este método no tendría consecuencias siempre que la instancia del servidor SOS realice una verificación gracias a la cual no se permita la inserción de sensores con la misma información asociada –metadatos–. En caso contrario, se obtendrían diferentes identificadores para cada petición, contraviniendo la esencia de la idempotencia.
- *DeleteSensor*: la ejecución reiterada de este método no tiene consecuencias, ya que las peticiones de eliminación posteriores intentarán eliminar un recurso que no existe y el servidor devolverá un código de error sin mayores consecuencias.
- *InsertObservation*: la ejecución reiterada de este método no tiene consecuencias, puesto que el servidor SOS ha de verificar si en el instante en el que se ha tomado dicha observación –phenomenonTime– ya existe una observación previa asociada al mismo identificador del *offering* indicado en la petición.
- *InsertResult*: se aplica la misma reflexión realizada para el método *InsertObservation*.
- *InsertResultTemplate*: la ejecución reiterada de este método no tendrá consecuencias siempre que la instancia del servidor SOS realice una verificación gracias a la cual no se permita la inserción de la misma plantilla –idénticos metadatos referentes al procedimiento, elemento de interés y propiedad observada–. En caso contrario, se contravendría la propiedad analizada.

Por otro lado, en cuanto a la seguridad y dejando a un lado la utilización del método *POST* –el cual por definición no es seguro– para la actualización del servidor, la utilización del método *GET* con estos mismos fines, provocando la ejecución de procesos que pueden conllevar modificaciones en el estado del servidor, vulnera la citada característica.

Por último, en lo que respecta a los conceptos implícitos dentro de la interfaz, tal y como aparecen recogidos en 3.2.1, se aborda su análisis en los siguientes puntos.

4.4.6.4.1. Recursos

Al igual que ocurría con las especificaciones de los estándares *WMS* y *WFS*, en *SOS* también se realiza una utilización de los *URI* que es meridianamente diferente de la promulgada en *REST*, siendo completamente aplicables las conclusiones mostradas para la primera en el punto 4.4.2.4.1. Por otro lado, la Tabla 4.36 muestra las tipologías de recursos que pueden diferenciarse en este caso:

- Servicio, accesible mediante la operación *GetCapabilities* que devuelve el fichero descriptor del mismo. En este caso, el *URI* corresponde con la ruta de acceso al recurso y, por tanto, se cumple con el requerimiento *REST* de utilización del protocolo *HTTP* como protocolo de aplicación, aunque no se saque partido de toda su potencialidad, puesto que no se utilizan los métodos *POST*, *PUT* y *DELETE* para crear, actualizar y eliminar, respectivamente, el recurso.
- Sensor, gestionado mediante las operaciones *InsertSensor*, *DeleteSensor* y *DescribeSensor* en lo que respecta a las operaciones de creación, eliminación y consulta. En este sentido, se contraviene lo dispuesto en *REST* sobre la utilización de los métodos *PUT*, *POST* y *DELETE* para las operaciones de creación, modificación y borrado del recurso.

Método	Recurso	Representación
GetCapabilities	Servicio	XML
DescribeSensor	Sensor	SensorML
InsertSensor	Sensor	XML
DeleteSensor	Sensor	XML
GetObservation	Observación	XML
GetObservationById	Observación	XML
InsertObservation	Observación	XML
GetFeatureOfInterest	Entidad de interés	XML
GetResult	Resultado	XML
InsertResult	Resultado	XML
GetResultTemplate	Plantilla resultado	XML
InsertResultTemplate	Plantilla resultado	XML

Tabla 4.36 Relación entre recursos, representaciones y métodos en SOS.

- Observación, gestionado mediante las operaciones *InsertObservation*, *GetObservation* y *GetObservationById* en lo que respecta a las operaciones de creación y consulta –dos últimas operaciones–. En este sentido, se contraviene lo dispuesto en *REST* sobre la utilización de los métodos *PUT* para la operación de creación, además de no aprovechar la versatilidad del protocolo *HTTP* para la actualización y eliminación del recurso.
- Entidad de interés, accesible mediante la operación *GetFeatureOfInterest*. En este caso se vuelve a utilizar el *URI* como ruta de acceso al recurso y, por tanto, se cumple con el requerimiento *REST* de utilización del protocolo *HTTP* como protocolo de aplicación, aunque no se saque partido de toda su potencialidad, puesto que no se utilizan los métodos *POST*, *PUT* y *DELETE* para crear, actualizar y eliminar, respectivamente, el recurso.
- Resultado, gestionado mediante las operaciones *InsertResult* y *GetResult* en lo que respecta a las operaciones de creación y consulta, respectivamente. Bajo esta óptica, los comentarios recogidos anteriormente para las observaciones son completamente aplicables.
- Plantilla de resultado, gestionado mediante las operaciones *InsertResultTemplate* y *GetResultTemplate* en lo que respecta a las operaciones de creación y consulta, respectivamente. En este sentido, los comentarios recogidos anteriormente para las observaciones son completamente aplicables.

4.4.6.4.2. Representaciones

El estándar SOS recoge una única representación para cada una de las tipologías de recursos comentadas anteriormente –ver Tabla 4.36–. En todos los casos, salvo para los sensores en las que se trata de una variante específica XML, denominada *SensorML*, la representación exigida es XML. No obstante, se deja abierta la opción a que los proveedores de aplicaciones servidoras y clientes puedan manejar otras tipologías de representaciones, informando obligatoriamente de las mismas en el descriptor del servicio. En cualquier caso, las reflexiones planteadas en 4.4.2.4.2 para el estándar WMS entorno a la utilización de JSON y XML, son enteramente aplicables.

4.4.6.4.3. Hipermedia

El estándar SOS muestra, en referencia a este apartado, el mismo comportamiento recogido en 4.4.2.4.3 para el estándar WMS. A título ilustrativo, se podría incorporar en la representación de un recurso de tipo sensor enlaces con los URI correspondientes a las observaciones y resultados vinculados con el mismo.

4.4.6.4.4. Mensajes autodescriptivos

El estándar SOS muestra, en referencia a este apartado, el mismo comportamiento recogido en 4.4.2.4.4 para el estándar WMS.

4.4.6.5. Estructuración en capas

En línea con lo recogido en 4.4.2.5, desde la perspectiva del estándar SOS se podría contemplar la existencia de nodos intermedios que realicen ciertas tareas, como por ejemplo la conversión entre diferentes unidades de medición de las variables observadas, allí donde sea aplicable.

4.4.6.6. Código a demanda

La especificación SOS no recoge consideración alguna con respecto a la posibilidad de descargar código en el cliente para la ejecución del mismo. Aunque esta restricción tenga su origen en la inexistencia de una motivación clara para establecer los mecanismos que permitieran abordarla, es cierto que el acceso directo a los repositorios en los que se alojan los datos sobre los sensores, las observaciones, etc., podría tener implicaciones importantes en el ámbito de la seguridad del sistema.

4.4.6.7. Conclusiones

El análisis del cumplimiento de los principios REST por parte de la interfaz KVP del estándar SOS puede resumirse en la Tabla 4.37.

Las características de las operaciones y de los recursos identificados en este estándar, sin embargo, permiten definir una interfaz REST, corrigiendo los aspectos en los que la actual especificación del estándar no se ajusta a los principios de esta interfaz:

Principio	Cumplimiento	Aspectos clave
División cliente-servidor	✓	
Ausencia de estado	✓	
Existencia de caché	✗	Aunque la interfaz no recoja la utilización de mecanismos de cacheado, es cierto que las características de las peticiones hacen que resulte posible sacar partido de los mecanismos de cacheado presentes en el protocolo <i>HTTP</i> .
Interfaz uniforme	✗	Se utiliza el protocolo <i>HTTP</i> como mero protocolo de transporte, no aprovechando las posibilidades que presentan los métodos que éste ofrece para la gestión de recursos.
Separación en capas	✓	
Código a demanda	✗	No se contempla esta posibilidad.

Tabla 4.37 Cumplimiento de los principios *REST* en *SOS*.

- Recursos homogéneos: todos los recursos tienen una característica común y es que no son virtuales, existen físicamente en diferentes repositorios. Por tanto, es perfectamente factible crear los *URI* de acceso a los mismos, disponiendo de los métodos *GET*, *POST*, *PUT* y *DELETE* para gestionar su consulta, actualización, creación y eliminación. De esta forma también se solucionan los problemas relacionados con el incumplimiento de las características de seguridad e idempotencia. La Tabla 4.38 muestra las correspondencias entre los métodos del protocolo *HTTP* y la interfaz *KVP* para la propuesta de interfaz *REST* de *SOS*.

Mostradas las equivalencias, la Tabla 4.39 presenta la propuesta de *URI*³⁹ para cada una de los recursos. Se considera que un mismo servidor *SOS* puede publicar más de un servicio.

- Representaciones según estándares oficiales y *de facto*: con carácter general, se ofrecerán en formato *XML* –con sus variantes *sensorML* y *TML* cuando la respuesta haga referencia a información sobre un sensor o un resultado, respectivamente, mediante “*streaming*”–.
- Inclusión de contenidos hipermedia e implementación de mensajes autodescriptivos: facilitando el cumplimiento de estas características de *REST*. En el primero de los aspectos y a título ilustrativo, en el fichero *XML* correspondiente a un sensor se podrían incluir el *URI* de las observaciones vinculadas al mismo.
- Aprovechamiento de los mecanismos de cacheado: la interfaz debe recoger la posibilidad de utilizar las técnicas de cacheado que ofrece el protocolo *HTTP*.

³⁹ Las partes del *URI* que son variables se identifican entre los símbolos “<” y “>”.

Recurso	Método HTTP	Método WFS KVP
Descriptor del servicio	GET	GetCapabilities
	POST	-
	PUT	-
	DELETE	-
Objeto de interés	GET	GetFeatureOfInterest
	POST	-
	PUT	-
	DELETE	-
Sensor	GET	DescribeSensor
	POST	-
	PUT	InsertSensor
	DELETE	DeleteSensor
Observación	GET	GetObservation / GetObservationById
	POST	-
	PUT	InsertObservation
	DELETE	-
Plantilla de resultado	GET	GetResultTemplate
	POST	-
	PUT	InsertResultTemplate
	DELETE	-
Resultado	GET	GetResult
	POST	-
	PUT	InsertResult
	DELETE	-

Tabla 4.38 Equivalencias entre operaciones HTTP y KVP para SOS.

Servicio	
<a href="http://<server>:<port>/<service>/capabilities">http://<server>:<port>/<service>/capabilities	
GET	Permite obtener el fichero descriptor del servicio, conteniendo los metadatos relativos al mismo – autoría del servicio, condiciones de uso, etc.–.
POST	Permite actualizar el contenido del fichero descriptor, especificando el contenido a modificar. Operación disponible para perfiles de usuario con privilegios de administración.
PUT	Permite crear el fichero descriptor, especificando el contenido del mismo en la misma acción. Operación disponible para perfiles de usuario con privilegios de administración.
DELETE	Permite eliminar el fichero descriptor. Operación disponible para perfiles de usuario con privilegios de administración.

Objeto de interés

<http://<server>:<port>/<service>/features/<featureID>>

GET	Permite obtener información sobre una determinada entidad de interés – <i>feature of interest</i> – correspondiente al identificador especificado. Si no se especifica el fragmento <i><featureOfInterestID></i> el servicio debe devolver un fichero con enlaces a la representación por defecto de cada una de las entidades registradas.
POST	Permite actualizar el contenido asociado a una entidad. Operación disponible para perfiles de usuario con privilegios de administración.
PUT	Permite crear el registro correspondiente a una nueva entidad, especificando sus metadatos. En este caso, no se especifica el fragmento <i><featureOfInterestID></i> , puesto que es el sistema el que asignará un identificador unívoco a la nueva entidad.
DELETE	Permite eliminar la entidad del repositorio. Operación disponible para perfiles de usuario con privilegios de administración.

Sensor

<http://<server>:<port>/<service>/features/<featureID>/<sensorID>>

<http://<server>:<port>/<service>/sensors/<sensorID>>

GET	Permite obtener información sobre un determinado sensor, pudiendo: <ul style="list-style-type: none"> ▪ Especificar el identificador de la entidad de interés a la que se encuentra asociado y el identificador concreto del sensor. En este caso, si no se especifica el fragmento <i><sensorID></i> el servicio debe devolver un fichero con enlaces a la representación por defecto de cada uno de los sensores asociados a la entidad referida. ▪ Especificar el identificador del sensor. En este caso, si no se especifica el fragmento <i><sensorID></i> el servicio debe devolver un fichero con enlaces a la representación por defecto de cada uno de los sensores registrados.
POST	Permite actualizar los metadatos registrados sobre el sensor especificado mediante su identificador unívoco.
PUT	Permite crear un nuevo sensor, asociándolo a la entidad cuyo identificador sea especificado en los metadatos del sensor. En este caso no se añadirá el fragmento <i><sensorID></i> , puesto que este será asignado por el servidor en el momento que se proceda al registro.
DELETE	Permite eliminar del registro el sensor con el identificador especificado en la petición.

Observación

<http://<server>:<port>/<service>/sensors/<sensorID>/observations/<observationID>>

<http://<server>:<port>/<service>/observations/<observationID>>

GET	Permite obtener información sobre una determinada observación, pudiendo: <ul style="list-style-type: none"> ▪ Especificar el identificador del sensor al que se encuentra asociada y el identificador concreto de la observación. En este caso, si no se especifica el fragmento <i><observationID></i> el servicio debe devolver un fichero con enlaces a la representación por defecto de cada una de las observaciones asociadas al sensor referido. ▪ Especificar el identificador de la observación. En este caso, si no se especifica el fragmento <i><observationID></i> el servicio debe devolver un fichero con enlaces a la representación por defecto de cada una de las observaciones registradas.
------------	--

POST	Permite actualizar la información asociada a la observación relacionada con el identificador indicado.
PUT	Permite crear una nueva observación, indicando toda la información asociada. En este caso no se añadirá el fragmento <code><observationID></code> , puesto que este será asignado por el servidor en el momento que se proceda al registro.
DELETE	Permite eliminar la observación asociada al identificador indicado.
Plantilla de resultado	
<a href="http://<server>:<port>/<service>/templates/<templateID>">http://<server>:<port>/<service>/templates/<templateID>	
GET	Permite obtener los metadatos que describen la plantilla de resultados especificada mediante su identificador. Si no se especifica el fragmento <code><templateID></code> el servicio debe devolver un fichero con enlaces a la representación de cada una de las plantillas registradas.
POST	Permite actualizar la información asociada a la plantilla asociada al identificador indicado.
PUT	Permite crear una nueva plantilla de resultado, indicando toda la información asociada.
DELETE	Permite eliminar la plantilla asociada al identificador indicado.
Resultado	
<a href="http://<server>:<port>/<service>/sensors/<sensorID>/observations/<observationID>/<resultID>">http://<server>:<port>/<service>/sensors/<sensorID>/observations/<observationID>/<resultID> <a href="http://<server>:<port>/<service>/observations/<observationID>/results/<resultID>">http://<server>:<port>/<service>/observations/<observationID>/results/<resultID>	
GET	Permite obtener información sobre un determinado resultado, partiendo de la identificación de la observación a la que se encuentra asociado, de ahí que existan dos alternativas, correspondientes a las dos mostradas para indicar una observación concreta. Si no se especifica el fragmento <code><resultID></code> el servicio debe devolver un fichero con enlaces a la representación por defecto de cada uno de los resultados asociados a la observación referida.
POST	Permite actualizar la información asociada al resultado asociado al identificador indicado.
PUT	Permite crear un nuevo resultado, indicando toda la información asociada. En este caso no se añadirá el fragmento <code><resultID></code> , puesto que este será asignado por el servidor en el momento que se proceda al registro.
DELETE	Permite eliminar el resultado asociado al identificador indicado.

