

Institut de Noves Tecnologies de la Imatge

Universitat Jaume I



Aproximación al tratamiento completo del ciclo de vida sobre datos provenientes de sensores mediante estándares GIS

Tesis Doctoral
Sergio Trilles Oliver

Trabajo realizado bajo la dirección de los Doctores **D. Joaquín Huerta Guijarro** y **D. Óscar Belmonte Fernández**, presentado en conformidad a los requisitos para obtener el grado de Doctor por la Universitat Jaume I con la mención Internacional en el título de Doctor.

Castelló de la Plana
Julio 2015

Este trabajo ha sido íntegramente financiado mediante una ayuda predoctoral
Vali+d de la Generalitat Valenciana (ref. ACIF/2012/112)

A la meua família

Agraïments

Aquesta tesi és l'esforç d'un grapat d'anys de la meua vida. En ella es veu reflectida cada una de les persones que han participat de forma directa o indirecta.

En primer lloc, voldria agrair a tots dos tutors per la seua ajuda en aquests tres anys. Reconèixer al Dr. Joaquín Huerta l'oportunitat i ajut per continuar endavant amb aquest treball. Al Dr. Óscar Belmonte, per ser capaç de fer-me trobar el camí. I tampoc voldria oblidar-me de la Dr. Laura Díaz, que va ser qui em va fer descobrir aquest meravellós camp de la tecnologia geoespacial.

També m'agradaria agrair al Dr. Sven Schade la seua hospitalitat durant el temps viscut en terres alpines. Agrair el companyerisme rebut per part de tots els components i excomponents del grup d'investigació GEOTEC, amb els quals he tingut el plaer de conviure durant aquests quatre anys i escaig. En especial a Alejandro Lujan per ajudar-me a fer de *SEnviro* una realitat. Al mateix temps, també vull reconèixer al Dr. Pablo Juan el seu ajut en aquelles parts de la tesi on l'estadística és present.

No vull oblidar a tots els meus amics i amigues que algun dia m'han considerat com a desaparegut. A tots i totes, gràcies per ser i estar amb mi. En especial a *Les Estakes* per fer-me més entretinguda la vida.

Finalment, voldria agrair aquest treball a la meua família. Als meus pares per donar-me tot el necessari per a poder arribar fins aquí. Als meus germans i nebodes. I com no, a Clara per ser qui ha de patir-me de més a prop. Gràcies de tot cor!

Resumen

El aumento de la necesidad por conocer cuál es el estado de cualquier entorno, ya sea por motivos medioambientales, económicos, como sociales, ha propiciado el crecimiento del uso de nodos de sensorización capaces de cuantificar y cualificar cualquier tipo de fenómeno observable. Esta necesidad, combinada con la disminución de los precios y tamaño de los componentes, ha favorecido el despliegue de dichos nodos con la finalidad de interconectarse entre ellos formando una red que proporciona los datos de sensorización.

Mayoritariamente, las redes de sensores disponibles actualmente ofrecen sus observaciones mediante protocolos y formatos propietarios. Esta situación, muestra la carencia de un estándar común para que facilite la gestión de los datos generados por los sensores de una forma interoperable. Además, dichos protocolos y formatos, tampoco favorecen el consumo de este tipo de contenido en entornos donde los recursos computacionales son más reducidos, como pueden ser teléfonos móviles.

La tendencia actual es utilizar los protocolos de Internet para la comunicación entre los nodos de sensorización. Dichos protocolos, permiten el uso de los sensores como una parte de Internet, donde el enfoque es llamado el Internet de la Cosas. Esto propicia un escenario idóneo para el control de los sensores y favorece el acceso a ellos, aunque actualmente tampoco existe un estándar capaz de ofrecer dicho enfoque de una forma universal. El uso de los protocolos de Internet ha favorecido el uso de sensores ofreciendo nuevos usos donde antes no era posible. Prueba de ello son las ciudades inteligentes, en las cuales la implantación de sensores en sus alrededores, se ha hecho con la finalidad de conocer y controlar lo que está sucediendo en ella.

De esta forma, cada día hay un mayor despliegue de redes de sensores midiendo gran cantidad de fenómenos diferentes. Estas redes proporcionan grandes

volúmenes de datos en múltiples formatos, resoluciones y escalas con una alta tasa de generación que no es posible analizar con las tecnologías actuales, por lo que es necesario ofrecer métodos de análisis adaptados a estas características. Además, también se necesita ofrecer dichos análisis mediante procesamientos que garanticen su reusabilidad e interoperabilidad. Esto puede lograrse mediante los procesamientos que siguen estándares, ya que establecen una interfaz predefinida para su ejecución.

Enmarcados en este escenario, se presenta la actual tesis. En ella, se utiliza como hilo conductor el ciclo de vida de cualquier contenido geoespacial, que coincide con el ciclo de los datos proporcionados por sensores. Para ello, se presentan cuatro etapas que son: adquisición, publicación, acceso y análisis. Para cada una de ellas se detallan los problemas existentes y se proponen diferentes soluciones.

Para la etapa de adquisición se presenta una plataforma de sensorización, basada en software y hardware libre, energéticamente autónoma e integrada en el Internet de las Cosas. En la etapa de publicación se diseña un procesamiento estándar para la publicación automatizada de contenido geoespacial en los diferentes servicios estándares del ámbito geoespacial. Además, en la etapa de acceso proponen diferentes formas de acceder a datos de sensores dependiendo del caso de uso en particular. Las interfaces propuestas se basan en seguir los paradigmas del Internet de las Cosas a través del uso de estándares geoespaciales mejorando la interoperabilidad, además de ofrecer un acceso ligero para dispositivos con restricciones limitadas o de ser capaces de servir grandes cantidades de observaciones. En la última etapa, el análisis, se aportan dos metodologías diferentes. La primera de ellas permite el análisis en tiempo real sobre grandes cantidades de datos y, la segunda, ofrece aplicar cualquier tipo de modelo utilizando un procesamiento estándar. Finalmente se proponen diferentes casos de uso donde son aplicadas todas las aportaciones que se describen.

Palabras clave: *Sensores, Internet de las Cosas, Publicación automatizada, Interoperabilidad, Estándares GIS, Análisis en Tiempo Real.*

Abstract

The ever-increasing changes in our environment, due to environmental, economic, or social reasons, has led to the use of sensing nodes able to quantify and qualify any type of observable phenomenon and monitor their status. This necessity, combined with the decline in prices and size of components, has favoured the development of these nodes in order to connect and create networks to perceive the environment.

The available sensor networks offer their observations using proprietary formats and protocols. This situation shows the lack of a common standard to facilitate the management of data generated by sensors in an interoperable way. In addition, such protocols and formats do not favour the consumption of this kind of data in environments where computing resources are lower, such as mobile phones.

The current trend is to use the Internet protocols for communication among sensor nodes. These protocols allow sensors to be a part of the Internet; this concept is called the Internet of things. Sensors being controlled by using the internet, although there are no universal standards for this approach. The use of Internet protocols has favoured the use of sensors offering new applications where previously this was not possible. The Smart cities paradigm provides a clear example of this, whereby the implementation of sensors have been established in order to know and control what is happening within the city.

In this way, every day sensor networks measure a large number of different phenomena. These networks provide large volumes of data in multiple formats, resolutions and scales with a high rate of generation that are not possible to analyse with existing technologies, so it is necessary to provide methods of analysis for these characteristics. In addition, they are needed to guarantee their usability and interoperability. This can be achieved through the standards, as they establish ways to use this kind of content.

In this scenario, the current thesis is presented. For this the life cycle of any geospatial content is used as thread driver, which it coincides with the life cycle of the content provided by sensors. Four stages are defined, these are: acquisition, publication, access and analysis. For each stage, the thesis defines the current problems and provides a possible solution.

For the acquisition stage a platform for sensing is presented, based on open software and hardware, that is energy-autonomous and integrated with the Internet of Things. In the second stage, a standard process to publish geospatial content is designed, it is used to publish sensor data. In addition, in the access stage, different ways of accessing data from sensors depending on the use case are provided. The proposed interfaces are based on the paradigms of the Internet of things through the use of geospatial standards to improve interoperability, as well as offering a lightweight access to devices with limited restrictions or that are able to serve large numbers of observations. In the last stage, two different methodologies are defined in order to analyse sensor data. The first one allows the analysis in real-time of large amounts of data, and the second methodology offers the application of any type of model using a standard processing. Finally, different use cases for each contribution are presented.

Keywords: *Sensors, Internet of Things, Automatic publication, Interoperability, GIS standards, Realtime analysis.*

Tabla de contenidos

Resumen	V
Abstract	VII
Acrónimos	XIX
1. Introducción	1
1.1. Visión general	1
1.2. Motivación	6
1.3. Contexto	7
1.4. Contribuciones y estructura de la tesis	10
2. Adquisición	13
2.1. Introducción	13
2.2. Conceptos relacionados	17
2.2.1. Redes de sensores	17
2.2.2. Plataformas actuales basadas en microcontrolador	20
2.3. La plataforma SEnviro	22
2.3.1. Visión general de SEnviro	22
2.3.2. Diseño de un SEnviro Thing	23
2.3.3. El comportamiento de SEnviro Thing	25
2.4. Trabajos relacionados para la adquisición	26
2.5. Conclusiones	30
3. Publicación	33

3.1. Introducción	33
3.2. Conceptos relacionados	35
3.2.1. Estándares para la definición de datos geoespaciales	36
Estándares para la definición de datos de sensores	38
3.2.2. Generación y publicación de metadatos	42
3.2.3. Geoprocesamiento	45
3.3. Geoss Service Factory	47
3.3.1. Publicación de datos geoespaciales	50
Publicación de datos de sensores	52
3.3.2. Generación y publicación de metadatos	53
3.4. Trabajos relacionados para la Publicación	57
3.4.1. Publicación de contenido geoespacial	57
3.4.2. Generación y publicación de metadatos	58
3.5. Conclusiones	59
4. Acceso	61
4.1. Introducción	61
4.2. Conceptos relacionados	64
4.2.1. Servicios Web generales	64
Interfaces mediante REST	64
4.2.2. Interfaz para IoT: OGC SensorThings API	65
4.2.3. Mecanismos para el tiempo real	67
4.3. Interfaces para acceder a sensores	71
4.3.1. Interfaz ligera para datos de sensores	72
Servicio de descarga	72
Servicio de visualización	76
4.3.2. OGC SensorThings API	77
4.3.3. Servicios para el acceso a datos de sensores con alta tasa de refresco	79
4.4. Trabajo relacionado	82
4.5. Conclusiones	84

5. Análisis	87
5.1. Introducción	87
5.2. Conceptos Relacionados	89
5.2.1. Análisis en tiempo real	89
5.2.2. Estándares para el geoprocesamiento	93
5.3. Metodología para el análisis en tiempo real	93
5.3.1. Enfoque bróker	95
5.3.2. Algoritmos de análisis	96
5.3.3. Resultados del análisis	97
5.4. Metodología para el análisis mediante procesamientos	97
5.4.1. Modelización de los parámetros	100
5.4.2. Arquitectura de los servicios de procesamiento	100
5.5. Trabajo Relacionado	102
5.5.1. Trabajos relacionados con el análisis de grandes cantidades de datos en tiempo real	103
5.5.2. Trabajos relacionados con el análisis del geoprocesamiento	103
5.6. Conclusiones	104
6. Casos de uso	107
6.1. SEnviro: prueba de concepto en la Universitat Jaume I	107
6.1.1. Contexto de la Universitat Jaume I	108
6.1.2. La plataforma SEnviro para la monitorización ambiental . .	109
Construyendo un SEnviro Thing	110
Detalles del comportamiento del SEnviro Thing	115
6.1.3. Fuente de alimentación autónoma	117
Consumo energético	117
Pruebas de vida de la batería	119
6.1.4. Un cliente Web capaz de conectarse a la API SensorThings	120
6.2. Ejemplo de ejecución del GSF	121
6.3. Propuesta para el acceso a redes heterogéneas	130
6.3.1. Fuentes de datos utilizadas	131

Información oficial de sensores: red de calidad del aire de la Generalitat Valenciana	131
Información voluntaria de sensores: Meteoclimatic	132
6.3.2. Pre-procesado de las fuentes	133
Módulo de publicación en base de datos	134
Módulo de publicación en SOS	135
6.3.3. Servicios	136
6.3.4. Cliente móvil	137
6.4. Detección de anomalías sobre datos de sensores	139
6.4.1. Detección de eventos en tiempo real usando CUSUM	139
6.4.2. Propuesta de implementación	141
Capa contenido	142
Implementación de la capa de servicios	143
Panel de control para la visualización	149
6.5. Predicción del riesgo de incendios	151
6.5.1. Materiales: datos para la predicción de riesgo de incendios forestales	154
6.5.2. Modelos para la predicción de riesgo de incendios forestales	155
Procesos puntuales espaciales	156
Procesos Poisson	160
Procesos Area-Interaction	162
6.5.3. Modelos definidos	164
6.5.4. Ejecución de los modelos	164
6.5.5. Un ejemplo de ejecución	170
7. Conclusiones y Trabajo Futuro	175
7.1. Conclusiones generales	175
7.1.1. Adquisición	175
7.1.2. Publicación	176
7.1.3. Acceso	177
7.1.4. Análisis	178
7.2. Líneas futuras	181

7. Conclusions and Future Work	183
7.1. General conclusions	183
7.1.1. Acquisition	183
7.1.2. Publication	184
7.1.3. Access	185
7.1.4. Analysis	186
7.2. Future research	188
A. Publicaciones	191
Bibliografía	195

Índice de Ilustraciones

1.1. Ciclo de vida del contenido geoespacial.	3
1.2. Forma de actuar en el contenido geoespacial.	3
1.3. Arquitectura por capas.	4
2.1. Diferentes nodos de sensorización que forman una red de sensores.	15
2.2. Diferentes nodos de sensorización para varios entornos. a) Nodo de calidad del aire. b) Nodo de monitorización de la salud. c) Nodo de parking inteligente.	16
2.3. Diferentes tipologías para redes de sensores. a) Estrella. b) Árbol. c) Malla.	18
2.4. Diseño conceptual de un nodo de <i>SEnviro Thing</i>	23
2.5. Diagrama del comportamiento de un <i>SEnviro Thing</i>	25
3.1. Arquitectura INSPIRE.	37
3.2. Componentes que forman el estándar SensorML.	40
3.3. Componentes que forman el estándar O&M.	42
3.4. Modelo UML de la interfaz WPS.	47
3.5. Posición de GSF dentro de la arquitectura INSPIRE.	48
3.6. Patrón factoría utilizado en Geoss Service Factory.	50
3.7. UML con los parámetros de entradas y salida del GSF.	51
3.8. Diferentes pasos para la publicación de datos geospaciales utilizando GSF.	53
3.9. Diferentes pasos para la publicación de datos de sensores utilizando GSF.	54

3.10. Diferentes pasos para la generación y publicación de metadatos utilizando GSF.	55
3.11. Parámetros de entrada y salida del <i>Generador de Metadatos</i>	55
3.12. Secuencia de pasos del <i>Generador de Metadatos</i>	56
4.1. Modelo de datos en OGC SensorThings API (fuente: http://ogc-iot.github.io/ogc-iot-api)	67
4.2. Modelos de comunicación. a) Punto a punto. b) Publicación/Suscripción.	69
4.3. Esquema general con todos los servicios, interfaces y formatos de salida.	72
4.4. Representación de los elementos de la interfaz. a) Punto de entrada. b) Elemento. c) Descripción del sensor.	75
4.5. Ejemplo de visualización para el servicio diseñado.	77
4.6. Patrón URI de OGC SensorThings API.	78
4.7. Ejemplo de OGC SensorThings API basada en una plataforma <i>SEnviro</i>	78
4.8. Visión general del Sensor Data Streaming.	81
4.9. Modelo punto a punto.	82
5.1. Forma de trabajar típica para procesar flujos de datos en tiempo real.	91
5.2. Vista general de la metodología presentada para analizar flujos de datos en tiempo real.	94
5.3. Enfoque conceptual de bróker.	96
5.4. Modelo de publicación / suscripción.	98
5.5. Ejemplo de las entradas y salidas del geoprocesamiento.	101
5.6. Diferentes capas para el geoprocesamiento.	101
5.7. Diagrama de secuencia para la ejecución del WPS.	102
6.1. Ubicaciones de cada <i>SEnviro Thing</i> dentro del campus de la Universitat Jaume I.	108
6.2. Red <i>SEnviro</i> para el campus de la Universitat Jaume I.	110
6.3. Montaje de un <i>SEnviro Thing</i>	112
6.4. Caja Pagoda.	115

6.5. Cliente desarrollado para mostrar las observaciones de la red <i>SEnviro</i> .	120
6.6. Representación de los incendios producidos en Castellón entre los años 2005 al 2010.	122
6.7. Captura de los datos publicados en Geoserver.	126
6.8. Captura de la información del dato publicado en Geoserver.	127
6.9. Captura de la publicación del metadato en GeoNetwork.	128
6.10. Mapa de las 61 estaciones de calidad del aire en la red Valencia. .	132
6.11. Mapa de las estaciones de la red Meteoclimatic.	133
6.12. Módulo de publicación en BBDD y SOS.	134
6.13. Esquema general con todos los servicios, interfaces y formatos de salida.	136
6.14. Casos de uso del cliente diseñado.	137
6.15. Diferentes pantallas. a) Lista de estaciones de calidad del aire. b) Información para una estación de calidad del aire. c) Observaciones de una estación de calidad del aire. d) Filtro temporal.	138
6.16. Diferentes pantallas. a) Últimas observaciones en mapas Bing. b) Selección de los fenómenos para mostrar. c) Últimas observaciones en mapas de ESRI. d) Selección de mapas.	139
6.17. Resumen tecnológico de la implementación del sistema para detectar anomalías.	142
6.18. La figura muestra cómo se utiliza RMS para las salidas, tanto para eventos como para observaciones.	148
6.19. Diferentes capturas del cliente. a) Muestra los nodos de sensores (o agrupación) como marcadores, indicando los eventos en el marcador. b) Muestra cómo se aplica la agrupación teniendo en cuenta el número de eventos.	150
6.20. Diferentes capturas del cliente. a) Muestra un menú para visualizar los fenómenos que componen cada nodo. b) Un gráfico para visualizar las últimas observaciones y los eventos lanzados	151
6.21. Parámetros de entrada y salida de los modelos.	166
6.22. Diagrama de secuencia para la ejecución del riesgo de incendio. .	169

- 6.23. Representación visual de los parámetros de entrada que son, a) el contorno, b) el incendio, c) la orientación, d) el uso de la tierra, e) la elevación y f) la pendiente. 171
- 6.24. Gráficas de todos los modelos. a) Gráfico del modelo de *Poisson* polinómicas con x, y . b) Gráficos de modelo de *Poisson* con $x, y, 2$. c) Gráficos de modelo de *Poisson* con covariable elevación. d) Gráfico del modelo de *Poisson* con el uso del suelo de covarianza. e) Gráfico del modelo de *Poisson* con todas las covariables. f) Gráfico del modelo de *Area-Interaction* con x, y polinómicas. g) Gráfico del modelo de *Area-Interaction* con covariable elevación. h) Gráfico del modelo de *Area-Interaction* con el uso del suelo de covarianza. i) Gráfico del modelo de *Area-Interaction* con todas las covariables. 172
- 6.25. La representación visual de los contenidos de un punto KML. . . . 173
- 6.26. Mapa con los riesgos de incendio. a) Mapa del modelo de *Poisson* polinómicas con x, y . b) Mapa de modelo de *Poisson* con $x, y, 2$. c) Mapa de modelo de *Poisson* con covariable elevación. d) Mapa del modelo de *Poisson* con el uso del suelo de covarianza. e) Mapa del modelo de *Poisson* con todas las covariables. f) Mapa del modelo de *Area-Interaction* con con x, y . g) Mapa del modelo de *Area-Interaction* con covariable elevación. h) Mapa del modelo de *Area-Interaction* con el uso del suelo de covarianza. i) Mapa del modelo de *Area-Interaction* con todas las covariables. 174

Acrónimos

IP	Protocolo de Internet
IoT	Internet de las Cosas
URI	Identificador de Recursos Uniforme
WSN	Redes de Sensores Inalámbricos
GIS	Sistemas de Información Geoespacial
ICT	Tecnologías de la Información y las Comunicaciones
SOA	Arquitecturas Orientadas a Servicios
SDI	Infraestructuras de Datos Espaciales
IEEE	Instituto de Ingenieros Eléctricos y Electrónicos
GEOSS	Global Earth Observation System of Systems
GMS	Global Monitoring for Environment and Security
INSPIRE	Infraestructure for Spatial Information in Europe
CDTI	Centro para el Desarrollo Tecnológico Industrial
INIT	Instituto de Nuevas Imágenes de la Tecnología
INLA	Aproximaciones Integradas Anidadas Laplace
JRC	Joint Research Center
SEnviro	Sense our ENVIRONment

ETL	Extraer, Transformar y Cargar
WoT	Web de las Cosas
Wi-Fi	Wireless Fidelity
RFID	Identificación por Radiofrecuencia
HTTP	Protocolo de Transferencia de Hipertexto
REST	Transferencia de Estado Representacional
GPIO	Entradas y Salidas de Propósito General
PIR	Infrarojos Pasivos
GSF	Geoss Service Factory
GII	Infraestructuras de Informació Geospacial
OGC	Open Geospatial Consortium
EO	Observación de la Tierra
WMS	Web Map Service
WFS	Web Feature Service
SOS	Sensor Observation Service
SES	Sensor Event Service
CSW	Catalogue Service para la Web
WPS	Web Processing Service
SWE	Sensor Web Enablement
XML	Lenguaje de Marcas Extensible
SensorML	Sensor Model Language
O&M	Observations and Measurements
TransducerML	Transducer Markup Language

SPS	Sensor Planning Service
SAS	Sensor Alert Service
CRS	Referencia de Coordenadas del Sensor
UML	Unified Modeling Language
SPP	Service Publication Profile
KML	Keyhole Markup Language
GML	Geography Markup Language
XSLT	Extensible Stylesheet Language Transformations
TCP/IP	Protocolo de control de transmisión / Protocolo de Internet
SOAP	Simple Object Access Protocol
JSON	JavaScript Object Notation
MOM	Message Oriented Middleware
QoS	Calidad del servicio
JMS	Java Message Service
MQTT	Message Queue Telemetry Transport
STOMP	Protocolo de mensajes
XMPP	Extensible Messaging and Presence Protocol
RMS	RealTime Message Service
SDS	Sensor Data Streaming
PEs	Elementos de Procesamiento
DSPS	Sistemas de procesamiento de flujos distribuidos
UJI	Universitat Jaume I
WAP	Punto de Acceso Inalámbrico

RTC	Reloj en Tiempo Real
UART	Transmisor-Receptor Asíncrono Universal
PLA	Ácido Poliláctico
ABS	Acrilonitrilo Butadieno Estireno
CSV	Valores Separados por Comas
WDT	Watch Dog Timer
CSS	Hojas de Estilo en Cascada
BBDD	Bases de Datos
RSS	Sindicación Realmente Simple
HTML	Lenguaje de Marcas de Hipertexto
CUSUM	CUmulative SUM
CLC	Corine Land Cover
IGN	Instituto Geográfico Nacional
AEMA	Agencia Europea de Medio Ambiente
CSR	Aleatoriedad Espacial Completa

Capítulo 1

Introducción

Este primer capítulo tiene como objetivo sintetizar en esencia cuál es el alcance de la actual tesis. Para ello, en la Sección 1.1 se propone una visión general del trabajo realizado. En la segunda sección (Sección 1.2) se detalla cuál ha sido la motivación a lo largo de la investigación propuesta. El contexto en el que se enmarca la tesis se muestra en la Sección 1.3. La Sección 1.4 enumera las contribuciones que aporta la investigación y detalla la estructura con la que se presenta el actual documento.

1.1. Visión general

Las necesidades de la sociedad actual por conocer qué sucede en su entorno, unido a las necesidades medioambientales como la preocupación por el cambio climático o la seguridad en los entornos industriales, ha generado un aumento en el despliegue de sensores capaces de cuantificar los estados de un entorno en particular (ONeill y otros, 1997).

Otra de las razones que ha favorecido la adaptación de este tipo de dispositivos, ha sido el descenso del precio de los componentes, por lo que, una gran cantidad de sensores han sido desplegados con la finalidad de crear redes con objetivos dispares.

El uso del protocolo de Internet (IP, en sus siglas en inglés) se ha utilizado para lograr la conectividad entre los nodos de sensores e Internet (Rodrigues y Neves, 2010). Mediante este enfoque, los sensores se interconectan utilizando protocolos principalmente abiertos, en los que cada dispositivo tiene su propia

dirección IP. Dicha tendencias se llama el Internet de las Cosas (IoT, en sus siglas en inglés) (Kortuem y otros, 2010) y ha favorecido el poder utilizar dichos sensores mediante los protocolos establecidos para Internet. Esta unión, ha permitido que actualmente se estén generando y sirviendo una gran cantidad de información producida directamente por los sensores utilizando Internet como medio. Siguiendo este paradigma, cada sensor es representado mediante un Identificador de Recursos Uniforme (URI, en sus siglas en inglés).

Por todas estas razones, hay una creciente demanda en la creación de redes de sensores inalámbricos (WSNs, en sus siglas en inglés) que proporcionan información actualizada sobre el estado del fenómeno que se desee monitorizar. El objetivo de los sensores, en la totalidad de sus aplicaciones, es tomar medidas (observaciones) sobre un fenómeno. Seguidamente, dicha observación se publica para realizar algún tipo de análisis con la finalidad de tomar decisiones y desencadenar las actuaciones necesarias.

Este contexto pone en duda las formas tradicionales en las que son gestionados los sensores, al igual que sus observaciones generadas, en las cuales se utilizaban formatos y protocolos propietarios adaptados a un uso en particular. Esta situación presenta un problema de conectividad, el cual es necesario resolver para adaptarse a estos nuevos requisitos.

El principal objetivo de esta tesis, es aplicar nuevos mecanismos para cubrir las necesidades descritas, además de ofrecer un tratamiento completo sobre los sensores y los datos generados de estos. Para ello, se define el ciclo de vida sobre los datos de sensores (Figura 1.1).

Dicho ciclo de vida, al igual que en cualquier contenido geoespacial, está dividido en cinco etapas diferentes: adquisición, publicación, descubrimiento, acceso y análisis; con la excepción de que la adquisición y el análisis comparten la misma funcionalidad, ya que el resultado de ambas es la producción de nuevo contenido. La adquisición únicamente se encuentra como fase inicial, considerándose cuando los sensores producen nuevo contenido. Las siguientes veces será el análisis quien genere nuevo contenido para su posterior publicación.

La necesidad de ofrecer los diferentes mecanismos para cada una de las etapas del ciclo de vida de la información de sensores, refleja el flujo de trabajo mostrado por la Figura 1.2, donde tras la adquisición y el análisis de los datos de sensores, estos deben ser publicados para un posterior descubrimiento y además,



Figura 1.1: Ciclo de vida del contenido geoespacial.

poder realizar cualquier tipo de actuación dependiendo de los propios datos o los resultados de su análisis.



Figura 1.2: Forma de actuar en el contenido geoespacial.

El presente trabajo se basa en el tratamiento de todas estas etapas desde la perspectiva de los Sistemas de Información Geoespacial (GIS, en sus siglas en inglés), adoptando estándares y ofreciendo las arquitecturas que recomiendan las organizaciones en dicho ámbito. De esta forma, los datos de sensores se consideran como cualquier otro tipo de contenido geoespacial.

En los últimos años, en el ámbito GIS, al igual que en muchas otras disciplinas del mundo de las tecnologías de la información y las comunicaciones (ICT, en sus siglas en inglés), se han movido desde aplicaciones de escritorio centralizadas, hacia Arquitecturas Orientadas a Servicios (SOA, en sus siglas en inglés) y distribuidas computacionalmente. Esta tendencia se sigue en las Infraestructuras de Datos Espaciales (SDI, en sus siglas en inglés) donde ofrece la posibilidad de acceder a recursos geoespaciales, además de estar distribuidos y ser heterogéneos (Masser, 2005).

Como se verá en posteriores capítulos, existen diferentes organizaciones que se encargan de ofrecer las herramientas necesarias para estandarizar la forma en la que se gestiona dicha información. Estas organizaciones siguen la arquitectura SOA. Para ello, establecen tres capas diferentes (Figura 1.3).

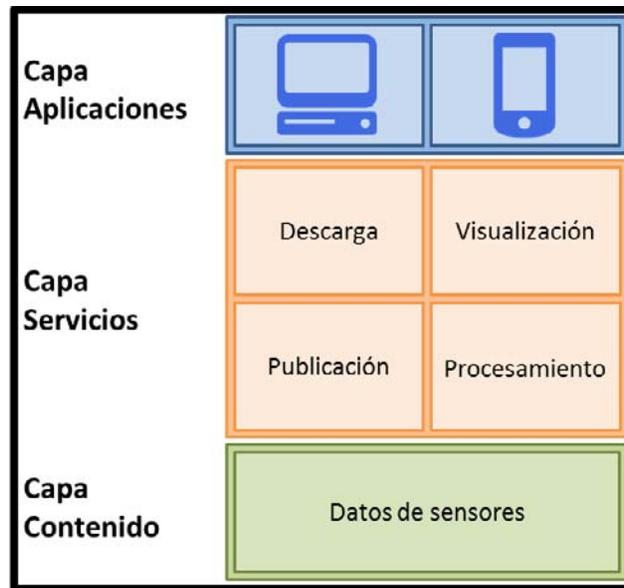


Figura 1.3: Arquitectura por capas.

En la capa superior residen los usuarios y las aplicaciones (clientes). Es en esta capa donde las aplicaciones y clientes consumen los servicios proporcionados por la capa inferior. Los servicios se encuentran en la capa media. Dichos servicios se categorizan siguiendo su funcionalidad, se proponen cuatro funciones diferentes que son: la descarga, la visualización, la publicación y el procesamiento. La capa de contenido se encuentra en la parte inferior, donde residen los datos en sí.

Una de las principales mejoras de la utilización en este tipo de arquitectura es

el aumento de la interoperabilidad. Esta es definida por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE), como la habilidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada. Uno de los objetivos de este trabajo es favorecer interoperabilidad entre los datos obtenidos por sensores, debido a que un sistema interoperable facilita la reusabilidad de los recursos que son ofrecidos mediante él (Granell y otros, 2010).

Siguiendo las etapas del ciclo de vida definido y las diferentes capas y funcionalidades previstas en la figura anterior (Figura 1.3), se pretende dotar de diferentes mecanismos que logren satisfacer las necesidades requeridas en cada una de estas etapas. La única etapa del ciclo de vida que no ha sido completada es la de descubrimiento, aunque ha sido incluida en la etapa de publicación.

Los primeros cuatro capítulos de la tesis definen las aportaciones para cada una de las etapas definidas en el ciclo de vida de los datos de sensores. La primera etapa diseña una plataforma genérica de sensorización para la construcción de nodos. Dichos nodos pueden medir una variedad de fenómenos usando sensores y dependiendo del uso en particular. La plataforma está pensada para que forme parte del IoT. Además, dicha plataforma se caracteriza por ser enérgicamente autónoma y estar construida tanto con hardware como con software libre.

Esta plataforma servirá como productor de las observaciones en crudo que posteriormente pasarán a ser analizadas para obtener algún tipo de resultado con valor añadido. Para ello, el segundo paso es la publicación de las observaciones, donde se presenta un mecanismo mediante un procesamiento estándar, que permite la publicación de contenido geoespacial entre estos datos de sensores y en los diferentes servicios estándares del ámbito GIS.

Tras la publicación, el siguiente paso es el acceso, en el cual se ofrecen varias interfaces para acceder a los datos en diferentes ambientes y usos particulares. Se define una interfaz para acceder a los datos de una forma ligera y estándar, que podrá ser usada en entornos con características computacionales más restringidas, además de utilizar el acceso mediante URI, siguiendo los paradigmas de IoT. También se utiliza una interfaz candidata a estándar para acceder a sensores siguiendo el IoT. Finalmente, se detalla una interfaz para el acceso a grandes cantidades de datos de sensores en tiempo real.

La última etapa en el ciclo previsto es el análisis de este tipo de contenido. Para ello se definen dos metodologías diferentes. La primera de ellas se utiliza para el

análisis de fuentes de sensores con altas frecuencias de actualización en tiempo real. Y la segunda metodología se define para la ejecución de cualquier tipo de modelo científico. Dicha metodología utiliza los estándares de procesamiento del ámbito GIS.

Finalmente se presenta un capítulo con los casos de uso donde se han utilizado cada una de las aportaciones detalladas con anterioridad.

1.2. Motivación

La motivación del actual trabajo, ha sido abarcar todo el ciclo de vida definido para los datos de sensores. Para ello, se han propuesto diferentes soluciones, todas ellas capaces de llevar a cabo sus objetivos de una forma eficaz, por lo que han sido consideradas diferentes motivaciones para cada uno de las etapas del ciclo de vida detallado.