Tabla 4.39 Propuesta de recursos y operaciones asociadas a la interfaz REST del estándar SOS.

A modo de resumen, la plataforma que se está diseñando debe mantener la interfaz *KVP* del estándar *OGC WMS*, junto con la interfaz *SOAP* definida en [Bro12] e incorporar el protocolo *REST*.

4.4.7 Servicio de geoprocésamiento WPS

La especificación *WPS* define una serie de interfaces para cada uno de estos métodos, las cuales aparecen recogidas en la Tabla 4.40. La existencia de la opción de utilizar el protocolo *SOAP* sobre *HTTP*, utilizando el método *POST* de este último, pone de manifiesto que, entre los presupuestos

arquitectónicos considerados por los diseñadores de este estándar, figuraba la arquitectura SOA. Pese a ello, es interesante analizar el grado de alineación de este estándar con respecto a los principios de la arquitectura ROA y el protocolo REST, según se recoge en 3.2.2 [Gra12].

Operación	Protocolos	Codificación
GetCapabilities	HTTP (Get)	KVP
	HTTP (Post)	XML
	HTTP (Post) + SOAP	XML
DescribeProcess	HTTP (Get)	KVP
	HTTP (Post)	XML
	HTTP (Post) + SOAP	XML
Execute	HTTP (Get)	KVP
	HTTP (Post)	XML
	HTTP (Post) + SOAP	XML

Tabla 4.40 Protocolos y codificaciones admitidas en las operaciones WPS.

4.4.7.1. División cliente-servidor

Esta restricción la cumple el estándar WPS puesto que los clientes y los servidores compatibles interactúan mediante el paradigma de WS, el cual promueve una clara distinción entre consumidores y proveedores en términos de roles y funcionalidades. La adición de un nuevo proceso a una instancia de un servicio WPS no implica realizar cambios en las capacidades o en la funcionalidad de la interfaz de usuario, porque la interfaz del servicio permanece inalterada.

4.4.7.2. Interacciones sin estado

Según se indicó en 4.4.2.2, la comunicación entre clientes y servidores en cualquier servicio OGC es por naturaleza sin estado. En general, la interacción con el servicio sigue el patrón petición-respuesta en el que no existe dependencia alguna con interacciones previas. En WPS, un agente usuario está capacitado para determinar los parámetros de entrada y salida de cualquier proceso, a partir de la descripción del mismo que se realiza en la respuesta a una petición *DescribeProcess*.

4.4.7.3. Existencia de caché

La capacidad para activar de forma selectiva el almacenamiento de las respuestas, de forma que los clientes puedan reutilizarlas posteriormente ante la necesidad de plantear las mismas peticiones, no se contempla en la especificación del estándar WPS. Este hecho viene justificado por la esencia propiamente dicha del cacheado, el cual tiene sentido cuando se recupera información estática, cosa que es altamente improbable para las peticiones de ejecución de procesos, aunque no cuando se trata de los métodos *GetCapabilities* y *DescribeProcess*.

Pese a ello, cabe tener en cuenta las opciones recogidas en 4.4.2.3 sobre el protocolo HTTP y los mecanismos que este habilita para el control de las cachés.

4.4.7.4. Interfaz uniforme

Como ya se ha reseñado anteriormente, la interfaz uniforme implica que el servidor expone una interfaz derivada de la semántica de los métodos del protocolo *HTTP*, actuando éste como protocolo de aplicación. En este caso, en cualquiera de las opciones de protocolo, método y codificación ofrecidas por *WPS*, no se cumple con este requerimiento. Se utiliza *HTTP* como mero protocolo de transporte sobre el cual se encapsulan peticiones en las que la semántica de las mismas reside en el *URI* –método *Get*– o en el fichero *XML* asociado –método *Post*–, respondiendo conceptualmente a las *RPC*.

Por otro lado, mientras que las operaciones *GetCapabilities* y *DescribeProcess* permiten recuperar información, la operación *Execute* puede alterar el estado del servidor. Encapsulando esta última operación a través del método *Get* se viola la propiedad de seguridad del mismo, tal y como ésta es definida en la especificación del protocolo *HTTP*. Además, dado que la repetición de una misma petición puede conllevar diferentes respuestas –aunque solo sea por el simple hecho de poder depender de recursos externos que actúan como parámetros de entrada y que pueden estar fuera del control del cliente que realiza la petición–, también se vulnera el principio de idempotencia.

Entrando en los conceptos implícitos dentro de la interfaz, tal y como aparecen recogidos en 3.2.1, se aborda su análisis en los siguientes puntos.

4.4.7.4.1. Recursos

Al igual que ocurría con otras especificaciones como *WMS*, *WFS* o *SOS*, en *WPS* también se realiza una utilización de los *URI* que diverge de la promulgada en *REST*, siendo completamente aplicables las conclusiones mostradas para la primera en el punto 4.4.2.4.1. Por otro lado, si se analizan los diferentes métodos, los únicos que asocian un *URI* a un recurso son los denominados *GetCapabilities* y *DescribeProcess*. En el otro, el *URI* hace referencia, a un punto de acceso a un servicio –tal y como puede observarse en la Figura 4.34–, como ya se apuntó en 4.4.2.4.

En otro orden de cosas, dejando a un lado la consideración conceptual anterior, el estándar *WPS* refiere que el identificador de un proceso debe ser un nombre que permita definir de forma unívoca dicho proceso dentro del servicio, añadiendo la opción de especificar un Nombre de Recurso Uniforme (*URN*) [Sch07], el cual permita categorizar dicho proceso. El identificador en cuestión no permite enlazar un recurso mediante un *URI* de forma directa. Por tanto, como resultado de una petición *GetCapabilities* en la que se refieren, entre otras cosas, los identificadores de los procesos alojados en el servicio, el cliente no obtiene directamente el *URI* del recurso, sino que ha de crearlo a partir del *URI* del servicio y del *URN* del proceso, lo cual implica que el cliente deba conocer las reglas de construcción de los *URI* y, por tanto, aumenta el grado de acoplamiento entre cliente y servidor.

A modo de ejemplo, la Figura 4.41 muestra un fragmento extraído de la respuesta a la ejecución de la petición <http://www.ign.es/wps-transformacion/servicios?REQUEST=GetCapabilities&SERVICE=WPS>, en el que se resume un proceso denominado *RasterStatistics* publicado por el Instituto Geográfico Nacional (IGN).

Para establecer el *URI* de la operación *DescribeProcess* será preciso concatenar, al valor del atributo *xlink:href* del elemento *ows:Get* o *ows:Post*, la *querystring* correspondiente:

<http://www.ign.es/wps-transformacion/servicios?REQUEST=DescribeProcess&SERVICE=WPS&IDENTIFIER=TransformCoordinates&VERSION=1.0.0>

```

- <ows:Operation name="Execute">
  - <ows:DCP>
    - <ows:HTTP>
      <ows:Get xlink:href="http://www.ign.es/wps-transformacion/servicios?"/>
      <ows:Post xlink:href="http://www.ign.es/wps-transformacion/servicios"/>
    </ows:HTTP>
  </ows:DCP>
</ows:Operation>
</ows:OperationsMetadata>
- <wps:ProcessOfferings>
  - <wps:Process wps:processVersion="1.0.0">
    <ows:Identifier>TransformCoordinates</ows:Identifier>
    - <ows:Title>
      Proceso para transformar una geometría GML de un sistema de referencia a otro
    </ows:Title>
    - <ows:Abstract>
      El propósito de este proceso es transformar geometrías o coberturas de un sistema de coordenadas en otro. Las versiones del estándar GML que soporta son 2.1.1, 3.0.1, 3.1.1 o 3.2.1. Se basa en el borrador de la guía técnica para los servicios de transformación de coordenadas Inspire, versión 2.0 de 09/07/2009. Para la transformación entre los Datum ED50 y ETRS89 se utiliza la rejilla avalada por el Grupo de Trabajo para la transición a ETRS89 compuesto por todas las Comunidades Autónomas y los Ministerios de Fomento (IGN) y Defensa, que se encuentra disponible para su descarga en www.ign.es. Consta de dos rejillas, una para la península (PENR2009.gsb, anteriormente llamada R2009v9.gsb siendo idéntica a ésta) y otra para Baleares denominada BALR2009.gsb. Ambas rejillas están en formato NTV2 y contienen los incrementos en longitud y latitud entre estos dos datum. Esta transformación no destruye topología alguna, es eficiente y continúa con un grado de acuerdo de solo algunos centímetros sobre todo el territorio.
    </ows:Abstract>
    <wps:Profile>urn:ogc:wps:1.0.0:BKG:TransformCoordinates:1.0</wps:Profile>
  </wps:Process>
</wps:ProcessOfferings>

```

Figura 4.41 Fragmento de respuesta *GetCapabilities* de un servicio WPS [IGN14].

4.4.7.4.2. Representaciones

El estándar WPS recoge una representación –siendo ésta en formato XML– para las operaciones *GetCapabilities* y *DescribeProcess*. Solo se admite una representación alternativa para la operación *Execute*, pudiendo en este caso optar entre los datos en crudo –*raw data*– o en formato XML. Todo ello, dejando a un lado consideraciones sobre la existencia previa del recurso o su generación como consecuencia de una petición *RPC* envuelta en una falsa cobertura *REST*.

4.4.7.4.3. Hipermedia

Abundando en lo indicado en 4.4.2.4.3, ante un error en la ejecución de un proceso, el cliente recibirá un mensaje XML con un código de excepción determinado, a partir del cual el cliente no obtiene información alguna que le permita saber qué hacer; salvo que se realice una programación específica con lo cual, nuevamente, se vuelve a aumentar el acoplamiento referido en 4.4.7.4.1.

Por otro lado, en relación a los códigos de excepción, sería interesante que se observara en toda su extensión la aplicación de lo dispuesto en [Sch05], en referencia a la adecuación de las excepciones a los códigos de estado que recoge el protocolo *HTTP* [Bau13], puesto que de esta forma se transmitiría una información más fiable y precisa sobre el estado del servidor.

4.4.7.4.4. Mensajes autodescriptivos

El estándar *WPS* muestra, en referencia a este apartado, el mismo comportamiento recogido en 4.4.2.4.4 para el estándar *WMS*.

4.4.7.5. Estructuración en capas

En línea con lo recogido en 4.4.2.5, desde la perspectiva del estándar *WPS*, no se realiza ninguna manipulación o transformación de los datos a los que se puede llegar acceder vía *HTTP* para obtener la información correspondiente a los parámetros de entrada de una operación *Execute*, de ahí que se considere que este estándar no aprovecha toda la potencialidad del protocolo.

4.4.7.6. Código a demanda

La especificación *WPS* no recoge consideración alguna con respecto a la posibilidad de descargar código en el cliente para la ejecución del mismo en un nivel diferente del servidor, lo cual parece contravenir la propia idiosincrasia de este tipo de servicios, en los que es la capacidad de cálculo del servidor la que es utilizada por los clientes, permitiendo minimizar los recursos necesarios en esta última vertiente. Sin embargo, disponer de habilidades para manipular los datos por parte del cliente –filtrado, agregación, fusión– puede resultar interesante en determinados escenarios.

4.4.7.7. Conclusiones

Recapitulando las consideraciones recogidas en el análisis del grado de cumplimiento de los principios de *REST* por parte de la actual especificación del protocolo *WPS*, se obtiene la Tabla 4.41 que permite colegir, sin género de dudas, que dicha interfaz no es compatible.

Principio	Cumplimiento	Aspectos clave
División cliente-servidor	✓	
Ausencia de estado	✓	
Existencia de caché	✗	Las características de las operaciones contempladas en este estándar no son susceptibles, a priori, de ser cacheadas.
Interfaz uniforme	✗	La especificación utiliza el protocolo <i>HTTP</i> como mero protocolo de transporte. Sin embargo, sería factible referenciar los diferentes recursos identificados, construyendo una interfaz uniforme.
Separación en capas	✓	
Código a demanda	✗	No se contempla esta posibilidad.

Tabla 4.41 Cumplimiento de los principios *REST* en el estándar *WPS*.

Las características funcionales de esta especificación son incompatibles con parte de los principios estudiados. Sin embargo, podrían plantearse un conjunto de cambios que facilitarían la adopción de una interfaz *REST*, atendiendo a las siguientes consideraciones:

- Recursos heterogéneos: el estudio de identificación de tipologías de recursos dentro de este estándar ofrece características peculiares. Conceptualmente hablando, un proceso es la instanciación física que da respuesta a una determinada función ofrecida por el sistema. Desde este punto de vista, por tanto, no es más que una herramienta que permite transformar una información de partida –parámetros– en unos resultados, siendo los propios parámetros y los resultados obtenidos los elementos homologables a recursos dentro de la terminología *REST*.

Para ilustrar esta afirmación se plantea el siguiente escenario. Supóngase que un usuario necesita conocer la geometría resultante de la intersección entre dos geometrías origen, e imagínese ahora que hay una organización que atiende este tipo de necesidades y ha desarrollado un algoritmo que permite realizar una intersección entre dos geometrías dadas –véase función “intersección” en la Tabla 4.15–. Desde un punto de vista tradicional, la organización encapsularía este algoritmo y permitiría, por ejemplo, su ejecución vía Web gracias a una interfaz *KVP* o *SOAP*. Estas interfaces serían invocables por el usuario, el cual pasaría los datos de entrada –la fórmula escogida variaría en función de la interfaz– y obtendría en la respuesta directamente la geometría resultante o un enlace a la misma –*URI* que la identifique unívocamente–. Por otro lado, al usuario podría interesarle conocer algunos detalles sobre la implementación de la función, cosa que conseguiría consultando el descriptor del proceso, en el cual se incluirían los metadatos del mismo.

Atendiendo a este planteamiento, dentro del estándar *WPS* se podrían diferenciar dos recursos que existen físicamente: el descriptor del servicio y el descriptor del proceso. A estas dos tipologías se añadiría una tercera, de carácter virtual, que es la del recurso resultado. Esta característica viene dada porque este recurso no existe físicamente hasta que se procede a su invocación –la cual conllevaría implícitamente la ejecución de un proceso– y estaríamos de nuevo ante la problemática ya tratada cuando se analizaba el estándar *WMS* y la relación existente entre la función ‘*getMap*’ y el recurso mapa. Tal y como ocurrió en ese caso, el *URI* del recurso resultado debe contemplar la utilización del componente *query string* para la especificación de los parámetros de entrada necesarios para la ejecución del proceso. Cabe reseñar que este hecho implicaría una capacitación del cliente para la construcción del *URI* que va más allá de las características de un cliente típico *REST*. Es decir, el cliente deberá descargar la información correspondiente al proceso –archivo descriptor del proceso–, procesar el apartado del descriptor en el que se especifiquen los parámetros de entrada y construir con esta información el *URI*.

Llegados a ese punto, en los recursos descriptores sería factible habilitar los métodos *GET*, *POST*, *PUT* y *DELETE* para gestionar la consulta, actualización, creación y eliminación, respectivamente, de los mismos. Por lo que respecta al recurso resultado, solo tiene sentido plantear las operaciones de consulta y borrado –*GET* y *DELETE*, respectivamente–.

La Tabla 4.42 muestra las correspondencias entre los métodos del protocolo *HTTP* y la interfaz *KVP* para la propuesta de interfaz *REST* de *WPS*.

Recurso	Método HTTP	Método WPS KVP
Descriptor del servicio	GET	GetCapabilities
	POST	-
	PUT	-
	DELETE	-
Descriptor del proceso	GET	DescribeProcess
	POST	-
	PUT	-
	DELETE	-
Resultado	GET	Execute
	POST	Execute
	PUT	-
	DELETE	-

Tabla 4.42 Equivalencias entre operaciones HTTP y KVP para WPS.

Por su parte, la Tabla 4.43 presenta la propuesta de *URI*⁴⁰ para cada una de los recursos. Se considera que un mismo servidor WPS puede publicar más de un servicio y cada servicio, a su vez, más de un resultado.

- Representaciones según estándares oficiales y *de facto*: con carácter general se ofrecerán en formato XML y JSON, aunque se permitirá la existencia de otros formatos para encapsular el contenido de los recursos resultantes de la ejecución de operaciones de análisis como las recogidas en 4.3.4, en cuyo caso se respetarán los formatos allí establecidos.
- Inclusión de contenidos hipermedia e implementación de mensajes autodescriptivos: facilitando el cumplimiento de estas características de REST. En el primero de los aspectos y a título ilustrativo, se podría incluir los URI correspondientes a posibles procesos que puedan encadenarse al actual, simulando de esta forma el concepto de estado y facilitando la navegación del cliente.
- Aprovechamiento de los mecanismos de cacheado: Aunque las técnicas de cacheado no son aplicables para la ejecución de procesos en la gran mayoría de los casos, puede plantearse la utilización de dichas técnicas contempladas dentro del protocolo HTTP en aquellas circunstancias que sean adecuadas, es decir, cuando se prevea la ejecución reiterada de un mismo proceso, con los mismos parámetros de entrada.

A modo de resumen, la plataforma que se está diseñando debe mantener la interfaz KVP y SOAP del estándar OGC WMS, e incorporar la propuesta REST.

⁴⁰ Las partes del URI que son variables se identifican entre los símbolos "<" y ">".

Servicio	
<a href="http://<server>:<port>/<service>/capabilities">http://<server>:<port>/<service>/capabilities	
GET	Permite obtener el fichero descriptor del servicio, conteniendo los metadatos relativos al mismo – autoría del servicio, condiciones de uso, etc.–.
POST	Permite actualizar el contenido del fichero descriptor. Operación disponible para perfiles de usuario con privilegios de administración.
PUT	Permite crear el fichero descriptor, especificando el contenido del mismo en la misma acción. Operación disponible para perfiles de usuario con privilegios de administración.
DELETE	Permite eliminar el fichero descriptor. Operación disponible para perfiles de usuario con privilegios de administración.
Descriptor del proceso	
<a href="http://<server>:<port>/<service>/processes/<processID>/description">http://<server>:<port>/<service>/processes/<processID>/description	
GET	Permite obtener el fichero descriptor del proceso correspondiente al identificador especificado. Si no se especifica el fragmento <i><processID></i> el servicio debe devolver un fichero con enlaces a la representación por defecto de cada una de los registros registrados.
POST	Permite actualizar el contenido del descriptor del proceso. Operación disponible para perfiles de usuario con privilegios de administración.
PUT	Permite crear el descriptor asociado a un proceso, especificando sus metadatos. Operación disponible para perfiles de usuario con privilegios de administración.
DELETE	Permite eliminar el descriptor del proceso. Operación disponible para perfiles de usuario con privilegios de administración.
Resultado	
<a href="http://<server>:<port>/<service>/results/<resultID>">http://<server>:<port>/<service>/results/<resultID>	
GET	Permite consultar el resultado o uno de los resultados obtenidos a partir de la ejecución de un proceso. Los datos de entrada que necesita el proceso para poder ser ejecutado podrían pasarse como parámetros del <i>URI</i> o bien mediante la simulación de una estructura de directorios: <a href="http://.../<resultID>?<parName1>=<valuePar1>&...&<parNameN>=<valueParN>">http://.../<resultID>?<parName1>=<valuePar1>&...&<parNameN>=<valueParN> <a href="http://.../<resultID>/<parName1>=<valuePar1>/.../<parNameN>=<valueParN>">http://.../<resultID>/<parName1>=<valuePar1>/.../<parNameN>=<valueParN> Si no se especifica el fragmento <i><resultID></i> el servicio debe devolver un fichero con enlaces a la representación por defecto de cada uno de los resultados obtenidos en la ejecución del proceso
DELETE	Permite eliminar el resultado con el identificador especificado en la petición.

Tabla 4.43 Propuesta de recursos y operaciones asociadas a la interfaz *REST* del estándar *WPS*.

4.4.8 Seguridad

Una vez acometido el estudio de las interfaces a utilizar en cada una de las tipologías de estándares que han de dar cobertura a la especificación funcional recogida en 4.3, se pasa a analizar dos aspectos arquitectónicos importantes en la definición de cualquier arquitectura. El primero de ellos es el de la seguridad.

En el mundo de las soluciones empresariales es imprescindible contar con mecanismos que permitan asegurar los canales de comunicación entre determinados servicios y sus clientes. Por su parte, en el ámbito geoespacial, hasta la aparición de las interfaces *SOAP*, las *KVP* sobre *HTTP* no contemplaban la posibilidad de arbitrar mecanismos de aseguramiento de las conexiones.

Dadas las circunstancias, se plantean distintas alternativas mediante la utilización de las capacidades de:

- *HTTP* a través de un conjunto de cabeceras *–headers–* que permiten evitar, entre otros, aspectos como:
 - Secuencias de comandos en sitios cruzados *–cross-site scripting–*: permite a una tercera parte la inyección, en una página web visitada por un usuario, de código malicioso *–en cualquier lenguaje de tipo script–*. Entre las consecuencias más comunes de este tipo de prácticas ilegales está la suplantación de identidad *–phishing–*. La cabecera que lucha contra esta técnica maliciosa se denomina *X-XSS-Protection*.
 - Secuestro de clic *–clickjacking–*: operación mediante la cual se muestra una interfaz de usuario falsa en la que son “secuestrados” los *clicks* que realiza el usuario en ciertos componentes de la interfaz *–botones, cajas de texto, etc.–* para la ejecución de acciones diferentes a las mostradas en la interfaz, mayoritariamente su redirección a páginas situadas en otra aplicación y/o dominio. La cabecera que lucha contra esta técnica maliciosa se denomina *X-Frame-Options*.
 - Olfateado *–sniff–*: técnica maliciosa que permite al atacante acceder al contenido que se transmite por la red entre un cliente y un servidor, de forma que puedan robarse credenciales u otros tipos de información confidencial. La cabecera que lucha contra esta técnica se denomina *X-Content-Type-Options*.

Es recomendable que se activen estas cabeceras en todas las interacciones que se realicen entre clientes y servidores.

Adicionalmente, existen otras opciones que arbitran mecanismos de seguridad más o menos robustos, como la generación de un símbolo o testigo *–token–* que es generado por el servidor y que vincula la utilización de un servicio con, por ejemplo, un determinado *URI* de origen de la petición. Este testigo debe incorporarlo la aplicación Web de origen en cada una de las peticiones *–en forma de parámetro–*. Esta es la opción utilizada por *ArcGIS for Server* para gestionar el acceso a geoservicios de uso restringido. El siguiente *URI* muestra un ejemplo de la implementación

realizada por esta tecnología en el escenario analizado –se ha resaltado en negrita la parte correspondiente al parámetro en cuestión–⁴¹.

[http://geoserveis.icc.cat/arcgis/rest/services/Utilities/Geometry/GeometryServer/areasAndLengths?f=json&polygons=\[{"rings":\[\[\[414376.17200000025,4592771.6384999999\],\[413336.32830000017,4592737.2634999999\],\[412932.42200000025,4592943.5134999999\],\[412666.01580000017,4593476.3259999999\],\[412726.17200000025,4593983.3572\],\[413087.10950000025,4594275.5447\],\[413559.76580000017,4594404.4509999999\],\[413645.70330000017,4594404.4509999999\],\[414702.73450000025,4593957.5759999999\],\[414702.73450000025,4593948.9822\],\[414376.17200000025,4592771.6384999999\]\]\],"spatialReference":{"wkid":25831}}\]&sr=25831&lengthUnit=9036&areaUnit={"areaUnit":"esriHectares"}&token=aR1sbWW2btcDrix5He3Jr5WUztFHR2C75C-ihovX8M9umT_vILxDFCdGo1o5WYlu](http://geoserveis.icc.cat/arcgis/rest/services/Utilities/Geometry/GeometryServer/areasAndLengths?f=json&polygons=[{)

- **HTTPS** que permite la encriptación en origen y la restauración en destino de todos los paquetes de información transmitidos, mediante la utilización de diferentes técnicas basadas en la generación de claves públicas y/o privadas, siguiendo los protocolos criptográficos *Secure Sockets Layer (SSL)* o *Transport Layer Security (TLS)* —sucesor del primero—. De esta forma, se considera que se establece un canal seguro entre ambos extremos, lo que se denomina “punto a punto”.

El principal problema de esta técnica reside en la necesidad de certificar la autenticidad de los servidores a los que se conecta el cliente –para evitar suplantaciones de identidad–, para lo cual existen entidades que emiten certificados de identidad, aunque el procedimiento administrativo necesario hace que muchas empresas opten por generarse sus propios certificados, con lo cuál el cliente debe fiarse de la bondad del certificado.

Otro aspecto que puede resultar problemático, viene relacionado con el carácter de seguridad “punto a punto” mencionado anteriormente. La posible necesidad de que el mensaje atraviese un servidor intermedio *proxy*, implica que éste acceda al contenido del mensaje para poder encaminarlo correctamente. En este escenario, ya no se puede garantizar la seguridad entre el extremo servidor y el consumidor, tan solo en el fragmento que va desde el servidor del servicio y el *proxy*. En tal caso, el servidor vería la petición que le llega del *proxy*, no la del cliente.

Asimismo, es factible habilitar ciertas capacidades de restricción de acceso en los componentes que gestionan el tráfico de una red, como por ejemplo en un cortafuegos –*firewall*–, en los cuales se pueden habilitar reglas que restrinjan las direcciones de los clientes –*IP address*– desde las que pueden provenir las peticiones que tienen acceso a un determinado servicio.

⁴¹ La petición permite al cliente obtener el área y el perímetro, en formato *JSON* –parámetro “*f*–, de los polígonos definidos mediante el parámetro “*polygons*” –siendo representadas las coordenadas de los vértices del mismo mediante el sistema *Universal Transversal de Mercator (UTM)*, huso 31N y dátum *ETRS89*, según se especifica en el parámetro “*spatialReference*–. El resultado se especificará en el mismo sistema de coordenadas en el que se definieron las geometrías –parámetro “*s*–, expresando el perímetro en kilómetros –parámetro “*lengthUnit*” igual a 9036– y el área en hectáreas –parámetro “*areaUnit*”–.

Sin embargo, dentro de la pila de protocolos *WS-I* en la que se sustenta *SOAP*, se encuentra el protocolo *WS-Security*. Este protocolo se centra en definir una plataforma que permita asegurar la integridad y la confidencialidad de un mensaje enviado a través de la red, desde su creación hasta que es consumido. En este sentido, va más allá de la protección que ofrece *HTTPS*, puesto que en este caso no solo se controla que el mensaje llegue al servidor o cliente correcto –hasta donde llegaría *HTTPS*–, sino que además se gestiona que solo pueda ser leído por el proceso al que va dirigido. Esto se consigue combinando este protocolo con otras extensiones de *WS* o protocolos de alto nivel que actúan en la capa de aplicación, en este caso incorporando características de seguridad en la cabecera de los mensajes *SOAP*; de esta forma, se permite implementar una amplia variedad de modelos y tecnologías relacionados con la seguridad en las comunicaciones Web.

Por tanto, en función de los requerimientos concretos planteados para cada uno de los geoservicios, se adoptarán las medidas de seguridad oportunas. Para conexiones que utilicen la interfaz *REST* o *KVP* la opción que ofrece mayor seguridad es la constituida por *HTTPS*. En cambio, para conexiones que hagan lo propio con *SOAP*, se obtiene, comparativamente, un mayor nivel de seguridad con *WS-Security*.

4.4.9 Composición

El último aspecto de la arquitectura que se va a analizar es el correspondiente a los mecanismos que favorezcan la composición u orquestación de los servicios. Como plataforma SIG distribuida en la que se pretende ofrecer servicios –grano fino– que creen valor añadido, la capacidad de combinación de los mismos permite crear nuevos servicios –grano grueso–, con los cuales dar respuesta a necesidades funcionalmente complejas.

Considerando las connotaciones semánticas recogidas en 3.3 y 4.3.1, así como la problemática de la composición esbozada en 3.4, la técnica más madura para la composición de servicios es la orquestación. Sin embargo, el lenguaje *WS-BPEL* está estrechamente ligado al estándar *WSDL* –cubriendo el ámbito de servicios *SOAP*–, aspecto que dificulta su aplicación a servicios de tipo *REST*. Dado que la arquitectura que se está definiendo precisa cubrir ambas interfaces –*SOAP* y *REST*–, se opta por la utilización de un motor compatible con *WS-BPEL* para dotar a la arquitectura distribuida de capacidades de composición de servicios *SOAP* y con la extensión *BPEL for REST* para realizar lo propio con servicios *REST* [Pau08b].

4.4.10 Estructura de capas

A lo largo del punto 4.4 se han analizado diferentes aspectos relacionados con la caracterización de la arquitectura distribuida óptima para un SIG. En 4.4.1 se estudiaron las ventajas e inconvenientes de los dos modelos de referencia que cuentan con mayor número de adeptos y de implementaciones, determinándose que era preciso analizar los diferentes estándares *OGC* que permiten dar cobertura a la especificación funcional recogida en 4.3. A tal fin se dedicaron los puntos 4.4.2, 4.4.3, 4.4.4, 4.4.5, 4.4.6 y 4.4.7, realizando un estudio del grado de implantación e idoneidad de las interfaces *SOAP* y *REST* en cada uno de los estándares referidos. La conclusión obtenida no fue otra que la necesidad de estructurar una plataforma SIG distribuida de carácter híbrido, mostrando una polivalencia que permitiera dar cobertura al amplio espectro de usuarios de IG y servicios geoespaciales, ya sean estos usuarios de aplicaciones empresariales más o menos complejas, o bien de aplicaciones tradicionales SIG.