En la etapa de adquisición, una de las motivaciones por las que se apostó por la creación de una plataforma de sensorización, fue que durante el análisis de los trabajos relacionados se detectó la carencia de una plataforma completamente genérica. De esta manera, el objetivo principal fue el diseño de una plataforma de sensorización preparada para cualquier tipo de propósito y capaz de ser extrapolable a cualquier ámbito de uso.

Otra de las motivaciones en esta etapa, fue cumplir con el hecho de que fuera energéticamente autónoma. De esta manera, se ha logrado una plataforma que no tenga ningún impedimento a la hora de ser instalada, siempre que se instale en un ambiente de exterior. Además, también se quería obtener una plataforma completamente libre, tanto a nivel hardware como software.

Ya en la etapa de publicación, el procesamiento presentado para dicho fin, tiene el objetivo de llenar una parte que es clave para el ciclo de cualquier tipo de contenido. Hoy en día, no hay un mecanismo sencillo para la publicación de datos geoespaciales ni tampoco de generación y publicación de metadatos, por lo que este mecanismo busca poner al alcance de todos una herramienta que ofrezca de forma fácil, la publicación de los datos, siendo en este caso datos de sensores en los diferentes servicios geoespaciales. Esto hace que realmente este proyecto presente un beneficio importante en el mundo GIS, además de ofrecer una interfaz conocida que ofrezca facilidad para ser utilizada.

En la fase de acceso, como ya se ha comentado anteriormente, existe una gran variedad de formatos y protocolos, lo que hace que exista una gran heterogeneidad a la hora de acceder a las redes de sensores. El trabajo presentado, tiene como motivación ofrecer las herramientas necesarias para integrar dichas fuentes heterogéneas y ofrecerlas mediante las normas y formatos GIS. Otro objetivo es facilitar el acceso de una forma ágil, además de proporcionar otros tipos de interfaces. Éstas son capaces de aligerar el uso de este tipo de datos para ser consumidos en los dispositivos con capacidades computacionales más restringidas, tales como los teléfonos móviles.

Además, otra de las motivaciones es ofrecer acceso a fuentes de datos con alta frecuencia de refresco en tiempo real. Dichas interfaces se caracterizan por ser más eficaces cuando las fuentes de datos producen un gran volumen de datos.

Finalmente, en la etapa de análisis se proponen dos metodologías diferentes para dos motivaciones diferenciadas. Cuando existen grandes cantidades de datos, al realizar análisis sobre ellos, no es posible abordarlo mediante las herramientas tradicionales (McAfee y Brynjolfsson, 2012). La primera metodología ofrece los mecanismos necesarios para analizar grandes cantidades de datos en tiempo real. La segunda metodología tiene como motivación poder utilizar modelos científicos en ambientes totalmente distribuidos, sin tener dependencia de ningún entorno particular. Para ello se utilizan estándares para el procesamiento, los cuales ofrecen la capacidad de reutilizar dichos modelos y su compartición.

1.3. Contexto

Los resultados extraídos de las investigaciones de la presente tesis doctoral han sido aportados a los siguientes proyectos:

Nueva generación de aplicaciones para la gestión de riesgos medioambientales: Gestión de incendios con Infraestructuras de Datos Espaciales con servicios de geoprocesamiento, visualización avanzada e integración de Servicios de la Web 2.0 Este proyecto es de ámbito local y tuvo una duración de 2 años. Sus objetivos se basaban en la explotación de servicios estándares para el procesamiento de modelos de predicción de riesgo de incendios. Fueron dos las aportaciones a a este proyecto. Por una lado, un geoprocesamiento estándar capaz de aplicar un modelo científico para la predicción de riesgos de incendios,

incluyendo dicha parte en la actual tesis. Y por otro lado, una aplicación móvil para delimitar las áreas quemadas tras un incendio.

Eurogeoss EuroGEOSS es un proyecto del Séptimo Programa Marco de la Comisión Europea, cuyo objetivo es la creación de un Sistema de Observación Ambiental de la Tierra que se desarrolla en las áreas temáticas de Cubierta Forestal, Sequía y Biodiversidad. Este proyecto cumple los requisitos de *Infrastructure for Spatial Information in Europe* (INSPIRE) y *Shared Environmental Information System* (SEIS), y las normas que aseguran la coherencia entre *Global Earth Observation System of Systems* (GEOSS), INSPIRE y *Global Monitoring for Environment and Security* (GMES) además de desarrollarse basándose en sistemas, recursos y servicios ya existentes o que tengan previsto su lanzamiento en breve en Europa. El proyecto persigue la interoperabilidad multidisciplinar, es decir, que la información de diferentes disciplinas sea comparable y compatible, facilitando la toma de decisiones.

Este proyecto tiene más de seis millones de euros de presupuesto y en él, participan 22 organismos de 10 países distintos. Los socios españoles son la Universidad de Zaragoza, la Universitat Jaume I de Castellón, el Instituto Geográfico Nacional (IGN) a través del Centro Nacional de Información Geográfica (CNIG), la Confederación Hidrográfica del Ebro, el Consejo Superior de Investigaciones Científicas (CSIC) a través del Instituto Pirenaico de Ecología y el Observatorio de la Sostenibilidad de España. Se estructura en siete paquetes de trabajo, donde además de los tres temáticos, hay otros cuatro grupos dedicados a coordinación, interoperabilidad multidisciplinar, evaluación de beneficios y diseminación.

La contribución a este proyecto ha sido un proceso automatizado para la publicación de contenido y generación de metadatos cumpliendo la normativa INSPIRE.

España virtual Es un proyecto CENIT, subvencionado por el Centro para el Desarrollo Tecnológico Industrial (CDTI) dentro del programa Ingenio 2010, orientado a crear un puente entre el mundo geográfico y las tecnologías de Internet.

Durante los últimos años, el mundo ha vivido una revolución en la manera en que los ciudadanos hacen uso de las tecnologías de la información geoespacial y 3D. Los satélites de observación de la Tierra, Internet, los dispositivos móviles y las tecnologías 3D y de código abierto, han universalizado el acceso a esta

información, rompiendo las fronteras entre el mundo físico y el virtual.

Su objetivo es la definición de la arquitectura, protocolos y estándares de la Internet Geográfica, con un foco especial en la visualización 3D, mundos virtuales e interacción entre usuarios. España Virtual incluye aspectos semánticos y tecnologías para el procesamiento masivo y almacenamiento de datos geográficos.

España Virtual ha representado, en gran parte, la contribución de España a dicha revolución, aún en curso, creando la tecnología para unir dos mundos: el geoespacial y el de servicios de Internet, desarrollando los máximos avances de la mano de los principales actores nacionales en cada campo.

La contribución a este proyecto, relacionada con la tesis actual, fue un servicio para la publicación de contenido y metadatos geoespaciales.

SEnviro - Sense our Environment Se trata de un pequeño proyecto, de carácter local y con duración de 6 meses, financiado por el Instituto de Nuevas Imágenes de la Tecnología (INIT), mediante la convocatoria de Prueba de concepto de 2014. El proyecto consta de dos partes. En la primera de ellas, se creó un dispositivo de bajo coste que aglutine diferentes sensores medioambientales, con el cual se formará una red de sensores capaz de ofrecer las mediciones en tiempo real de forma inalámbrica.

Dicho dispositivo, se desarrolla mediante un microcontrolador, como Arduino, al cual se conectarán los diferentes sensores para cada uno de los componentes a monitorizar. Dichos componentes se dividen en dos categorías, meteorológicos y de calidad del aire. Ejemplos de sensores meteorológicos pueden ser: temperatura, humedad ambiente, viento, precipitación, etc. Y de calidad del aire podemos encontrar: gases (SO₂, NO, NO₂, CO o CO₂), partículas en suspensión o ruido. Para el acceso a las mediciones en tiempo real se utilizan tecnologías ágiles e interoperables, capaces de ser utilizadas en diferentes dispositivos y con limitación de recursos.

La segunda fase es la encargada de realizar varios análisis sobre los datos obtenidos en tiempo real. Un primer paso será la creación de modelos para mostrar el estado actual en cada momento. Posteriormente, se investigará la aplicación de este tipo de datos para la obtención de mapas de predicción espacial y variación temporal. Para ello, se usará la metodología de Aproximaciones Integradas Anidadas Laplace (INLA, en sus siglas en inglés). Con ella, se será capaz de modelar los datos obtenidos de los sensores mediante una metodología bayesiana. Ésta

permite la inclusión de cualquier tipo de covariable en el modelo, desde continuas (temperaturas, humedades, etc.) a categóricas (tipo de suelo, de vegetación, etc).

La presente tesis incluye también como aportación el primer objetivo del proyecto, que es el desarrollo de una plataforma de sensorización medioambiental, con las características de ser de bajo coste y energéticamente autónoma.

BD4Env - BigData for Environment Previamente llamado como DEaP - Digital Earth Platform, se trata de un proyecto institucional del Joint Research Center (JRC). Tiene como objetivo desarrollar componentes hacia una plataforma interactiva para el análisis de políticas y de toma de decisiones, basadas en la convergencia de servicios web, tecnologías de sensores y en los medios de comunicación social con el análisis de Big Data. Contribuye, de una forma profunda, a través de estudios piloto dirigidos a la tierra con perspectivas teóricas, métodos analíticos y actividades de previsión. Se desarrollan nuevas soluciones tecnológicas para la visualización, la experimentación y el análisis de datos, al tiempo que aborda también las consideraciones de privacidad.

La contribución a este proyecto, es una de las metodologías de análisis incluidas en la presente tesis, utilizada para el análisis de grandes cantidades de datos en tiempo real sobre datos provenientes de sensores.

1.4. Contribuciones y estructura de la tesis

Las contribuciones obtenidas del presente trabajo son presentadas a continuación. Para ello se han utilizado las etapas del ciclo de vida definidas anteriormente para agruparlas, que a su vez, se corresponden con los capítulos principales en los que se compone la tesis.

- *Adquisición.* El Capítulo 2 detalla la etapa de adquisición de los sensores. Como contribuciones en la etapa de adquisición, se presenta una propuesta de plataforma de sensorización, llamada *SEnviro*. Esta se caracteriza por:
 1. Ser capaz de tomar observaciones en varios escenarios, tales como, en las ciudades inteligentes, el medio ambiente, la salud o la agricultura, entre otros.
 2. Considerarse de bajo coste, energéticamente autónoma y construida tanto con hardware como con software libre.

3. Por estar formada por los paradigmas de la IoT, utilizando el protocolo IP.
- *Publicación.* El tercer capítulo (Capítulo 3) se centra en la etapa de publicación de los datos. Las aportaciones en la etapa de publicación se resumen en un procesamiento para la publicación de contenido geoespacial:
 1. Ayuda a publicar contenido geoespacial, entre ellos datos de sensores, en los diferentes servicios estándares.
 2. Posibilita la creación y publicación de metadatos sobre los datos publicados.
 3. Ofrece interfaces estándares, por lo que facilita su reutilización y uso en diferentes ambientes.
 - *Acceso.* El Capítulo 4 se encarga de diseñar varias interfaces para facilitar el acceso a las observaciones de los sensores. Se ofrecen diferentes formas de acceso, las cuales aportan los siguientes beneficios:
 1. Se ofrece una forma de acceder a los datos de sensores en entornos con características modestas como teléfonos móviles, sin renunciar a los estándares GIS, por lo que no afecta a la interoperabilidad.
 2. Está pensada para ofrecer un acceso a los datos de sensores siguiendo los principios de IoT y mediante conexión IP.
 3. Permite el acceso a sensores con altas frecuencias de actualización y ofrece flujos mediante diferentes protocolos.
 - *Análisis.* El Capítulo 5 se centra en la etapa de análisis de los datos. Como parte del análisis se presentan dos metodologías diferentes. La primera de ellas está diseñada para el análisis de grandes cantidades de datos en tiempo real, la cual tiene las siguientes contribuciones:
 1. Tiene la capacidad de conectar con fuentes de datos, independientemente del tipo del sensor.
 2. Posibilita el análisis de grandes cantidades de datos de sensores en tiempo real.
 3. Aporta un mecanismo para transformar los datos de sensores a formatos estándares.

4. Proporciona la capacidad de conectarse con diferentes protocolos mediante una solución de bróker.
5. Ofrece la capacidad de servir los resultados de los análisis mediante varios protocolos de conexión en tiempo real.

La segunda metodología utiliza procesamientos estándares para la ejecución de modelos científicos. Sus principales contribuciones son:

1. Encapsular cualquier modelo científico para ser ejecutado en un entorno distribuido y desacoplado de un ambiente particular.
2. Ofrecer la capacidad de reutilizar dichos modelos debido a que se usan estándares definidos para el procesamiento.

Los demás capítulos se estructuran de la siguiente manera. El Capítulo 6 recopila todos los casos de uso donde se han aplicado las contribuciones presentadas a la largo de la tesis. En primer lugar, la Sección 6.1 define un ejemplo de uso de la plataforma de sensorización diseñada en el capítulo de adquisición (Capítulo 2). La segunda sección (Sección 6.2), muestra un ejemplo de uso del proceso de publicación de contenido geoespacial y de la generación y publicación de metadatos. La Sección 6.3, muestra un ejemplo de integración de fuentes de sensores heterogéneas, las cuales se publican en los servicios estándares para su posterior disposición mediante las interfaces de acceso definidas en el Capítulo 4. La cuarta sección (Sección 6.4) propone una implementación tecnológica de la metodología de análisis de gran cantidad de datos en tiempo real, para la detección de anomalías sobre datos de sensores de calidad del aire. Finalmente, la Sección 6.5 utiliza la metodología para el procesamiento mediante estándares, diseñada en la Sección 5.4 para la aplicación de un modelo científico dedicado a la predicción del riego de incendio.

Finalmente, Capítulo 7, resume las principales aportaciones de esta tesis, presentando las conclusiones alcanzadas y donde se discuten las limitaciones y futuras líneas de investigación, que permitirán continuar con la investigación iniciada.

Adicionalmente, en el Anexo A se presentan las publicaciones más representativas derivadas del trabajo de investigación realizado en la presente tesis, las cuales han sido un referente externo y continuo de evaluación durante el transcurso de la misma.

Capítulo 2

Adquisición

El capítulo que empieza tiene como objetivo llenar el primer paso dentro del ciclo de vida de datos provenientes de sensores, que es la adquisición de las observaciones. Como principal aportación se presenta una solución mediante una plataforma genérica de sensorización. Dicha plataforma llamada en su forma inglesa como *Sense our ENVIRONment (SEnviro)* es capaz de adaptarse a cualquier propósito de monitorización de una forma rápida y económica. El capítulo está estructurado de la siguiente manera. La Sección 2.1 muestra una pequeña introducción a la adquisición de observaciones mediante redes de sensores. La segunda sección (Sección 2.2) detalla los conceptos necesarios para un correcto transcurso del capítulo. La Sección 2.3 presenta la aportación principal de este capítulo con la plataforma de sensorización *SEnviro*. La cuarta sección (Sección 2.4) realiza un estudio de los diferentes trabajos que actualmente se encuentran en la literatura. Finalmente, en la Sección 2.5 se detallan las conclusiones devenidas de este capítulo.

2.1. Introducción

Hoy en día, uno de los retos de la sociedad es conocer en cualquier momento qué está sucediendo alrededor de ella, y cómo esto puede afectar en su vida diaria. La creciente preocupación por el cambio climático, los desastres naturales, el calentamiento global o el aumento de epidemias, hacen que la observación del medio ambiente sea un aspecto importante tanto en los países desarrollados como en los en vía de desarrollo (ONeill y otros, 1997). Por estas razones,

existe una creciente demanda en el despliegue de dispositivos de sensorización (comúnmente llamados sensores) con la finalidad de proporcionar información actualizada sobre el estado del entorno.

Un sensor es un dispositivo que tiene la capacidad de cuantificar el estado en un entorno específico. De forma sencilla, el funcionamiento de un sensor es entrar en contacto con un tipo de señal cuantificable y producir un valor como salida. Los valores que obtiene son llamados observaciones, las cuales pueden estar asociadas a una posición geográfica y a un instante en el tiempo. Las observaciones pueden ser tanto cualitativas como cuantitativas. Cada sensor puede estar configurado para la toma de observaciones de una forma periódica.

En el mercado, existen varios tipos de sensores dependiendo de su construcción física (tecnología). Entre ellos están los mecánicos (para medir cantidades como posición, forma, velocidad, fuerza, presión, entre otros), eléctricos (para medir voltaje, corriente, carga y conductividad), magnéticos (para medir campo, flujo o permeabilidad magnética) o térmicos (para medir temperatura, flujo, conductividad o calor). Existen otros tipos de sensores más recientes, como son los acústicos, químicos, ópticos, de radiación, láser o de fibra óptica. También se puede considerar a los propios ciudadanos como sensores (Goodchild, 2007), ya que gracias a las redes sociales cada ciudadano se convierte en un productor de éstas.

Existen dos tipos de categorías de sensores dependiendo del entorno de uso, los in-situ y los de teledetección. La diferencia entre ellos, se debe a que los sensores in-situ están en contacto con el medio que se desea cuantificar (o cualificar) y los de teledetección lo hacen a distancia, como por ejemplo desde satélites. El presente estudio está centrado en el primer tipo de sensores, aunque el trabajo podría ser extrapolable a ambos tipos.

En la actualidad, los sensores están presentes en una gran variedad de escenarios. Dos de los motivos que han favorecido la asimilación de estos dispositivos son:

- El aumento de la capacidad de procesamiento de los microcontroladores y la reducción de su tamaño, sumado a la reducción de su precio. Este aspecto da lugar al contexto idóneo para permitir el uso de sensores en muchos campos en los que no podía ser aplicado anteriormente.
- El movimiento a favor del hardware libre. En los últimos años han apare-

cido varios proyectos que han puesto a disposición de la comunidad los esquemas eléctricos de los componentes, con los que cualquiera puede construirlos por si mismo, favoreciendo el uso de estos dispositivos. Uno de los proyectos más destacables es Arduino¹. Arduino es una plataforma con un microcontrolador de bajo coste, donde los usuarios pueden añadir y programar cualquier tipo de dispositivo de forma fácil. Este proyecto cuenta con una gran comunidad de usuarios y desarrolladores, la cual aporta múltiples fuentes de conocimiento y ejemplos a los que recurrir.

Se entiende como sensor un dispositivo simple capaz de cuantificar un único fenómeno o varios si están directamente relacionados. Un sensor o un grupo de ellos forman un nodo de sensorización, teniendo la característica en común de pertenecer a un mismo objeto. Por cada uno de los sensores de un nodo de sensorización se obtiene una serie de observaciones (Figura 2.1). En dicha figura (Figura 2.1), se pueden ver diferentes series de observaciones generadas por cada nodo de sensorización.

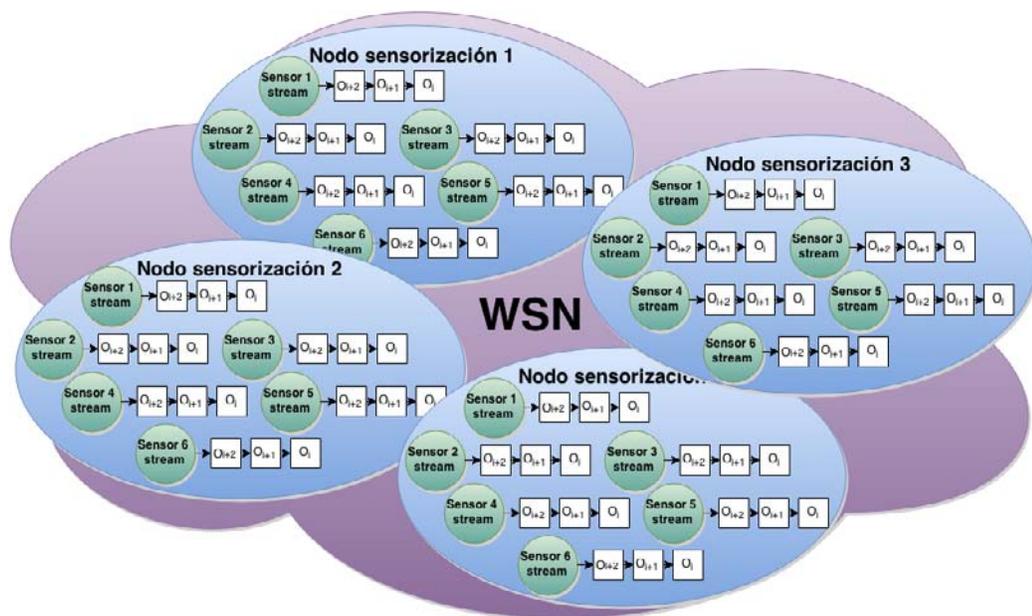


Figura 2.1: Diferentes nodos de sensorización que forman una red de sensores.

La unión mediante tecnologías de comunicación (inalámbricas) de varios nodos

¹<http://www.arduino.cc/>

de sensorización, construye una WSN con el objetivo de trabajar conjuntamente.

Para ello, se establecen diferentes estrategias o modos de conexión entre los diversos sensores. Las actuales WSNs se utilizan en muchos ámbitos, como pueden ser en la meteorología (temperatura, humedad, etc.), en la calidad del aire (CO₂, CO, PPM, entre otros) (Figura 2.2 a)), en la salud (pulso, presión arterial, etc.) (Figura 2.2 b)) y/o en las ciudades inteligentes (ocupación de plazas de aparcamiento o control de iluminación, entre otros) (Figura 2.2 c)). En la Sección 2.2.1 se detalla más profundamente el concepto de red de sensores.

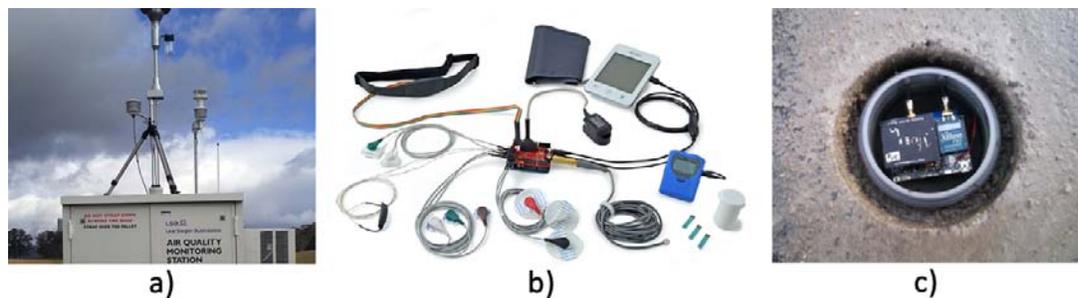


Figura 2.2: Diferentes nodos de sensorización para varios entornos. a) Nodo de calidad del aire. b) Nodo de monitorización de la salud. c) Nodo de parking inteligente.

Desafortunadamente, la mayoría de WSNs disponibles públicamente no ofrecen un acceso estándar a los datos que producen. Cuando una red de sensores no ofrece un acceso ni un formato estándar, se debe ofrecer un preprocesamiento particular para acceder a las fuentes de datos. Dicho preprocesado se encarga de extraer, transformar y cargar (ETL, en sus siglas en inglés) y permite a las organizaciones mover datos desde múltiples fuentes, además reformatearlos y limpiarlos. Para facilitar esta labor y reutilizar herramientas existentes, en este trabajo se definen diferentes categorías dependiendo del formato utilizado para las codificaciones de las fuentes de sensores. De esta forma se definen tres categorías: WSNs estructuradas, WSNs semiestructuradas y WSNs no estructuradas.

Ante el escenario descrito, se presenta como principal aportación de este capítulo, una plataforma de sensorización que será utilizada en la fase de adquisición de observaciones. Esta plataforma puede utilizarse para el estudio de diferentes fenómenos y para una gran diversidad de usos, como por ejemplo para fines medioambientales, ciudades inteligentes o en salud, entre otros. Esta platafor-

ma es llamada en su forma inglesa como *Sense our ENVIRONment* (*SEnviro*). Puede ser considerada como un nodo de sensorización de bajo coste, ya que está compuesta por elementos con un coste reducido. También está pensada para ser autónoma, por lo que no necesita alimentación de energía eléctrica y además, ofrece versatilidad para añadir o quitar sensores. Finalmente, *SEnviro* está pensada para integrarse con los paradigmas de IoT y la Web de las Cosas (WoT, en sus siglas en inglés).

2.2. Conceptos relacionados

En esta sección se definen los conceptos básicos para entender el trabajo definido en el presente capítulo. En la Sección 2.2.1 se detallan las nuevas tendencias para la creación de WSNs siguiendo el paradigma de IoT. En la Sección 2.2.2 se enumeran las diferentes opciones para la construcción de nodos de sensorización mediante plataformas de desarrollo basadas en microcontroladores.

2.2.1. Redes de sensores

Como se ha indicado en la introducción, las WSNs se componen de nodos de sensorización, los cuales a su vez pueden agrupar diferentes sensores. Cada uno de estos sensores es capaz de medir un fenómeno distinto. Existen varios fines por los cuales se usan las WSNs según Sohrawy y otros (2007), siendo estos de cuatro tipos diferentes: la recopilación de datos, el seguimiento, la vigilancia y la telemetría médica.

Cada nodo de sensorización es capaz de recoger las observaciones y transferirlas a otro nodo, pudiendo ser éste un nodo final o intermedio. Al igual que en las redes tradicionales, las redes de sensores tienen diferentes topologías, como son: la de estrella, la de árbol o la de malla (Figura 2.3) (Liu y Ning, 2003). Para construir estas topologías, deben definirse diferentes componentes dentro de la misma red. Además de los nodos de sensorización, las redes de sensores están compuestas de otros elementos, que son los enrutadores y los coordinadores (Faludi, 2010). El enrutador intercambia las observaciones entre los nodos y el coordinador tiene el control de la red.

En cuanto a los protocolos utilizados en las WSNs, estos se caracterizan por tener una alta heterogeneidad, existiendo soluciones propietarias y no propietarias.

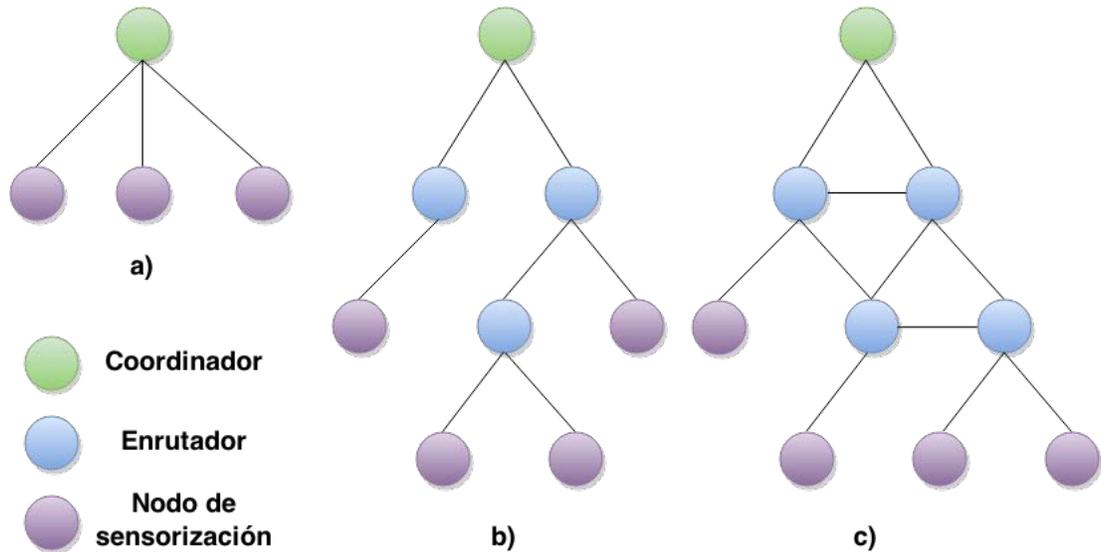


Figura 2.3: Diferentes tipologías para redes de sensores. a) Estrella. b) Árbol. c) Malla.

Tradicionalmente, las WSNs han sido basadas sobre protocolos propietarios y no seguían el protocolo de Internet (No-IP, en sus siglas en inglés), como son *Zigbee* (Alliance, 2008), *Z-wave* (Z-Wave, 2007) o *Insteon* (Darbee, 2005), entre otros.

Una de las principales dificultades en este ámbito es garantizar la interoperabilidad entre los diferentes protocolos (Zorzi y otros, 2010), cosa que es difícil de alcanzar con el uso de soluciones propietarias. Debido a esto, las nuevas tendencias en WSNs se centran en el uso de diferentes tecnologías de comunicación comunes en otros ámbitos (Buratti y otros, 2009) como son Bluetooth, Identificación por Radiofrecuencia (RFID, en sus siglas en inglés), Wireless Fidelity (Wi-Fi), datos móviles, etc. Estas tecnologías implican una conexión directa con los nodos de sensorización, con el fin de obtener las observaciones de forma inmediata. Además, el uso de estas tecnologías no descarta la posibilidad de comunicar con los diferentes nodos, con el fin de crear una variedad de topologías y estrategias de trabajo.

Estas tecnologías ofrecen una nueva posibilidad: el uso del protocolo de Internet (IP, en sus siglas en inglés) para la comunicación entre los nodos e Internet (Rodrigues y Neves, 2010). Este enfoque constituye el Internet de las Cosas (IoT, en sus siglas en inglés) (Kortuem y otros, 2010). IoT describe que en el mundo

real todas las cosas físicas pueden ser integradas en el mundo virtual de los bits y bytes. Este concepto ofrece la posibilidad de crear objetos inteligentes y formar redes de ellos conectados mediante el protocolo IP. Los nodos son conectados para construir WSNs mediante estándares abiertos y cada uno se identifica con una dirección IP diferente. De esta manera, los sensores pueden considerarse objetos inteligentes ya que permiten interactuar directamente con ellos (Kortuem y otros, 2010).

Según Bröring y otros (2012) cada nodo puede ser considerado un objeto inteligente que es capaz de sentir, comunicar y actuar. De esta forma, cada nodo está dotado de memoria y de la inteligencia suficiente para ser considerado como un objeto con cierta inteligencia. El término fue acuñado por primera vez en el trabajo de David Brock en 2001 (Uckelmann y otros, 2011). Posteriormente, se definió el término de la Web de las Cosas (WoT, en sus siglas en inglés) que describe la evolución de IoT y la integración de los objetos de IoT mediante los protocolos de la Web (Guinard y otros, 2009).

De esta manera, los protocolos Web existentes se utilizan como lenguaje común para la comunicación entre diferentes nodos. El protocolo de transferencia de hipertexto (HTTP, en sus siglas en inglés) se utiliza como capa de aplicación, en lugar de capa de transporte, tal y como es comúnmente utilizada en los servicios Web habituales. Cada nodo tiene asociada una URI a la que se puede acceder a través de las operaciones conocidas en HTTP (*GET*, *PUT*, *POST* y *DELETE*). Uno de los principales beneficios del uso de HTTP en las WSNs, es que permite el uso de arquitecturas de servicios Web, como por ejemplo la Transferencia de Estado Representacional (REST, en sus siglas en inglés) (Pautasso y Wilde, 2010). Fue Fielding y Taylor (2002) quien propuso utilizar el uso del paradigma REST para acceder a los nodos que siguen el paradigma de IoT. Además, los servicios REST son más adecuados en WSNs, donde los recursos computacionales son más limitados que en otros ámbitos más habituales (Yazar y Dunkels, 2009).

Comúnmente, hay dos métodos para acceder a la WSNs cuando se utiliza una interfaz REST, siendo estos el acceso directo y el indirecto (Guinard y otros, 2009). En el caso de acceso directo, cada nodo está conectado directamente a Internet. El acceso indirecto utiliza la red privada que realiza una función de pasarela.

2.2.2. Plataformas actuales basadas en microcontrolador

Una de las opciones más utilizadas para la creación de nodos de sensorización son las plataformas de desarrollo basadas en microcontroladores. La razón se debe a que son muy versátiles y permiten añadir cualquier tipo de sensor o módulo para diferentes propósitos (conectividad, almacenamiento, etc.). Además permiten realizar cualquier tipo de procesamiento sobre sus entradas con el fin de producir una salida. También, como anteriormente ya ha sido mencionado, permiten la reducción del coste y el incremento de la popularidad de este tipo de plataformas, propiciando la proliferación de varias alternativas. Actualmente existen diferentes opciones, siendo las más destacables las siguientes: Arduino, Raspberry Pi², BeagleBone³ y MSP430 Launchpad⁴. Las plataformas nombradas son completamente o parcialmente hardware libre. Seguidamente, se detalla cada una de ellas. La Tabla 2.1 realiza una comparación más detallada.

	<i>Arduino UNO</i>	<i>Raspberry Pi</i>	<i>BeagleBone</i>	<i>MSP430 Launchpad</i>
Modelo	R3	B	A5	1.5
Microcontrolador	ATmega328	ARM11	ARMCortex-A8	TI M430G2553
Arquitectura	8 Bit	32 Bit	32 Bit	16 Bit
Velocidad	16MHz	700MHz	700MHz	16MHz
RAM	2 KB	256MB	256MB	512 B
Flash	32KB	SD	4GB	16KB
Min. power	42mA	700mA	700mA	0.5uA
Entradas dig.	14	8	66	8
Entradas anal.	6	N/A	7	8
Ethernet	No	Yes	Yes	No
Leng. de prog.	Wiring-based	Python, C y Basic	Python y C	C/C++
IDE	Arduino tool	IDLE, Scratch, Linux	Python, Scratch, Linux	IAR Embedded
Coste	\$29.95	\$35.00	\$199.95	\$4.30
Hardware libre	Completamente	Parcialmente	Completamente	Completamente

Tabla 2.1: Comparativa entra las diferentes plataformas de desarrollo basadas en microcontrolador.

- *Arduino*. Es una plataforma completamente diseñada como hardware libre. Está diseñado para leer sobre sus entradas, realizar procesamientos no muy pesados y producir una salida. En la actualidad, existen una gran variedad de

²<https://www.raspberrypi.org/>

³<http://beagleboard.org/>

⁴<http://www.ti.com/>

sensores y actuadores que son compatibles con Arduino. La principal ventaja frente a otras plataformas, es la gran cantidad de recursos disponibles, tanto a nivel de hardware como de software. Otra ventaja es el bajo consumo que ofrece, ya que el nivel de procesamiento es más restringido si se compara con sus rivales. Arduino es la plataforma más utilizada y es empleada en muchas aplicaciones distintas. Existen varias versiones de Arduino con diferentes características.

- *Raspberry Pi*. Es una plataforma considerada parcialmente hardware libre. Está diseñada para ofrecer más procesamiento que Arduino, estando más enfocada a aplicaciones multimedia ya que lleva un conector de Interfaz Multimedia de Alta Definición (HDMI, en sus siglas en inglés). También ofrece la posibilidad de añadir sensores gracias a las entradas, llamadas entradas y salidas de propósito general (GPIO, en sus siglas en inglés). Raspberry funciona con Linux, lo que favorece el desarrollo de aplicaciones. Esta plataforma es más cara que Arduino y tiene un mayor consumo energético.
- *BeagleBone*. Es una plataforma completamente hardware libre y tiene menor popularidad que las anteriores. Es similar a la Raspberry Pi ya que comparten muchas características, tanto en términos de procesamiento como en conectores de entrada y salida. Tanto BeagleBone como Raspberry Pi son programadas con niveles más altos de abstracción en comparación con Arduino.
- *MSP430 Launchpad*. Es una plataforma basada en un microcontrolador similar a Arduino. Sus características son más limitadas que la Raspberry Pi o la BeagleBone. Al igual que Arduino, la MSP430 está diseñada para aplicaciones donde el procesamiento de datos no es muy elevado. La diferencia con Arduino es que la MSP430 no tiene la misma comunidad de usuarios y desarrolladores, por lo tanto no existe gran variedad de complementos para dicha plataforma.

Para el caso de uso presentado en la Sección 6.1, se ha elegido Arduino UNO como plataforma. Si comparamos Arduino UNO con las otras plataformas, como son las Raspberry Pi o la BeagleBone, Arduino UNO es más apropiado para trabajar de forma energéticamente autónoma, ya que tiene un menor consumo.

Además, para un escenario donde únicamente se requiera la recogida de observaciones y su posterior envío, no es necesario un alto poder de procesamiento. Otra de las razones por la que se ha escogido Arduino UNO, a diferencia de la MSP430 Launchpad, es que tiene gran cantidad de placas de expansión (shields en inglés) que facilitan su desarrollo y su uso.

2.3. La plataforma SEnviro

En esta sección se presenta la plataforma de sensorización diseñada para la adquisición de datos sensoriales. La Sección 2.3.1 empieza con una pequeña vista general de la plataforma. Dicha sección (Sección 2.3.2) muestra de manera conceptual el diseño del hardware necesario para el desarrollo de la plataforma. Finalmente, la Sección 2.3.3 detalla cuál es el comportamiento a nivel de software de la plataforma.

2.3.1. Visión general de SEnviro

SEnviro tiene como objetivo ser una plataforma de bajo coste, libre y energéticamente autónoma. Cada nodo *SEnviro* utiliza una plataforma basada en microcontroladores a modo de núcleo con el fin de agrupar varios sensores, capaces de acoplarse y desacoplarse de una forma sencilla. La plataforma *SEnviro* sigue los paradigmas de IoT y WoT. Para ello utiliza el protocolo IP como comunicación entre diferentes nodos. Con estas características se puede considerar que cada nodo de *SEnviro* es un objeto inteligente. Cada uno de estos nodos es llamado *SEnviro Thing*, donde una red *SEnviro* está formada por varios *SEnviro Things*.

La plataforma propuesta conforma una solución barata, ya que está pensada para ser construida mediante componentes y sensores de bajo coste. La plataforma *SEnviro* puede considerarse abierta porque utiliza tanto hardware libre como software libre y no añade ninguna restricción de licencias. Otra característica importante de esta plataforma es que es energéticamente autónoma, ya que cuenta con una batería propia. Además, es una placa solar la que se encarga de proporcionar la energía necesaria para recargar la batería.

Para ofrecer una plataforma autónoma energéticamente, cada *SEnviro Thing* incorpora una batería y un panel solar para proporcionar alimentación en todo momento. De esta forma se ofrece suficiente comodidad a la hora de la instalación

de los nodos, sin que haya dependencia de darle corriente directa de suministro eléctrico.

Un ejemplo de desarrollo completo de *SEnviro* se describe en detalle en la sección 6.1. En ella, se presenta un ejemplo de *SEnviro Thing* tanto a nivel hardware como software. Para este primer caso de uso, se ha creado un prototipo de *SEnviro Thing* para la monitorización del medioambiente, añadiendo sensores ambientales, como son: temperatura, humedad o CO₂, entre otros. Los nodos cuentan con un módulo Wi-Fi con el fin de enviar las observaciones para su posterior publicación, acceso y análisis.

2.3.2. Diseño de un *SEnviro Thing*

Cada *SEnviro Thing* ha sido diseñado para ser un nodo de sensorización que actúe como un objeto inteligente y proporcione observaciones in-situ. Cada uno de ellos está formado por diferentes componentes, los cuales están organizados en cuatro grupos dependiendo de su funcionalidad: *Núcleo*, *Sensores*, *Alimentación* y *Comunicación* (Figura 2.4).

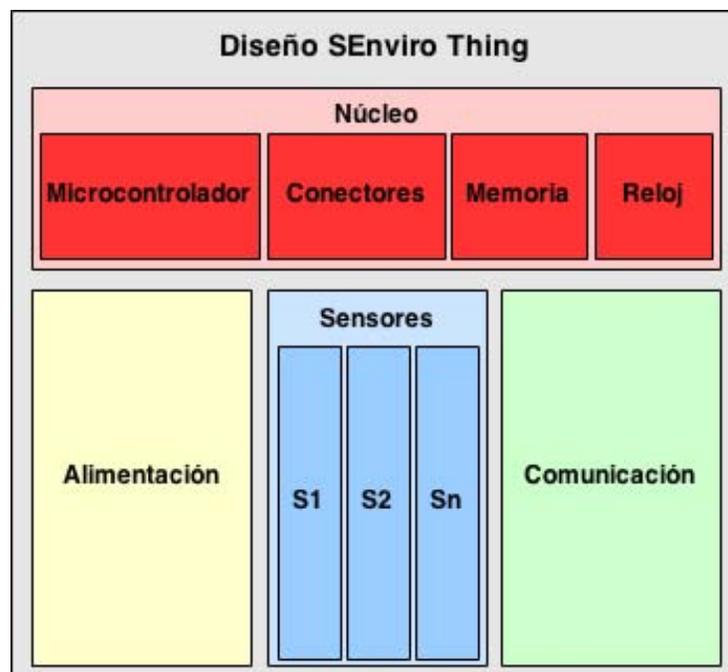


Figura 2.4: Diseño conceptual de un nodo de *SEnviro Thing*.

El *Núcleo* de *SEnviro Thing* es el encargado de recoger, almacenar y enviar las

observaciones provenientes de los diferentes sensores. La estructura del *Núcleo* proporciona las interfaces de hardware y protocolos de comunicación adecuados para conectar con los sensores. A su vez, el *Núcleo* está formado por cuatro partes diferentes. Seguidamente se detallan cada una de ellas:

- El *Microcontrolador* es el componente más importante del *Núcleo*, ya que procesa los datos provenientes de los sensores y realiza las diferentes interacciones con cualquiera de los componentes integrados en el nodo.
- Los *Conectores* ofrecen varios conectores para añadir de forma fácil y rápida los diferentes *Sensores* al *Núcleo*. Para ello, ofrecen diferentes tipos de interfaces y así poder garantizar la interoperabilidad entre los *Sensores* y el *Núcleo*.
- Además, cada *SEnviro Thing* incluye la capacidad de almacenar información para diversos fines. Uno de los principales objetivos es que cada *SEnviro Thing* debe conocer su estado actual, por lo que necesita una *Memoria* para almacenar su propio estado. Por otra parte, la *Memoria* se puede utilizar para guardar diferentes tipos de datos, como por ejemplos las observaciones obtenidas por los *Sensores* en de caso que no exista conexión estable.
- Cada *SEnviro Thing* necesita, además, conocer la fecha y hora para ser asociado a su estado. Para este propósito se incluye un *Reloj* en cada uno de ellos. Por ejemplo, éste podría ser necesario para etiquetar cada una de las observaciones a la fecha y hora en las que se han obtenido.

El *Núcleo* proporciona la base adecuada para construir un sistema en el que los *Sensores* pueden ser acoplados y desacoplados en función de los fenómenos que se quieran medir en cada momento y para cada uso en particular. Como se ha indicado, el *Núcleo* ofrece conectores interoperables, que facilitan conectar y desconectar los *Sensores* de la plataforma de una forma fácil.

Con el fin de dotar de conectividad a cada *SEnviro Thing* se ha incluido un módulo de *Comunicación*. Cada *SEnviro Thing* debe utilizar un canal de comunicación, ya sea para comunicar con otros *SEnviro Thing* o para comunicar con un coordinador central (servidor).

Para tal propósito, el módulo de comunicación proporciona al *Núcleo* la capacidad de enviar y recibir datos al exterior. Para cumplir con los principios de IoT, se

proporciona una comunicación mediante el protocolo IP. Este módulo ofrece la posibilidad de intercambiar el tipo de comunicación de una forma fácil y ofrecer otros tipos de comunicación sin necesidad de cambios en el diseño.

2.3.3. El comportamiento de SEnviro Thing

Como se ha indicado, cada *SEnviro Thing* ha sido diseñado para ser modular tanto a nivel hardware como de software. Con el fin de proporcionar una fácil actualización de los componentes y que sea además personalizable dependiendo del uso particular, *SEnviro* permite realizarlo con muy poco esfuerzo. De esta manera, se asegura que el *SEnviro Thing* continuará trabajando cada vez que se añada o se quite un nuevo sensor. De esta forma, *SEnviro Thing* es muy versátil, ya que se pueden adaptar fácilmente a diferentes escenarios y ámbitos. El nodo es capaz de cambiar su comportamiento, elegir qué sensores quieren que se consulten y con qué frecuencia, sin la necesidad de volver a programarlo. La Figura 2.5 muestra cuál es el flujo de trabajo que realiza un *SEnviro Thing*.

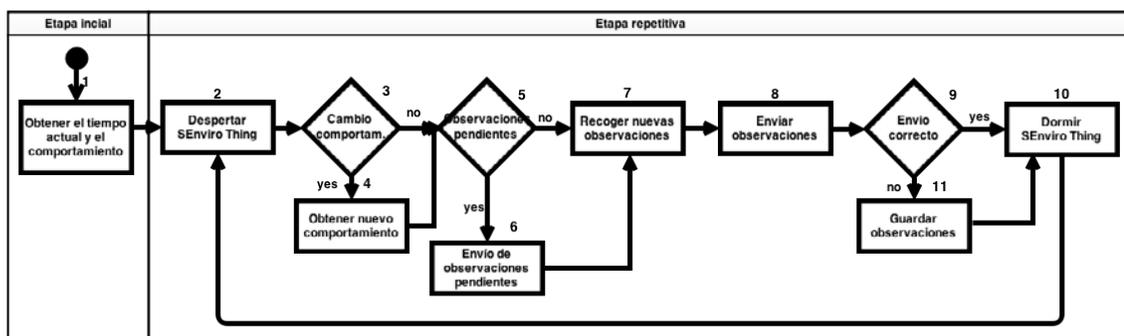


Figura 2.5: Diagrama del comportamiento de un *SEnviro Thing*.

El comportamiento de un *SEnviro Thing* está diferenciado en dos etapas, una *inicial* y una *repetitiva*. En la etapa *inicial*, se ejecutan un conjunto de métodos necesarios para la correcta inicialización del nodo, por ejemplo, un método para inicializar el *Reloj* (Figura 2.5 1)) con la hora actual.

En la etapa *bucle* se definen los métodos que deben repetirse durante el ciclo de vida de cada *SEnviro Thing*. Éste se puede dividir en cinco etapas diferentes, las cuales son: a) El *SEnviro Thing* se despierta de un período de latencia (Figura 2.5 2)); b) se comprueba si debe modificar su comportamiento (Figura 2.5 3)). En caso afirmativo, se cambia su comportamiento (Figura 2.5 4)); c) se comprueba si hay

observaciones pendientes en la memoria que previamente no se habían enviado (Figura 2.5 5)). Si existen dichas observaciones, se realiza el envío (Figura 2.5 6)); d) se obtienen las nuevas observaciones de cada uno de los sensores activos (Figura 2.5 7)) y se tratan de enviar dichas observaciones (Figura 2.5 8)); y e) después de tratar de enviar las observaciones (Figura 2.5 9)), se vuelve a dormir el nodo (Figura 2.5 10)). Si hay algún problema al enviar las observaciones, se guardan en la *Memoria del SEnviro Thing* (Figura 2.5 11)) antes de dormirlo. Estas observaciones se intentarán volver a enviar en los siguientes ciclos. Finalmente, el ciclo de vida del *SEnviro Thing* comienza de nuevo cuando se vuelve a despertar (Figura 2.5 2)).

2.4. Trabajos relacionados para la adquisición

En la literatura, existen varios trabajos similares a la plataforma de sensorización *SEnviro* que se presenta en este capítulo, como parte de la adquisición de datos provenientes de sensores. Para ello, se ha realizado un estudio de las principales aportaciones. La siguiente lista muestra una pequeña descripción de cada uno de ellos. Como característica común en estos, es que todos utilizan hardware libre.

- El proyecto *SenseBox* (Bröring y otros, 2012) propone que ciertos objetos del mundo real puedan ser utilizados a través de la Web. Los autores presentan un caso de uso donde se utiliza una plataforma con un microcontrolador Intel Atom conectado a un Arduino. La función del Arduino consiste en conectar los sensores y la plataforma del Intel Atom. El objetivo del trabajo es contar la carga de tráfico en una carretera.
- El artículo Mendez y otros (2011) presenta una red de sensores en la que utilizan la plataforma Arduino. Cada Arduino tiene instalado un módulo Bluetooth. La plataforma ofrece observaciones para diferentes fenómenos como CO, CO₂, temperatura y humedad. Un operario equipado con un teléfono móvil se conecta mediante Bluetooth para recoger las observaciones y enviarlas a un servidor central.
- Los autores en Ferdoush y Li (2014) también proponen una red de sensores usando la plataforma Arduino mediante una conexión Zigbee. Cada plataforma incluye dos sensores diferentes que miden temperatura y humedad. Una Raspberry Pi recoge las observaciones y las publica mediante

Internet. También se presenta un cliente Web para la visualización de las observaciones.

- En Abraham y Pandian (2013) se presenta una solución de bajo coste utilizando la plataforma Arduino. Cada Arduino está equipado con varios sensores, que miden fenómenos CO, CO₂, hidrógeno, metano o ruido. Además, cada plataforma incorpora un Sistema de Posicionamiento Global (GPS, en sus siglas en inglés) para añadir la situación geográfica. Los autores no especifican qué tipo de conexión inalámbrica utilizan.
- En Baker (2014) se presenta un prototipo de plataforma de sensorización con una placa Arduino Mega. Dicha plataforma ofrece temperatura y humedad, recopilando los datos en intervalos de 5 minutos. Aunque el dispositivo está equipado con una tarjeta digital donde se almacenan las observaciones, se puede trabajar en tiempo real mediante una conexión Ethernet. Cuenta con baterías que duran hasta 8 horas.
- Una red de sensores se diseña en Dines y otros (2012) la cual utiliza también la plataforma Arduino. Se realizan varias pruebas utilizando dos configuraciones diferentes de topología de red. La primera configuración utiliza ZigBee y la otra Wi-Fi. En el caso de la configuración Wi-Fi, la tipología empleada es en estrella centralizada por un punto de acceso inalámbrico (WAP, en sus siglas en inglés).
- En Davidovic y otros (2013) se presenta una solución en la que los usuarios son los responsables de la recogida de las observaciones, siguiendo técnicas de *crowdsourcing*. Cada nodo de la red proporciona datos de temperatura y humedad. El sistema utiliza la plataforma Arduino, con la cual se accede mediante un módulo Bluetooth. Los usuarios pueden obtener las observaciones utilizando la conexión Bluetooth de sus teléfonos móviles. Cuando un usuario recibe una observación, se envía a un servidor y a continuación éstas se publican, siendo accesibles mediante un servidor OGC SOS. Cada observación se asocia a las coordenadas del teléfono móvil en el instante en que se recibe la medida.
- El trabajo presentado en Gaddam y otros (2014) muestra una red de sensores para la monitorización de la temperatura, la humedad del ambiente y la

del suelo. Diferentes nodos se conectan a través de un coordinador ZigBee, que a su vez, está conectado a un ordenador con conexión a Internet.

- Los autores en Voigt y otros (2003) presentan el desarrollo de una plataforma de sensorización autónoma utilizando un MSP430 Launchpad. Dicha plataforma está equipada con un acelerómetro, un sensor de luz, uno de Infrarrojos Pasivo (PIR, en sus siglas en inglés), uno de temperatura y un micrófono. Este sistema se caracteriza por tener un panel solar que siempre mantiene la batería cargada.
- En el último trabajo analizado Demuth (2012) se detalla un prototipo de plataforma para la monitorización agrícola. Los autores utilizan un Arduino Mega equipado con GPS. Además, se añade un módulo Wi-Fi a través del puerto Ethernet. La plataforma ofrece diferentes sensores que recogen medidas de temperatura, humedad del suelo y/o luz. Este trabajo es especialmente relevante en el contexto del trabajo presentado, ya que sigue los paradigmas de IoT.

Con el fin de comparar los trabajos anteriormente revisados, la Tabla 2.2 muestra una comparación entre ellos. Se proponen las siguientes características para ser analizadas en cada trabajo:

- *Plataforma*. Hace referencia al tipo de plataforma utilizada, basada en un microcontrolador.
- *Tiempo real*. Indica si trabaja en tiempo real. Escala: Sí/No.
- *Conexión*. Se refiere al tipo de conexión que utiliza. Escala: Wi-Fi, Bluetooth, ZigBee, entre otros.
- *Fenómenos*. Se refiere al tipo de fenómenos que puede detectar. Escala: Temperatura, Humedad, Partículas, Barómetro o Ruido, entre otros.
- *Coste*. Indica el coste en euros del desarrollo de la plataforma. Escala con Euros.
- *IoT-WoT*. Se refiere a si el proyecto sigue los paradigmas de IoT y WoT. Escala: Sí/No.

Tabla 2.2: Comparación entre los diferentes trabajos analizados.

Referencia	Plataforma	Tiempo real	Conexión	Fenómenos	Coste	IoT-WoT
Bröring y otros (2012)	Arduino y Intel Atom	Si	UMTS-3G USB	N/A ^a	375€ sensores)	Si
Mendez y otros (2011)	Arduino	No	Bluetooth	CO, CO ₂ , Temp. y Hum.	N/A	No
Ferdoush y Li (2014)	Arduino y Raspberry	Si	ZigBee/Ethernet	Temp. y Hum.	N/A	No
Abraham y Pandian (2013)	Arduino	Si	N/A	CO, CO ₂ , Hidrogeno, Metano y Ruido	N/A	No
Baker (2014)	Arduino Mega	No	Ethernet	Temp. y Hum.	N/A	No
Dines y otros (2012)	Arduino Uno	Si	Wi-Fi	Temp., Hum., Bar. y Gases	820€	No
Davidovic y otros (2013)	Arduino Uno	Si	Bluetooth	Temp. and Hum.	N/A	No
Gaddam y otros (2014)	Arduino Uno	Si	ZigBee	Temp., Hum. y Hum. suelo	N/A	No
Voigt y otros (2003)	Texas Inst. MSP430	N/A	Not specified	Temp., Hum., PIR, Ruido y Accel.	N/A	No
Demuth (2012)	Arduino Mega	Si	Ethernet	Temp., Hum. suelo y luz	185€	Si
SEnviro	Arduino UNO	Si	Wi-Fi	Temp., Hum., Particulas, Bar., Ruido, Gases, Luz, Pluviómetro y Anemómetro	286.28€	Yes

^aN/A: No especificado.

Seguidamente se comparan las diferentes aportaciones analizadas con el trabajo presentado en este capítulo. Para ello, se utilizan las mismas características definidas para detallar los trabajos analizados en la Tabla 2.2. Al igual que la actual aportación, la mayoría de los trabajos analizados utilizan la plataforma Arduino; como ya se ha dicho anteriormente, esto se debe a que Arduino tiene una mayor aceptación entre los usuarios, porque dispone de una gran variedad de recursos libres que facilitan su uso, además de ofrecer una gran variedad de sensores diferentes.

Muchos de los trabajos analizados, al igual que esta aportación, ofrecen las observaciones en tiempo real. Esto es posible gracias a que la plataforma de sensorización presentada está provista de conexión a Internet utilizando el protocolo IP. *SEnviro* es la plataforma que mayor número de fenómenos ofrece entre todas las analizadas. Como se verá en el Capítulo 6, *SEnviro* es capaz de monitorizar diferentes fenómenos, tanto climáticos (temperatura, humedad, dirección y velocidad del viento o cantidad de lluvia) como de calidad del aire (ruido, gases, luminosidad y partículas).

En cuanto al coste de la plataforma, *SEnviro* es más barato que otros proyectos, ya que tiene un coste por nodo de 286,28 €. La plataforma presentada en Demuth (2012) es menos costosa, pero no incluye los sensores para los diferentes fenómenos. Otra característica que se analiza, es la comparación del consumo de energía, pero sólo uno de los trabajos analizados (Dines y otros, 2012) ofrece dicha información y su consumo es de 268 mA sin sensores. *SEnviro* tiene un mejor consumo energético, ya que únicamente consume 90 mA incluyendo todos los componentes.

Sólo dos proyectos analizados (Bröring y otros, 2012; Demuth, 2012) siguen los paradigmas de IoT y WoT. *SEnviro* sigue dichos paradigmas y ofrece una interfaz REST para ello. Como se verá en el Capítulo 6, el presente proyecto tiene como objetivo ser interoperable y, para este propósito, se ha elegido un estándar con el fin de acceder a cada *SEnviro Thing*.

2.5. Conclusiones

Este capítulo tiene como objetivo la creación de una plataforma de sensorización para la toma de observaciones. Esta plataforma constituye la aportación para la adquisición dentro del flujo de trabajo de los datos provenientes de sensores. Dicha

plataforma de sensorización se llama *SEnviro* y cumple dos objetivos principales: (1) diseñar una plataforma mediante una arquitectura basada en hardware libre y (2) que sea totalmente desacoplada a un caso de uso en concreto, pudiéndose extrapolar a cualquier ámbito de monitorización.

La plataforma *SEnviro* utiliza hardware libre ofreciendo una gran variedad de beneficios. Por una parte, la plataforma se beneficia de tener una gran comunidad de desarrolladores y facilita el uso del hardware utilizado. Además, permite opciones de expansión, ya que hay muchos componentes compatibles, debido a que los esquemas están totalmente disponibles. Otra ventaja de utilizar hardware libre es el bajo coste de estos componentes, por el hecho de existir muchos fabricantes que proporcionan el mismo componente. Por otra parte, ofrece la posibilidad de replicar la solución, ya que el hardware es completamente libre. Esto a su vez, retroalimenta la propuesta permitiendo a los nuevos usuarios sumar sus aportaciones y hacer crecer la plataforma *SEnviro*.

SEnviro ofrece una conexión fácil a la hora de actualizarlo con nuevos sensores tanto en términos de hardware como de software. A nivel de hardware, ofrece una conexión *plug and play* gracias a un *shield* de expansión. Y a nivel de software, *SEnviro* desarrolla un *Núcleo*, con el que añadir o quitar nuevos *Sensores* que puede ser abordado con poco esfuerzo.

Con el objetivo de evitar cualquier tipo de restricción en el despliegue de los *SEnviro Things*, uno de los retos de este proyecto, es ofrecer una plataforma capaz de ser completamente autónoma energéticamente hablando. Otro punto a tener en cuenta es la capacidad del propio nodo para cambiar su comportamiento mientras está trabajando. Esto evita el reemplazo de los nodos para su posterior reprogramación, cada vez que se quiera cambiar la frecuencia de observaciones o a la hora de elegir qué *Sensores* utilizar.

Capítulo 3

Publicación

Tras la propuesta de un diseño de plataforma de sensorización (Capítulo 2), como fuente de producción de datos provenientes de sensores, este tercer capítulo tiene como objetivo alcanzar un segundo paso dentro del ciclo de vida definido en la Introducción (Capítulo 1), que es la publicación. Para ello, se detalla un mecanismo genérico para la publicación de datos geospaciales, entre los que se incluyen los datos adquiridos por sensores. Además de la publicación de los propios datos, también se contempla la generación y publicación de metadatos. Para alcanzar los dos objetivos, se presenta un procesamiento geoespacial para la publicación llamado *Geoss Service Factory* (GSF). El capítulo se compone de la siguiente manera. La primera sección (Sección 3.1) muestra una pequeña introducción sobre la publicación de datos geospaciales, además de la creación y publicación de metadatos. La segunda sección (Sección 3.2) detalla los conceptos necesarios para un correcto entendimiento del capítulo. La Sección 3.3 presenta la aportación principal de este capítulo con el diseño del procesamiento GSF. La cuarta sección (Sección 3.4) realiza un estudio de los diferentes trabajos que existen en la literatura, relacionados con la contribución presentada. Finalmente, en la Sección 3.5 se detallan las conclusiones devenidas de este capítulo.

3.1. Introducción

Tras la adquisición de las observaciones provenientes de los propios sensores, el flujo natural es su publicación, con el fin de una posterior difusión. Para su publicación es necesario transformar las observaciones, de modo que sigan alguno

de los formatos de codificación o normas existentes. Estos se llaman estándares y son producto de diferentes organizaciones, algunas solo para uso interno, otras para uso por grupos de gente, grupos de compañías, o una subsección de una industria. En el campo de los sensores, mayoritariamente para la definición de las observaciones se utilizan formatos propietarios dependiendo del uso en particular, definidos por los fabricantes de los sensores o por las propias agencias que los utilizan (Arnaboldi y otros, 2013).

La carencia de un estándar común, capaz de aglutinar diferentes propósitos y que sea establecido de forma universal, conlleva un serio problema para el consumo de cualquier tipo de información, siendo igual en los datos de sensores (Botts y otros, 2008). Para solucionar dicho inconveniente, en las Infraestructuras de Información Geoespacial (GII, en sus siglas en inglés), existen varias propuestas de estandarización para los diferentes propósitos, en los cuales se encuentran los datos de sensores. Estos estándares tienen como objetivo homogeneizar tanto el formato de los propios datos, como la forma de ofrecerlos (Botts y otros, 2008).

La tendencia actual en GII es la de implementar y organizar los datos geoespaciales, entre ellos los datos de sensores, dentro de lo que se conoce como SDI (Masser, 2005). Las SDIs en Europa tienen un movimiento (directiva) reconocido, llamado INSPIRE (Directive, 2007). INSPIRE propone un marco general para las SDIs a efectos de las políticas ambientales europeas. Dicha directiva contempla todo lo relacionado con datos ambientales, agrupados en 34 temas diferentes, entre los que se encuentran por ejemplo: las redes de transporte, la ocupación del suelo o la hidrografía. Este movimiento puede considerarse como la contribución Europea a GEOSS. GEOSS tiene como propósito la observación íntegra de la Tierra, para la mejora del seguimiento y la predicción de su comportamiento.

Como se ha comentado anteriormente este trabajo contempla, además de la publicación de datos geoespaciales, la generación y publicación de los metadatos que se describen a los datos. En el contexto de la directiva INSPIRE, tanto la creación como el mantenimiento de los metadatos que se generen, deben ser publicados en los servicios de descubrimiento (Discovery) (Craglia y otros, 2007). De acuerdo con la nombrada directiva, por cada dato que se publique se ha de generar y publicar un metadato. Dichos metadatos deben ser validados y descritos de forma adecuada y precisa (Zarazaga-Soria y otros, 2003).

Actualmente, existe una gran cantidad de datos geoespaciales públicos, pero no todos ellos tienen asociados el metadato que los describe, por lo que su

descubrimiento no es una tarea fácil (Nogueras-Iso y otros, 2005). La falta de metadatos se debe a la dificultad en su creación. Por una parte, hacerlo de una forma manual resulta un proceso laborioso debido a la complejidad de los estándares utilizados para generar los metadatos. Y por otra parte, se debe a la carencia de mecanismos para automatizar la generación de los metadatos en el mismo momento en que se genera el dato, siendo este el mejor momento para su creación y publicación (Zarazaga-Soria y otros, 2003). Desafortunadamente, en la actualidad no existe ningún mecanismo automatizado para la creación de metadatos que no requiera usuarios expertos para su correcto funcionamiento (Díaz y Schade, 2011).

En este tercer capítulo se presenta un servicio de publicación llamado GSF (Díaz y Schade, 2011), mediante el cual se quiere dar solución a la publicación de datos y metadatos. Dicha herramienta se caracteriza por ser genérica y ofrecer una interfaz estándar. Está pensada para ser empleada por usuarios tanto expertos en el dominio geoespacial, como usuarios no experimentados.

GSF en comparación con las herramientas de publicación actuales, se implementa mediante una interfaz estándar y asiste en la publicación de diferentes tipos de datos en los casos de servicios basados en estándares existentes. GSF oculta la complejidad al usuario para la publicación de nuevo contenido y, por lo tanto, facilita el intercambio de ellos, sin olvidar el cumplimiento de los estándares con el fin de garantizar la interoperabilidad. Las características principales de GSF son: 1) ofrecer la posibilidad de publicar diferentes datos geoespaciales, entre ellos datos provenientes de sensores; 2) seguir un modelo de diseño utilizando el patrón factoría, mediante el cual se puede conformar un grupo de fábricas que proporcionan operaciones para publicar nuevo contenido en cada uno de los servicios típicos en el ámbito geoespacial; 3) se desarrolla mediante un geoprocesamiento, el cual ofrece la capacidad de ser ejecutado mediante interfaces predefinidas, aportando una mejora en interoperabilidad. GSF ocupa un papel importante dentro de la actual tesis, ya que tras su adquisición se desarrolla el trabajo de publicación de los datos de sensores.

3.2. Conceptos relacionados

En esta sección se introducen los conceptos básicos utilizados a lo largo del capítulo. Dicha sección está compuesta por tres secciones diferentes. La primera

sección (Sección 3.2.1) define de forma general los estándares utilizados para la creación de datos geoespaciales. La Sección 3.2.2 detalla los estándares utilizados para los datos provenientes de sensores. Finalmente, la Sección 3.2.3 habla de los estándares para el geoprocesamiento GIS.

3.2.1. Estándares para la definición de datos geoespaciales

El principal propósito de la iniciativa GEOSS es ofrecer servicios con interfaces comunes para mejorar su interoperabilidad y la escalabilidad (Lautenbacher, 2006). Para ello, se definen unos principios comunes, técnicas y estándares que deberán ser asumidos por todos aquellos que quieran cumplir dicha iniciativa. Además, deberán ser los usuarios quienes asuman dichos principios a la hora de generar o consumir datos geoespaciales.

El organismo Open Geospatial Consortium (OGC) es el que se encarga de la estandarización de servicios y datos geoespaciales, con el fin de alcanzar los principios de interoperabilidad y escalabilidad. OGC es un consorcio internacional de empresas, agencias gubernamentales y universidades que participan en un proceso continuo para el consenso y desarrollo de especificaciones, con el fin de establecer los estándares. Se trata de la principal organización de estandarización de la comunidad GIS.

En el contexto Europeo, existen varias iniciativas con la finalidad de apoyar de forma técnica, política y social el desarrollo de sistemas de GII y Observación de la Tierra (EO, en sus siglas en inglés). Como se indicó en la Sección 3.1 una de ellas es INSPIRE. Además de ésta, existe un programa llamado Copernicus, anteriormente llamado GMES, cuyo objetivo es dotar a Europa de una capacidad operacional y autónoma sobre este tipo de temas. En mayor medida, el propósito de Copernicus es ofrecer servicios de información basados tanto en datos provenientes de satélites como de la tierra (in-situ). Dicha iniciativa es liderada por la Agencia Espacial Europea y la Comisión Europea. Copernicus e INSPIRE son en cierta medida la aportación Europea al movimiento GEOSS (Díaz y Schade, 2011).