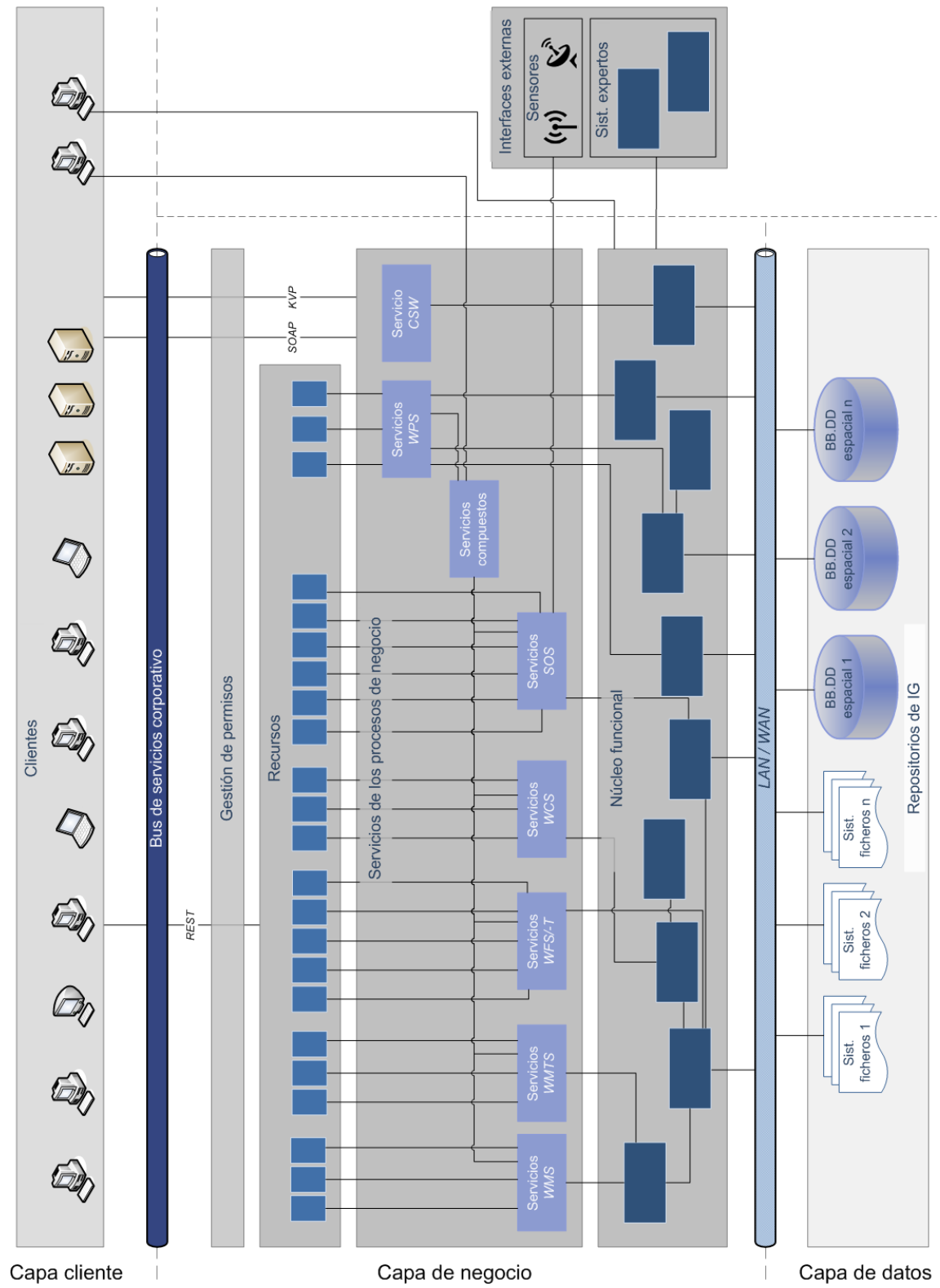


Figura 4.42 Modelo arquitectónico propuesto para SIG distribuidos [fuente: propia].

Consecuentemente, llegados a este extremo, en el presente punto se aborda la definición de las diferentes capas de la solución arquitectónica propuesta, presentadas en el esquema mostrado en la Figura 4.42. La granularidad expuesta es fruto del compromiso del autor por elaborar un modelo que mostrara un nivel de detalle suficiente para entender conceptualmente la propuesta, sin añadir elementos propios de un mayor refinamiento que no aportarían información relevante para tal fin. A título ilustrativo, la capa de datos podría dividirse en una que contiene propiamente los repositorios y otra que englobe los elementos que componen la interfaz de la misma con la capa de negocio.

4.4.10.1. Capa de datos

Esta capa se encuentra en el nivel inferior de la arquitectura y engloba todos aquellos repositorios de datos, accesibles mediante una red *LAN* o *WAN*. Estos repositorios pueden encontrarse repartidos en diferentes servidores físicos, alojando estos sus respectivos *DBMS* –instancias de las bases de datos y aplicaciones para la administración de las mismas– y sistemas de ficheros.

La existencia de múltiples repositorios de ambas tipologías parecería contravenir lo expuesto a lo largo del punto 2.1, en cuanto a la centralización de los diferentes repositorios departamentales. En un escenario ideal, todos estos repositorios han pasado a residir en uno o varios *DBMS* o sistemas de ficheros que pueden ubicarse físicamente en distintos servidores a efectos operativos, dando cobertura a todos los departamentos. Este es el escenario diseñado en esta propuesta.

4.4.10.2. Capa de negocio

Se trata de una capa intermedia que, a su vez, según el nivel de detalle –granularidad– del estudio, podría descomponerse en subcapas. En ella se hallan todos los componentes del SIG que implementan la lógica de negocio. En este sentido, haciendo referencia a la descripción de las arquitecturas *SOA* y *ROA* realizadas en 3.1.4 y 3.2.2, respectivamente, se diferencian las siguientes tipologías:

- Núcleo funcional: conjunto de elementos que procesan la lógica de negocio, para lo cual necesitan interactuar con:
 - Capa de datos: para abastecerse de la IG necesaria para los procesos en cuestión.
 - Sistemas expertos: catálogo de componentes que completan la plataforma de aplicaciones operativas dentro de la organización. En esta categoría se pueden encontrar sistemas de Planificación de Recursos Empresariales (*ERP*), Gestión de las Relaciones con los Clientes (*CRM*), minería de datos –*data mining*–, etc. La interacción con ellos permite ofrecer procesos que explotan la información empresarial desde un punto de vista geoespacial, por ejemplo para la realización de estudios de mercadotecnia geoespacial –*geomarketing*–.
 - Servicios de los procesos de negocio: la administración de estos servicios se realiza mediante componentes del núcleo funcional.

Estos componentes del núcleo funcional pueden interactuar entre ellos para implementar funciones de la lógica de negocio de mayor complejidad, en aras a conseguir un nivel óptimo de cohesión interna de los componentes y de acoplamiento entre ellos.

Asimismo, estos componentes precisan interfaces de administración para que los usuarios con los roles adecuados puedan gestionarlos. Este aspecto viene recogido en el diagrama mediante el enlace entre la caja que los agrupa y una de las estaciones de trabajo de la capa cliente. La interacción del cliente con un determinado componente se realiza utilizando conexiones externas al bus de servicios.

- Servicios de los procesos de negocio: engloba el conjunto de componentes que encapsulan en interfaces Web las diferentes funciones recogidas en la especificación funcional –presentada en el punto 4.3–. De conformidad con dicha especificación, se han representado los diferentes servicios OGC que permiten dar respuesta a la misma. Entre ellos encontramos unos servicios de carácter básico –WMS, WMTS, WFS-T, WCS, SOS y CSW– y otros de mayor complejidad. Entre estos últimos se encuentran los servicios creados mediante composición de los anteriores para lograr funciones más avanzadas. Estos servicios compuestos se harán públicos a través de los correspondientes servicios WPS.

Al igual que pasaba con los componentes nucleares, en este caso también se muestran interacciones. La primera atañe exclusivamente a los servicios SOS que permiten capturar información a partir de la red de sensores externos que se esté monitorizando.

La segunda afecta a los servicios de composición, disponiendo de una interfaz de usuario accesible desde una estación de trabajo cliente que permite definir los flujos de trabajo en forma de procesos que intervienen e interacciones entre los mismos –en diferentes niveles jerárquicos, puesto que pueden conjugarse servicios WPS con otros servicios básicos para crear nuevos servicios WPS–; el lenguaje propuesto para esta definición es WS-BPEL, atendiendo a lo recogido en 4.4.9. La tercera y última es la que muestra las interfaces Web entre todo el conjunto de servicios y la capa cliente, formada por todos aquellos consumidores de los servicios.

- Recursos: componentes que implementan desde un punto de vista lógico la esencia de la interfaz REST de la plataforma. Para cada uno de los servicios que dispone de la interfaz indicada, se ha definido en el diagrama un componente por cada una de las tipologías de recursos que tiene el servicio –cumpliendo con lo estudiado a lo largo del punto 4.4–. Indicar además, que los recursos son creados y gestionados por el servicio con el que están relacionados que, a su vez, es administrado por un determinado componente nuclear.
- Gestión de permisos: dando cumplimiento a lo indicado en 4.4.8, se habilita en la arquitectura un componente para la administración de los permisos necesarios para controlar el acceso de los clientes a los servicios/recursos ofrecidos.

Las interacciones que se producen entre la capa de negocio y la capa cliente se gestionan a través de la figura del bus de servicios corporativo. Este componente es clave –tal y como se analizó en 3.1.4– para garantizar aspectos como, por ejemplo, la calidad del servicio, atendiendo a los posibles acuerdos SLA firmados con los clientes.

4.4.10.3. Capa de cliente

Engloba al conjunto de consumidores de los *WS* ofrecidos por la plataforma, a través de cualquiera de las interfaces que ofrece la plataforma, en función de la tipología de servicio –*HTTP + KVP*, *HTTP + XML*, *SOAP* y *REST*–⁴². Estos consumidores o clientes también pueden dividirse en diferentes tipologías en función de la plataforma de computación utilizada –estaciones de trabajo, terminales, dispositivos móviles, etc.– y también en función del tipo de comunicación establecida, en referencia a si esta se produce entre una persona y el sistema, o entre diferentes sistemas –como ocurre por ejemplo en aquellos servicios que son utilizados por otras organizaciones. Este último caso se recoge en el diagrama mediante las figuras gráficas de los servidores.

Reseñar por último, que en el caso de la interacción entre las personas y el sistema, aunque excepcionalmente esta puede realizarse de forma “manual” –la construcción de los *URI* se acomete sin ayuda, lo cual en la práctica se circunscribe únicamente a fases muy tempranas de desarrollo en la que los programadores quieren chequear algún aspecto del servicio–, se realiza mediante interfaces de usuario de aplicaciones –ligeras o pesadas– que incorporan la lógica necesaria para gestionar la citada interacción con los servicios.

⁴² Con la finalidad de mejorar la legibilidad del diagrama correspondiente al modelo arquitectónico propuesto, se indica el término *KVP* en representación a las interfaces tradicionales de los estándares *OGC: HTTP + KVP* y *HTTP + XML*, utilizando los métodos *HTTP Get* y *Post*, respectivamente.

5 Líneas futuras de investigación

El estudio abordado a lo largo de los anteriores capítulos ha puesto de manifiesto certezas, pero también incertidumbres con respecto al futuro de las plataformas distribuidas. En cuanto a las certezas, es indudable concluir que los requerimientos actuales y futuros de computación, también en el ámbito de los SIG, seguirán precisando de plataformas distribuidas que permitan afrontar los retos funcionales y logísticos que se derivan de las crecientes necesidades por parte del ámbito empresarial, con modelos de negocio cada vez más complejos.

En líneas generales, se plantean las siguientes líneas de investigación:

- Modelos arquitectónicos: en lo que a las incertidumbres respecta, la principal es sin duda la evolución que sufrirán los actuales esquemas arquitectónicos de referencia: *SOA* y *ROA*. El camino de *SOA* ha sido más largo y presenta un número de investigaciones que han concluido en numerosos estándares que permiten afirmar que se trata de una plataforma estable y correctamente dimensionada para los retos referidos. Si acaso, cabrá observar la evolución que pueden sufrir dichos estándares y otros nuevos que, sin duda, surgirán fruto de iniciativas institucionales o del ámbito empresarial. El mercado irá dictaminando cuales de estas iniciativas acabarán triunfando y habrá que adaptar la plataforma SIG distribuida a esta evolución.

ROA, por su parte, tiene en su juventud su principal virtud y a la vez su principal hándicap. Aunque los fundamentos se encuentran en la propia naturaleza de Internet y del protocolo *HTTP*, es cierto que hasta hace poco tiempo eran pocas las plataformas de computación que optaban por adoptar esta arquitectura. De hecho, existen muchas que implementan los principios de *REST* de forma incompleta o haciendo unas interpretaciones manifiestamente subjetivas. Esta bisoñez deberá ir dando paso a un periodo de madurez, a medida que las iniciativas de estandarización existentes y futuras vayan consolidándose, permitiendo definir mecanismos que palien algunos de los puntos débiles que se achacan a *ROA* desde el punto de vista de los abogados de *SOA*. Dicho esto, no hay que perder de vista que hablamos de principios conceptuales muy diferentes y que, por tanto, siempre encontraremos soluciones en las que una propuesta híbrida sea la más conveniente, tal y como se ha concluido en este estudio.

- Estándares del ámbito geoespacial: la principal incógnita es la evolución que sufrirán los estándares regulados principalmente por el *OGC*, aunque también aquellos promulgados por *ISO* y por algunas organizaciones de carácter más regional como *INSPIRE*. El estudio realizado de dichos estándares manifiesta importantes deficiencias en cuanto a la adopción de interfaces *REST* y es de esperar que en los próximos años se vayan solucionando.

Sin embargo, hay que ser consciente que en el ámbito de la normalización, la evolución de los estándares siempre va por detrás de las necesidades del mercado y en ocasiones esta demora puede conllevar una pérdida de competitividad que puede resultar crucial en el futuro de una organización. Sirva como ejemplo el estándar *WMTS*, aprobado en 2010 por el *OGC* y cuyo fundamento es en último extremo el mismo que el defendido en 2006 por la propuesta de perfil

WMS denominada WMS-C. Ésta, a su vez, intentaba remediar las grandes limitaciones que en términos de rendimiento ofrecía WMS para ciertas aplicaciones y perfiles de usuario, que ya habían encontrado en 2005 a *Google Maps* una plataforma que permitía extender el uso de la cartografía en línea como nunca antes se había conseguido.

- **Semántica:** es largo el camino que todavía queda para consensuar mecanismos que faciliten la vertebración de la Web semántica. Los pasos que se han dado han sido importantes, pero ha quedado claro a lo largo de este estudio que en el ámbito geoespacial los estándares no cubren en absoluto esta problemática.
- **Composición:** la mayor complejidad de los procesos de negocio a la que se ha hecho referencia al inicio de este capítulo pone de manifiesto la necesidad de disponer de componentes que permitan administrar la composición de los servicios de grano fino, para conseguir servicios de mayor complejidad que encuentren en la suma de los mismos la respuesta a dicha complejidad. Pero dicha administración debe poder realizarse de la forma más automática posible y en este punto la semántica de las descripciones de los servicios juega un papel predominante, como ponen de manifiesto las numerosas investigaciones que se están llevando a cabo.

Queda por delante un panorama esperanzador, son numerosos los retos existentes y los que sin duda aparecerán dada la velocidad a la que evolucionan las tecnologías de la información, pero el papel que debe ocupar en este escenario los SIG es primordial. Por ello, es necesario analizar las actuales carencias e intentar encontrar soluciones a las mismas, innovando y observando con atención las iniciativas que puedan surgir para darles solución.

6 Conclusiones

Con este capítulo se da por finalizado el estudio presentado en este TFM, analizando la consecución de los retos planteados en un inicio. Esta labor se realizará en sentido inverso a como fueron planteados dichos retos, comenzando por el grado de consecución de los objetivos secundarios y acabando con el corolario que supone el objetivo principal.

El primer objetivo secundario planteado fue el de la determinación del abanico funcional que debía cubrir la arquitectura distribuida. Este aspecto se ha abordado en dos fases:

- Estudio del marco histórico en el que se caracteriza la evolución de los SIG y las necesidades presentes y futuras –con el riesgo que supone predecir qué ocurrirá en el futuro, especialmente en áreas tecnológicas de tan rápida evolución– a las que han tenido, tienen y tendrán que dar respuesta estos sistemas. Estas necesidades solo pueden ser resueltas mediante la concepción de sistemas distribuidos.
- Caracterización de un caso de estudio: la determinación de la panoplia funcional a cubrir precisaba abordar el estudio con algún caso práctico que permitiera extraer y extrapolar los requerimientos funcionales. En este sentido, se ha realizado un análisis exhaustivo de la oferta y demanda que registra una organización productora de cartografía y servicios asociados, con competencias oficiales en el ámbito territorial de la región catalana y, por tanto, con unas obligaciones claras de servicio a los ciudadanos en general y al tejido industrial de la IG en particular. Las conclusiones funcionales obtenidas son perfectamente extrapolables al resto de organismos regionales, nacionales e internacionales, así como a las empresas que necesiten poner a disposición de sus clientes –ya sean éstos de la propia organización o externos– contenidos cartográficos y servicios que permitan desarrollar, en el marco de las aplicaciones al uso, la lógica de negocio empresarial.

En segundo lugar era preciso conocer en detalle las arquitecturas de carácter distribuido que gozan de mayor implantación en la actualidad, poniendo de manifiesto sus virtudes y también sus debilidades. En este sentido, se han analizado *SOA* y *ROA* y se ha llegado a la conclusión de que la arquitectura a diseñar debería englobar aspectos de ambas. No se puede afirmar que una de ellas sea mejor que la otra, son fruto de planteamientos conceptualmente muy diferentes y, a juicio del autor, son claramente complementarias.

SOA está más focalizada en dar respuesta a lógicas de negocio complejas, en las que sea preciso convivir con sistemas heredados que cubren las necesidades de diferentes áreas de las organizaciones y en las que los requerimientos de seguridad y calidad del servicio sean importantes.

En cambio, *ROA* es una propuesta basada en unos principios que persiguen lograr una interfaz de publicación muy sencilla –denominada *REST*– aprovechando las características del protocolo *HTTP* al máximo. Para ello, habilita el concepto de recurso, como elemento encapsulado en cada uno de los servicios ofrecidos; permitiéndose la interacción con el mismo mediante las operaciones básicas de *HTTP* que permiten su creación, eliminación, actualización y consulta. Su simplicidad –aunque en algunos casos sea simplemente aparente a juicio de algunos estudiosos– la hace especialmente

atractiva, puesto que las labores de descubrimiento de los recursos existentes es sencilla, siempre y cuando se respeten todos los principios de diseño en los que se sustenta *ROA*. Sin embargo, adolece de ciertas limitaciones que son propias del protocolo *HTTP*, como por ejemplo en el ámbito de la seguridad.

Dadas las circunstancias, y tal y como se ha comentado, se llega a la conclusión de que lo que resulta más interesante es la fusión de ambas arquitecturas, creando una estructura híbrida que aúna las virtudes de ambas y da completa cobertura a los requerimientos funcionales.

Sin embargo, quedaba otro aspecto muy importante a resolver. En el ámbito geoespacial ha sido muy costoso llegar a disponer de estándares que permitieran lograr la ansiada interoperabilidad entre plataformas SIG servidoras de IG y servicios. La creación del *OGC* supuso un paso muy importante en ese camino, fruto del cual son numerosos los estándares existentes en la actualidad. Algunos no han tenido mucho éxito, francamente, pero otros sí que han sido mayoritariamente aceptados por la comunidad geoespacial. Por otro lado, iniciativas surgidas del ámbito institucional, como la normativa *INSPIRE* que emana de la UE y que es de obligado cumplimiento en todos los estados miembro, han supuesto un nuevo aldabonazo en el ámbito regulatorio.

El diseño de la nueva plataforma SIG debía enfrentarse a las características de los estándares adoptados de forma prácticamente universal, como *WMS*, *WMTS*, *WFS*, *WCS*, *SOS* y *WPS*. Las especificaciones de cada uno de ellos dan cobertura a unas determinadas interfaces. En algunos casos este hecho se fundamenta en la antigüedad de la especificación, la cual ha recogido las tendencias del momento. En este sentido, todos disponen de una interfaz *KVP* y de otra *SOAP* –añadida más recientemente–, con lo cual quedarían cubiertos los requerimientos de la nueva arquitectura que provienen de *SOA*. Sin embargo, solo uno de ellos dispone de una interfaz *REST* aprobada. Esta teórica deficiencia cabe analizarla en detalle para determinar si es fruto de una “falta de dedicación” o si realmente es debida a una incoherencia conceptual. El estudio realizado ha puesto de manifiesto que en algunos casos es cuestionable plantear una interfaz *REST* –por ejemplo para los estándares *WMS* o *WPS*–, puesto que su concepción choca con algunos principios arquitectónicos en los que se basa esta interfaz, aunque en otros su existencia parece inherente a las características de la información gestionada. En cualquier caso, el estudio muestra una propuesta de interfaz en la que se indican los recursos que deberían ofrecerse en cada uno de los estándares, puesto que su existencia ampliará la interoperabilidad de la plataforma.

Por tanto, se puede concluir a juicio del autor que se han conseguido todos los objetivos parciales que conducen a alcanzar el objetivo principal de creación de una plataforma adaptada al ámbito de los SIG distribuidos. Simplemente cabe, tal y como se recoge en el capítulo anterior, investigar y prestar atención a las investigaciones de terceros, con respecto a la evolución de determinadas variables que conducirán los modelos arquitectónicos en un futuro.

7 Agradecimientos

Ha sido largo el camino recorrido desde que empecé a dibujar el contexto de la que sería mi propuesta de TFM. En el mismo ha habido momentos de profunda reflexión sobre el enfoque a realizar para abordar la investigación de los diferentes aspectos que subyacen en el estudio y otros en los que la línea de trabajo estaba marcada y las pautas perfectamente definidas. Sin embargo, sea cual fuera el momento, para poder llegar a este punto de conclusión de mi trabajo ha resultado indispensable contar con la ayuda de algunas personas, a las que quiero hacer mención de forma explícita.

En primer lugar, como no podría ser de otra forma, a mi familia que me ha servido de sustento. En el caso de Maribel –mi mujer– y de Darío y Teresa –mis padres– por el respaldo que me han dado, supliéndome en las innumerables horas que he tenido que dedicarme a este proyecto, haciéndose cargo de aspectos logísticos cotidianos, especialmente de todos aquellos que conlleva ser padre de dos maravillosas criaturas. En segundo lugar a mis hijos Blanca y Iago, en los cuales encuentro siempre alicientes para seguir progresando personal y profesionalmente, con el ánimo de poder ofrecerles cada día mayor calidad de vida, en todos los sentidos. Ellos son el motor de mi existencia.

En tercer lugar, a la persona que ha sido mi guía desde un punto de vista intelectual, mostrándome de forma eficaz el camino a seguir y haciéndome ver aquellos aspectos en los que debía mejorar mi trabajo. Por su paciencia, por su sabiduría y por su ejemplo, mi sincero agradecimiento a la Dra. Elena Ruíz Larrocha.

Y, por último, a todos los autores que aparecen reflejados en el apartado bibliográfico, porque su trabajo ha sido fundamental para poder conducir a buen puerto el mío.

A todos, muchas gracias.

8 Bibliografía

Identificador	Referencia bibliográfica
[ACM02]	ACM. <i>ACM Transactions on Internet Technology. 2º volumen.</i> Nueva York (Estados Unidos de América) : Association for Computing Machinery, 2002.
[Ami10]	Amini, A. y Sadjedy, S. <i>SOMGIS - A Service Oriented Model for GIS Software Architecture. Proceedings of the 2nd International Conference on Computer Engineering and Technology .</i> Chengdu (China) : IEEE, 2010.
[Bac97]	Bacon, F. <i>Essayes: Religious Meditations. Places of Perswasion and Disswasion. Seene and Allowed.</i> Londres (Reino Unido) : Humfrey Hooper, 1597.
[Bar10]	Baryannis, G. y Plexousakis, D. <i>Automated Web Service Composition: State of the Art and Research Challenges.</i> Iraklio : Foundation for Research & Technology. Institute of Computer Science, 2010. Technical Report ICS-FORTH/TR-409.
[Bau10]	Baumann, P. (ed.) . <i>OGC GML Application Schema-Coverages.</i> Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2010. OGC 09-146r1.
[Bau12]	Baumann, P. (ed.) . <i>OGC WCS 2.0 Interface Standard - Core: Corrigendum.</i> Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2012. OGC 09-110r4.
[Bau13]	Baumann, P. y Dumitru, M.A. (ed.) . <i>OGC Web Coverage Service Interface Standard - REST Protocol Extension.</i> Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2013. OGC 12-174.
[Ben05]	Benatallah, B., Casati, F. y Traverso, P. <i>Service-Oriented Computing: A research roadmap. Proceedings of the 3rd International Conference on Service Oriented Computing.</i> Amsterdam (Holanda) : Springer, 2005.
[Bia11]	Bianco, P., Lewis, G., Merson, P. y Simanta, S. <i>Architecting Service-Oriented Systems.</i> Pittsburgh, Pensilvania (Estados Unidos de América) : Carnegie Mellon University, 2011.
[Bic65]	Bickmore, D.P. y Shaw, M.A. <i>The Atlas of Great Britain and Northern Ireland.</i> Nueva York (Estados Unidos de América) : The Academy of Political Science, 1965.
[Bot07]	Botts, M., y otros. <i>OGC Sensor Web Enablement: Overview and High Level Architecture (OGC White Paper).</i> Wayland, Massachusetts (Estados Unidos América) : Open Geospatial Consortium, 2007. OGC 07-165.
[Bot07b]	Botts, M. (ed.) . <i>OpenGIS Sensor Model Language (SensorML) Implementation Specification.</i> Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2007. OGC 07-000.
[Bro12]	Bröring, A., Echterhoff, J. y Stasch, C. (ed.) . <i>OGC Sensor Observation Service Interface Standard.</i> Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2012. OGC 12-006.
[Bro99]	Brown, J.S., Gold, R., Weiser, M. The origins of ubiquitous computing research at PARC in the late 1980s. <i>IBM Systems Journal.</i> 1999, Vol. 38, Edición 4.
[Bur11]	Burghardt, D., Hahmann, S. y Weber, B. "80% of All Information is Geospatially Referenced"??? Towards a Research Framework: Using the Semantic Web for (In)Validating this Famous Geo Assertion. Utrecht (Holanda) : AGILE, 2011

Identificador	Referencia bibliográfica
[But08]	Butler, H., y otros. The GeoJSON Format Specification. <i>GeoJSON</i> . [En línea] GeoJSON, 16 de Junio de 2008. http://geojson.org/geojson-spec.html .
[Cap09]	Capretz, M., Felgar de Toledo, M.B., Higthino, W.A. <i>REST and Resource-Oriented Architecture. Proceedings 1st International Symposium on Services Science ISSS'09</i> . Berlín (Alemania) : Alt, R.; Fähnrich, K.P.; Franczyk, B. Editores, 2009.
[Cea14]	Cearley, D.W. <i>The Top 10 Strategic Technology Trends fro 2014</i> . Stamford, Connecticut (Estados Unidos de América) : Gartner Inc., 2014.
[Cle93]	Clementini, E., Di Felice, P. y Van Oosterom, P. <i>A small set of formal topological relationships suitable for end-user interaction</i> . Berlin : Springer, 1993.
[Coc08]	Cocero, D. y Santos, J.M. <i>Los SIG raster: herramienta de análisis medioambiental y territorial</i> . Madrid : Editorial UNED, 2008.
[Coo08]	Cook, J. y Singh, R. (ed.) . <i>Summary of the OGC Web Services, Phase 5 (OWS-5) Interoperability Testbed</i> . Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2008. OGC 08-073r2.
[Coo12]	Coors, V., Hagedom, B. y Schilling, A. (ed.) . <i>OGC 3D Portrayal Interoperability Experiment. Final Report</i> . Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2012. OGC 12-075.
[Cop91]	Coppock, J.T. y Rhind, D.W. <i>Geographical Information Systems</i> . Hoboken, New Jersey (Estados Unidos de América) : John Willey & Sons, 1991.
[Cox11]	Cox, S. (ed.) . <i>Observations and Measurements - XML Implementation</i> . Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2011. OGC 10-025r1.
[Dal01]	Daly, M. <i>OpenGIS Implementation Specification: Coordinate Transformation Services</i> . Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2001. OGC 01-009.
[Dal01b]	Daly, M. (ed.) . <i>OpenGIS Implementation Specification: Coordinate Transformation Services</i> . Mayland, Massachusetts (Estados Unidos) : Open Geospatial Consortium, 2001. OGC 01-009.
[Del06]	De la Beaujardiere, J. (ed.) . <i>OpenGIS Web Map Service (WMS) Implementation Specification</i> . Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2006. OGC 06-042.
[Dia10]	Díaz, L., Gould, M. y Granell, C. Service-oriented applications for environmental models: reusable geospatial services. <i>Environmental Modelling & Software</i> . 2010, Vol. 25, 2.
[Dia11]	Díaz, L., y otros. <i>Managing user generated information in geospatial cyberinfrastructures. Edición 3</i> , Amsterdam : Future Generation Computer Systems. Elsevier, 2011, Vol. 27.
[DRQ14]	D2RQ. <i>D2RQ</i> . [En línea] 2014. http://d2rq.org/ .
[Dus05]	Duschene, P. y Sonnet, J. (ed.) . <i>OGC WMS Change Request: Support for WSDL & SOAP</i> . Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2005. OGC 04-050r1.
[Ech11]	Echterhoff, J. y Simonis, I. (ed.) . <i>OGC Sensor Planning Service Implementation Standard</i> . Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2011. OGC 09-000.
[Ech11b]	Echterhoff, J. (ed.) . <i>OpenGIS SWE Service Model Implementation Standard</i> . Waylanda, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2011. OGC 09-001.