Por otra parte, INSPIRE, a nivel técnico, ofrece una amplia variedad de estándares que garantizan la interoperabilidad para una futura integración de los servicios i/o productos generados a partir de ellos (Mykkänen y Tuomainen, 2008). Actualmente, INSPIRE recomienda las especificaciones y estándares OGC existentes.

La arquitectura técnica INSPIRE define tres capas diferentes, que son: la capa de contenido geoespacial (datos, metadatos y modelos), la capa de servicios y la capa de aplicaciones (aplicaciones y geoportales), como se ilustra en la Figura 3.1.

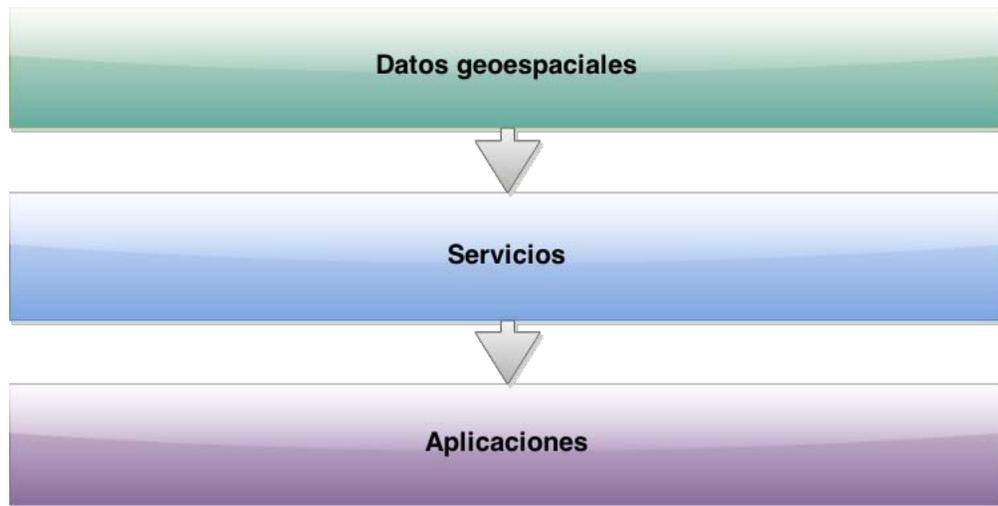


Figura 3.1: Arquitectura INSPIRE.

En el ámbito GIS existen servicios propios (denominados servicios geoespaciales), que por su particularidad son diferenciados de los servicios de propósito general (Mykkänen y Tuomainen, 2008). INSPIRE regula diferentes categorías dependiendo de la funcionalidad: *Discovery*, *View*, *Download*, *Transformation* y *Invocation*.

OGC ha realizado un laborioso trabajo con el fin de proponer diferentes estándares para dar solución a cada una de las categorías previstas por INSPIRE, entre los cuales se encuentran: Web Map Service (WMS) (de la Beaujardiere, 2006), Web Feature Service (WFS) (Vretanos, 2005), Sensor Observation Service (SOS), Sensor Event Service (SES) o Catalogue Services para la web (CSW) (Nebert, 2004), entre otros. Otras especificaciones, como Web Processing Service (WPS) (Schut, 2008) proporcionan una interfaz para realizar cualquier tipo de geoprocésamiento. La Tabla 3.1 muestra una visión general sobre las especificaciones de los estándares OGC indicando a qué servicio INSPIRE corresponden.

Está fuera del objetivo de esta tesis englobar todo tipo de contenido geoespacial que no sea utilizado para el tratamiento de datos de sensores, por lo que

<i>Descripción</i>	<i>Especificación OGC</i>	<i>Servicios INSPIRE</i>
Catalogue Services	CSW	Discovery
Web Mapping Service	WMS	View
Vector Data Download Service	WFS	Download
Raster Data Download Service	WCS	Download
Sensor Data Download Service	SOS	Download
Web Processing Service	WPS	Invoke
Sensor Event Service	SES	N/A

Tabla 3.1: Diferentes servicios geoespaciales y sus estándares

únicamente se detallarán los estándares que tengan relación con sensores.

Estándares para la definición de datos de sensores

Al igual que en el resto de datos geoespaciales, los datos de sensores también se organizan siguiendo las iniciativas GII para lograr formar las SDIs. OGC establece un grupo de trabajo llamado Sensor Web Enablement (SWE) (Botts y otros, 2008), el cual define un conjunto de especificaciones relacionadas con sensores, proponiendo modelos de datos y servicios web, con el fin de ser usados como puente entre los sensores y los usuarios, permitiendo que los sensores sean accesibles y controlables a través de la web (Sheth y otros, 2008).

SWE alberga un conjunto de especificaciones en un esquema estándar abierto mediante lenguaje de marcas extensible (XML, en sus siglas en inglés) y servicios Web, que permiten: 1) encontrar sensores y datos del sensor; 2) describir sistemas y datos del sensor; 3) recuperar tanto observaciones del sensor a tiempo real como históricos; 4) asignar sensores y sistemas de tratamiento del sensor; 5) agregar simulaciones y recuperar los resultados de simulaciones; 6) notificar resultados y alertas de los fenómenos de interés; 7) permitir la intercomunicación, la asignación y la descripción del sensor y además obtener, observaciones de sensores mediante la Web.

SWE incluye diferentes estándares, entre los cuales están: Sensor Model Language (SensorML) (Botts y Robin, 2007), Observations and Measurements (O&M) (Cox, 2007), SOS, Transducer Markup Language (TransducerML) (Havens, 2007), Sensor Planning Service (SPS) (Simonis, 2007), Sensor Alert Service

(SAS) (Simonis, 2006), entre otros. Sin embargo, en este trabajo sólo se usan las primeras tres especificaciones, la cuales son detalladas a continuación.

SensorML El estándar SensorML es uno de los más importantes dentro del SWE, ya que proporciona la información necesaria para definir un sensor. Con él se define cómo encontrar, procesar y registrar observaciones, de forma que se establece un modelo de datos y esquema XML para el control de los sensores mediante la Web. De esta forma, SensorML detalla un esquema estándar describiendo cualquier tipo de sensor, ya sea estacionario o dinámico, in-situ o remoto y activo o pasivo. Con esto se obtiene un mecanismo completo para representar sensores, apoyar el proceso y analizar los valores de las observaciones, georeferenciarlas, además de suministrar características de actuación sobre los propiedades de cada sensor. Son tres los ámbitos que define SensorML:

- *Definición de las observaciones.* Se detallan aspectos como el tipo de fenómeno observado (temperatura, presión atmosférica, dirección viento, etc.), así como la precisión, exactitud o tipo de los valores de las observaciones.
- *Definición de las geometrías.* Se detallan las características geoespaciales que define el sensor.
- *Definición del sensor.* Se detalla la información sobre el sensor.

SensorML describe cada sensor utilizando nueve componentes principales (Figura 3.2). De ellos los requeridos son:

- *identifiedAs.* Define el identificador principal del sensor, proporcionando un único identificador, nombre y tipo del sensor.
- *documentConstrainedBy.* Indica el tiempo de validez para el que una observación es válida, el nivel de seguridad y restricciones a nivel de derechos, además del uso comercial.
- *measures.* Describe las características de las observaciones, parámetros del modelo de sensor y de respuesta del sensor relativas a esa observación.

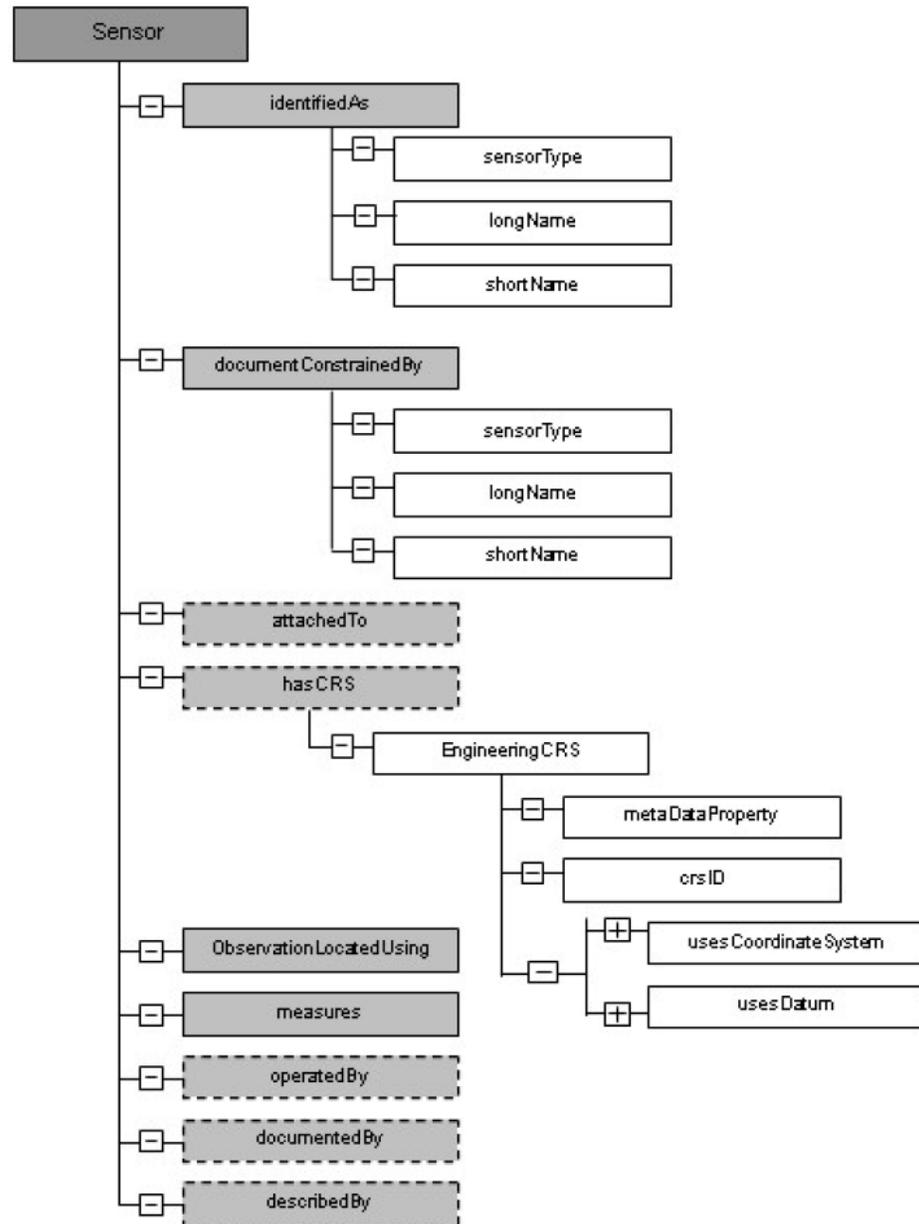


Figura 3.2: Componentes que forman el estándar SensorML.

- *observationsLocatedUsing*. Indica el estado del sensor, mostrando no sólo su localización geoespacial, sino también su orientación, velocidad y aceleración (si se trata de un sensor móvil), además de las transformaciones de coordenadas y operaciones necesarias para su uso.

Además de los componentes detallados anteriormente, existen otros opcionales que son:

- *AttachedTo*. Define la plataforma a la que se sujeta el sensor.
- *hasCRS*. Define el sistema de Referencia de Coordenadas del Sensor (CRS, en sus siglas en inglés).
- *operatedBy*. Indica la información sobre los responsables del sensor y cómo se gestionan las observaciones.
- *documentedBy*. Añade los metadatos necesarios para describir el autor del documento, qué modificaciones se han hecho, quién las ha realizado y cuándo, además de otras referencias.
- *describedBy*. Añade los metadatos sobre el modelo, fabricante, número de serie e historia del sensor.

O&M Proporciona un modelo estándar para representar e intercambiar observaciones de sensores. El estándar presenta un modelo de datos codificado mediante XML, el cual describe la relación entre diferentes aspectos del proceso de captura de datos. El esquema O&M define tanto las observaciones como los fenómenos. O&M puede incluir metadatos, como ayuda al análisis e interpretación de los datos. La Figura 3.3 muestra la representación mediante UML (Unified Modeling Language) del estándar.

Las especificaciones del O&M se requieren expresamente para todos los estándares del grupo SWE, así como el soporte general para sistemas sometidos a OGC que trabajan con medidas técnicas en ciencias e ingeniería.

SOS Es un servicio para el almacenamiento de observaciones mediante interfaz Web, siendo el principal modelo de servicio del grupo SWE. Su objetivo es ofrecer acceso a las observaciones de los sensores y sistemas de sensores de una forma estándar que sea consistente para todos los sistemas de sensores, incluyendo todas las categorías posibles de estos. Para ello, SOS utiliza los estándares definidos anteriormente (SensorML y O&M).

SOS tiene las siguientes tres operaciones básicas obligatorias que proporcionan su funcionalidad básica. La primera de ellas, *GetCapabilities*, ofrece la descripción

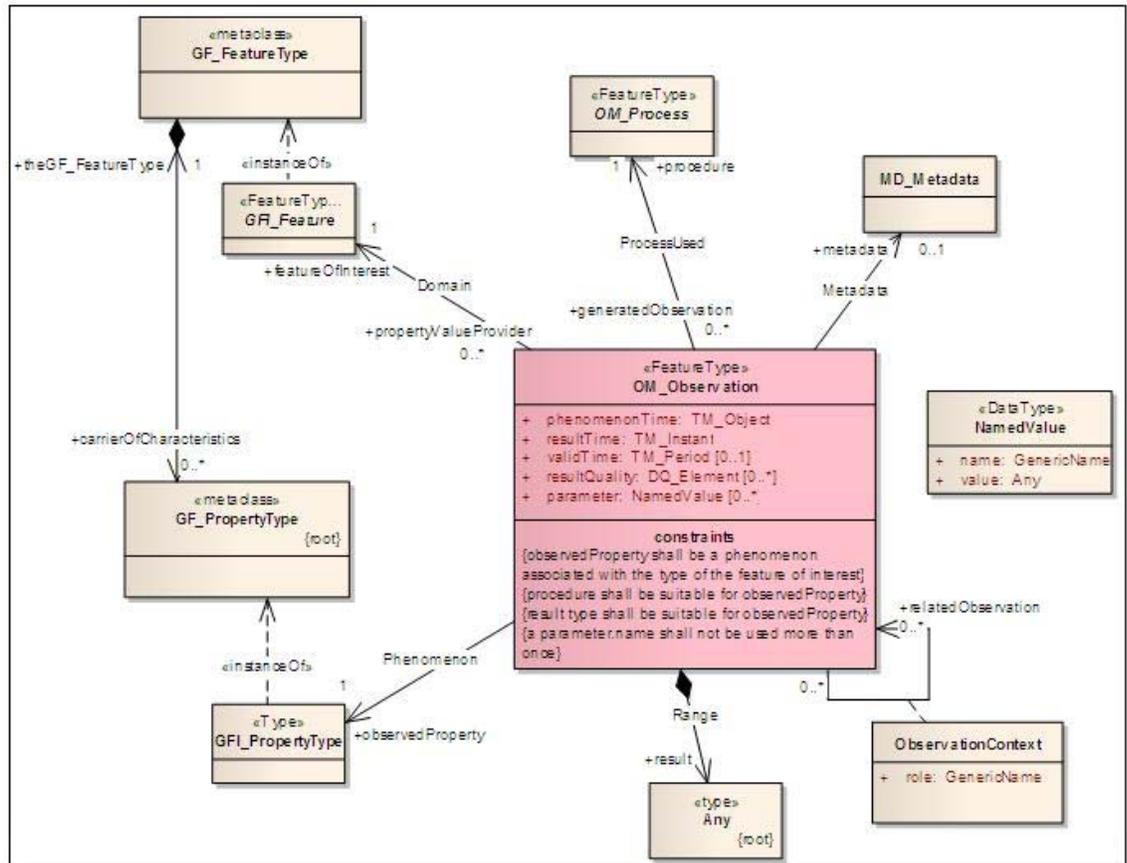


Figura 3.3: Componentes que forman el estándar O&M.

espacial y temporal de las observaciones que han sido almacenadas, así como una lista de los sensores y las características disponibles. La operación *DescribeSensor* se utiliza para la descripción de los sensores mediante SensorML. Finalmente, la operación *GetObservation* ofrece acceso a las observaciones y por lo tanto, proporciona la funcionalidad principal del SOS. Esta operación devuelve las observaciones en el estándar O&M.

3.2.2. Generación y publicación de metadatos

Un metadato viene del griego $\mu\epsilon\tau\alpha$, meta, que significa “más allá de”, que juntamente con dato significa “más allá del dato”. Pero el significado más acuñado es el de “datos sobre datos” (Group y otros, 1996; Kildow, 1996). Un metadato tiene el objetivo de describir otros datos. En general, un grupo de metadatos

se refiere a un grupo de datos, llamado recurso. El concepto de metadatos es análogo al uso de índices para localizar objetos en vez de datos. Por ejemplo, en una biblioteca el uso de fichas que especifican autores, títulos, editoriales y lugares para buscar libros. Así, los metadatos ayudan a ubicar datos.

Los metadatos en el ámbito de GIS, se aplican sobre los datos que tienen, de forma explícita o implícita, información geoespacial. En otras palabras, están asociados a una posición geoespacial. Estos objetos pueden ser almacenados en un sistema de información geoespacial llamado catálogo o, simplemente, pueden ser documentos, datos, imágenes u otros objetos.

Además, los metadatos tienen como objetivo facilitar la interoperabilidad en un entorno distribuido como son las SDIs.

Los metadatos sobre información geoespacial informan a los usuarios sobre los datos existentes, describiendo: el sistema de referencia espacial, su calidad, su distribución, su formato, restricciones de seguridad o frecuencia de actualización, entre otros aspectos, de tal manera que sirven para describir un conjunto de datos geoespaciales y contestan las preguntas: “qué”, “cuándo”, “dónde”, “quién son”, “dónde son” y “cómo” se han generado los datos.

Los objetivos que se persiguen con la creación de los metadatos son los siguientes:

- Que se pueda buscar y encontrar los conjuntos de datos, es decir, saber qué datos existen, qué datos hay disponibles en una determinada zona y para un tema determinado, a una escala o en general con unas características específicas que el usuario demanda.
- Que se pueda valorar la calidad del conjunto de datos, valoración requerida para ser utilizada como fuente de datos en otros procesos dentro de una misma organización o para el intercambio entre organizaciones.
- Que se pueda elegir cuál es el conjunto de datos más idóneo, comparando los distintos conjuntos de datos entre sí, de modo que se pueda seleccionar cuáles cumplen los requisitos del usuario de manera óptima para el propósito que se persigue.
- Evitar la duplicidad de trabajo notificando sobre la información existente, su ubicación y su disponibilidad.

- Facilitar la utilización de los datos, mediante la descripción de todas las características técnicas relevantes de los datos, del modo más objetivo, más amplio y completo, haciendo posible su explotación y su ayuda a los usuarios de los datos tanto en la obtención de resultados como en su mantenimiento y actualización.

Los metadatos se ciñen a las normas de la Organización Internacional de Normalización (ISO, en sus siglas en inglés). Existe una serie de metadatos, llamada ISO 191xx, encargada de definir los metadatos geoespaciales. ISO 19115 define la forma de describir la información geoespacial y los servicios asociados, incluyendo el contenido espacio-temporal, la calidad de los datos, el acceso y los derechos de uso. La norma define más de 400 elementos de metadatos y 20 elementos. Concretamente, la versión que se utiliza para generar el metadato es la ISO 19139. Ésta, define los esquemas XML necesarios para generar los metadatos conforme a ISO 19115 en un formato de XML.

Los metadatos, como ya se ha dicho, son creados por los proveedores de datos y son almacenados en catálogos siguiendo unos estándares. Esto se hace para que un usuario pueda encontrar el dato que quiera entre millones de ellos. Pero el problema de los metadatos principalmente recae en la creación y mantenimiento para que sea encontrado de forma efectiva. La razón de este inconveniente es la poca sincronización entre datos y metadatos, además de la complejidad existente para automatizar la extracción de características del dato para generar el metadato (Bulterman, 2004; Manso y otros, 2004).

Existen tres técnicas para la generación de metadatos que son: manual, automático y mixto (Campbell, 2004; Currier, 2004; JORUM, 2004). Si únicamente se analiza la generación automática, hay dos formas de generación de metadatos, que son por extracción y deducción (Manso y otros, 2004).

- *Extracción.* Los datos geoespaciales, como son los ráster o vectoriales, se caracterizan por tener una estructura claramente definida, lo que facilita la extracción de metadatos. Existen algunas herramientas capaces de extraer directamente, mediante mecanismos ad-hoc, la información necesaria para la generación de los metadatos (Manso y otros, 2004).
- *Deducción.* Para la deducción, existen mecanismos de inferencia que utilizan técnicas de minería de datos, recuperación de datos, técnicas de razona-

miento, etc. Por ejemplo, podemos conocer el *Bounding Box* de un dato conociendo su topónimo u obtener un buen resumen a partir del nombre.

Una vez generado el metadato, éste tiene que ser publicado para que otro usuario pueda encontrarlo. La publicación de metadatos es una de las partes más importantes en el ciclo de la información, ya que gracias a la publicación los datos son visibles. En el mundo de la información geoespacial es muy importante la publicación de los metadatos porque sirve para poner a disposición de los usuarios la información de los datos geoespaciales. Además, la publicación de metadatos está regida por la OGC y se realiza en catálogos.

Un catálogo de metadatos contiene los metadatos del contenido geoespacial. Los catálogos permiten realizar búsquedas, a través de opciones como nombre, área geográfica, coordenadas, categoría temática ó tipo de dato. A través del catálogo de metadatos se puede descargar la ficha de metadatos de los distintos recursos.

El servicio OGC, que se encarga de definir cómo ha de realizarse el proceso de publicación, se llama Servicio de Catalogo (CSW, en sus siglas en inglés). De esta forma los metadatos se publican siguiendo unos estándares, que en primer lugar definen cómo ha de ser el metadato y seguidamente cómo ha de producirse la publicación. Los catálogos más famosos son Geonetwork¹ y Deegree².

3.2.3. Geoprocesamiento

Otro de los conceptos necesarios para el actual capítulo son los servicios de geoprocesamiento. Los geoprocesamientos se distinguen de cualquier otro proceso debido a la funcionalidad en que son utilizados, ya que se aplican para realizar procesos utilizando contenido geoespacial. Estos, al igual que los procesos de uso general, ofrecen la capacidad de realizar cualquier tipo de proceso/ algoritmo mediante la Web, ofreciendo una ejecución concurrente para cada uno de los usuarios que lo ejecutan.

La ventaja del procesamiento remoto para compartir rutinas y reutilizar métodos sin la necesidad de mantenimiento de aplicaciones de forma local, compite con la desventaja de su rendimiento. En general, cuando se trabaja con datos espaciales,

¹<http://geonetwork-opensource.org/>

²<http://www.deegree.org/>

estos suelen ser datos complejos y de gran tamaño, pudiendo hacer inviable el procesamiento remoto de este tipo de datos debido al sobre coste que implica enviar grandes cantidades de datos desde el consumidor al proveedor del servicio de procesamiento. Por lo tanto, las técnicas de generalización complementan perfectamente a los servicios de geoprocésamiento ya que permiten la extracción de la información relevante que evite el tráfico de grandes volúmenes de datos.

Si hablamos del estándar OGC encargado de ello, éste se llama WPS. Dicha especificación fue diseñada para estandarizar la forma de ofrecer procesamientos GIS a través de Internet. WPS proporciona diferentes reglas para describir cualquier proceso, así como la forma de realizar peticiones al servicio y la respuesta que ofrece. También define cómo un cliente puede solicitar la ejecución de dichos servicios. La especificación del WPS fue publicada por OGC en 2004 y en 2007 se publicó la versión 1.0.0.

Mediante el uso de servicios de geoprocésamiento distribuido, los usuarios pueden acceder y compartir modelos. El acceso a servicios de geoprocésamiento distribuidos incrementa su reutilización, disminuyendo el coste de mantenimiento y haciendo más eficiente el trabajo de los expertos en el contexto de las SDIs. La comunicación con estos servicios se realiza mediante peticiones HTTP con XML, garantizando de este modo la interoperabilidad y la independencia entre sistemas. Si bien en un principio WPS fue diseñado para trabajar con datos espacialmente referenciados, actualmente su especificación no limita el tipo de procesos únicamente a este tipo de datos, sino que es posible la ejecución de procesamientos sobre datos vectoriales, ráster, literales, u otro tipo de datos, entre ellos los que siguen el estándar Geography Markup Language (GML).

El modelo propuesto por OGC se basa en arquitecturas abiertas e interoperables, que permiten crear aplicaciones a partir de la concatenación de operaciones atómicas, tal y como se muestra en la Figura 3.4, en contraste con las aplicaciones GIS monolíticas donde toda la funcionalidad está implementada dentro de la propia aplicación. El servicio WPS proporciona tres operaciones: *GetCapabilities*, *DescribeProcess* y *Execute*. La implementación de estas operaciones es obligatoria por parte del servidor. A continuación se detallan cada uno de ellos.

- *GetCapabilities*. Informa sobre las capacidades del geoprocésamiento.
- *DescribeProcess*. Indica los parámetros que requiere, sus formatos y los resultados que produce.

- *Execute*. Realiza la ejecución del geoprocesamiento. Se le proporcionan los parámetros de entrada, quedando a la espera de los resultados.

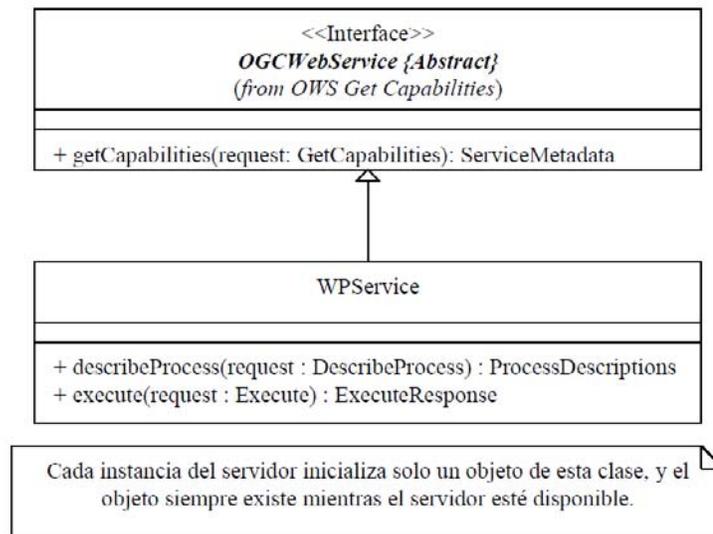


Figura 3.4: Modelo UML de la interfaz WPS.

3.3. Geoss Service Factory

Esta sección presenta la propuesta para la publicación de datos geoespaciales, llamada GSF (Díaz y Schade, 2011). GSF extiende los servicios GII (Sección 3.2.1) y propone la publicación como un servicio más. GSF proporciona un componente interoperable, que en forma de geoprocesamiento, ofrece la funcionalidad de publicar datos geoespaciales en servicios estándares, cumpliendo con la directiva europea INSPIRE y favoreciendo el movimiento hacia las SDIs.

Dicho geoprocesamiento propone una solución para la publicación de contenido geoespacial orientada al consumidor de datos, facilitando la publicación automática de los mismos y proporcionando una herramienta de colaboración para el usuario final.

La publicación forma parte del flujo natural de trabajo para el tratamiento de información. Tras la adquisición de los datos es necesaria su publicación para poder ser reutilizados por otros usuarios interesados. GSF mejora la etapa de

publicación del contenido geoespacial, simplificando y automatizando el proceso de publicación.

El resultado de la ejecución de GSF son los puntos de entrada a los servicios donde los datos han sido publicados, ya sea para la descarga (Download), visualización (View) o descubrimiento (Discovery). De esta forma se favorece el cumplimiento de los estándares y de las diferentes iniciativas como INSPIRE.

La Figura 3.5 muestra donde se encuentra GSF dentro de la arquitectura definida por INSPIRE. GSF extiende la publicación como un servicio GIS más en la capa de servicios. La capa de aplicación incluye los clientes para el usuario final, que van desde los sistemas de apoyo de decisiones ambientales complejas a simples clientes en dispositivos móviles. Finalmente la capa de contenido incluye los datos, metadatos y modelos.

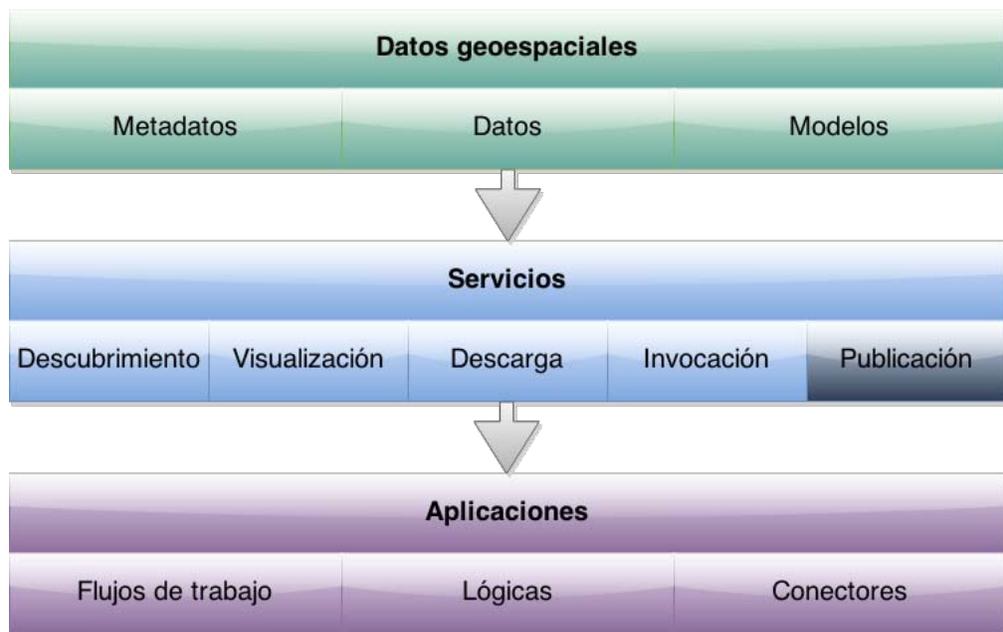


Figura 3.5: Posición de GSF dentro de la arquitectura INSPIRE.

GSF está implementado como un modelo de procesamiento ofreciendo una interfaz estándar. Estructuralmente sigue un patrón factoría (Gamma y otros, 1994). El patrón factoría es un patrón creacional que proporciona un mecanismo escalable para crear nuevas entidades según unos criterios determinados. Con este patrón, es posible crear una interfaz común para conjuntos o familias de

objetos que dependen mutuamente de factorías. En este caso, el patrón factoría se utiliza para diseñar una serie de factorías donde cada una se ocupa de la gestión de los datos para cada uno de los servicios definidos en INSPIRE (Díaz y Schade, 2011). De esta forma, GSF aprovecha cada factoría para diseñar un método de publicación para cada uno de los servicios establecidos por INSPIRE.

La Figura 3.6 muestra el diagrama del patrón factoría que sigue GSF. En él, se utiliza la herencia para cada uno de los servicios de INSPIRE y GEOSS (parte central e izquierdo de la Figura 3.6). Las factorías *View* y *Download* son las encargadas de la publicación de los datos en cada uno de los servicios. Para ello, se implementan las especificaciones de cada uno de los estándares definidos por OGC. De esta forma, WMS se utiliza para implementar un servicio de visualización y WFS para implementar, a su vez, un servicio de descarga.

El GSF tiene asociado un perfil de publicación o Service Publication Profile (SPP), que se compone de la información necesaria para la publicación de los datos. El SPP es un documento estructurado (XML) que puede ser enviado como parámetro del geoprocesamiento.

El alcance de este trabajo está limitado a la publicación de los datos exclusivamente a los servicios OGC. Dichas especificaciones ofrecen dos posibilidades de comunicación que son:

- *Interfaces transaccionales.* Algunos estándares OGC (como WFS, SOS, y CSW) incluyen operaciones transaccionales en su especificación. En este caso, cada factoría utiliza dicha operación para la publicación de los datos. De esta forma, la implementación es independiente de las instancias de los servicios, siendo capaz de desplegar contenido en cualquier instancia del servicio que implemente la interfaz transaccional.
- *Interfaces no transaccionales.* Otras especificaciones OGC no proporcionan operaciones transaccionales. En estos casos, la publicación debe realizarse por otros medios. Por ejemplo, el servicio WMS, no ofrece dicha opción, por lo que la implementación de la factoría dependerá de la instancia del servicio en particular.