Identificador	Referencia bibliográfica
[Ege89]	Egenhofer, M.J. y Herring, J.R. <i>A mathematical framework for the definition of topological relationships. Fourth International Symposium on Spatial Data Handling. Edición 2</i> , Dordrecht (Países Bajos) : Kluwer Academic Publishers, 1989, Vol. 19.
[Erl08]	Erl, T. <i>SOA: Principles of service design</i> . New York (Estados Unidos de América) : Prentice Hall, 2008.
[ESR10]	ESRI Inc. <i>GeoServices REST Specification. Version 1.0</i> . Redlands, California (Estados Unidos de América) : ESRI Inc., 2010.
[ESR14]	ESRI Inc. ArcGIS REST API. <i>ArcGIS Resources</i> . [En línea] 2014. http://resources.arcgis.com/en/help/arcgis-rest-api/index.htm .
[Eur14]	European Commission. Data Specifications. <i>Infraestructure for Spatial Information in the European Community</i> . [En línea] European Commission, 2014. http://inspire.ec.europa.eu/index.cfm/pageid/2 .
[Eva03]	Evans, J.D. (ed.) . <i>Web Coverage Service (WCS), Version 1.0.0</i> . Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2003. OGC 03-065r6.
[FGD14]	Federal Geographic Data Committee. Standards publications. <i>FGDC</i> . [En línea] Federal Geographic Data Committee, 2014. http://www.fgdc.gov/standards/standards_publications/index_html .
[Fie00]	Fielding, R. <i>Architectural Styles and the Design of Network-based Software Architectures. PhD Tesis</i> . Irvine (Estados Unidos de América) : University of California, 2000.
[Fie99]	Fielding, R., y otros. Status Code Definitions. <i>W3C</i> . [En línea] Junio de 1999. http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html .
[Flo09]	Florczyk, A, y otros. <i>INSPIRE-able Services</i> . Hannover : Association of Geographic Information Laboratories for Europe, 2009.
[Fra94]	Frank, A.U. y Timpf, S. <i>Multiple representations for cartographical objects in a multi-scale tree - an intelligent graphical zoom. Edición 6</i> , Amsterdam : Computers & Graphics. Elsevier, 1994, Vol. Vol. 18.
[Fre12]	Fredrich, T. RESTful Service Best Practices. Recommendations for Creating Web Services. <i>REST API Tutorial</i> . [En línea] 29 de Mayo de 2012. http://www.restapitutorial.com/media/RESTful_Best_Practices-v1_1.pdf .
[Gar08]	Garbajosa, J., Soriano, F.J. y Moreno, J.J. <i>Informe de Vigilancia Tecnológica de madrid+d en Tecnologías. Tecnologías Software Orientadas a Servicios</i> . Madrid (España) : Fundación madrid+d para el Conocimiento, 2008.
[Gen06]	Genong, Y. y otros. Semantic Augmentations for Geospatial Catalogue Service. <i>IGARSS 2006 : IEEE International Geoscience and Remote Sensing Symposium</i> . New York (Estados Unidos de América) : Institute of Electrical and Electronics Engineers, 2006.
[Gha10]	Ghaemi, A, y otros. <i>Implementing a Municipal SDI with Service Oriented Architecture</i> . Singapore : GSDI, 2010.
[Gom10]	Gomadam, K., Ranabahu, A. y Sheth, A. SA-REST: Semantic Annotation of Web Resources. <i>W3C</i> . [En línea] World Wide Web Consortium, 5 de Abril de 2010. http://www.w3.org/Submission/SA-REST/#sec_2_3 .
[Goo97]	Goodchild, M.F. NCGIA Core Curriculum 1990 Version. <i>UBC</i> . [En línea] 30 de Agosto de 1997. http://www.geog.ubc.ca/courses/klink/gis.notes/ncgia/toc.html .

Identificador	Referencia bibliográfica
[Gra12]	Granel, C., y otros. <i>Assessment of OGC Web Processing Services for REST principles</i> . s.l. : International Journal of Data Mining, Modelling, 2012.
[Gra12b]	Granel, C., Huerta, J. y Tamayo, A. <i>Measuring complexity in OGC Web Services XML Schemas: Pragmatic use and solutions</i> . Oxford : International Journal of Geographical Information Science. Taylor & Francis, 2012.
[Gre10]	Greenwood, J. y Whiteside, A. (ed.) . <i>OGC Web Services Common Standard</i> . Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2010. OGC 06-121r9.
[Gro12]	Gröger, G., y otros. <i>OGC City Geography Markup Language (CityGML) En-coding Standard</i> . Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2012. OGC 12-019.
[Gum08]	Gumusay, M.U. y Sahin, K. <i>Service Oriented Architecture (SOA) based Web Services for Geographic Information Systems. Proceedings of the 21st congress of the International Society for Photogrammetry and Remote Sensing</i> . Beijing (China) : ISPRS, 2008.
[Guo10]	Guo, L. y otros. <i>Study on GIS architecture based on SOA and RIA. Proceedings of the 3rd International Conference on Information Sciences and Interaction Sciences</i> . Chengdu (China) : IEEE, 2010.
[Han92]	Hane, C., Hane, F. y Hane, P. <i>An introduction to GIS: linking maps to databases</i> . s.l. : Database, 1992, Vol. 2
[Her10]	Herring, J.R. (ed.) . <i>OpenGIS Implementation Standard for Geographic Information - Simple Feature Access - Part 2: SQL option</i> . Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2010. OGC 06-104r4.
[Her11]	Herring, J.R. (ed.) . <i>OpenGIS Implementation Standard for Geographic Information - Simple feature access - Part 1: Common architecture</i> . Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2011. OGC 06-103r4.
[Her12]	Herring, J. y Perry, M. (ed.) . <i>OGC GeoSPARQL - A Geographic Query Language for RDF Data</i> . Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2012. OGC 11-052r4.
[IBM13]	IBM. The Four V's of Big Data. <i>IBM Big Data Hub</i> . [En línea] 2013. http://www.ibmbigdatahub.com/infographic/four-vs-big-data .
[ICG14]	Instituto Cartográfico y Geológico de Cataluña. Servicios en línea (geoservicios). <i>ICGC</i> . [En línea] 2014. Instituto Cartográfico y Geológico de Cataluña.
[ICG14b]	Instituto Cartográfico y Geológico de Cataluña. Diferencias entre base topográfica y mapa topográfico. <i>ICGC</i> . [En línea] 2014. http://www.icc.cat/esl/Home-ICC/Mapas-escolares-y-divulgacion/Preguntas-y-respuestas/Diferencias-entre-base-topografica-y-mapa-topografico .
[IET14]	Instituto de Estudos do Território. Infraestructura de datos espaciais de Galicia. <i>IET</i> . [En línea] Xunta de Galicia, 2014. http://sitga.xunta.es/sitganet/catalogo/inicio.html?lang=gl .
[IGN14]	Instituto Geográfico Nacional. Infraestructura de Datos Espaciales. <i>IGN</i> . [En línea] 2014. http://www.ign.es/ign/layoutIn/actividadesIDEServicios.do .
[INS08]	INSPIRE Thematic Working Group Coordinate Reference Systems and Geographical Grid Systems. <i>D2.8.1.1 Data Specification On Coordinate Reference Systems</i> . Ispra (Italia) : Joint Research Center (Comision Europea), 2008.

Identificador	Referencia bibliográfica
[IOC11]	Initial Operating Capability Task Force for Network Services. <i>Technical Guidance for the implementation of INSPIRE View Services.</i> Ispra (Italia) : Joint Research Center (Comisión Europea), 2011.
[IOC11b]	Initial Operating Capability Task Force Network Services. Technical Guidance for the implementation of INSPIRE View Services. <i>INSPIRE.</i> [En línea] 2011. http://inspire.ec.europa.eu/documents/Network_Services/TechnicalGuidance_ViewServices_v3.1.pdf .
[IOC13]	Initial Operating Capability Task Force for Network Services. <i>Technical Guidance for the implementation of INSPIRE Download Services.</i> Ispra (Italia) : Joint Research Center (Comisión Europea), 2013.
[ISO11]	ISO/IEC. <i>Information technology - Database languages -SQL Multimedia and Application Packages - Part 3: Spatial.</i> Ginebra (Suiza) : ISO/IEC, 2011. ISO/IEC 13249-3:2006.
[Jul10]	Julià, N., Masó, J. y Pomakis, K. <i>OpenGIS Web Map Tile Service Implementation Standard.</i> Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2010. OGC 07-057r7.
[Khr14]	Khronos Group. WebGL: OpenGL ES 2.0 for the Web. <i>Khronos Group.</i> [En línea] 2014. https://www.khronos.org/webgl/ .
[Lle08]	Lleopart, A. Critical analysis of the ICC BT-5M dataset matching process for Road Network. <i>CONFERENCE CALL: November 20, 2008 - TWG-DM Transport Network.</i> [En línea] 2008. http://www.gis4eurepository.corila.it/Reserved/03_Meetings/AR_ConferenceCallList.asp?Id=94 .
[Luc08]	Lucchi, R., Millot, M. <i>Resource Oriented Architecture and REST. Assesment of impact and advantages on INSPIRE.</i> Ispra (Italia) : Joint Research Center (Comisión Europea), 2008.
[Man08]	Mandel, L. Describe REST Web services with WSDL 2.0. <i>Developer Works.</i> [En línea] IBM, 29 de Mayo de 2008. http://www.ibm.com/developerworks/library/ws-restwsdl/ .
[Mar09]	Martell, R. (ed.). <i>OGC CSW-ebRIM Registry Service - Part 1: ebRIM profile of CSW.</i> Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2009. 07-110r4.
[Mau09]	Maué, P. (ed.). <i>Semantic annotations in OGC standards.</i> Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2009. OGC 08-167r1.
[Min14]	Ministerio de Fomento. Preguntas frecuentes. <i>IDEE.</i> [En línea] 2014. http://www.idee.es/faq .
[Neb07]	Nebert, D., Vretanos, P. y Whiteside, A. (ed.). <i>OpenGIS Catalogue Services Specification.</i> Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2007. OGC 07-006r1.
[NSD08]	Network Services Drafting Team. <i>INSPIRE Network Services Architecture.</i> Ispra (Italia) : Joint Research Center (Comisión Europea), 2008.
[NSD10]	Network Services Drafting Team. <i>Draft Technical Guidance for INSPIRE Coordinate Transformation Services.</i> Ispra (Italia) : Joint Research Center (Comisión Europea), 2010.
[Nun12]	Nunes, A. <i>Diccionari terminològic de sistemes d'informació geogràfica.</i> Barcelona : Instituto Cartográfico de Cataluña, 2012.
[OAS04]	OASIS. UDDI XML Community. Universal Description, Discovery and Integration v3.0.2 (UDDI). OASIS. [En línea] 19 de octubre de 2004. https://www.oasis-open.org/committees/uddi-spec/doc/spec/v3/uddi-v3.0.2-20041019.htm#_Toc85907969 .

Identificador	Referencia bibliográfica
[OGC06]	OGC. <i>The OpenGIS Abstract Specification. Topic 6: Schema for coverage geometry and functions.</i> Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2006. OGC 07-011 / ISO 19123.
[OGC11]	Open Geospatial Consortium. 3D Portrayal IE. <i>Open Geospatial.</i> [En línea] Open Geospatial Consortium, 2011. http://www.opengeospatial.org/projects/initiatives/3dpi .
[OGC11b]	Open Geospatial Consortium. The OGC Forms REST and WFS/FE Standards Working Groups. OGC. [En línea] 2011. http://www.opengeospatial.org/node/1497 .
[OGC11c]	Open Geospatial Consortium. How to model your observation data in SOS 2.0. <i>OGC Network.</i> [En línea] 2011. http://www.ogcnetwork.net/sos_2_0/tutorial/om .
[OGC14]	Open Geospatial Consortium. OGC Mission, Vision & Goals. OGC. [En línea] 2014. http://www.opengeospatial.org/ogc/vision .
[OGC14b]	Open Geospatial Consortium. Web Coverage Service. OGC. [En línea] 2014. http://www.opengeospatial.org/standards/wcs .
[Ore11]	O'Reilly, T. (ed.). <i>OGC PUCK Protocol Standard Version 1.4.</i> Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2011. OGC 09-127r2.
[OSG12]	Open Source Geospatial Foundation. Tile Map Service Specification. <i>OSGeo Wiki.</i> [En línea] 16 de Mayo de 2012. http://wiki.osgeo.org/wiki/Tile_Map_Service_Specification .
[OSM14]	Fundación OpenStreetMap. OpenStreetMap. [En línea] 2014. http://www.openstreetmap.org .
[OSM14b]	Humanitarian OpenStreetMap Team. Haiti. [En línea] Fundación OpenStreetMap Team, 2014. http://hot.openstreetmap.org/projects/haiti-2 .
[Pan11]	Panareda, J.M., Rabella, J.M. y Ramzzini, G. <i>Diccionari terminològic de cartografia.</i> Barcelona : Instituto Cartográfico de Cataluña, 2011.
[Pap08]	Papazoglou, M.P. <i>Web services: Principles and technologies.</i> Essex : Pearson Education, 2008.
[Par00]	Parent, Ch., Spaccapietra, S. y Vangenot, Ch. <i>GIS Databases: From Multiscale to MultiRepresentation. Proceedings of the 4th International Symposium, SARA 2000.</i> Horseshoe Bay, Texas (Estados Unidos de América) : Springer, 2000.
[Pau08]	Pautasso, C., Zimmermann, O. y Leymann, F. <i>RESTful Web Services vs. "Big" Web Services: Making the Right Architectural Decision.</i> Beijing : ACM. WWW Conference, 2008.
[Pau08b]	Pautasso, C. <i>BPEL for REST.</i> Milán (Italia) : Springer, 2008.
[Per02]	Percivall, G. (ed). <i>The OpenGIS Abstract Specification. Topic 12: OpenGIS Service Architecture.</i> Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2002.
[Pet08]	Peters, D. <i>Building a GIS. System Architecture Design Strategies for Managers.</i> Redlands, California (Estados Unidos de América) : ESRI Press, 2008.
[Pon12]	Ponce, Isabel. Definición de redes sociales. <i>Observatorio Tecnológico.</i> [En línea] 17 de Abril de 2012. http://recursostic.educacion.es/observatorio/web/es/internet/web-20/1043-redes-sociales?start=1 .
[Por13]	Portele, C. y Sankaran, S. <i>GeoServices REST API – RFC comments.</i> Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2013. OGC 12-064.
[Ric07]	Richardson, L. y Ruby, S. <i>RESTful Web Services.</i> Sebastopol (Estados Unidos de América) : O'Reilly, 2007.