Como se ha comentado, GSF tiene la capacidad de publicar datos geoespaciales en los diferentes servicios, pero además, tiene la característica de generar los metadatos asociados a los datos publicados y publicarlos en los catálogos.

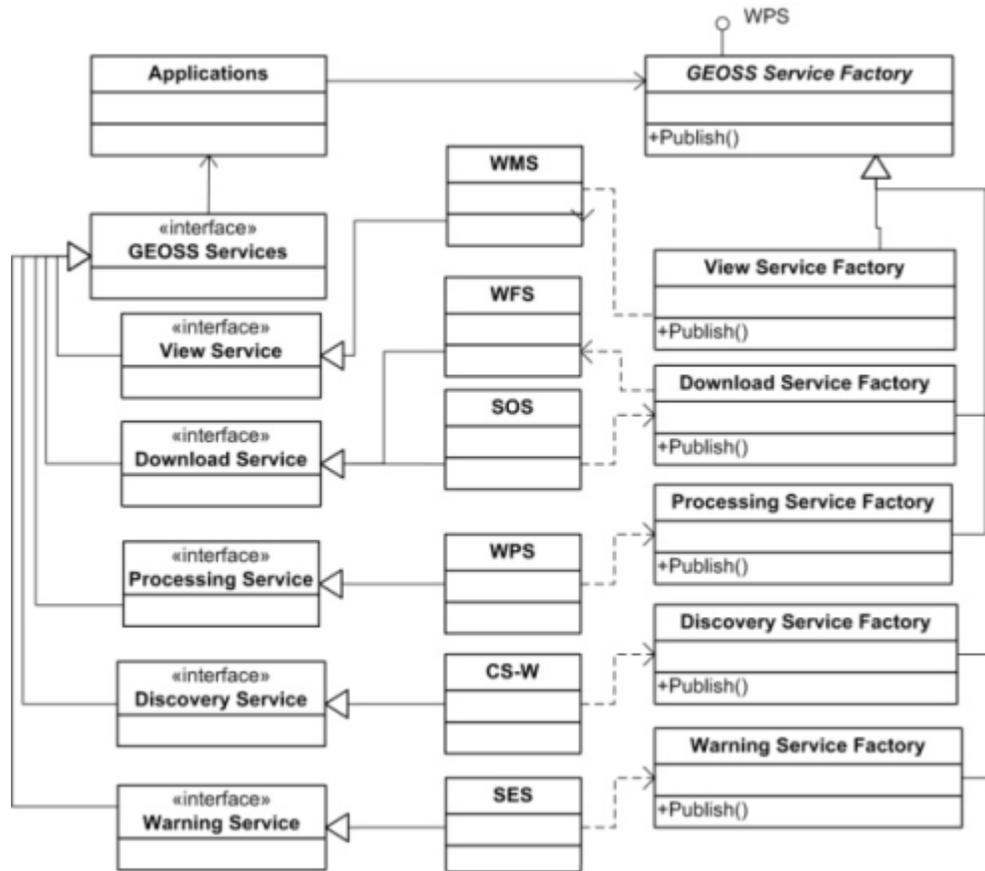


Figura 3.6: Patrón factoría utilizado en Geoss Service Factory.

Las siguientes secciones describen cómo se realiza la publicación de los datos y cómo se generan y publican los metadatos.

3.3.1. Publicación de datos geospaciales

Para la publicación en los diferentes servicios, mediante el patrón factoría, GSF provee un método para ello llamado *Publish*, siendo éste diferente para cada servicio. Estos métodos son encapsulados en un geoprocesamiento estándar. Como cualquier geoprocesamiento, GSF establece unos parámetros de entrada y unos de salida. La Figura 3.7 muestra el diagrama de clases UML con una simplificación de la interfaz de GSF, además de los parámetros de entrada y salida. A continuación se detalla cada uno de ellos:

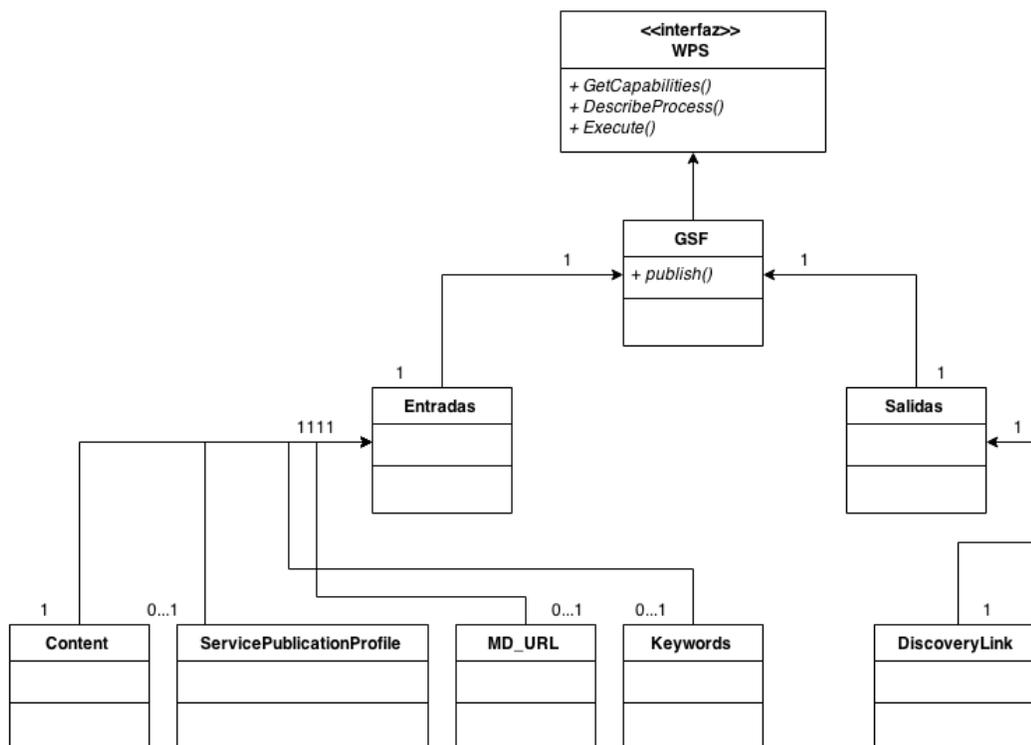


Figura 3.7: UML con los parámetros de entradas y salida del GSF.

- *Contenido*. Es el único parámetro obligatorio. Puede indicarse tanto como valor, como referencia, en cuyo caso se utiliza un enlace para indicar la localización del dato. Permite publicar varios tipos, que van desde formato vectorial, a ráster o metadatos.
- *ServicePublicationProfile*. Es un fichero codificado como XML, que describe la configuración con la que se realizará la publicación. Este parámetro incluye información sobre cada uno de los servicios donde han de ser publicados los datos.
- *MD_URL*. Este parámetro indica que el contenido ya ha sido publicado con anterioridad, por lo que ya existen metadatos disponibles que deben ser reutilizados. De esta forma el dato a publicar será asociado al metadato que se indique en el enlace.
- *Palabras clave*. Proporciona las palabras que definen al dato y serán utilizadas para la generación del metadato.

- *DiscoveryLink*. Este es el único parámetro de salida. Contiene la información necesaria para descubrir el contenido publicado en las diferentes estancias de los servicios, por lo que contendrá los diferentes enlaces a los servicios, al igual que el enlace al catálogo, donde se haya almacenado el metadato generado.

Como ya se ha indicado, el fichero SPP define el comportamiento que seguirá GSF para decidir dónde se publicará para cada servicio. Por ejemplo, el SPP determina que dependiendo del formato del dato, el dato sea publicado para la visualización y la descarga. Además de aportar la información de dónde residen los diferentes servicios en donde publicar, el uso del SPP permite que el GSF sea más escalable y flexible.

La Figura 3.8 muestra cuál es el flujo que se sigue para la publicación de datos geoespaciales. Cuando se invoca el geoprocesamiento mediante la operación *Execute* (Figura 3.8 1)), se obtiene la información del SPP para establecer cuál será la configuración para realizar la publicación (Figura 3.8 2)). Dependiendo del tipo de dato, se procederá a publicarlo en los servicios correspondientes, por lo que se llamará a la factoría encargada de proporcionar y crear dichos servicios (Figura 3.8 3)). Por cada una de las factorías, mediante el método de publicación, el dato será enviado a publicar en cada uno de los servicios (Figura 3.8 4)). Finalmente, se realizará la conexión con cada una de las instancias de los servicios donde se haya indicado en el SPP (Figura 3.8 5)). Como resultado, se obtendrán los enlaces donde los datos han sido publicados. Posteriormente, dichos enlaces serán necesarios para la generación del metadato.

Como se comentó anteriormente, GSF tiene la habilidad de publicar tanto datos ráster como vectoriales. GSF es capaz de publicar Shapefiles (datos vectoriales) en el servicio de WFS. Además, puede publicar datos ráster como es GEOTIFF. También soporta Keyhole Markup Language (KML) y Geography Markup Language (GML) para ser publicados en el servicio WMS. Finalmente, también ha sido extendido para la publicación de observaciones de sensores que sigan el estándar O&M.

Publicación de datos de sensores

Como se ha comentado, GSF también es capaz de publicar datos provenientes de sensores. Cuando una nueva observación se genera, con el fin de ser publicada

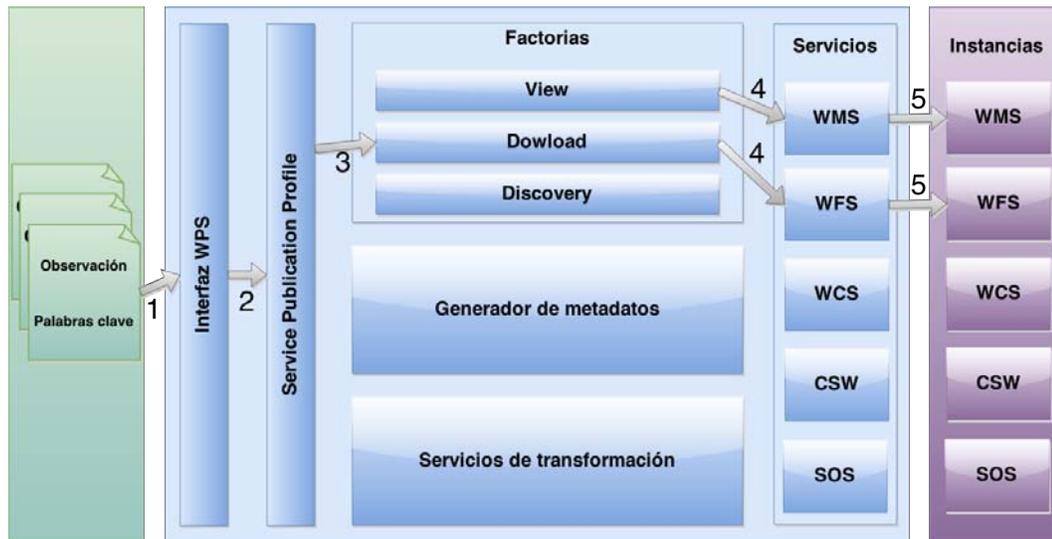


Figura 3.8: Diferentes pasos para la publicación de datos geospaciales utilizando GSF.

para un posterior acceso, se invoca el geoprocesamiento GSF (Figura 3.9 1)). Seguidamente, al igual que se realiza en los otros servicios, el SPP almacena la información necesaria para la publicación en los servicios estándares, para este caso donde reside el servicio SOS (Figura 3.9 2)). La Factoría encargada de publicar en dicho servicio es la de *Download* (Figura 3.9 3)). Gracias al método *Publish*, de forma interna, conecta con la instancia indicada en el SPP (Figura 3.9 5)). De esta forma, mediante la operación *InsertObservation* la observación se publica en el SOS. Además, también se permite la opción de registrar nuevos sensores mediante la operación *RegisterSensor*.

3.3.2. Generación y publicación de metadatos

A parte de publicar los datos en sí, GSF ofrece la posibilidad de generar el metadato de cada uno de los datos geospaciales que se publican. Este proceso se realiza de forma automática y completamente desatendida por parte del usuario. El usuario solo ha de indicar la información sobre los sitios donde se publica el metadato y algunas palabras clave que definan el propio dato. Para ello se utiliza el fichero de configuración SPP. El módulo encargado de la generación del metadato, se llama *Generador de Metadatos*.

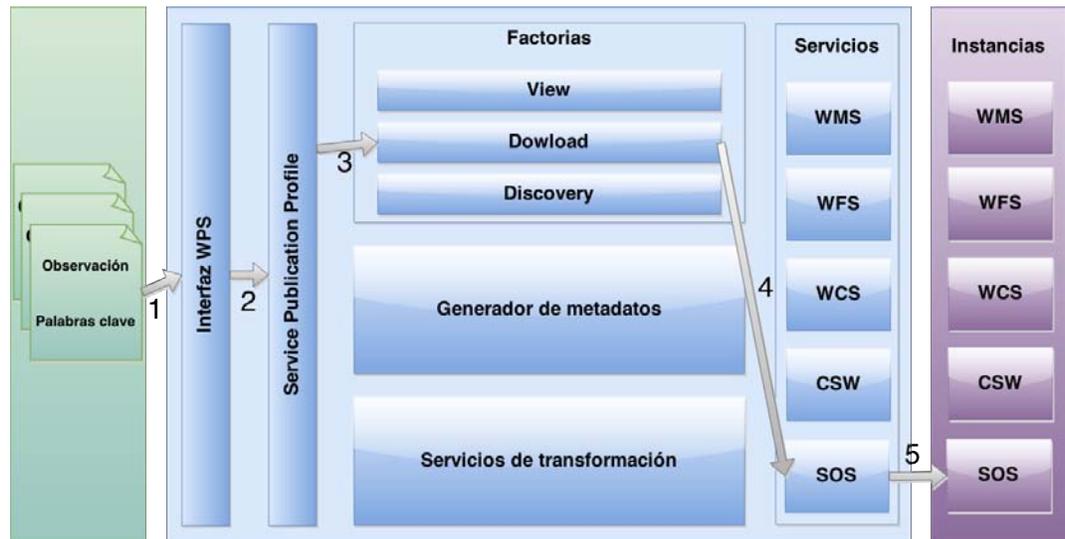


Figura 3.9: Diferentes pasos para la publicación de datos de sensores utilizando GSF.

Para poder generar el metadato es necesario conocer información referente al dato y al sitio donde se ha publicado. Parte de esta información puede ser extraída de los servicios en los que se ha publicado el dato. Por esta razón, el *Generador de Metadatos* tendrá que actuar cuando se haya publicado el dato, si no se le indica lo contrario. La factoría encargada de ejecutar el *Generador de Metadatos* es la *Discovery*.

Cuando se termina de publicar un dato, si el geoprocesamiento no ha recibido el parámetro *MD_URL* con un metadato, se llama al *Generador de Metadatos*. Éste, como parámetros de entrada, necesita por parte del procesamiento de publicación de los enlaces a los servicios donde residen los datos para generar las peticiones, además de las palabras clave. Como único parámetro de salida del GSF, tenemos un enlace de acceso al servicio de catálogo donde se haya publicado el metadato.

El primer paso del generador será consultar los servicios de visualización y descarga (Figura 3.10 2)), para obtener la información necesaria y crear el metadato. Esto se realizará mediante las peticiones oportunas (Figura 3.10 3)). Una vez consultada la información necesaria, se aplicará el proceso de generación del metadato (Figura 3.10 4)) y procederá a la publicación en el catálogo (Figura 3.10 5)). Finalmente el geoprocesamiento devolverá el enlace resultante de la publicación (Figura 3.10 6-7)).

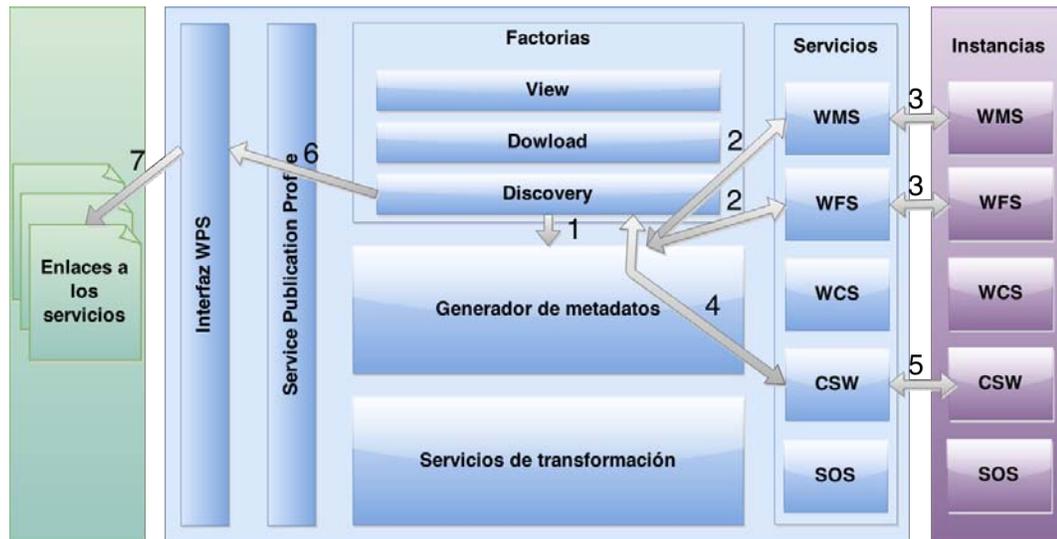


Figura 3.10: Diferentes pasos para la generación y publicación de metadatos utilizando GSF.

A continuación se detalla el proceso de generación y publicación del metadato. La Figura 3.11 muestra las entradas y salidas del *Generador de Metadatos*. Los diferentes pasos realizados para la generación y publicación se muestran en la Figura 3.12. A continuación se detalla dicho proceso.

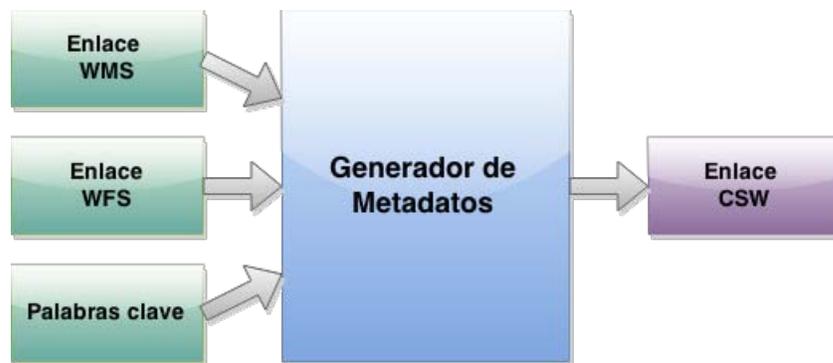


Figura 3.11: Parámetros de entrada y salida del *Generador de Metadatos*.

1. Como primer paso, se realiza una petición *GetCapabilities* a los servicios donde el dato ha sido previamente publicado. Por ejemplo, si el contenido publicado es un dato vectorial que ha sido publicado para su visualización



Figura 3.12: Secuencia de pasos del *Generador de Metadatos*.

en un WMS y para su descarga en un WFS, el *Generador de Metadatos* solicita primero un *GetCapabilities* al servicio de visualización. Esta petición devuelve un XML con las características del dato que serán necesarias para la creación del metadato. A continuación, hace lo mismo, pero con el servicio de descarga para añadir más información al metadato.

2. El siguiente paso es aplicar plantillas de transformación Extensible Stylesheet Language Transformations (XSLT). La transformación tendrá como entrada el XML obtenido del *GetCapabilities*. El resultado que se obtiene tras la transformación es otro documento XML que forma el metadato cumpliendo los estándares ISO 19139 e INSPIRE.
3. El tercer paso consiste en analizar el XML del metadato para rellenar campos que no ha sido posibles rellenarlos en la transformación, como las palabras clave o los enlaces a los servicios. Tras este paso el metadato se considera completado.
4. Seguidamente se publica el metadato en el catálogo que esté definido en el SPP. Este paso se realiza de forma análoga a la publicación de datos, pero en este caso al servicio de Catálogo, que de nuevo y siguiendo las reglas de

<i>Metadato</i>	<i>Campo del metadato</i>	<i>Descripción del campo</i>
	Language	Lenguaje
	CharacterSet	Tipo de codificación
	HierarchyLevel	Nivel jerárquico de los datos
	Contact	Responsable de la Información
	DateStamp	Fecha de creación
	StandardVersion	Estándar utilizado
	IdentificationInfo	Datos de información: nombre, datos, resumen
	DistributionInfo	Enlaces a los servicios
	DataQuality	Información sobre el estándar

Tabla 3.2: Diferentes campos del metadato generado.

implementación, serán normalmente servicios que implementan la interfaz CSW. Para ello se utiliza el perfil transaccional del servicio CSW (CSW-T).

5. El último paso es obtener el enlace que identifica el metadato publicado. Tras obtener el enlace el *Generador de Metadatos*, se considerará como finalizado.

Se debe mencionar que el metadato generado no tiene todos los campos de la especificación, ya que algunos no se pueden obtener de forma automática. Se han elegido los campos que se muestra en la Tabla 3.2.

3.4. Trabajos relacionados para la Publicación

En esta sección se analizan los trabajos relacionados tanto con la publicación de datos geoespaciales, incluyendo los datos de sensores (Sección 3.4.1), como la generación y publicación de metadatos (Sección 3.4.2).

3.4.1. Publicación de contenido geoespacial

Si se analizan los trabajos relacionado sobre la publicación de contenido geoespacial, existen multitud de trabajos ya que la publicación es una parte muy importante en el ciclo de vida de este tipo de contenido. Aunque todos ellos asumen que es un proceso trivial (Beltran y otros, 2009; Chen y otros, 2009), debido a que la publicación en sí no supone ninguna novedad destacable, siendo esto a su

vez aplicable a los datos de sensores. En el trabajo presentado la novedad recae en el uso de un procesamiento estándar para la publicación. Por esa razón no se ha realizado un estudio bibliográfico de la publicación de contenido geoespacial.

3.4.2. Generación y publicación de metadatos

Por lo referente a la generación de metadatos, en la literatura hay trabajos que abordan esta cuestión proponiendo herramientas automáticas para la generación de metadatos (Bulterman, 2004; Manso y otros, 2004; Beltrán y Martín, 2010). Existen diferentes métodos de generación: manual, semiautomático o automático (Campbell, 2004; Halliday, 2004). Beard (1996) propuso cinco métodos para la generación de metadatos: 1) de forma manual (tecleando), 2) mediante la ampliación de la información almacenada con los valores obtenidos a través de consultas de tipo geoespacial, 3) mediante mediciones automáticas y observaciones, 4) extraídos y calculados, y finalmente, 5) deducido a partir de otros elementos. Otros autores clasifican la generación automática en dos clases: extracción de metadatos y de inferencia (Beltran y otros, 2009).

Herramientas como CatMDEdit (Zarazaga-Soria y otros, 2003) se utilizan para la extracción automática de metadatos en diferentes formatos, como se refleja en el trabajo de Taussi (2007). La cantidad de información que puede ser extraída, principalmente depende del modelo de representación utilizado y su formato de archivo. Por lo tanto, hay elementos que únicamente se pueden extraer de ciertos tipos de datos y archivos, mientras que otros, tales como el tamaño de los datos, se pueden obtener en todas las circunstancias.

También hay trabajos que describen la forma de integrar la generación de metadatos dentro de los GIS. En esta línea de investigación existen mecanismos ad-hoc que procesan los datos para extraer información (Manso y otros, 2004), que se utilizan en el editor de metadatos de gvSIG (Beltran y otros, 2009).

Por otra parte, las técnicas de generación pueden describir la naturaleza de las herramientas que proporcionan la funcionalidad en la generación de metadatos. Normalmente, esta funcionalidad está incluida en los servicios de catálogos; estos son, por ejemplo, Geonetwork o ESRI ArcCatalog (incluido en ArcGIS). Dichas herramientas permiten la creación automática de campos básicos y la sincronización de datos de actualización y metadatos.

3.5. Conclusiones

En este capítulo se ha presentado una herramienta para la publicación de contenido geoespacial, además de posibilitar la generación automática y la publicación de metadatos, extendiendo el trabajo previo de Díaz y Schade (2011). Además, sirve como componente para la publicación de datos de sensores en servicios estándares. De esta forma, GSF es capaz de publicar observaciones codificadas mediante el estándar O&M.

GSF ofrece la capacidad de publicar datos geoespaciales en los diferentes servicios establecidos por la iniciativa INSPIRE, utilizando un geoprocesamiento, como es WPS, que a su vez ofrece una interfaz estandarizada. Además, facilita el proceso para la generación de los metadatos, que tradicionalmente se ha considerado un trabajo duro (Beltran y otros, 2009). Dicha herramienta puede ser utilizada tanto por usuarios expertos como por usuarios ocasionales.

El proyecto global aborda la mejora en la disponibilidad de datos en el entorno de las SDI así como su mantenimiento. En particular, la generación y publicación de metadatos facilita que un dato sea encontrado, ya que se generará y publicará el metadato que lo define y facilitará su localización. Se ha dado una posible solución al problema de disponibilidad de datos y metadatos y, lo que es más importante, su enlace, de forma que cada vez que se publicamos un dato y/o metadato, estos están enlazados para mejorar, no solo la visibilidad y descubrimiento del dato, si no también su acceso.

La creación de metadatos mejora la relación entre los datos y metadatos con el fin de facilitar la visibilidad de los datos y la accesibilidad. Este enfoque se utiliza en el servicio de publicación propuesto. GSF utiliza un geoprocesamiento mediante un estándar OGC para alcanzar dicho objetivo. Este estándar proporciona reglas para describir cualquier cálculo (proceso), así como la forma de realizar peticiones al servicio y la respuesta que ofrece.

GSF ofrece una solución para la publicación de contenido geoespacial destinada a aumentar la accesibilidad de datos para mejorar el intercambio de estos. Este sistema permite la publicación automática y proporciona una herramienta para el usuario final. GSF mejora la etapa de publicación de contenido al simplificar y automatizar el proceso de publicación.

Además se ha añadido la capacidad de publicar las observaciones producidas

mediante sensores. Tras la adquisición de las observación, éstas pueden ser convertidas a un formato estándar, por ejemplo O&M. Mediante el geoprocesamiento propuesto en este capítulo se permite su publicación en un servicio estándar, como lo es SOS. Además, al igual que en los otros tipos de datos geoespaciales, también se permite la generación y publicación del metadato que defina el dato.

De esta forma GSF sirve para cubrir una de las partes más importantes dentro del flujo natural del tratamiento de datos de sensores, como es la publicación, ya que sin una correcta generación y publicación del metadato que defina el dato es difícil el posterior acceso a él.

Capítulo 4

Acceso

Una vez publicadas las observaciones de sensores (Capítulo 3), el siguiente paso es ofrecer los servicios adecuados para que puedan ser consumidas por los usuarios o que estén listas para ser utilizadas para diferentes análisis. Este cuarto capítulo tiene como objetivo definir dichos servicios. Para ello se recurrirá al uso de estándares, los cuales favorecen la interoperabilidad. El capítulo se compone de la siguiente manera. La Sección 4.1 realiza una pequeña introducción sobre la forma de acceder a las observaciones de sensores. La segunda sección (Sección 4.2), detalla los conceptos que serán utilizados a lo largo del capítulo. La Sección 4.3 presenta la aportación principal de este capítulo, definiendo las interfaces y servicios diseñados para el acceso a los datos de sensores. La cuarta sección (Sección 4.4), realiza un estudio de los diferentes trabajos que actualmente se encuentran en la literatura. Finalmente, en la Sección 4.5 se detallan las conclusiones extraídas del actual capítulo.

4.1. Introducción

En la gran mayoría de redes de sensores disponibles, desafortunadamente, sus observaciones son servidas utilizando formatos no estándares (Capítulo 2). Estos formatos pueden ser diferenciados en dos aspectos. Por un lado la propia codificación de las observaciones, por ejemplo utilizando un XML con una estructura predefinida. Y por otro lado la forma de conexión, es decir, los protocolos utilizados para acceder a las observaciones en sí.

No seguir estándares impide un acceso reusable a dicha información, tradu-

ciéndose en una carencia de interoperabilidad. Ofrecer un acceso interoperable sobre los datos de sensores es crucial para un posterior análisis, además de facilitar la extracción de información útil, para la toma de decisiones sobre los resultados obtenidos en dichos análisis (Gualtieri y Tartaglia, 1998).

Si la publicación de los datos es una pieza importante, el acceso a ellos no lo es menos. Ofrecer mecanismos de conexión entre las fuentes de datos y los usuarios es uno de los objetivos del actual capítulo. Los mecanismos presentados, se caracterizan por ser diseñados siguiendo estándares predefinidos con el fin de aumentar la interoperabilidad. Además, también están pensados para adaptarse a las nuevas tendencias de consumo y así poder dar soporte a dispositivos donde sus prestaciones sean limitadas.

Uno de los objetivos principales que se plantean a lo largo de este capítulo, es proporcionar dichos servicios interoperables para garantizar el acceso a fuentes de datos heterogéneas procedentes de redes de sensores. Este hecho servirá para ofrecer un nuevo punto de entrada a dichas redes, pero en este caso utilizando estándares con los beneficios que eso conlleva.

Como ya se adelantó en el capítulo anterior, en el ámbito GIS, existen ciertas organizaciones encargadas de establecer estándares, una de estas es OGC. OGC mediante SWE, ofrece un abanico de estándares para el tratamiento de observaciones de sensores. En particular, OGC establece el estándar SOS para el almacenamiento de las observaciones mediante O&M. Cada uno de los sensores registrados se describe mediante el estándar SensorML. SOS, ofrece una interfaz de servicios Web estándar que permite a los clientes acceder a las descripciones de los sensores asociados y a las observaciones recogidas.

Como se concluye en Tamayo y otros (2011), el uso de los estándares SWE, sobre todo del servidor SOS, compromete el rendimiento en dispositivos con características reducidas como pueden ser teléfonos móviles. Para dar solución a esa problemática y como primera aportación de este capítulo, se presenta una interfaz para facilitar el acceso a datos de sensores, capaz de ser usada en dispositivos con capacidad computacional más limitadas.

La Sección 4.3.1 detalla cómo se aborda dicha interfaz sin comprometer el rendimiento, ya que utiliza interfaces ligeras que ya han demostrado su correcto funcionamiento en esos entornos. Además, tampoco comprometen la interoperabilidad que ofrecen los estándares SWE, ya que está basada en ellos. Ésta

se puede definir como: 1) ofrece un rendimiento en entornos con características con capacidad computacional más limitadas; 2) no deja de ofrecer estándares SWE, por lo que no afecta a la interoperabilidad; 3) está pensada siguiendo los principios de IoT. Dicha interfaz se presenta en un ámbito más genérico para servir datos de cualquier tipo de sensor.

Tras el aumento de la popularidad del IoT, dentro de las organizaciones de estandarización, han aparecido nuevas iniciativas con el fin de ofrecer estándares acordes a dichas tendencias. Por ejemplo, OGC ha propuesto un candidato a estandarización llamado OGC SensorThings API¹. Este candidato define un modelo de datos y una interfaz, adaptados a los paradigmas de IoT y WoT, basándose en los estándares consolidados SWE tales como SensorML o O&M.

Dicha interfaz será adoptada como segunda forma de acceso en este capítulo (Sección 4.3.2). El objetivo de esta interfaz es ofrecer un perfecto entendimiento con los paradigmas de IoT y WoT presentados en el Capítulo 2. Con OGC SensorThings API se proporciona un acceso interoperable a los datos del sensor siguiendo dichos paradigmas teniendo el apoyo de una de las mayores organizaciones de estandarización, como es OGC.

Finalmente, en la Sección 4.3.3 se presentan unos componentes para poder consumir observaciones de sensores con altas tasas de refresco. Como ya se ha dicho, actualmente hay desplegadas cantidades enormes de redes de sensores, las cuales generan un flujo constante de observaciones, dependiendo de la frecuencia de actualización de cada una de ellas. Además, el uso del protocolo de control de transmisión / Protocolo de Internet (TCP/IP, en sus siglas en inglés) para la adaptación de los paradigmas de IoT y WoT ha favorecido la disponibilidad de esos flujos de observaciones.

Para ello se define un mecanismo para el tratamiento de estos tipos de flujos, con el que se facilite el acceso a ellos y se pueda realizar cualquier tipo de análisis en tiempo real. Esta metodología permite: 1) conectar con diferentes fuentes de datos, independientemente del tipo de datos; 2) permitir conexiones más adecuadas cuando los sensores tengan mayores frecuencias de actualización; 3) y la capacidad de ofrecer dichos flujos mediante diferentes protocolos.

¹OGC SensorThing API,<http://ogc-iot.github.io/ogc-iot-api>

4.2. Conceptos relacionados

En esta sección se detallan los conceptos que son necesarios para una posterior presentación de las aportaciones propuestas en este capítulo. La sección se divide en tres secciones diferentes. En la primera de ellas (Sección 4.2.1) se detallan las nuevas tendencias para favorecer la conexión entre fuentes de datos y clientes mediante servicios Web. En la Sección 4.2.2 se detalla una interfaz candidata a estándar que sigue los paradigmas de IoT y WoT. Finalmente, la Sección 4.2.3 analiza las diferentes alternativas para el consumo de observaciones cuando se trabaja con fuentes de sensores con altas frecuencias de refresco.

4.2.1. Servicios Web generales

En los últimos años, han aparecido nuevos paradigmas para el diseño de aplicaciones utilizando la Web, siendo más conocidos como servicios Web. Un servicio Web se define como una aplicación que tiene la habilidad de ofrecer datos o servicios de procesamientos a otras aplicaciones (Erl, 2004). Existen diferentes estilos o arquitecturas, como por ejemplo, Simple Object Access Protocol (SOAP) (Fielding, 2000; Mein y otros, 2002) o Representational State Transfer (REST) (Pautasso y Wilde, 2010).

Tanto SOAP como REST definen cómo han de ser cursadas las peticiones de los servicios a los servidores, la forma en la cual se envían los resultados y cómo se han de publicar los servicios Web. En los servicios SOAP, la unidad básica en la comunicación es el contenido y no la operación (Silva de la Cruz, 2013). Por este motivo, usualmente son llamados servicios orientados a objetos. En relación a los servicios REST estos establecen las operaciones básicas del protocolo HTTP, como por ejemplo *GET*, *PUT*, *POST* y *DELETE*. REST, se basa en el formato XML y JavaScript Object Notation (JSON) como base para el intercambio de información. Seguidamente se detalla con más profundidad este tipo de arquitectura, ya que será la empleada para la definición de los servicios en las siguientes secciones.

Interfaces mediante REST

En Fielding (2000) se define REST como: *"La transferencia de estado representacional (REST), es un estilo de arquitectura para sistemas hipertexto distribuidos.*

REST proporciona un conjunto de restricciones arquitectónicas que, cuando se aplica como un todo, hace hincapié en la escalabilidad de las interacciones de los componentes, la generalidad de interfaces, implementación independiente de los componentes y componentes de mediación, con el fin de reducir la latencia de la interacción, reforzar la seguridad y encapsular los sistemas heredados.”

REST se basa en tres conceptos diferentes, que son: la representación de los datos, los estados y las transferencias. Se detallan a continuación:

- *Representación de los datos.* Los datos son definidos y distribuidos mediante un modelo de datos. Estas representaciones son transferidas entre clientes y servidores.
- *Estados.* REST no utiliza estados en la parte servidora, por lo que deberá ser el cliente el encargado de gestionarlos. De esta forma, los estados deben ser proporcionados con la solicitud y gestionados por el cliente.
- *Transferencias.* Las representaciones y los estados se pueden transferir entre clientes y servidores.

REST es un estilo arquitectónico que se puede implementar de manera eficiente como una combinación del protocolo de transferencia de hipertexto (HTTP, en sus siglas en inglés) y TCP/IP. En el estilo REST, las peticiones HTTP se utilizan para transferir representaciones de recursos entre clientes y servidores. Unas URI se utilizan para codificar los diferentes estados.

Siguiendo el estilo REST, en este capítulo se propone una interfaz para servir los datos de sensores (Sección 4.3.1). Gracias a las URI, cada sensor puede ser identificado siguiendo los principios de IoT y WoT (Guinard y otros, 2011).

4.2.2. Interfaz para IoT: OGC SensorThings API

Como se detalla en el Capítulo 2, las nuevas tendencias en las redes de sensores apuntan a utilizar los protocolos TCP/IP y HTTP. Gracias a ello se puede tener acceso directo mediante Internet. A partir de esta situación se definieron los paradigmas de IoT y WoT. Dichos paradigmas también han sido aplicados a la hora de gestionar las redes de sensores, entre muchos otros ámbitos.

Debido a la necesidad de adaptar los estándares de sensores actuales a las nuevas tendencias y necesidades en el campo de las redes de sensores,

los organismos como OGC, proponen nuevos candidatos los cuales posibiliten la adaptación a dichos principios. Una de estas propuestas es el candidato a estandarización OGC SensorThings API. Éste puede ser considerado como una interfaz ligera para gestionar las redes de sensores si se compara con otros servicios SWE, como SOS.

OGC SensorThing API está pensado para seguir los paradigmas de WoT, ofreciendo una forma de acceso mediante los protocolos web, basado sobre la corriente de SWE y mediante un estilo arquitectónico REST. Su objetivo es proporcionar una forma estandarizada para representar el mundo real, en este caso los sensores y sus observaciones, en el mundo del IoT.

Dicho candidato cuenta con dos capas diferentes para conectar los sensores. La primera de ellas, llamada capa de modelo de datos IoT, se encarga de conectar con gran variedad de dispositivos IoT, y consiste en el modelo de datos estándar basado en describir las entidades y sus relaciones. La segunda capa define la interfaz del servicio. Para ello, hace uso de patrones URI para ofrecer los recursos mediante las operaciones CRUD (Create, Read, Update y Delete).

Con el fin de proporcionar un patrón común de acceso a los datos y capacidades de cada sensor IoT, OGC SensorThings API define un modelo de datos donde el núcleo de la entidad es el *Thing*. Cada *Thing* puede tener una o más localizaciones en el espacio y el tiempo. También, cada *Thing* puede tener uno o más flujos de datos. El modelo de datos contiene dos perfiles:

- *Sensing Profile*. Permite que los dispositivos y aplicaciones IoT puedan definirse mediante las operaciones CRUD.
- *Tasking Profile*. Permite que las aplicaciones puedan tener el control de los dispositivos mediante el servicio REST.

El modelo de datos del candidato a estándar OGC SensorThings API se muestra en la Figura 4.1. Cada *Thing* tiene una localización (*Location*) en el espacio y en el tiempo. Puede tener varios flujos de datos (*Datastreams*), que son colecciones de observaciones agrupadas por el mismo *Observed Property*. Una Observación es un evento ejecutado por el Sensor, que produce un resultado de la *Observed Property* para la característica de interés (Feature of Interest). Un *Thing* puede tener varios *Tasking Capabilities*, que son ejecutados mediante el actuador. El

usuario puede crear cualquier cantidad de tareas para ser ejecutadas en el servicio.

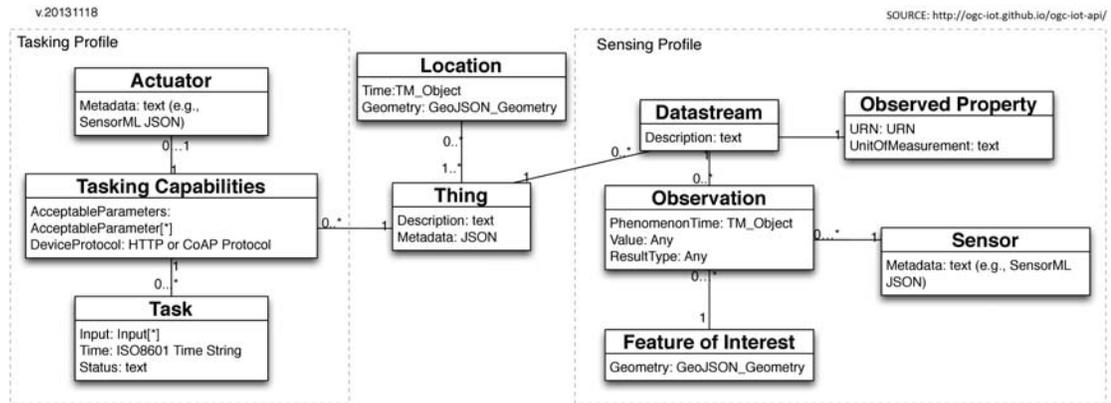


Figura 4.1: Modelo de datos en OGC SensorThings API (fuente: <http://ogc-iot.github.io/ogc-iot-api/>)

4.2.3. Mecanismos para el tiempo real

Hoy en día, el mundo digital está generando grandes cantidades de datos cada segundo (Mayer-Schönberger y Cukier, 2013), siendo gran parte de estos generados por sensores. La forma de publicar datos en Internet ha cambiado en los últimos años (Gubbi y otros, 2013). Internet se ha movido de las tradicionales páginas estáticas hacia la web 2.0. En el pasado, los contenidos ofrecidos a través de Internet eran casi estáticos, es decir, no tenían altas tasas de actualización (refresco). Actualmente, el contenido se actualiza de forma constante. Este cambio se manifiesta, por ejemplo, en el uso de los espacios personales. La popularización de los blogs introdujo actualizaciones diarias. Más recientemente podemos considerar las redes sociales, como por ejemplo Twitter, como un servicio de micro-blogging, donde los datos se están actualizando cada minuto. Otra tendencia que también ha favorecido el aumento de la tasa de refresco, es aplicando el protocolo TCP/IP para conectar las redes de sensores (Bahga y Madiseti, 2014). Dichas dos tendencias han producido lo que se ha llamado en la actualidad como Tsunami de datos (Minelli y otros, 2012).

Existen varias herramientas para gestionar contenido con altas tasas de refresco, siendo una de ellas el llamado Message Oriented Middleware (MOM) (Curry,

2004). MOM es un tipo de software capaz de transferir mensajes entre clientes en un sistema distribuido. Tradicionalmente, MOM se ha utilizado para establecer canales (flujos) entre clientes, con el fin de ofrecer los mensajes en tiempo real. MOM tiene las siguientes características: (1) comunicación síncrona y asíncrona; (2) transformación del formato de datos; (3) flexibilidad para la comunicación; (4) procesamiento en paralelo de los mensajes; (5) varios niveles de calidad del servicio (en inglés QoS) y (6) la entrega de datos en tiempo real a través de multicast. En el contexto MOM, existen diferentes conceptos importantes para entender su funcionamiento. A continuación se detalla cada uno de ellos:

- *Proveedor*. Se trata de una instancia de alguna de las implementaciones disponibles de MOM.
- *Mensaje*. Un *Mensaje* puede ser cualquier objeto o contenido que necesita ser transportado.
- *Cliente*. Una aplicación o proceso que produce y/o recibe *Mensajes*.
- *Productor / Editor*. Un *Cliente* que crea y envía *Mensajes*.
- *Consumidor / Subscriptor*. Un *Cliente* que recibe *Mensajes*.
- *Cola*. Un espacio para almacenar temporalmente los *Mensajes* que se han enviado y están esperando a ser leídos. Una *Cola* garantiza que cada *Mensaje* se procese sólo una vez.
- *Tópico*. Forma de distribución para la publicación de los *Mensajes* que se entregan a múltiples *Subscriptores*.

MOM ofrece los siguientes paradigmas de comunicación:

1. *Punto a punto*. En este modelo los *Mensajes* son enviados por el *Productor* al *Consumidor* mediante una *Cola*. En este caso los *Mensajes* no desaparecen hasta que el *Consumidor* los procese (Figura 4.2 a)).
2. *Publicación / Suscripción*. Este modelo soporta la publicación de *Mensajes* a un *Tópico* en particular. Los *Subscriptores* deben registrarse para recibir los *Mensajes* que se envíen a partir de ese momento al *Tópico* (Figura 4.2 b)).

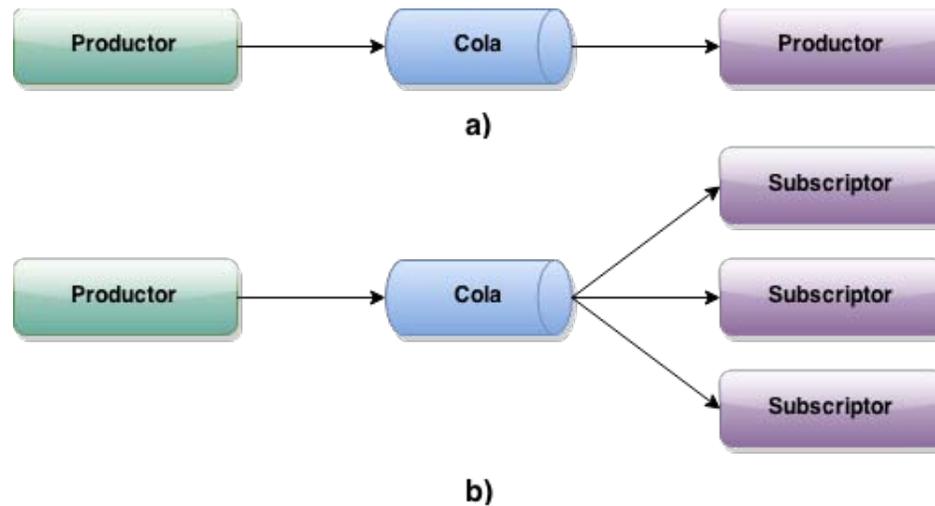


Figura 4.2: Modelos de comunicación. a) Punto a punto. b) Publicación/Suscripción.

Una implementación estándar de MOM es Java Message Service (JMS) (Hapner y otros, 2002). JMS describe las interfaces para comunicar a los productores con los consumidores. Existen diferentes herramientas que siguen JMS y habitualmente son llamados Message Brokers. Algunos de ellos son:

- *ActiveMQ*. Es el tipo de Message Broker más utilizado. Implementa la especificación JMS y ofrece numerosas características como el soporte para los patrones de integración empresarial, como el framework Spring. Por motivos de persistencia de mensajes, se utilizan técnicas de redundancia de datos. Utilizan una base de datos basada en archivos Kaha y ofrece un rendimiento muy bueno ya que se ejecuta en la JVM del broker y requiere muy pocos recursos, además de estar optimizada para los mensajes.
- *Apollo*. Es uno de los Message Brokers más recientes, basado en la experiencia del proyecto ActiveMQ, con el fin de ser más rápido, más robusto y más fácil de mantener que ActiveMQ. Esa arquitectura se basa en el lenguaje de programación Scala, que soporta bien el desarrollo de sistemas concurrentes. Apollo considera dos almacenes diferentes de mensajes persistentes. Por un lado una LevelDB NoSQL y por otro una base de datos Berkeley.
- *HornetQ*. Era conocido bajo el nombre de JBoss Messaging hasta que se creó un proyecto independiente bajo el nombre de HornetQ. HornetQ se

puede utilizar independientemente del servidor de aplicaciones JBoss. Tiene diferentes características y gran variedad de ajustes y es una implementación completa de la especificación JMS.

- *RabbitMQ*. Es un Message Broker ligero, fiable, escalable y portátil. Pero a diferencia de muchos de Message Brokers, no está basado en JMS. En su lugar, las aplicaciones se comunican con él a través de un protocolo a nivel de cable.
- *Kafka*. Es un Message Broker y ha sido diseñado originalmente por LinkedIn. Está escrito en Scala y no sigue las normas de JMS, por lo tanto, ofrece un protocolo propio para conectar con los clientes. Kafka está diseñado para el procesamiento de datos de la secuencia de actividad en tiempo real. Tiene las siguientes características: mensajería persistente, de alto rendimiento, compatible tanto con la semántica de temas, como con la de colas. El sistema de Kafka divide los mensajes con el fin de separar y paralizar las tareas de computación.

JMS sólo describe la especificación y no define el formato de los mensajes que se intercambian, por esta razón los sistemas JMS no puede considerarse un mecanismo completamente interoperable. Para mejorar la interoperabilidad de los Message Brokers se ofrecen múltiples protocolos. Cada Message Brokers ofrece diferentes protocolos de conexión, como AMQP², Message Queue Telemetry Transport (MQTT)³, OpenWire⁴, REST, simple (o Streaming) orientada a texto del Protocolo de mensajes (STOMP)⁵ o Extensible Messaging and Presence Protocol (XMPP)⁶. A continuación se detallan cada uno de ellos. La Tabla 4.1 muestra qué protocolos soporta cada Message Brokers.

- *AMQP*. Permite la colaboración de los agentes y clientes de diferentes fabricantes y plataformas. En él, se describe un formato binario y la lógica necesaria para la aplicación. Actualmente hay muchas empresas que utilizan AMQP, pero hay algunas críticas debido a su complejidad.

²<http://www.amqp.org>

³<http://mqtt.org>

⁴<http://activemq.apache.org/openwire.html>

⁵<http://activemq.apache.org/stomp.html>

⁶<http://xmpp.org>

	<i>ActiveMQ</i>	<i>Apollo</i>	<i>HornetQ</i>	<i>RabbitMQ</i>
AMQP	Si	Si	No	Si
MQTT	Si	Si	No	Si
OpenWire	Si	Si	No	No
Rest	Si	Si	Si	Si
STOMP	Si	Si	Si	Si
XMPP	Si	No	No	No

Tabla 4.1: Comparativa entre protocolos disponibles en los Message Brokers.

- *STOMP*. Ofrece un formato interoperable que permite a los clientes comunicarse con casi todos Message Broker disponibles. STOMP es fácil de implementar y ofrece flexibilidad, ya que es independiente del idioma, es decir, clientes desarrollados en diferentes idiomas pueden enviar y recibir mensajes.
- *MQTT*. Es un protocolo de mensajería para su uso en la parte superior del protocolo TCP/IP. Está diseñado para conexiones con lugares remotos donde exista un ancho de banda limitado. En 2013 IBM presentó MQTT v3.1 a la organización, OASIS realizó su estandarización.
- *XMPP*. Es un protocolo de comunicaciones para JMS basado en XML. El protocolo fue originalmente llamado Jabber, y fue desarrollado por la comunidad de código abierto Jabber en 1999.

4.3. Interfaces para acceder a sensores

En esta sección se detallan las interfaces diseñadas para servir los datos de sensores. Cada una de ellas está pensada para ser ejecutada en un entorno en concreto. La primera Sección 4.3.1 es basada en algunos de los estándares SWE detallados en las secciones anteriores y se caracteriza por utilizar dichas interfaces en un ámbito más genérico, aunque está pensada para entornos donde haya prestaciones reducidas, como teléfonos móviles. La segunda interfaz (Sección 4.3.2) que se propone será utilizada para adaptarse a los nuevos paradigmas de IoT y WoT. Finalmente, la tercera interfaz propuesta (Sección 4.3.3), está pensada

para dar acceso a grandes cantidades de datos en tiempo real. Seguidamente se detallan cada una de ellas.

4.3.1. Interfaz ligera para datos de sensores

Las interfaces diseñadas trabajan sobre los diferentes servicios geospaciales de descarga (*Download*) y visualización (*View*) (Figura 4.3). El primero de ellos, el servicio de descarga, permite obtener las observaciones de los sensores, mientras que el servicio de visualización se utiliza para la representación de los datos de dichos sensores. A continuación se define más concretamente cada uno ellos.

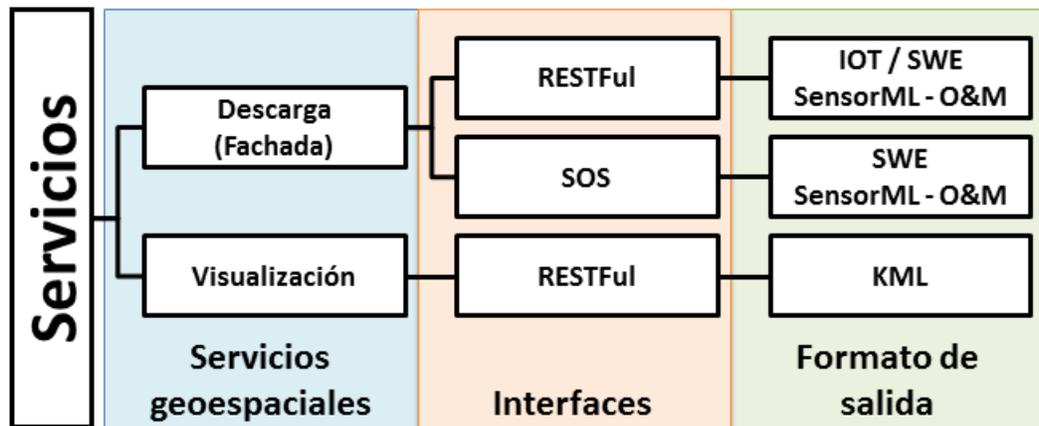


Figura 4.3: Esquema general con todos los servicios, interfaces y formatos de salida.

Servicio de descarga

Como ya se ha indicado, se ofrece una interfaz ligera para la descarga de datos de sensores. Dicha interfaz sigue la arquitectura REST y está adaptada a los estándares SWE. Además, se ofrece un servicio SOS para garantizar la interoperabilidad del sistema, por si es requerido para algún caso en particular. Seguidamente se detallan dichos servicios.

Para conseguir esta finalidad, se sigue un patrón Fachada (Schmidt y otros, 2013) encargado de distribuir el acceso a las dos interfaces nombradas. Con el fin de elegir una interfaz u otra, el patrón Fachada comprueba los tipos de parámetros y elige la interfaz correcta. De esta manera, se ofrece un único punto de entrada a las dos interfaces.

Este sistema ofrece a dichas dos interfaces para un mismo uso, ofrecer observaciones de sensores. La interfaz REST ofrece un acceso ligero, por lo que se puede utilizar en entornos en los que las funciones se encuentran restringidas, aunque esta interfaz también está basada en estándares para los formatos de salida.

Por otro lado, la interfaz SOS busca proporcionar acceso completo a las mismas fuentes de datos pero en ambientes donde no hay restricciones. También ofrece algunas de las operaciones que la interfaz REST no soporta.

La decisión de ofrecer dos interfaces diferentes se debe, como se concluye en Tamayo y otros (2011), al consumo de datos provenientes de un servidor SOS que tiene un bajo rendimiento en dispositivos móviles. De esta manera, la propuesta diseñada ofrece una solución más ligera para el acceso de datos de sensores sin perder las ventajas que proporciona un servicio completo como es SOS. A continuación se detalla cada una de ellas.

Interfaz REST Para ofrecer las funcionalidades del servicio de descarga se ha diseñado una interfaz REST para así poder obtener las observaciones de los sensores. Dicha interfaz facilita el acceso desde cualquier dispositivo, incluso desde teléfonos móviles con características limitadas (Mizouni y otros, 2011; Aijaz y otros, 2009).

Para ello se ha utilizado un servicio web REST que sigue el paradigma de la IoT y más específicamente el de WoT. El estilo arquitectónico REST permite acceder a los recursos de los sensores mediante una URI. El servicio proporciona la lista con todos los sensores y ofrece las observaciones. Los sensores disponibles son listados mediante el punto de entrada de la interfaz. El formato de salida es JSON, que es un formato de intercambio de datos basado en texto ligero, facilita su comprensión tanto para máquinas como para seres humanos (Crockford, 2006).

Los datos del sensor se devuelven al acceder al recurso correspondiente a través de su URI. El patrón del esquema de URI se define como:

http://<servidor>/<sensorId>/<metodo>

- *<servidor>*. El punto de entrada a la interfaz. Proporciona un conjunto de sensores que han sido añadidos en el sistema.
- *<sensorId>*. Se refiere a un identificador para un sensor específico. Cuando

se accede a este recurso debe enumerar una colección de todos los métodos disponibles que se aplican para este sensor.

- *<método>*. Representa el método de interacción con el sensor.

Por ejemplo, un método llamado *data* enumera las observaciones del sensor indicado. Este método también ofrece la posibilidad de proporcionar varias observaciones entre dos fechas introducidas como parámetros (*dateINIT*, *dateEND*). En este caso, el servicio devuelve una lista con las observaciones. Esta interfaz se puede ampliar con otras características, como la configuración o calibración; estos se añadirán en el futuro.

En el estado actual, son posibles dos tipos diferentes de salida. Por un lado, el punto de entrada enumera todos los sensores disponibles que el sistema contiene. Por otro lado, un único elemento que representa la última observación del sensor. Los atributos de cada sensor conectado pueden ser descubiertos y evaluados de forma automática, mediante el análisis de la información recogida en el punto de entrada y el elemento de datos de la plataforma de sensores. La Tabla 4.2 muestra los URI diseñados por los recursos y los métodos.

<i>Petición</i>	<i>Respuesta</i>
GET SensorREST/	Proporciona un conjunto de sensores que están unidos a la plataforma.
GET SensorREST/codigo	Proporciona un conjunto de métodos disponibles para este código de sensor.
GET SensorREST/codigo/metodo	Proporciona los valores para el método indicado.

Tabla 4.2: URI y métodos diseñados para proporcionar acceso a los sensores.

Punto de entrada El punto de entrada se encarga de listar la colección de sensores registrados en el sistema (Figura 4.4 a)). Por cada sensor, se suministra el ID del sensor y una URI para obtener la información sobre el sensor correspondiente, además de una descripción del propio sensor.

La URI para obtener información del sensor genera un SensorML (Figura 4.4 c)) para describir el sensor. El SensorML se codifica mediante un fichero estructurado XML siguiendo el estándar OGC. Se ha decidido utilizar una versión reducida del SensorML con los campos más generales. Para ello, se han utilizado cinco campos que son: identificación, descripción, localización, entradas y salidas de cada

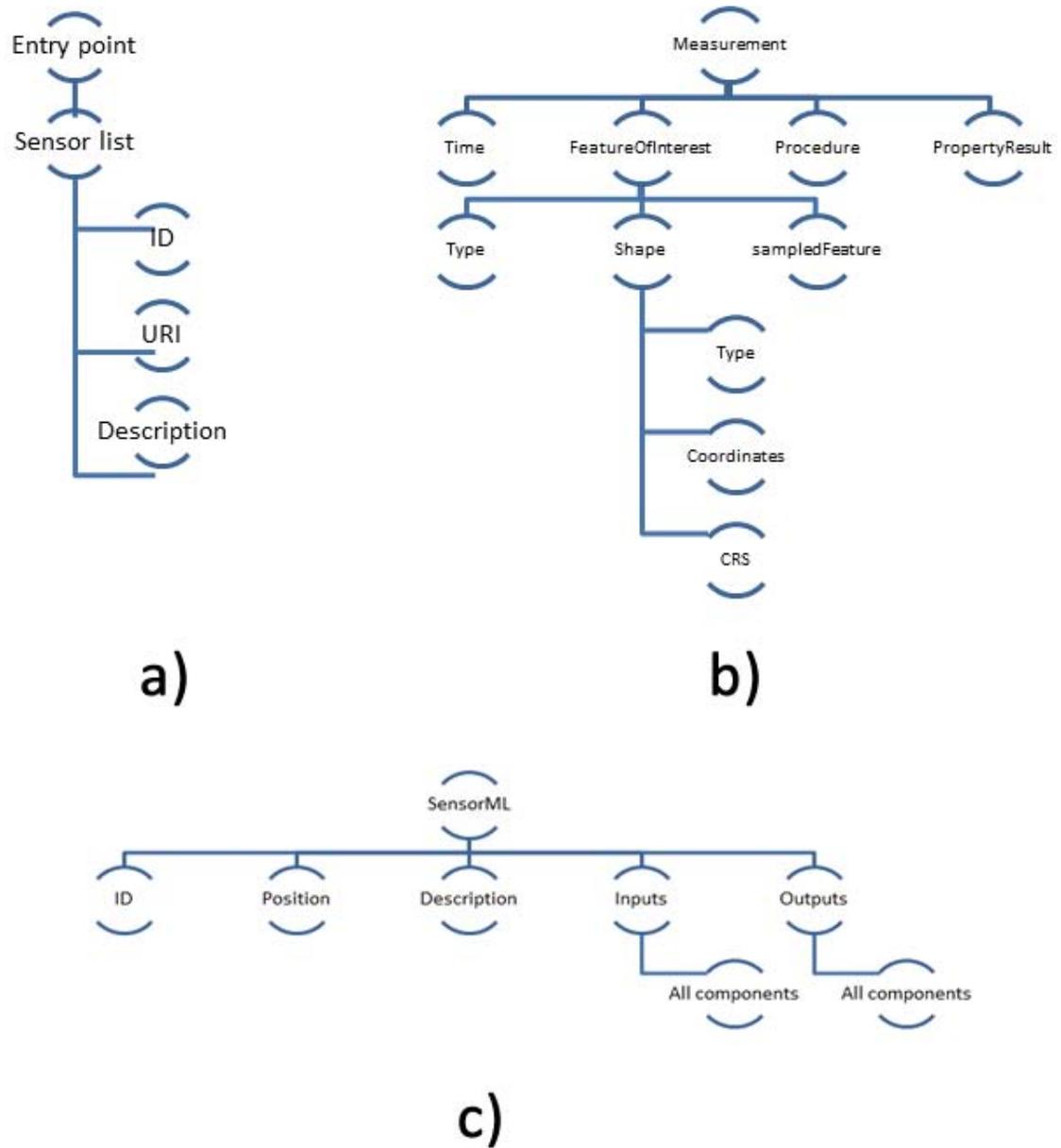


Figura 4.4: Representación de los elementos de la interfaz. a) Punto de entrada. b) Elemento. c) Descripción del sensor.

componente del sensor. El uso de dicho estándar aumenta la interoperabilidad de la interfaz, aunque únicamente se ha utilizado el estándar a modo de intercambio de codificación.

Elemento El elemento de datos (Figura 4.4 (b)) representa la última observación del sensor con el identificador indicado, codificada mediante el estándar O&M.

El modelo O&M se utiliza para representar e intercambiar resultados. Una observación es un evento que ocurre en una determinada posición espacio-temporal y genera un valor para el fenómeno observado. La salida tiene el formato de JSON en lugar de la representación típica que es XML.

Sensor Observation Service Además de la interfaz REST, este sistema también proporciona el acceso a un servidor SOS. De esta manera se garantiza ofrecer también la opción completa de los estándares OGC, con la finalidad de alcanzar una interfaz interoperable. El estándar SOS tiene tres operaciones que proporcionan su funcionalidad básica: *GetCapabilities*, *DescribeSensor* y *GetObservation*.

La operación *GetCapabilities* devuelve una descripción del servicio que contiene información acerca de la interfaz y los datos de los sensores disponibles. La operación *DescribeSensor* devuelve una descripción de un sensor específico, de un sistema de sensores o los datos producidos de los procedimientos como la posición del sensor, la calibración, entradas y salidas, etc. La respuesta es codificada en SensorML. La operación *GetObservation* proporciona el acceso a las observaciones de los sensores.

Servicio de visualización

Como se ha comentado, también se proporciona un servicio de visualización. Dicho servicio permite visualizar las observaciones de los sensores dentro de un mapa. Para ello, se ha diseñado una operación en la interfaz REST, que es capaz de exportar los datos a un archivo utilizando el KML (Figura 4.5). Esta operación tiene 2 parámetros, *DateINIT* con la fecha inicial y *DateEND* con la fecha final, siendo aplicada a un sensor en particular. Dicha funcionalidad es ofrecida mediante un nuevo método.

El archivo KML, generado mediante gráficos de columnas, visualiza los valores para los sensores. Estas columnas se muestran en diferentes colores basados en los seis niveles definidos: bueno, moderado, poco saludable para grupos sensibles, insalubres, muy insalubres y peligrosos. Estos niveles se definen siguiendo el

índice AQI⁷, para los fenómenos de calidad del aire. Para el resto de fenómenos no se utiliza una escala basada en niveles. El KML generado permite que sea reproducida en Google Earth y las representaciones de las observaciones van aumentando y disminuyendo en el tiempo.

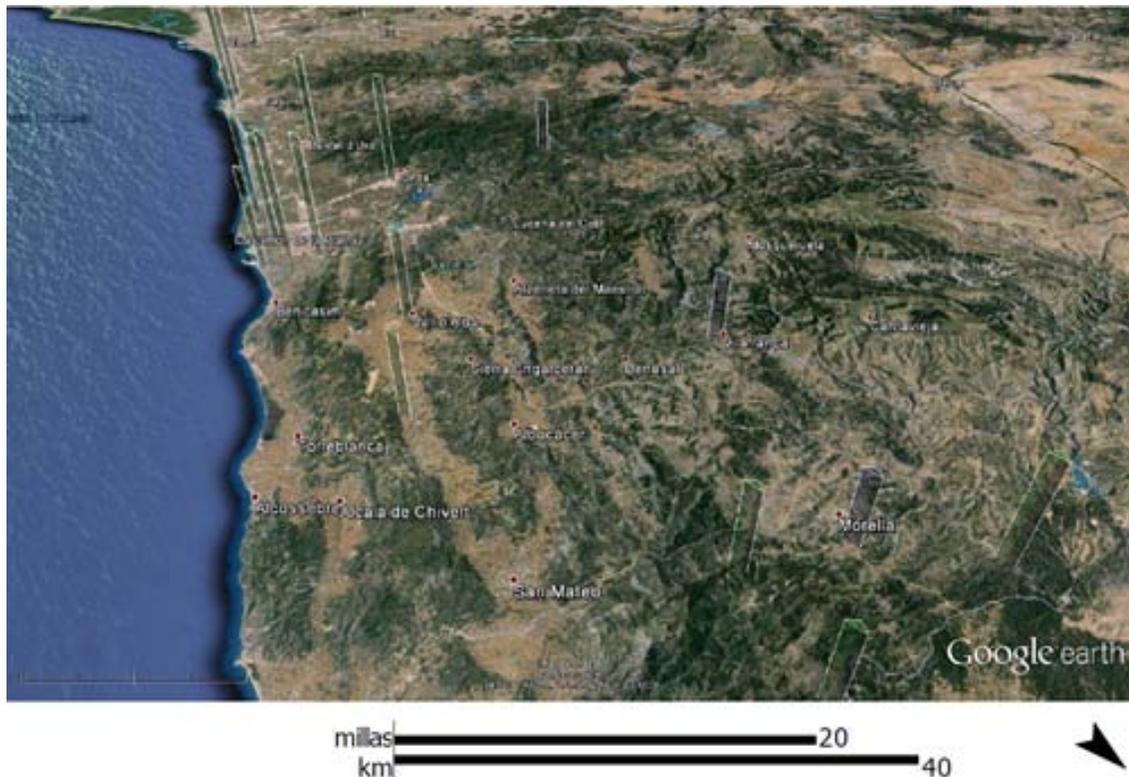


Figura 4.5: Ejemplo de visualización para el servicio diseñado.