Identificador	Referencia bibliográfica
[Sch05]	Schut, Peter. <i>Web Processing Service (WPS) Specification</i> . Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2005. OGC 05-007.
[Sch07]	Schut, Peter. <i>OpenGIS Web Processing Service</i> . Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2007. OGC 05-007r7.
[Sla14]	SlashDB. How it works. <i>SlashDB</i> . [En línea] SlashDB, 2014. http://www.slashdb.com/how-it-works/ .
[SOA14]	SOA. SOA Patterns. [En línea] Arcitura Education Inc., 2014. http://www.soapatterns.org/ .
[Sta11]	Stavropoulos, T., Vlahavas, I. y Vrakas, D. <i>A survey of service composition in ambient intelligence environments</i> . Norwell, Massachusetts (Estados Unidos de América) : Kluwer Academic Publishers Norwell, 2011.
[Tal08]	Talegaonkar, P. <i>Service Oriented Architecture for GIS Applications. Proceedings of the 12th International Conference of International Association for Computer Methods and Advances in Geomechanics</i> . Goa (India) : IACMAG, 2008.
[TOG09]	The Open Group. <i>The SOA Source Book</i> . Zaltbommel : Holanda, 2009.
[Vre02]	Vretanos, P. (ed.). <i>Web Feature Service Implementation Specification</i> . Wayland, Massachusetts (Estados Unidos) : Open Geospatial Consortium, 2002. OGC 02-058.
[Vre10]	Vretanos, P. (ed.). <i>OpenGIS Web Feature Service 2.0 Interface Standard</i> . Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2010. OGC 09-025r1 / ISO 19142.
[Vre11]	Vretanos, P. <i>A REST binding for WFS 2.0</i> . Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2011. OGC 09-025r1.
[Whi09]	Whiteside, A. (ed.). <i>Web Coverage Service (WCS) - Transaction operation extension</i> . Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2009. OGC 07-068r4.
[Wil08]	Wilson, T. <i>OGC KML</i> . Wayland, Massachusetts (Estados Unidos de América) : Open Geospatial Consortium, 2008. OGC 07-147r2.
[WWW01]	World Wide Web Consortium. <i>Web Services Description Language (WSDL) 1.1. W3C</i> . [En línea] 2001. http://www.w3.org/TR/wsdl .
[WWW04]	World Wide Web Consortium. <i>Semantic Markup for Web Services. World Wide Web Consortium</i> . [En línea] 22 de noviembre de 2004. http://www.w3.org/Submission/OWL-S .
[WWW07]	World Wide Web Consortium. <i>SOAP Version 1.2. W3C</i> . [En línea] 2007. http://www.w3.org/TR/soap .
[WWW14]	World Wide Web Consortium. <i>RDF Store Benchmarking. W3C</i> . [En línea] 2014. http://www.w3.org/wiki/RdfStoreBenchmarking .
[Zha08]	Zhao, T. y otros. <i>5th International Conference, GIScience 2008. Ontology-based Geospatial Data Query and Integration</i> . Park City, Utah (Estados Unidos de América) : Springer, 2008. Vols. Lecture Notes in Computer Science, volumen 5266.
[Zho04]	Zhou, X. y otros. <i>Multiresolution Spatial Databases: Making Web-based Spatial Applications Faster</i> . Berlín : Advanced Web Technologies and Applications. Lecture Notes in Computer Science. Springer, 2004. Vol. 3007.

9 Glosario

base topográfica *f.* Conjunto de datos vectoriales que recoge las características del relieve y de los elementos físicos o aparentes de un área geográfica mediante elementos que se almacenan con geometría bidimensional o tridimensional de tipo punto, línea y polígono, se clasifican según un catálogo y se enriquecen con atributos [ICG14b].

cartografía *f.* Ciencia, técnica y arte que trata la elaboración, la lectura e interpretación, el análisis y la explotación, la búsqueda, la historia, la conservación y el estudio en general de los mapas, planos, cartas, imágenes, perfiles, modelos tridimensionales o globos que representan el territorio a una escala determinada [Pan11].

cobertura *f.* Representación cartográfica continua de un aspecto o característica en el espacio, por oposición a la representación por medio de elementos [Nun12].

datum *m.* Cualquier nivel de superficie, línea o punto, utilizado como referencia de una dimensión, especialmente la altitud [Pan11].

datum geodésico *m.* Datum que define los parámetros que determinan las dimensiones del elipsoide de referencia [Pan11].

elipsoide terrestre *m.* Elipsoide de revolución que se aproxima al geoide, utilizado como base de referencia en trabajos geodésicos y cartográficos y, especialmente, para aplicar a los diferentes sistemas de proyección cartográfica y dar cobertura a los sistemas de navegación global por satélite [Pan11].

fotomapa *m.* Fotografía vertical de un territorio en la cual se han incorporado algunos datos, como toponimia, símbolos cartográficos, curvas de nivel, etc. [Pan11].

geodesia *f.* Ciencia que se encarga de medir y representar la figura y el campo gravitacional terrestre y de otros cuerpos espaciales, así como sus variaciones con el paso del tiempo [Pan11].

georeferenciar *v tr.* Situar sobre la Tierra un mapa, otro documento cartográfico o la representación de un objeto geográfico mediante el valor de sus coordenadas [Pan11].

imagen *f.* Documento gráfico de carácter continuo y aspecto fotográfico que representa un territorio o una escena, obtenido mediante un sensor remoto, como por ejemplo un equipo óptico-electrónico, una cámara fotográfica, un radar, etc. [Pan11].

mapa *m.* Representación proporcional del territorio en la cual la información se expresa de forma selectiva y generalizada mediante un lenguaje propio y convencional [Pan11].

mapa topográfico *m.* Representación gráfica bidimensional del conjunto de características del relieve y de los elementos físicos o aparentes de una área geográfica. Generalmente se obtienen por derivación de las bases topográficas [ICG14b].

metadato *m.* Cada una de las informaciones descriptivas que documentan una característica concreta de unos datos o, en general, de un recurso de información [Nun12].

ortofotografía *f.* Documento fotográfico que se obtiene a partir de fotografías aéreas en las cuales se han corregido las deformaciones perspectivas de la imagen y se ha restituido la imagen del terreno según una proyección ortogonal vertical [Pan11].

ortofotomapa *m.* Fotomapa, obtenido por la unión de ortofotografías, al cual se han añadido otras informaciones cartográficas, como curvas de nivel, cotas, toponimia, etc. [Pan11].

plano *m.* Documento cartográfico a gran escala, normalmente superior a 1:20.000, que representa una obra arquitectónica, urbanística, de ingeniería, etc., con dibujos de la estructura y la forma general o de cada parte o pieza, así como la distribución o las proporciones del conjunto [Pan11].

raster *m.* Red regular de celdas organizadas en filas y columnas por medio de la cual se representa un fenómeno geográfico muestreado según un intervalo de distancia constante, de manera que cada celda tiene asociado el valor del atributo medido en la posición correspondiente [Nun12].

relación topológica *f.* relación espacial invariante a las transformaciones de forma. La relación topológica básica de límite –topología arco-nodo o topología polígono-arco– sirve para estructurar los datos espaciales, mientras que la relación topológica básica de límite compartido –conectividad, adyacencia– se utiliza para operaciones de análisis espacial [Nun12].

resolución *f.* Grado de detalle de una representación cartográfica que se mide, de forma inversamente proporcional, por la dimensión mínima de los objetos representados [Nun12].

servicio de mapas *m.* servicio a través de Internet que permite visualizar e interactuar dentro de la ventana del navegador Web o de otros clientes con mapas proporcionados por un servidor remoto que se actualizan dinámicamente según las acciones del usuario del servicio [Nun12].

sistema de información geográfica *m.* Conjunto de datos organizados por medio de un software y un hardware que permiten almacenar, cartografiar, analizar, modificar y relacionar cualquier tipo de información geográfica con la finalidad de facilitar la gestión, el estudio y la planificación territorial [Pan11].

sistema de posicionamiento global *m.* Sistema de posicionamiento que, por medio de satélites, permite localizar las coordenadas terrestres y la altura sobre el nivel del mar en que se encuentra el receptor [Pan11].

sistema geodésico de referencia *m.* Conjunto de parámetros, como por ejemplo el elipsoide o el datum, que definen la forma y las dimensiones de la Tierra y el origen y orientación de los sistemas de coordenadas para determinar la posición de un punto en el espacio, así como la metodología para realizar las observaciones pertinentes [Pan11].

tesela *f.* Cada una de las unidades zonales en las que se descompone un espacio según una teselación regular o irregular [Nun12].

teselación *f.* Partición de un área según un conjunto de unidades poligonales que forman un mosaico de áreas disjuntas sin solapes ni vacíos [Nun12].

teselación de servicio de mapas *f.* Procedimiento consistente en dividir un mapa en secciones de manera que, para cada una de las escalas fijas de visualización del servicio de mapas y para cada una de las teselas en las que se divide la extensión geográfica del servicio, se genera una imagen prerenderizada del mapa que ahorra la generación del mapa para cada operación de visualización o navegación en el cliente [Nun12].

topología *f.* Conjunto de relaciones topológicas que se aplican a la definición y construcción de estructuras de datos que registran explícitamente y organizan los elementos espaciales representados de forma vectorial, de acuerdo con estas relaciones [Nun12].

topología arco-nodo *f.* información topológica y estructura de datos que garantiza la conectividad de las líneas a través de los nodos [Nun12].

topología polígono-arco *f.* información topológica y estructura de datos que garantiza la adyacencia de los polígonos a través de los arcos que los delimitan [Nun12].

10 Acrónimos

Término	Descripción
2D	Bidimensional
3D	Tridimensional
ASCII	Código Estándar Estadounidense para el Intercambio de Información (<i>American Standard Code for Information Interchange</i>)
BPEL	Lenguaje de Ejecución de Procesos de Negocio (<i>Business Process Execution Language</i>)
BSD	Distribución de Software Berkeley (<i>Berkeley Software Distribution</i>)
BTC	Base Topográfica de Cataluña
EBNF	Forma Extendida Backus-Naur (<i>Extended Backus-Naur Form</i>)
CAD	Diseño Asistido por Ordenador (<i>Computer Aided Design</i>)
CAM	Fabricación Asistida por Ordenador (<i>Computer Aided Manufacturing</i>)
CGIS	Sistema de Información Geográfico de Canadá (<i>Canada Geogrpahic Information System</i>)
CORBA	Arquitectura Común de Intermediarios en Peticiones a Objetos (<i>Common Object Request Broker Architecture</i>)
CRM	Gestión de Relaciones con los Clientes (<i>Customer Relationship Management</i>)
CRS	Sistema de Referencia de Coordenadas (<i>Coordinates Reference System</i>)
CRUD	Crear, Recuperar, Actualizar y Borrar (<i>Create, Read, Update and Delete</i>)
CSW	Servicio de Catálogo para la Web (<i>Catalogue Service for the Web</i>)
CS IDEC	Centro de Soporte a la IDEC
CTS	Servicio de Transformación de Coordenadas (<i>Coordinate Transformation Service</i>)
DARPA	Agencia de Proyectos de Investigación Avanzados de Defensa (<i>Defense Advanced Research Projects Agency</i>)
DAML-S	Agente de Lenguaje de Marcado de DARPA para Servicios (<i>DARPA Agent Markup Language for Services</i>)
DBMS	Sistema Gestor de Bases de Datos (<i>Database Management System</i>)
DE-9IM	Modelo de 9 Intersecciones Extendido Dimensionalmente (<i>Dimensionally Extended nine-Intersection Model</i>)
DGN	<i>Downers Grove North</i>

Término	Descripción
DXF	Formato de Intercambio de Dibujos (<i>Drawing Exchange Format</i>)
ebXML	Comercio Electrónico utilizando XML (<i>Electronic Business using XML</i>)
ebRIM	Modelo de Datos Registrales ebXML (<i>ebXML Registry Information Model</i>)
ED50	Datum Europeo de 1950 (<i>European Datum 1950</i>)
ERP	Planificación de Recursos Empresariales (<i>Enterprise Resource Planning</i>)
ESRI	Instituto de Investigación de Sistemas Medio Ambientales (Environmental Systems Research Institute)
ETRS	Sistema de Referencia Terrestre Europeo de 1989 (<i>European Terrestrial Reference System 1989</i>)
EVRS	Sistema de Referencia Vertical Europeo (<i>European Vertical Reference System</i>)
FGDC	Comité Federal de Datos Geográficos (<i>Federal Geographic Data Committee</i>)
FTP	Protocolo de Transferencia de Ficheros (<i>File Transfer Protocol</i>)
GDAL	Librería de Abstracción de Datos Geoespaciales (<i>Geospatial Data Abstraction Library</i>)
GeoJSON	Notación JavaScript de Objetos Geoespaciales (<i>Geospatial JavaScript Object Notation</i>)
GeoRSS	RSS Geográfico (<i>Geographic RSS</i>)
GIS4EU	SIG para Europa (<i>GIS for Europe</i>)
GML	Lenguaje de Marcado Geográfico (<i>Geography Markup Language</i>)
GPS	Sistema de Posicionamiento Global (<i>Global Positioning System</i>)
GPX	Formato de Intercambio GPS (<i>GPS Exchange Format</i>)
HATEOS	Hipermedia como Motor del Estado de la Aplicación (<i>Hypermedia as the Engine of the Application State</i>)
HTTP	Protocolo de Transferencia de HiperTexto (<i>Hypertext Transfer Protocol</i>)
ICGC	Instituto Cartográfico y Geológico de Cataluña
IDE	Infraestructura de Datos Espaciales (<i>Spatial Data Infrastructure</i>)
IDEC	Infraestructura de Datos Espaciales de Cataluña
IDEE	Infraestructura de Datos Espaciales de España
IDEG	Infraestructura de Datos Espaciales de Galicia
IE	Experimento de Interoperabilidad (<i>Interoperability Experiment</i>)
IETF	Grupo de Trabajo de Ingeniería de Interent (<i>Internet Engineering Task Force</i>)

Término	Descripción
IG	Información Geográfica
IGN	Instituto Geográfico Nacional
IIOIP	Protocolo Inter-ORB para Internet (<i>Internet Inter-ORB Protocol</i>)
IMS	Servidor de Mapas por Internet (<i>Internet Map Server</i>)
INSPIRE	Infraestructura para la Información Espacial en Europa (<i>Infrastructure for Spatial Information in Europe</i>)
IoE	Internet de Todo (<i>Internet of Everything</i>)
IOPE	Entradas, Salidas, Precondiciones y Efectos (<i>Inputs, Outputs, Preconditions and Effects</i>)
IOPR	Entradas, Salidas, Precondiciones y Resultados (<i>Inputs, Outputs, Preconditions and Results</i>)
IoT	Internet de las Cosas (<i>Internet of Things</i>)
IP	Protocolo de Internet (<i>Internet Protocol</i>)
ISO	Organización Internacional de Normalización (<i>International Organization for Standardization</i>)
ITRS	Sistema de Referencia Terrestre Internacional (<i>International Terrestrial Reference System</i>)
JSON	Notación JavaScript de Objetos (<i>JavaScript Object Notation</i>)
JTS	Suite de Topología Java (<i>Java Topology Suite</i>)
KML	Lenguaje de Marcado "Keyhole" (<i>Keyhole Markup Language</i>)
KMZ	Lenguaje de Marcado "Keyhole" Comprimido (<i>Keyhole Markup Language Zipped</i>)
KVP	Par Clave-Valor (<i>Key Value Pair</i>)
LAN	Red de Área Local (<i>Local Area Network</i>)
MDT	Modelo Digital de Terreno
MIME	Extensión multipropósito de correo de Internet (<i>Multi purpose Internet Mail Extension</i>)
MM	Multimedia
MTC	Mapa Topográfico de Cataluña
NCGIA	Centro Nacional para los Sistemas de Información Geográfica y el Análisis (<i>National Center for Geographic Information Systems and Analysis</i>)
OASIS	Organización para el Desarrollo de Estándares de Información Estructurada (<i>Organization for the Advancement of Structured Information Standards</i>)
OGC	Consorcio Geoespacial Abierto (<i>Open Geospatial Consortium</i>)