4.3.2. OGC SensorThings API

En esta sección se detalla brevemente como se utiliza la interfaz candidata OGC SensorThings API, ya que para el trabajo actual no se realiza ninguna modificación sobre ella, sino que únicamente se utiliza para dar acceso a los sensores.

Siguiendo los principios REST, OGC SensorThings API utiliza URI para acceder a los diferentes recursos. La Figura 4.6 muestra los tres componentes definidos por la URI del servicio REST, que son: la raíz, el recurso y las opciones de

⁷<http://airnow.gov/index.cfm?action=aqibasics.aqi>

consulta. Cuando los usuarios realizan una operación de lectura sobre un recurso, se ofrecen algunas opciones de consulta, como por ejemplo, ordenar o filtrar por diferentes criterios. De esta manera, el elemento más importante son las URIs, en él recae la importancia, pues gracias a ellas se puede acceder a los sensores disponibles a través del servicio de OGC SensorThings API.

`http://serviceURI/Things(1)/Sensors?$stop=2&orderby=ID`

URI del servicio
Ruta del recurso
Opciones para filtros

Figura 4.6: Patrón URI de OGC SensorThings API.

Figura 4.7 muestra un ejemplo del uso de OGC SensorThings API para *SEnviro Thing* (Capítulo 2).

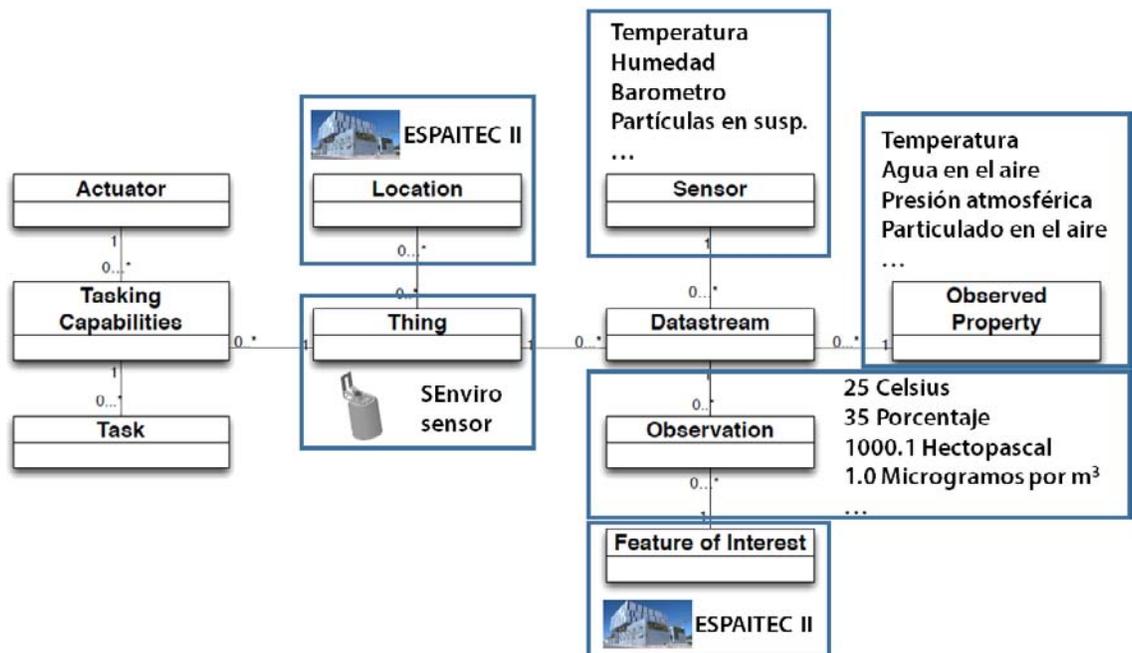


Figura 4.7: Ejemplo de OGC SensorThings API basada en una plataforma *SEnviro*.

Cada *Thing* puede estar asociada a una o más ubicaciones; en este caso el *Location* es la referencia geográfica al lugar donde está instalado el sensor. La codificación se realiza mediante el formato geoJSON⁸. Un *Thing* puede tener

⁸<http://geojson.org/>

muchos *Datastreams*. Un *Datastream* contiene información de un fenómeno. En este caso en particular, hay un *Datastream* para el fenómeno de cada sensor. Cada *Datastream* contiene un *Sensor* y un *ObservedProperty*. La primera se refiere a cada uno de los instrumentos que pueden observar un fenómeno; en el ejemplo, estos serían la temperatura, humedad, presión atmosférica y así sucesivamente. Un *ObservedProperty* especifica el fenómeno y también contiene la unidad de medida. Un *Datastream* puede tener varios *Observations* que indican el valor de estos fenómenos. Está codificado usando el formato O&M. En el ejemplo, son los valores de medición del sensor.

Finalmente, el *FeatureOfInterest* identifica las características de la *Thing*. La Tabla 4.3 muestra todas las entidades que se han utilizado, donde cada propiedad indica el tipo de formato con el que se ha codificado.

<i>Entidad</i>	<i>Propiedad</i>	<i>Tipo</i>
<i>Thing</i>	Descripción	Character String
<i>Location</i>	Tiempo	ISO8601
<i>Location</i>	Geometría	GeoJSON geometry
<i>Datastream</i>	Descripción	Character String
<i>ObservedProperty</i>	URI	URI
<i>ObservedProperty</i>	UnitOfMeasurement	Character String
<i>Observation</i>	Tiempo	ISO8601
<i>Observation</i>	ResultType	O&M Result Type
<i>Observation</i>	ResultValue	Any (depends on the ResultType)
<i>Sensor</i>	Metadato	Character String
<i>FeatureOfInterest</i>	Descripción	Character String
<i>FeatureOfInterest</i>	Geometría	GeoJSON geometry

Tabla 4.3: Propiedades de perfil de OGC SensorThings API.

4.3.3. Servicios para el acceso a datos de sensores con alta tasa de refresco

En esta sección se presentan los servicios definidos para servir una gran cantidad de observaciones. Dichos servicios son ideales para sensores con altas tasas de refresco. Los servicios ofrecen las observaciones de los sensores en

tiempo real, es decir, tan pronto como se generen los nuevos valores, estos estarán disponibles. De esta manera se crean flujos de datos por cada sensor.

Tradicionalmente, para acceder a este tipo de datos se utiliza el protocolo HTTP, con el que el cliente realiza peticiones a un servidor y se devuelve el resultado, en este caso la observación. Este tipo de comunicación se llama petición-respuesta, el cual debe repetirse cada vez que el cliente quiera obtener una nueva observación. Para fuentes con altas tasas de actualización, como son las de sensores, requieren frecuentes peticiones al servidor con la finalidad de obtener los valores más recientes.

Existen enfoques de comunicación más eficaces y eficientes para hacer frente en estos casos. Uno de los mecanismos que favorece la comunicación cuando existe una alta tasa de actualización son los mecanismos de *long polling*, donde el cliente envía de forma repetitiva nuevas peticiones al servidor. Si el servidor no tiene nuevos datos, entonces se envía el mensaje de finalización y se cierra la conexión. Pasado cierto tiempo, el cliente realiza una nueva petición para recibir los nuevos valores.

Para dar solución a estos objetivos, se ha diseñado un componente transversal siguiendo dicho principio, que será utilizado para transferir las observaciones de los sensores. Para ello, se utilizará un Message Broker (Sección 4.2.2), donde dicho componente será llamado como *RealTime Message Service* (RMS) (Figura 4.8).

Como se ha comentado, los Message Brokers tienen varios protocolos de conexión, algunos de ellos siguen las propiedades ya descritas y ofrecen conexiones *long polling*. Dicho componente será utilizado en diferentes partes de la interfaz presentada en esta sección, con el fin de ofrecer los diversos flujos de observaciones.

Utilizando el RMS, se define un componente llamado *Sensor Data Streaming* (SDS), el cual tiene dos objetivos claramente marcados. Por un lado, el SDS permite conectarse con diferentes fuentes de datos heterogéneas, de las categorías definidas en el Capítulo 2. Para ello se utilizan las técnicas de integración ETL. El SDS realiza la función de bróker (Buschmann y otros, 1996), ya que admite fuentes de entrada con diferentes naturalezas. De esta forma, el SDS define conectores específicos para cada una de las fuentes. Para las redes de sensores que siguen un estándar, dicho conector puede ser reutilizado. De esta

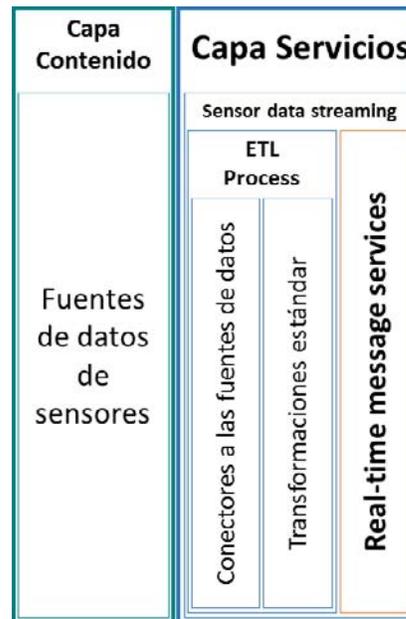


Figura 4.8: Visión general del Sensor Data Streaming.

manera, el SDS contiene la información necesaria para establecer los diferentes conectores. Además, esta información contiene los enlaces para establecer la conexión y la frecuencia de actualización de los sensores. Gracias a ellos se realizarán conexiones periódicas para obtener los nuevos datos. Dichas fuentes de datos serán accesibles mediante formatos y protocolos distintos.

Por otro lado, el segundo objetivo es aplicar las transformaciones necesarias para obtener los datos de los sensores en un formato estándar, con el fin de garantizar la interoperabilidad a nivel de codificación. Para ello, se aplicará un proceso ETL sobre cada observación perteneciente a una fuente no estándar. La Figura 4.8 muestra la visión general del componente SDS.

Para crear los flujos de observaciones el SDS utiliza el componente RMS. Para ello, se utiliza un Message Broker con el modelo *Punto a Punto* (Figura 4.9), por lo que se garantiza la entrega del *Mensaje* al *Consumidor*. Este paso tiene como resultado un flujo (*Cola*) por sensor y nodo de sensorización, a los que los clientes se conectan para consumir las observaciones provenientes de los sensores. La Figura 4.9 ilustra este uso particular del RMS.

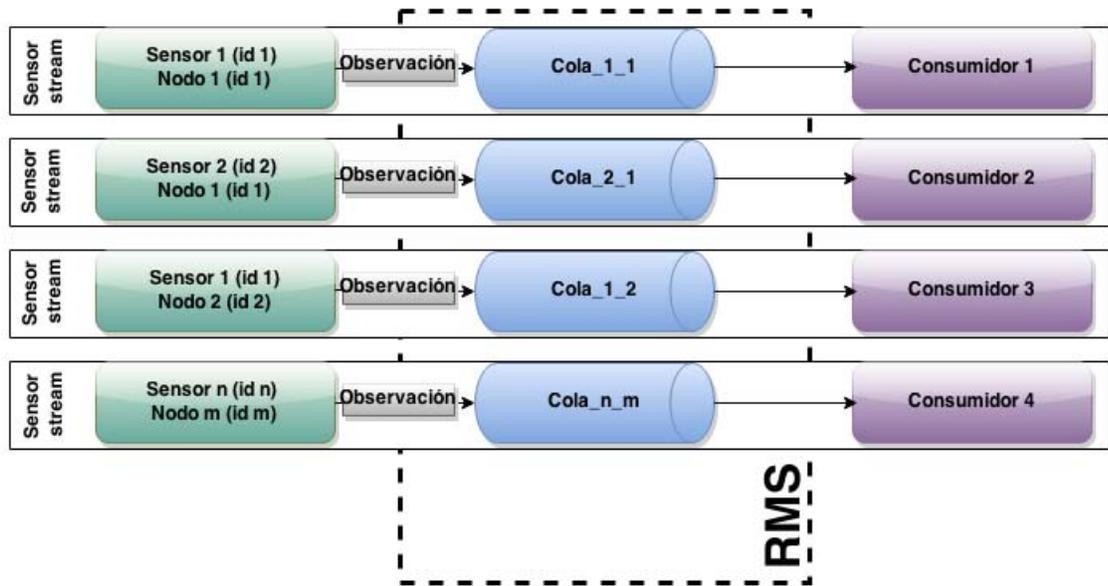


Figura 4.9: Modelo punto a punto.

4.4. Trabajo relacionado

Esta sección analiza diferentes trabajos relacionados para ofrecer datos provenientes de sensores. Los trabajos analizados se centran únicamente en aquellos que ofrecen datos de sensores en múltiples formatos. La Tabla 4.4 detalla cada uno de los estudios analizados en cinco aspectos diferentes.

Hay dos obras que proporcionan una revisión general del tema. El trabajo de Bernard y otros (2005) está dirigido a diseñar un geoportal a nivel Europeo para ofrecer datos de sensores. Bröring y otros (2011) muestra y analiza la reciente evolución de la estructura de la especificación SWE.

Otros proyectos más específicos como Li y Shue (2004) utilizan datos procedentes de sensores de calidad del aire en tiempo real. Los autores investigan un método para el análisis de los datos extraídos por 71 sensores distribuidos en Taiwán, accediendo directamente a los sensores para recuperar datos. El objetivo final de este trabajo es utilizar estos datos con el fin de analizar las partículas en suspensión (PM, en sus siglas en inglés) utilizando una red neuronal.

Alesheikh y otros (2005) trabajan con sensores de calidad del aire en tiempo real en Teherán. Este enfoque ofrece las observaciones codificadas mediante

<i>Referencia</i>	<i>SOS</i>	<i>Formato de los datos</i>	<i>Fuente de datos</i>	<i>Tiempo real</i>
Li y Shue (2004)	No	No	Sensores	No
Alesheikh y otros (2005)	No	GML	Base de datos	No
Viqueira y otros (2010)	Si	XML	Sensors, Base de datos	No
Poorazizi y otros (2011)	Si	KML, GeoRSS, GeoJSON	Base de datos	No
Pebesma y otros (2011)	Si	XML (UncerML)	Ninguno	No
Trabajo actual	Si	JSON (SensorML, O&M) y KML	Base de datos	No

Tabla 4.4: Comparación entre los diferentes trabajos.

el estándar GML y son almacenadas en una base de datos para su posterior recuperación.

El trabajo de Viqueira y otros (2010) tiene como objetivo ofrecer el acceso interoperable a datos meteorológicos utilizando un servicio de SOS. Este trabajo garantiza el acceso interoperable utilizando estándares SWE, además de ofrecer un fácil acceso a través de formatos ágiles, como XML. Estos acceden directamente a los datos de los sensores y los almacenan en una base de datos.

Un sistema en el que se insertan las observaciones de los sensores en una base de datos se propone en Poorazizi y otros (2011). Estos datos son servidos por un servidor SOS para su posterior uso en la creación de un mapa en tiempo real. El mapa utiliza cartografía de Google y se divide en regiones. Utilizan KML, RSS (GeoRSS) y geoJSON⁹ para el intercambio de datos.

Pebesma y otros (2011) ofrecen una solución interoperable que integra un servicio de SOS y crea un formato de intercambio de datos llamado UncerML. Utilizan un WPS para el procesamiento de sus observaciones.

Actualmente, existen algunos proyectos de investigación abiertos que son de especial interés, ya que ofrecen soluciones para la recolección, procesamiento y publicación de los datos provenientes de redes de sensores, como son Openiot¹⁰ o Global Sensor Networks¹¹. Pero se consideran fuera del objetivo, ya que la preocupación principal de estos proyectos es proporcionar una plataforma para los sensores y sus redes, en los que no se contempla el uso de estándares con el objetivo de ofrecer interfaces interoperables.

⁹<http://geojson.org/>

¹⁰Open Source cloud solution for the Internet of Things, <http://www.openiot.eu/>

¹¹<http://lsir.epfl.ch/research/current/gsn/>

El sistema de acceso a datos de sensores que se propone, en contraste con otros, tiene el reto de garantizar la interoperabilidad. Para lograr esto, ofrece una interfaz REST, pero con una mejora importante ya que se basa en formatos de datos de intercambio de OGC, como SensorML u O&M, a la vez que se ofrece un servidor SOS. La primera interfaz que se presenta en este capítulo, tiene como objetivo proporcionar dos servicios diferentes (visualización y descarga), manteniendo la interoperabilidad de los estándares SWE. Este enfoque se basa principalmente en considerar cada nodo sensor como un recurso REST, utilizando el formato de datos ligero JSON como una alternativa al XML. Rouached y otros (2012) realizan una comparación entre los servicios de SOS e interfaces REST, llegando a la conclusión de que las interfaces REST son mucho más ligeras. Según Poorazizi y otros (2012); Tamayo y otros (2011) el servidor SOS cumple los estándares OGC y proporciona interoperabilidad, pero cuando contiene muchas observaciones su rendimiento disminuye considerablemente.

Finalmente se ofrecen también dos interfaces para diferentes ámbitos. La primera de ellas describe un candidato a estándar de OGC llamado SensorThings API, el cual está basado en los actuales SWE y define una interfaz REST siguiendo los paradigmas de IoT y WoT. Y la segunda propone una metodología para servir datos de sensores cuando se generan con altas tasas de refresco, aplicando herramientas de Message Brokers en el ambiente de los sensores.

4.5. Conclusiones

En este capítulo de la tesis se han presentado diferentes formas de acceder a los datos de sensores. Cada una de las formas se especifica para un ámbito diferente, para cubrir unos requisitos concretos. Uno de los principales problemas es la variedad de formatos definidos para servir los datos de sensores, además de ofrecer dichos datos mediante estándares. Parte de este esfuerzo es llevado a cabo por las grandes organizaciones de estandarización, como OGC o ISO.

Desafortunadamente, cuando se definen nuevos estándares, se busca ofrecer la máxima universalidad y generalización posible. Esta situación conlleva al aumento de la complejidad de los mismos. Cuando estos estándares han de consumirse en entornos donde las características de computación son reducidas, el rendimiento global no es el adecuado (Tamayo y otros, 2011).

Ante estas dos situaciones, se define un servicio web como primera aportación

para el acceso a datos de sensores. Éste sigue un patrón Fachada que es capaz de ofrecer dos interfaces diferentes, dependiendo de las necesidades. Por una parte se propone un interfaz REST capaz de representar cada sensor mediante una URI, siguiendo los principios de IoT y WoT. Además, también se contempla el uso de estándares como la codificación de las observaciones y de los sensores, mediante el uso de los estándares O&M y SensorML. Dicha interfaz es creada para entornos donde no exista suficiente potencia de procesamiento, como pueden ser teléfonos móviles (Aijaz y otros, 2009). Por otra parte, se ofrece también un servidor SOS para ofrecer todas las propiedades del estándar.

Como segundo objetivo de este capítulo, se utiliza una interfaz candidata a estándar OGC que sigue los paradigmas de IoT y WoT basándose en los estándares SWE. Dicha interfaz ofrece un acceso fácil y ágil a los datos de los sensores. A pesar de que es un candidato a estándar OGC, tiene alta probabilidad de ser uno de los primeros estándares para redes de sensores que siga el paradigma de IoT.

Como último objetivo del capítulo, se presentan herramientas para situaciones donde las observaciones se generen con altas tasas de refresco. Dichas herramientas ofrecen la posibilidad de establecer flujos de observaciones, ofreciendo diferentes puntos de accesos y donde cada uno de ellos ofrece múltiples protocolos. Estas herramientas son realmente útiles cuando se están produciendo constantes observaciones, ya que no siguen el mecanismo de petición-respuesta, sino que se establece un canal entre el servidor y el cliente con el fin de tener siempre actualizado éste último. Para ello se aplican los Message Brokers para el consumo de este tipo de datos. Estas herramientas serán utilizadas para las metodologías definidas en el próximo capítulo, donde en una de ellas se realizan diferentes análisis sobre gran cantidad de datos en tiempo real.

Capítulo 5

Análisis

Tras la etapa de acceso (Capítulo 4) en la que se ha detallado las diferentes interfaces para consumir datos de sensores, se presenta la última etapa que se define en el ciclo de vida de datos de sensores, que es el análisis. En esta fase se diseñan diferentes metodologías para el procesado de datos de sensores, con la finalidad de obtener algún tipo de resultado que aumente el valor de los datos iniciales. Para tal fin se presentan dos metodologías de trabajo completamente diferenciadas. En primer lugar, una metodología para el análisis en tiempo real de grandes cantidades de datos de sensores. Y en segundo lugar, se describe cómo realizar cualquier tipo de análisis, siguiendo los estándares GIS para el procesamiento de contenido geoespacial. El capítulo se compone de las secciones siguientes. La Sección 5.1 muestra una pequeña introducción para el análisis de datos de sensores y datos geoespaciales en general. La segunda (Sección 5.2) detalla los conceptos necesarios para el correcto desarrollo del capítulo. La Sección 5.3 diseña una metodología para el análisis de grandes cantidades de datos en tiempo real. La cuarta (Sección 5.4) presenta el análisis mediante estándares geoespaciales. La Sección 5.5 realiza un estudio de los diferentes trabajos que actualmente se encuentran en la literatura. Finalmente, en la Sección 5.6 se detallan las conclusiones devenidas en este capítulo.

5.1. Introducción

En este capítulo se llega a la última fase del ciclo de vida para los datos de sensores definido en la introducción (Capítulo 1), que es el análisis. El análisis

sobre datos de sensores es muy interesante, ya que permite obtener valor añadido a los datos. Para ello, es necesario ofrecer diferentes técnicas para facilitar la aplicación de los modelos científicos sobre este tipo de datos.

Hoy en día, como se comentó en anteriores capítulos, se ha disparado el aumento de redes de sensores con el fin de monitorizar el estado actual de variables o conjuntos de variables de interés. Dicho aumento, en parte, es debido al aumento de la concienciación de la sociedad por los diferentes problemas medioambientales, como el cambio climático (White y otros, 2005).

Actualmente, existen modelos que a partir de los datos actuales pueden alertar sobre situaciones de riesgo o incluso predecir posibles amenazas en el futuro. Desafortunadamente estos modelos se ejecutan en entornos muy particulares, lo cual imposibilita su reutilización y publicación (Granell y otros, 2010).

Otra de las necesidades actuales, es obtener cualquier tipo de valor añadido sobre las observaciones de diferentes tipos de sensores, ya sean de calidad del aire, meteorológicos o de salud, entre otros. Dicha tarea no es trivial, ya que debido a la gran cantidad de sensores que producen observaciones hace difícil obtener un mecanismo capaz de analizar cada uno de ellos y generar resultados a partir de los mismos (McAfee y Brynjolfsson, 2012).

Para ello, en este capítulo se proponen dos metodologías diferentes. La primera metodología, se basa en el análisis en tiempo real de grandes cantidades de datos (Big data) provenientes de sensores. Dicha metodología utiliza nuevas técnicas de procesamiento de gran volumen de datos, con la peculiaridad que permite aplicar progresivamente cualquier tipo de modelo y obtener los resultados a medida que los sensores los producen.

Dicho mecanismo pretende ofrecer el marco necesario para analizar una gran cantidad de datos provenientes de sensores en tiempo real. Este sistema está basado en nuevas técnicas existentes para este uso, que garantizan que todos los datos serán procesados completamente, además de permitir aplicar cualquier tipo de algoritmo sobre el flujo de entrada de datos. Este enfoque puede ser aplicado a cualquier tipo de red de sensores. Las principales características de la metodología son: (1) posibilitar diferentes herramientas para conectar con las fuentes de datos de sensores, independientemente del tipo de datos; (2) analizar grandes cantidades de datos de los sensores en tiempo real y aplicar cualquier modelo sobre ellos; (3) conectar con diferentes protocolos mediante una solución

de bróker; 4) un mecanismo para transformar los datos de sensores a formatos estándares; (5) y un marco para servir los resultados del análisis en tiempo real con diferentes protocolos.

En la segunda metodología diseñada, se pretende que mediante el uso de un servicio de geoprocésamiento estándar, también de una forma general, ser capaz de encapsular cualquier modelo científico. El beneficio es exponer los modelos mediante un servicio web para que sean ejecutados en un entorno distribuido y desacoplarlo de un entorno de ejecución específico, pudiendo ser ejecutado desde cualquier cliente.

5.2. Conceptos Relacionados

En esta sección se detallan brevemente los conceptos necesarios para el desarrollo del presente capítulo. Con tal finalidad, la sección se divide en dos partes. La primera de ellas (Sección 5.2.1) enumera las diferentes técnicas para el análisis de grandes cantidades de datos en tiempo real. La Sección 5.2.2 describe los diferentes procesamientos actuales que siguen los estándares GIS.

5.2.1. Análisis en tiempo real

Según la Wikipedia un sistema en tiempo real es descrito como:

Es aquel sistema digital que interactúa activamente con un entorno con dinámica conocida en relación con sus entradas, salidas y restricciones temporales, para darle un correcto funcionamiento de acuerdo con los conceptos de predictibilidad, estabilidad, controlabilidad y alcanzabilidad.

Ofrecer un sistema capaz de cumplir con esta definición para el análisis de datos de sensores supone un gran desafío. Actualmente, tanto el cambio del contenido estático hacia el dinámico en la web, como el uso del protocolo TCP/IP para conectar sensores, han producido el aumento de flujos de información potencial (Capítulo 4). Dichos flujos ponen en desafío a los sistemas de análisis de datos existentes hoy en día, tanto para el análisis en tiempo real, como para el procesamiento por lotes (batch en inglés). Desencadenando la situación en la que no puedan ser gestionados mediante las herramientas y métodos disponibles, siendo

tan grande su importancia, que en dominios científicos y sectores empresariales ha sido llamado como “el problema del Big Data” (Manovich, 2012). Tres son las propiedades definidas para este tipo de problemas:

1. *Volumen*. La capacidad para procesar datos en la gama alta (terabytes o petabytes).
2. *Variedad*. La necesidad de combinar datos de diferentes formatos y fuentes.
3. *Velocidad*. La capacidad de analizar los datos en el momento en que se generan.

Para resolver este problema se requieren nuevas metodologías. Un problema de procesamiento sobre un gran volumen de datos se puede resolver a través de una paralelización masiva (Garland y otros, 2008). Esto puede ser factible gracias a dos recientes avances tecnológicos, como son la caída del precio del hardware y el uso de la computación distribuida, como la computación en la nube.

Aunque la computación distribuida ya ha sido una práctica común en la investigación y en el mundo empresarial, como en la física de partículas o la predicción del tiempo, no fue hasta la publicación de algoritmo Map-Reduce de Google (Dean y Ghemawat, 2008), donde grandes cantidades de datos también podían ser analizados en otros ámbitos sin esta tradición. A partir de este algoritmo se han desarrollado sistemas capaces de explotar el paralelismo para manejar grandes volúmenes de datos, tales como Hadoop¹. Después de Hadoop, ha habido otros proyectos que han mejorado el procesamiento de Big Data, Zookeeper² es sólo un ejemplo de ello.

La velocidad combinada con el volumen, son los dos aspectos más importantes cuando se trata de un análisis de los flujos de datos en tiempo real, donde los datos tienen que ser procesados tan pronto y tan rápido como sea posible.

Actualmente existen muchas plataformas de apoyo, como los sistemas de procesamiento de flujos distribuidos (DSPS, en sus siglas en inglés) (Sirish y otros, 2003), estos se han potenciado en los últimos años con el fin de garantizar este tipo análisis. En esencia, este tipo de plataformas tiene dos partes bien diferenciadas: un elemento de producción de los datos (Figura 5.1 círculo verde);

¹<http://hadoop.apache.org/>

²<http://zookeeper.apache.org/>

y un elemento que se encarga de procesarlos (Figura 5.1 círculos azules). El primer elemento asegura un flujo de datos continuo, que puede considerarse como una serie continua de valores. El segundo elemento aplica diferentes algoritmos de procesamiento sobre el flujo de datos generado.

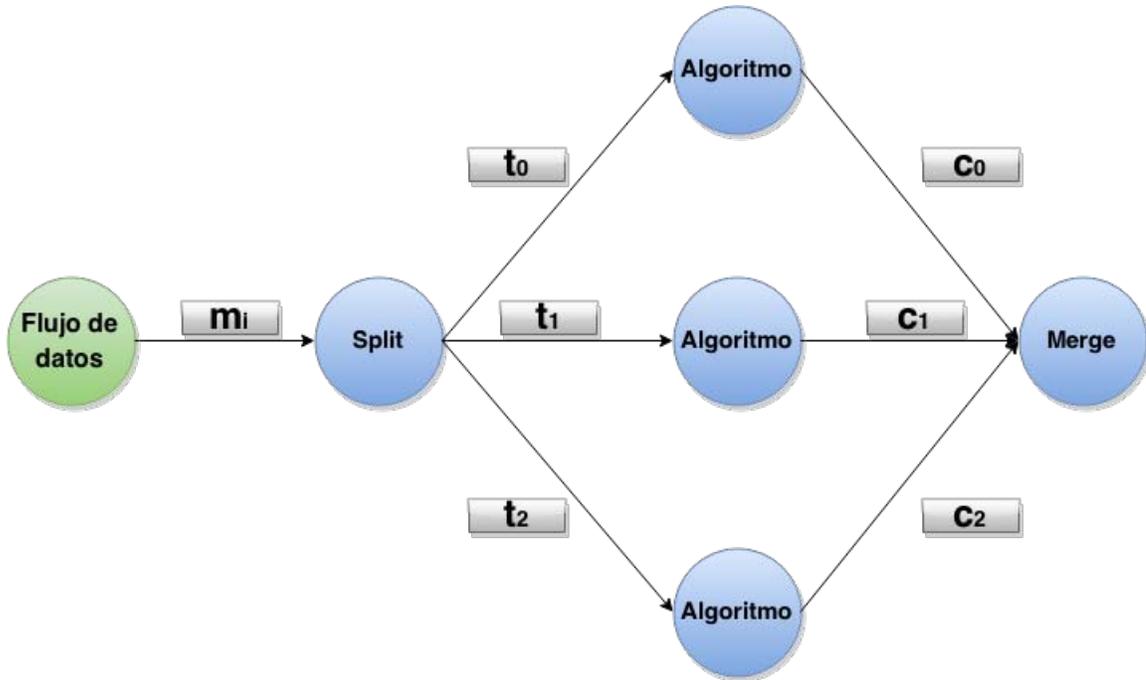


Figura 5.1: Forma de trabajar típica para procesar flujos de datos en tiempo real.

La Figura 5.1 muestra la forma natural de procesar flujos de datos. Los primeros nodos conectan con las fuentes de datos con el fin de consumir el contenido, en este caso las observaciones de los sensores. Tradicionalmente, se utilizan los Message Brokers para transmitir las observaciones (Sección 4.2.2) y crear los flujos. Estos flujos (en inglés streams) están conectados a los nodos de procesamiento, para realizar cualquier análisis. El resultado del análisis se transfiere a los procesos posteriores o se entrega como salida del sistema.