Término	Descripción
OpenGL	Librería de Gráficos Abierta (<i>Open Graphics Library</i>)
OpenGL ES	Librería de Gráficos Abierta para Sistemas Embebidos (<i>OpenGL for Embedded Systems</i>)
OSGeo	Fundación Geoespacial de Software Libre (<i>Open Source Geospatial Foundation</i>)
OSM	<i>OpenStreetMap</i>
OWL	Lenguaje Web de Ontologías (<i>Web Ontology Language</i>)
OWL-S	Lenguaje Web Semántico de Ontologías (<i>Web Ontology Semantic Language</i>)
OWL-Q	Lenguaje Web de Ontologías para la Calidad del Servicio (<i>Web Ontology Language for Quality of Service</i>)
P2P	Par a Par (<i>Peer to Peer</i>)
PC	Ordenador Personal (<i>Personal Computer</i>)
QoS	Calidad del Servicio (<i>Quality of Service</i>)
RDF	Marco de Descripción de Recursos (<i>Resource Description Framework</i>)
RDFS	Esquema <i>RDF</i> (<i>RDF Schema</i>)
RIA	Aplicación de Internet Enriquecida (<i>Rich Internet Application</i>)
RISC	Computador con Conjunto de Instrucciones Reducidas (<i>Reduced Instruction Set Computer</i>)
ROA	Arquitectura Orientada a Recursos (<i>Resource Oriented Architecture</i>)
RPC	Llamada a Procedimiento Remoto (<i>Remote Procedure Call</i>)
RSS	Sindicación Realmente Simple (<i>Really Simple Syndication</i>)
SA-REST	Anotaciones Semánticas para <i>REST</i> (<i>Semantic Annotation REST</i>)
SAS	Servicio de Alertas de Sensores (<i>Sensor Alert Service</i>)
SDE	Motor de Datos Espaciales (<i>Spatial Data Engine</i>)
SDO	Opción de Datos Espaciales (<i>Spatial Data Option</i>)
SensorML	Lenguaje de Modelado de Sensores (<i>Sensor Model Language</i>)
SHP	Fichero de formas de <i>ESRI</i> (<i>ESRI Shapefile</i>)
SI	Sistema de Información
SIG	Sistema de Información Geográfico (<i>Geographic Information System, GIS</i>)
SLA	Acuerdo de Nivel de Servicio (<i>Service Level Agreement</i>)

Término	Descripción
SOA	Arquitectura Orientada a Servicios (<i>Service Oriented Architecture</i>)
SOAP	Protocolo de Acceso a Objetos Simples (<i>Simple Object Access Protocol</i>)
SOC	Computación Orientada a Servicios (<i>Service Oriented Computing</i>)
SOS	Servicio de Observación de Sensores (<i>Sensor Observation Service</i>)
SPARQL	Protocolo SPARQL y Lenguaje de Consultas RDF (<i>SPARQL Protocol and RDF Query Language</i>)
SPS	Servicio de Planificación de Sensores (<i>Sensor Planning Service</i>)
SQL	Lenguaje de Consultas Estructurado (<i>Structured Query Language</i>)
SSL	Capa de Conexión Segura (<i>Secure Sockets Layer</i>)
SWE	Habilitación de Sensores Web (<i>Sensor Web Enablement</i>)
TFM	Trabajo Fin de Máster
TI	Tecnologías de la Información
TLS	Seguridad de la Capa de Transporte (<i>Transport Layer Security</i>)
TMS	Servicio de Mapa en Teselas (<i>Tile Map Service</i>)
TML	Lenguaje de Marcado de Transductores (<i>Transducer Markup Language</i>)
UDDI	Integración, Descubrimiento y Descripción Universal (<i>Universal Description, Discovery and Integration</i>)
UE	Unión Europea
UML	Lenguaje de Modelado Unificado (<i>Unified Modelling Language</i>)
URI	Identificador de Recurso Universal (<i>Universal Resource Identifier</i>)
URN	Nombre de Recurso Uniforme (<i>Uniform Resource Name</i>)
UTM	Universal Transversal de Mercator (<i>Universal Transverse Mercator</i>)
VRS	Sistema de referencia vertical (<i>Vertical Reference System</i>)
W3C	Consortio Mundial para la Web (<i>World Wide Web Consortium</i>)
W3DS	Servicio Web 3D (<i>Web 3D Service</i>)
WADL	Lenguaje Descriptivo de Aplicaciones Web (<i>Web Description Application Language</i>)
WAN	Red de Área Amplia (<i>Wide Area Network</i>)
WCS	Servicio Web de Coberturas (<i>Web Coverage Service</i>)

Término	Descripción
WebGL	Librería de Gráficos Web (<i>Web Graphics Library</i>)
WGS84	Sistema Geodésico Mundial de 1984 (<i>World Geodetic System 1984</i>)
WFS	Servicio Web de Objetos Geográficos (<i>Web Feature Service</i>)
WFS-T	Servicio Web de Objetos Geográficos Transaccional (<i>Web Feature Service – Transactional</i>)
WKB	Binario Bien Conocido (<i>Well Known Binary</i>)
WKT	Texto Bien Conocido (<i>Well Known Text</i>)
WMS	Servicio Web de Mapas (<i>Web Map Service</i>)
WMS-C	Servicio Web de Mapas con Cacheado de Teselas (<i>Web Map Service Tile Caching</i>)
WMTS	Servicio Web de Mapas en Teselas (<i>Web Map Tile Service</i>)
WNS	Servicio de Notificación Web (<i>Web Notification Service</i>)
WPS	Servicio Web de Procesamiento (<i>Web Processing Service</i>)
WS	Servicio Web (<i>Web Service</i>)
WSIO	Organización para la Interoperabilidad de los Servicios Web (<i>Web Services Interoperability Organization</i>)
WS-BPEL	BPEL para servicios Web (<i>Web Services BPEL</i>)
WS-CDL	Lenguaje para la Descripción de Coreografías de Servicios Web (<i>Web Services Choreography Description Language</i>)
WS-CF	Marco de trabajo para la Coordinación de Servicios Web (<i>Web Services Coordination Framework</i>)
WS-I	Interoperabilidad entre Servicios Web (<i>Web Services Interoperability</i>)
WVS	Servicio de Visualización Web (<i>Web View Service</i>)
XML	Lenguaje de Marcado Extensible (<i>Extensible Markup Language</i>)
XSRL	Lenguaje XML de Peticiones a Servicios (<i>XML Service Request Language</i>)

11 Anexo - Geoservicios del ICGC

La Tabla 11.1 recoge el conjunto de servicios publicados actualmente^{43 44}, agrupados éstos por tecnología servidora y recogiendo la interfaz y los protocolos soportados.

Nombre	Descripción	Tecnología	Interfaz	Protocolo
amplitud_termica	Mapa de amplitud térmica	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI ⁴⁵	REST
anomalies_bouguer	Mapa de las anomalías de Bouguer	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST
MGC_MapaBase	Mapas geológicos con selección de producto en función de nivel de zoom ⁴⁶ .	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST
BGC50mv1r01	Base geológica 1:50.000	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST
BGC250mv2	Base geológica 1:250.000	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST
estructural	Mapa estructural geotérmico	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST

⁴³ Relación obtenida de [ICG14] 1 de junio de 2014.

⁴⁴ Todos los productos se restringen al ámbito geográfico de la Comunidad Autónoma de Cataluña.

⁴⁵ Interfaz propietaria de ESRI que no sigue ningún estándar internacional.

⁴⁶ Se trata de un servicio en el que se combinan en dos capas diferentes los mapas geológicos 1:50.000 y 1:250.000. En función de la escala de visualización, solo uno de ellos muestra información.

Nombre	Descripción	Tecnología	Interfaz	Protocolo
flux_masses_agua	Mapa de flujos de masas de agua	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST
GeometryServer	Ejecución de procesos con geometrías de objetos geográficos ⁴⁷	ESRI ArcGIS Server 10.1	ESRI	REST
icc_atlm	Archivo temporal de límites municipales	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST
icc_bseccen	Base de secciones censales	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST
icc_bt5m	Base topográfica 1:5.000	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST
icc_bt25m	Base topográfica 1:25.000	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST
icc_bt50m	Base topográfica 1:50.000	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST
icc_limadmin	Base municipal 1:50.000	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST

⁴⁷ Dispone de herramientas para calcular paralelas, distancias, longitudes y áreas, cierre de polígonos, densificar vértices, dividir, etiquetar puntos, generalizar, intersectar, recortar o extender, modificar, simplificar, unir, determinar polígono convexo, proyectar, determinar zona afectada y relación espacial entre geometrías.

Nombre	Descripción	Tecnología	Interfaz	Protocolo
icc_bv5m	Base del callejero ⁴⁸	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST
icc_ct1m	Cartografía topográfica 1:1.000 y 1:2.000	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST
icc_mdt	Modelos Digitales del Terreno con resolución 15x15m y 5x5m	ESRI ArcGIS Server 10.1	WCS 1.1.1	HTTP (Get) HTTP (Post)
icc_polind	Base de polígonos industriales	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST
icc_talls	Distribución de hojas de las diferentes series cartográficas	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST
icc_vertexs	Base de vértices geodésicos	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST
icc_100cims	Base de los 100 picos de montaña más emblemáticos	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST
instal_bombes_calor	Mapa de instalaciones geotérmicas	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST
manifestacions_termals	Mapa de indicios geotérmicos	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST

⁴⁸ La propiedad de la base de calles está compartida entre el ICGC y otros organismos de la administración autonómica y local, razón por la cual no se publica en este servicio las geometrías de los ejes de calle y de los portales, simplemente se visualiza los textos de los nombres de calle y números de portal.

Nombre	Descripción	Tecnología	Interfaz	Protocolo
MGC50mv1r01	Mapa geológico 1:50.000	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST
MGC250mv2	Mapa geológico 1:250.000	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST
parametres_termics	Mapa de parámetros térmicos	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST
profunditat_superficies	Mapa de profundidades de las superficies	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST
riscos_geologics	Mapa de riesgos geológicos	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST
salt_termic	Mapa de salto térmico estacional	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST
sismicitat	Mapa sismológico	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST
Sondeigs	Mapa de sondeos	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST
subvencions_ICAEN	Mapa de instalaciones geotérmicas	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST
tecniques_geofisiques	Mapa de geofísica aplicada	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST

Nombre	Descripción	Tecnología	Interfaz	Protocolo
temperatura	Mapa de temperaturas a diferentes profundidades	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST
Tsup_reducida	Mapa de temperatura superficial reducida	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST
Zones_potencial	Mapa de zonas de potencial geotérmico profundo	ESRI ArcGIS Server 10.1	WMS 1.3.0	HTTP (Get) HTTP (Post)
			ESRI	REST
icgcTransfCoord	Transformación de coordenadas entre diferentes sistemas de referencia	GDAL ⁴⁹	WPS 1.0.0	HTTP (Get) HTTP (Post)
geotreballs	Distribución de hojas de publicación de los geotrabajos	Geoserver	WMS 1.3.0	HTTP (Get) HTTP (Post)
nivoallaus	Base de aludes del Pirineo	Geoserver	WMS 1.3.0	HTTP (Get) HTTP (Post)
icc_fonstopografic	Mapas topográficos representados mediante tonalidades de gris ⁵⁰ .	LizardTech Express Server 9	WMS 1.1.1	HTTP (Get) HTTP (Post)
icc_mapesbase	Mapas topográficos y ortofotos vigentes a diferentes escalas.	LizardTech Express Server 9	WMS 1.1.1	HTTP (Get) HTTP (Post)
icc_ndvicolor	Cobertura temática que muestra el índice de diferencia normalizada de vegetación.	LizardTech Express Server 9	WMS 1.1.1	HTTP (Get) HTTP (Post)
icc_ortohistorica	Ortofotos históricas a diferentes escalas.	LizardTech Express Server 9	WMS 1.1.1	HTTP (Get) HTTP (Post)

⁴⁹ Librería de Abstracción de Datos Geoespaciales (GDAL) distribuida bajo licencia de código libre por la *Open Source Geospatial Foundation* que permite realizar transformaciones entre diferentes formatos de datos geoespaciales. Dispone de API para C++, C#/.Net, Java, Perl Python, R, Ruby y VB6.

⁵⁰ Permite dar servicio a aquellos usuarios para los que la cartográfica topográfica actúa como marco de referencia sobre el cual precisan incorporar sus propias capas temáticas que, de esta forma, aparecerán resaltadas y se mejorará la legibilidad del conjunto.

Nombre	Descripción	Tecnología	Interfaz	Protocolo
icc_mapesmultibase	Mapas topográficos y ortofotos con selección de producto en función de nivel de zoom ⁵¹ .	MapCache	WMS 1.3.0	HTTP (Get) HTTP (Post)
			WMTS 1.0.0	REST
icc_ortoxpres	Ortofotos provisionales a diferentes escalas ⁵² .	SIGRID StereoWebMap	WMS 1.1.1	HTTP (Get) HTTP (Post)
geocodificador⁵³	Transformación de direcciones en posiciones y viceversa ⁵⁴	ICGC Servlet	ICGC	HTTP (Post) + SOAP

Tabla 11.1 Relación de geoservicios del ICGC.

⁵¹ Se trata de un servicio en el que se integran todos los mapas topográficos en una única capa de información que, en función del nivel de zoom, muestra un producto topográfico de una escala u otra. Este comportamiento se replica para las diferentes coberturas de ortofoto y para las versiones codificadas en gris de éstas y de los mapas topográficos. En los cuatro casos se ofrece la versión en coordenadas geográficas y proyectadas *UTM* con datum *ETRS89*.

⁵² Obtenidas mediante un proceso de rectificación óptica en tiempo real a partir de los fotogramas originales. Este sistema genera unas ortofotos que, aunque con una menor precisión, permiten a los usuarios acceder a este tipo de información poco tiempo después de haberse realizado el vuelo y mucho antes de que se publique la versión oficial.

⁵³ Servicio de uso restringido a las administraciones públicas.

⁵⁴ La transformación de una dirección postal –nombre de calle y número– en una localización geográfica se conoce con el término de geocodificación directa –*geocoding*–. El caso contrario, detectar cual es la dirección postal asociada con una ubicación concreta, se denomina geocodificación inversa –*reverse geocoding*–.