Ya hay disponibles soluciones para el análisis de datos en tiempo real, ejemplos de ellos son Borealis (Cetintemel y otros, 2006) o Millwheel (Akidau y otros, 2013), entre otros. Para este estudio, sólo se han tomado en cuenta los sistemas más extendidos, siendo estos los siguientes:

- *Storm*³. Es un marco de procesamiento distribuido de flujos en tiempo real, que se puede utilizar para analizar, filtrar y normalizar las series de datos o aplicar muchos otros procesamientos también sobre los datos en tiempo real. El marco es tolerante a fallos y garantiza que todos los datos serán procesados. Storm tiene su propia terminología, empezando por el concepto de *Topologías*. Una *Topología* define la forma en que se procesará el problema. Estas *Topologías* se componen con diferentes elementos, *Spouts* y *Bolts*. Los *Spouts* gestionan la recolección de los datos y generan las *Tuplas* en la *Topología*, además de enviar cada una de ellas a los *Bolts*. De este modo, se conectan el flujo de entrada y envían los mensajes a los *Bolts* que son los responsables del procesamiento de flujos. Cada *Bolt* procesa los flujos que recibe de un *Spout* y aplica cualquier tipo de procedimiento para generar su flujo de salida. La mejora real de Storm en comparación con otras soluciones, es que estas operaciones (*Spouts* y *Bolts*) pueden ser paralelizadas. La paralelización se maneja en el nivel de un solo *Bolt* y el factor de paralelización se define como parte de la *Topología*.
- *S4*⁴. Se caracteriza por ser de propósito general, en tiempo real, distribuida, escalable, descentralizada y modular para el procesamiento de datos en streaming. S4 emplea Zookeeper como capa de comunicación para coordinar los nodos del clúster. Zookeeper es una coordinación distribuida no tolerante a fallos que se ha convertido en una de las principales plataformas distribuidas. El procesamiento en S4 se realiza mediante el llamado elementos de procesamiento (PEs, en sus siglas en inglés). S4 transmite mensajes entre los diferentes PEs en forma de eventos. El estado de cada PE es inaccesible para las demás PEs, y la emisión y consumo de eventos son el único modo de interacción entre los diversos PEs que lo forman. Un clúster de S4 contiene PEs para procesar eventos. Puesto que los datos se transmiten entre los PEs, no se requiere control en el disco. De este modo, únicamente se proporciona una tolerancia a pérdidas de forma parcial.
- *Samza*⁵. Fue creado por LinkedIn⁶ y se ha convertido en un proyecto de Apache. Samza es un marco para ejecutar flujos de datos principalmente

³<http://storm.apache.org/>

⁴<http://incubator.apache.org/s4/>

⁵<http://samza.apache.org/>

⁶<http://www.linkedin.com/>

de Kafka⁷. Samza proporciona garantías de entrega de mensajes utilizando técnicas de copia de seguridad para el manejo de los fallos en los mensajes. Los nodos de procesamiento están representados por Samza *Jobs*. Cada uno de estos *Jobs* tiene un conjunto de flujos de entrada y de salida. Contiene una lista con los descriptores de entrada, salida y las tareas a realizar sobre los datos de entrada. Un *Job* crea una topología de red que procesa los mensajes. Las rutas se dividen en subflujos y se distribuyen a través de los *Jobs* de procesamiento en paralelo.

5.2.2. Estándares para el geoprocesamiento

Para el análisis mediante estándares de geoprocesamiento se utilizarán los conceptos ya introducidos en la Sección 3.2.3, por lo que no será necesario volverlos a presentar. En esta sección se detallan el uso de geoprocesamientos para la ejecución de procesos relacionados con las características geoespaciales mediante la Web. Como se indicó, uno de los geoprocesamientos más utilizados es WPS. Éste define diferentes reglas para describir cualquier proceso, así como la forma de realizar peticiones al servicio y la respuesta que ofrece, con la finalidad de poder reutilizar y estandarizar los diferentes procesamiento mediante un entorno distribuido.

5.3. Metodología para el análisis en tiempo real

En esta sección se detalla la metodología propuesta para analizar flujos de datos de sensores en tiempo real. Para ello, se piensa dar solución a la problemática de analizar grandes cantidades de datos (Big data) provenientes de los sensores en tiempo real, con el fin de obtener algún tipo de valor añadido sobre los datos en sí.

La Figura 5.2 presenta un resumen de la metodología propuesta. Distingue tres capas: capa de contenido, capa de servicios y capa de aplicación de contenidos. La capa de contenido (Capítulo 2) acoge las diferentes categorías de fuentes de datos de sensores que el sistema utilizará para aplicar el análisis. La capa de servicios permite conectar con las fuentes de datos del sensor, estandariza las observaciones y aplica los algoritmos para el análisis. También es la responsable

⁷<http://kafka.apache.org>

de servir los resultados obtenidos del análisis. En esta capa serán utilizadas las interfaces detalladas en la Sección 4.3.3. La última capa, aplicaciones, se conecta con la capa de servicios con el objetivo de visualizar los resultados.

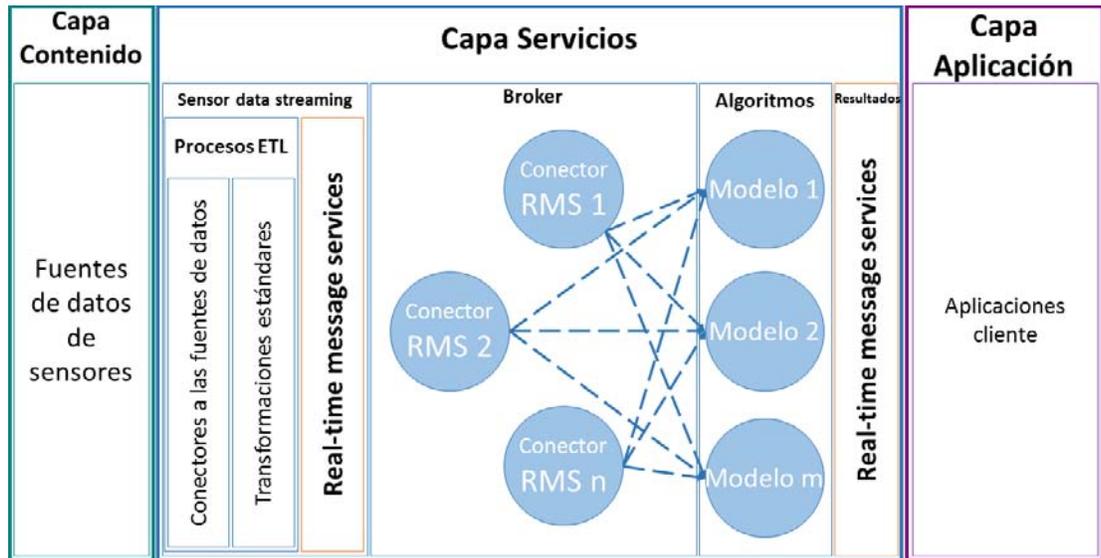


Figura 5.2: Vista general de la metodología presentada para analizar flujos de datos en tiempo real.

Referente a la capa de contenido, la metodología está diseñada para trabajar con datos suministrados por sensores. Se puede aplicar sobre cualquier tipo de fenómenos, como meteorológicos (temperatura, humedad, etc.), calidad del aire (CO₂, CO, PPM, etc.), salud (pulso, presión arterial, etc.), ciudades inteligentes (aparcamiento inteligente, luz inteligente, etc.). Así pues, se puede utilizar con otros tipos de datos, el sistema considera como entrada cualquier tipo de secuencia de datos (serie de valores). De esta forma el alcance no se limita únicamente a sensores, si no que se abre a cualquier productor de datos constantes.

Como se ha comentado en el Capítulo 2, desafortunadamente existen gran variedad de formatos a la hora de servir los datos de sensores. La metodología presentada pretende integrar todas las fuentes de datos posibles independientemente de su categoría (Sección 2.1).

La capa de servicios se conecta a las fuentes de datos del sensor, analiza las observaciones y proporciona los resultados. Esta capa se divide en cuatro componentes: Sensor de datos Streaming (SDS) (Sección 4.3.3), bróker (Sección

5.3.1), algoritmos (Sección 5.3.2) y resultados (Sección 5.3.3).

Como se vio en el capítulo anterior, se proponía un mecanismo llamado SDS, el cual se utilizaba para servir los datos de sensores en tiempo real. Este componente será utilizado como primera fase dentro de la capa de servicios. Con él, se ofrecerán los diferentes flujos de observaciones que serán analizados por los algoritmos o modelos. Como se ha comentado, el SDS realiza la tarea de integración de diferentes fuentes de sensores heterogéneas, además de ofrecerlas mediante mecanismos eficaces para datos con altas frecuencias de actualización. Para ello se utiliza el componente RMS (Sección 4.3.3).

En las siguientes secciones se definirá el enfoque de bróker (Sección 5.3.1) y cómo son aplicados los algoritmos (Sección 5.3.2).

5.3.1. Enfoque bróker

La metodología propuesta, con la finalidad de conectar con diferentes RMS (Sección 4.3.3) propone un patrón bróker (Nativi y otros, 2012; Buschmann y otros, 1996). Como muestra la Figura 5.3, el patrón bróker permite a los clientes conectarse con múltiples productores (fuentes de datos). La metodología propuesta sigue dicho patrón para conectar con el RMS, de tal modo, que permita utilizar diferentes instancias del RMS, pudiendo cada una de ellas funcionar con un protocolo diferente.

La motivación principal para usar un patrón bróker es la posibilidad de conectarse con fuentes heterogéneas y mediante diferentes protocolos, de esta forma se aumenta la flexibilidad del sistema, su escalabilidad, reusabilidad e interoperabilidad. Además, otra funcionalidad del patrón es servir como único punto de entrada cuando se apliquen los posteriores análisis.

La metodología ofrece nodos de conexión (Figura 5.2 círculos azules) para los diferentes protocolos (Sección 4.2.1), que se llaman *Conectores RMS*, los cuales tienen un papel de *Consumidor*. Para ello, cada nodo proporciona un cliente que es capaz de conectar con uno de los diferentes protocolos soportados. El bróker está pensado para ofrecer la conexión con diferentes protocolos, y no soporta múltiples formatos de codificación de datos, porque como se definió, el SDS únicamente ofrece como codificación un formato estándar. Cada *Conector RMS* establece una conexión entre los diferentes flujos de sensores proporcionados por las instancias del RMS. De esta manera, las observaciones serán transferidas

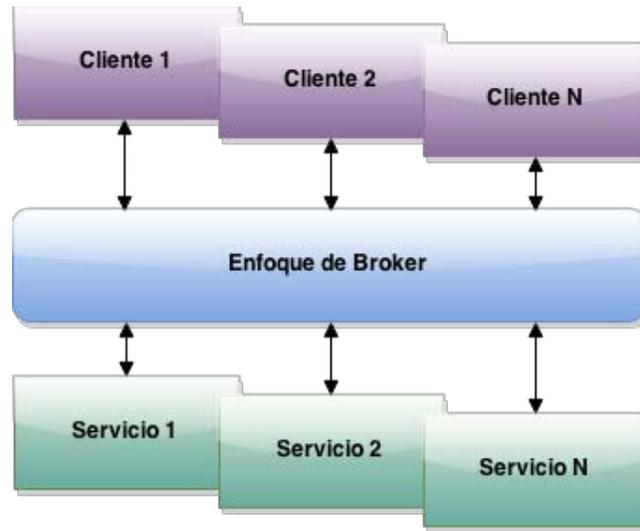


Figura 5.3: Enfoque conceptual de bróker.

al mismo ritmo que éstas se producen. Finalmente, estos nodos permiten reenviar las observaciones obtenidas a los algoritmos de análisis para el siguiente paso del proceso.

5.3.2. Algoritmos de análisis

Cuando las conexiones con el RMS están listas, el siguiente paso es aplicar diferentes algoritmos sobre los flujos de observaciones. La metodología presentada permite realizar cualquier algoritmo. Estos algoritmos serán codificados como nodos diferentes (similar a los mostrados en la Figura 5.2) para paralelizar la computación. Las salidas de estos nodos pueden ser transferidas a los nodos posteriores o entregarse como salidas del sistema. El algoritmo es el responsable de adaptarse a la naturaleza de los datos proporcionados por las fuentes de origen con el fin de aplicar el análisis requerido. El uso de estándares para la codificación de los mensajes transportados por los flujos de datos facilita la preparación de ellos para el análisis. Dependiendo del tipo de análisis, el algoritmo puede tener diferentes tipos de salidas.

Debe tenerse en cuenta cómo se propagan las observaciones para aplicar los algoritmos dentro del flujo de trabajo del sistema. Para flujos de datos de sensor, el orden en que se ha generado cada observación es muy importante.

5.4. METODOLOGÍA PARA EL ANÁLISIS MEDIANTE PROCESAMIENTOS 97

Dependiendo del tipo de análisis, el sistema puede necesitar preservar dicho orden. La metodología presentada tiene en cuenta esta característica y garantiza que todos los mensajes serán analizados, además de ofrecer la posibilidad de mantener el orden en que se procesan los mensajes.

5.3.3. Resultados del análisis

Algunos resultados del algoritmo serán las salidas de la capa de servicios. Tras el análisis, la cantidad de datos se ve reducida razonablemente. De esta manera, las salidas ya no se consideran como Big data, ahora son consideradas como datos manejables (Mayer-Schönberger y Cukier, 2013), siendo válidos para su posible visualización.

Las salidas se codificarán usando el estándar más apropiado. Estas salidas deben ser servidas por el RMS (Sección 4.3.3) para mejorar la interoperabilidad, ya que mediante el cual se ofrece una variedad de protocolos de comunicación. Para ello, se utiliza el RMS con un modelo de publicación/suscripción. La Figura 5.4 muestra un ejemplo para usar el RMS. En este caso, tiene salidas por sensor y nodo de detección. El modelo publicación/suscripción permite la conexión de varios clientes o suscriptores.

5.4. Metodología para el análisis mediante procesamiento estándares

En esta sección se presenta la metodología para el análisis mediante procesamiento que se rijan por estándares GIS, llamados comúnmente como geoprocesamientos.

Los expertos medioambientales, tales como los expertos forestales, necesitan acceder de manera eficiente y compartir no sólo al contenido, sino que también necesitan tener herramientas capaces de generar nueva información que ayude en la toma de decisiones. Por lo tanto, el objetivo de la metodología que se presenta es ofrecer la posibilidad de compartir un modelo para que sea accesible y reutilizable por los diferentes usuarios, además de proporcionar otros beneficios, como la interoperabilidad necesaria para que el modelo generado pueda ser ejecutado por el mayor número de usuarios posibles.

En un dominio geoespacial, se proporciona la capacidad de descubrir, acceder

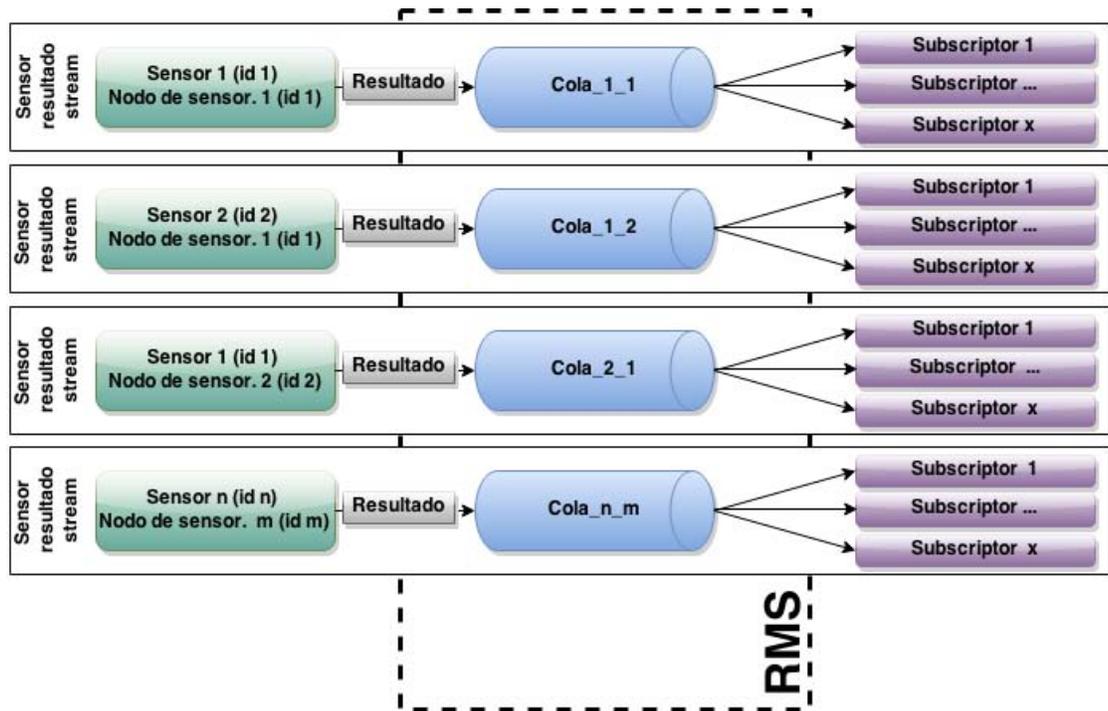


Figura 5.4: Modelo de publicación / suscripción.

y compartir recursos geospaciales entre una amplia gama de actores a diferentes escalas (Masser y otros, 2008; Rajabifard y otros, 2002). Las SDIs comprenden un conjunto de políticas y normativas con la finalidad de promover la creación de servicios que utilicen contenido con información espacial para ayudar a las diversas comunidades de usuarios a la recopilación, el intercambio, el acceso y la explotación de los recursos espaciales (Bishop y otros, 2000; Nebert, 2004).

La tendencia actual en aplicaciones ambientales, se encuentra en el despliegue de aplicaciones en la parte superior de las SDIs con el acceso a los recursos a través de servicios Web estándares, los cuales son eficaces y sencillos de usar y están disponibles de una manera adhoc (Granell y otros, 2010) con la finalidad de promover la interoperabilidad (Percival, 2008).

En la comunidad SDI, sin embargo, se identifican dos cuestiones principales que deben abordarse con la finalidad de proporcionar a los usuarios aplicaciones ambientales útiles. En primer lugar, la falta de capacidades en la ejecución de modelos en línea, ya que actualmente son ejecutados en otros ambientes más particulares, tales como aplicaciones de escritorio (Granell y otros, 2010). Y en

5.4. METODOLOGÍA PARA EL ANÁLISIS MEDIANTE PROCESAMIENTOS 99

segundo lugar, la falta de mecanismos para facilitar la publicación de los resultados para ser compartido con otros interesados (Díaz y otros, 2011).

La interoperabilidad, dentro de la comunidad geoespacial, se mejora a través de los esfuerzos de normalización en su mayoría procedentes de OGC, que se encarga de definir las especificaciones de las interfaces estándares para cada uno de los servicios, con la finalidad de garantizar la interoperabilidad de los datos y modelos de una forma distribuida (Percival, 2008). Las especificaciones existentes han sido probadas para ayudar en la creación de servicios espaciales para compartir recursos espaciales distribuidos (Bernard y otros, 2005). Por ejemplo, el servicio de procesamiento WPS (Schut, 2008) (Sección 2.2.2), el cual especifica una interfaz para acceder a la funcionalidad de procesamiento como servicios Web distribuidos. Además, proporciona una interfaz para describir cualquier cálculo (proceso) y el protocolo de comunicación para enviar solicitudes de servicio, recuperando los resultados del modelo (Schut, 2008).

La especificación WPS se define generalmente para proporcionar acceso basado en estándares para una amplia variedad de cálculos de contenidos geoespaciales utilizando servicios Web. Un servicio basado en WPS puede ofrecer operaciones para tratar tanto datos vectoriales como ráster, almacenados estos de forma local o distribuidos a través de Internet.

El uso de interfaces basadas en estándares para servicios Web, tales como servicios de procesamiento Web, aumenta la interoperabilidad. Actualmente, esta funcionalidad puede estar disponible a través de un servicio, que le permite ser invocado y reutilizado en diferentes escenarios. Los procesos se ejecutan en el servidor, así que no hay necesidad de instalar o almacenar ningún dato localmente. Una de las ventajas, es que gracias a los servicios basados en estándares y arquitecturas orientadas a estos, hace que construir aplicaciones clientes sea una tarea más fácil. Las aplicaciones tienen que conectar con los servicios con la finalidad de que los usuarios puedan interactuar de forma transparente con las operaciones ofrecidas por los modelos. Estas aplicaciones cliente pueden ser tanto de escritorio, Web o incluso una aplicación móvil.

En este sentido, este trabajo inicialmente se presenta con el beneficio de ofrecer estos modelos mediante un servicio basado en estándares para ser reutilizados en otros escenarios, por lo que los usuarios son capaces de reutilizar sus modelos y compartirlos ante la comunidad científica.

Siguiendo los principios de las SDIs, se presenta una metodología mediante el uso de servicios de procesamiento basada en estándares. Para ello se utiliza el estándar de OGC WPS (Schut, 2008).

Las siguientes secciones detallan el mecanismo para aplicar modelos científicos sobre los geoprocесamientos, en este caso se utiliza la especificación WPS. La Sección 5.4.1 muestra cómo se realiza la modelización de los parámetros necesarios para ser utilizados mediante un geoprocесamiento. Y la Sección 5.4.2 detalla la arquitectura en forma general del sistema propuesto.

5.4.1. Modelización de los parámetros

Los geoprocесamientos describen la forma en que se han de modelar los parámetros tanto de entrada como de salida. Esto beneficia el tratamiento tanto de dichas entradas como de las salidas, ya que previamente se conoce con qué tipo de formato se trabaja. Además, se ofrecen conversores tanto para leer las entradas como para generar las salidas dependiendo del formato prefijado.

La Figura 5.5 muestra cómo se representan las entradas y salidas en un geoprocесamiento. Además, se ha de indicar tanto qué tipo de codificación seguirán las entradas y las salidas, así como si es un parámetro requerido u opcional.

El ejemplo representado en la Figura 5.5 muestra cinco entradas, todas ellas requeridas. Tras la ejecución del geoprocесo, el resultado debería verse como dos salidas diferentes.

5.4.2. Arquitectura de los servicios de procesamiento

Uno de los principales objetivos de este trabajo es facilitar la reutilización de los modelos científicos que serán definidos. Actualmente estos modelos o algoritmos se ejecutan en máquinas de escritorio en un entorno particular, siendo inaccesibles para un usuario no especializado. La actual propuesta pretende encapsular los modelos científicos y exponerlos mediante un servicio Web basado en estándares, y así facilitar su accesibilidad.

La Figura 5.6 muestra dónde se encuentra el sistema propuesto de acuerdo con los principios de las SDIs. La directiva INSPIRE (Directive, 2007) define cómo funcionalidad para el procesamiento, los tipos de servicios *Invoke spatial* y *Transformation* (ver parte superior de la Figura 5.6). Dichos tipos son en los que

5.4. METODOLOGÍA PARA EL ANÁLISIS MEDIANTE PROCESAMIENTOS 101

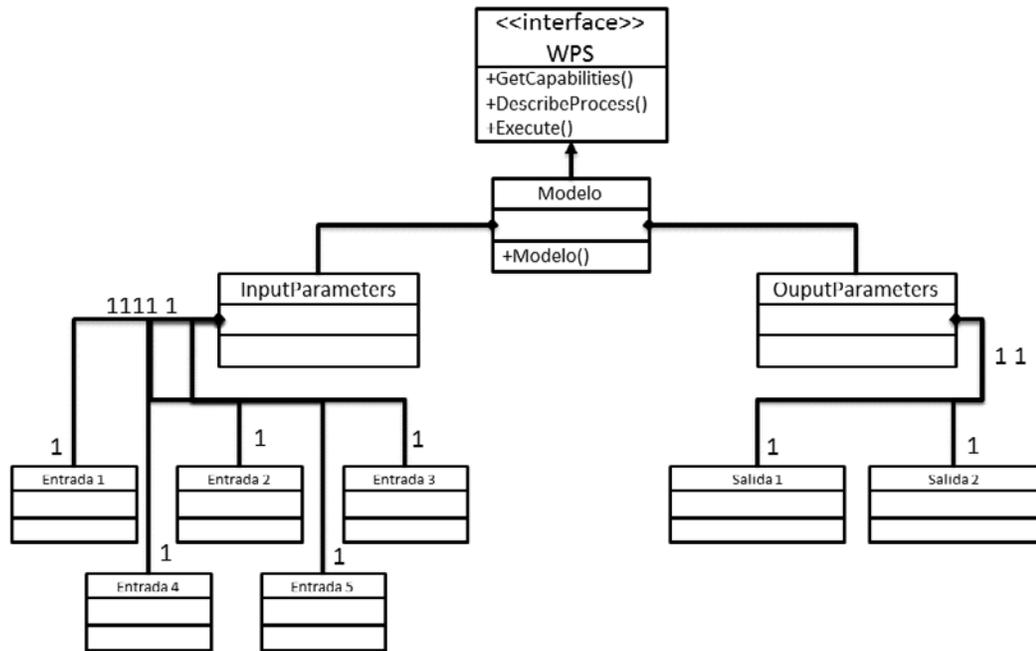


Figura 5.5: Ejemplo de las entradas y salidas del geoprocesamiento.

se basan para ofrecer el servicio de procesamiento. OGC, por su parte, implementa una especificación (WPS) particular sobre este tipo de servicio. Es sobre esta implementación donde se encapsularán los diferentes modelos científicos, generando varias instancias para cada uno de ellos.

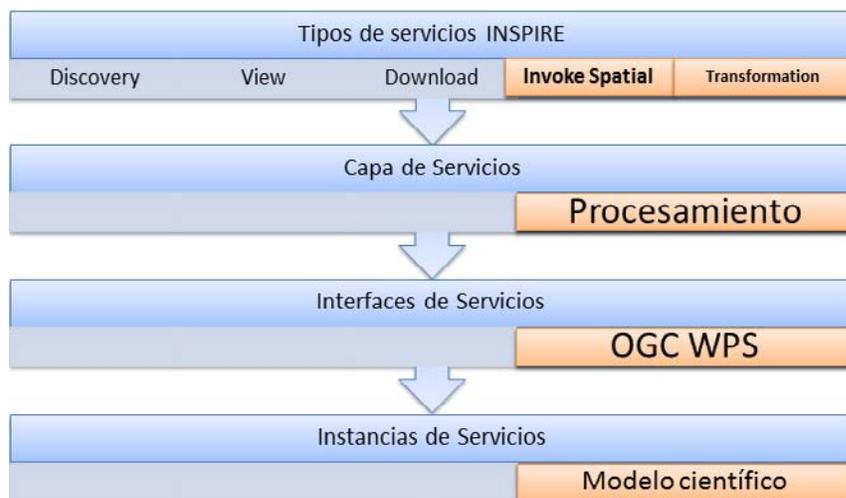


Figura 5.6: Diferentes capas para el geoprocesamiento.

A continuación, se describen cuáles son los pasos para la ejecución del geoprocesamiento para un modelo genérico (Figura 5.7). En el primer paso, el usuario puede invocar el modelo mediante una petición *Execute* que proporciona la interfaz WPS. En dicha petición se requieren todos los parámetros para definir las entradas. El segundo paso es verificar si todas las entradas son correctas para el buen funcionamiento del modelo. Además de verificar, también puede ser necesario realizar alguna transformación en las entradas, dependiendo de los requisitos del modelo en particular.

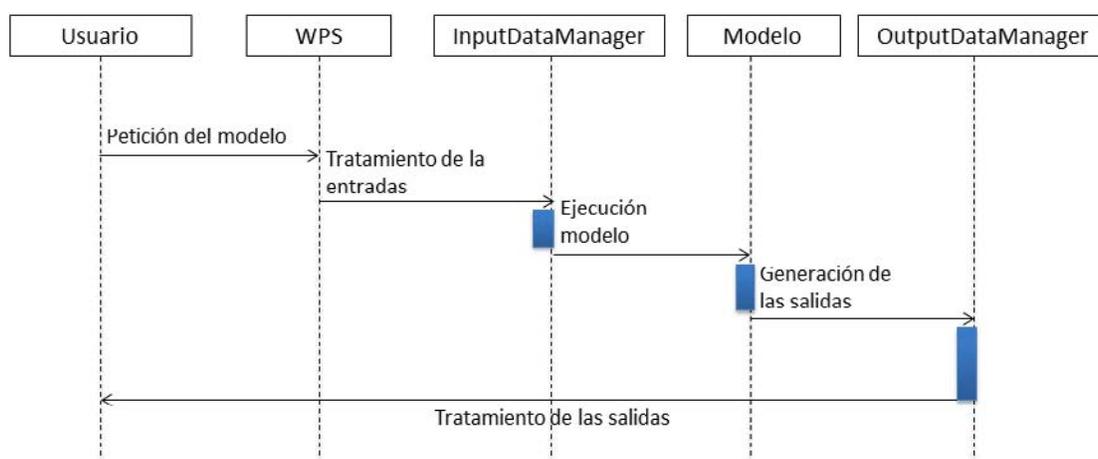


Figura 5.7: Diagrama de secuencia para la ejecución del WPS.

Una vez preparadas las entradas, el siguiente paso es la ejecución del modelo. Para ello, se utilizará o bien algún tipo de conector capaz de ejecutar el modelo en su entorno en particular mediante algún tipo de conector, o bien se implementará dentro del propio WPS. La última parte de la ejecución es obtener los resultados y realizar las transformaciones necesarias para devolverlos como salidas.

5.5. Trabajo Relacionado

En esta sección se detallará el trabajo relacionado con las dos metodologías definidas para el análisis sobre datos de sensores. La primera de ellas sigue un enfoque para analizar grandes cantidades de datos en tiempo real, mientras que la segunda se basa en un geoprocesamiento para aplicar modelos científicos utilizando estándares. Siguiendo cada una de estas metodologías, se presentan

dos secciones para analizar los trabajos relacionados de cada una de ellas.

5.5.1. Trabajos relacionados con el análisis de grandes cantidades de datos en tiempo real

La metodología definida para el análisis de grandes cantidades de datos en tiempo real se basa en los sistemas DSPS (Sección 5.2.1). Algunos trabajos que utilizan dichos sistemas ya han sido desarrollados con anterioridad. Un posible uso de la metodología presentada es analizar los datos proporcionados por las propias personas. En este caso, los sensores son las personas y éstas proporcionan los flujos de datos. Fue para este uso la primera aplicación del framework Storm, ya que se desarrolló inicialmente por Twitter Inc. (Toshniwal y otros, 2014), para analizar los tweets que sus usuarios generaban. Simoncelli y otros (2013) utiliza Storm para extraer tendencias a partir de los hashtags generados en Twitter.

Enfoques similares (utilizando también Storm) ya han sido propuestos, principalmente para analizar los datos proporcionados por diferentes tipos de sensores. Ji y otros (2014) utiliza Storm para analizar los datos proporcionados por las redes de sensores magnéticos; Kumar (2014) propone un sistema para detectar fugas en una red de distribución de agua. Los autores en Nandan (2013) analizan la ubicación de los jugadores y el balón durante un partido de fútbol por sensores de radio. Storm se utiliza para dicho análisis en tiempo real.

Storm también ha sido utilizado para la detección de valores anómalos. En Sunderrajan y otros (2014), por ejemplo, se utiliza para detectar valores anómalos en el consumo de energía de enchufes inteligentes. Yutan y otros (2014) utiliza Storm para la monitorización, el flujo de datos y para localizar el origen de los ataques o anomalías en tiempo real, mediante la búsqueda de las direcciones IP específicas. El análisis en tiempo real también ha sido utilizado para ser aplicado en la minería de datos (DeFrancisci Morales, 2013). Sitaram y otros (2013) aplica dichas técnicas para el reconocimiento de voz en tiempo real.

5.5.2. Trabajos relacionados con el análisis del geoprocesamiento

El trabajo presentado en la Sección 5.4 tiene como objetivo aprovechar las posibilidades que ofrecen los geoprocesamientos, tanto en reusabilidad como

en interoperabilidad, gracias a los estándares que los rigen. Michaelis y Ames (2009) argumenta que los servicios basados en geoprocesamientos WPS son adecuados para realizar cálculos geoespaciales. Kiehle (2006); Yang y otros (2008) describen modelos geoespaciales desplegados en la parte superior de las SDIs para proporcionar a los usuarios el acceso a los datos y modelos implementados como servicios web, aumentando la interoperabilidad y la reutilización de ellos mismos.

Hasta el momento son muchos los campos en los que se han utilizado los geoprocesamientos. Uno de ellos es Granell y otros (2010), donde describe una aplicación basada en SDI de vigilancia ambiental, en el dominio de la hidrología. Más recientemente, en el ámbito forestal, McInerney y otros (2012) describe un portal web que proporciona acceso a una SDI, el cual ofrece datos sobre los recursos forestales a diversas escalas espaciales, así como también proporciona capacidades analíticas para el control y la validación de los cambios en el medio ambiente.

Modelos para el análisis de deslizamientos de tierra también han sido intensamente estudiados. En Cannata y otros (2012) utilizan WPS para la evaluación del deslizamiento de tierra de poca profundidad en una aplicación en tiempo real. Thiebes y otros (2013) proporciona un modelo para el análisis de deslizamientos mediante WPS. El objetivo es la integración de este modelo en los sistemas de alerta temprana de deslizamientos. Dubois y otros (2013) presentan la aplicación e-Hábitat que permite la delimitación de los hábitats ecológicos para especies de interés.

5.6. Conclusiones

Como conclusión al capítulo de análisis, decir que se han detallado dos formas diferentes para el análisis de datos de sensores. La primera de ellas se basa en una metodología para el análisis de grandes cantidades de datos (big data). Y en la segunda de ellas, se utiliza un geoprocesamiento estándar con la finalidad de encapsular un modelo, aumentando sus ventajas a la hora de su ejecución.

Referente al primero de ellos, se ha desarrollado una metodología general para analizar flujos de datos de redes de sensores en tiempo real. La metodología incluye puntos de acceso a diferentes fuentes heterogéneas de datos de sensores, las cuales ofrecen múltiples protocolos. Es compatible con una rica gama de

herramientas para conectar con fuentes de datos de sensores independientemente de los tipos de datos particulares que ofrecen. Este enfoque también permite utilizar un patrón bróker para conectar con diferentes interfaces en tiempo real, de modo que el sistema puede analizar grandes cantidades de datos de sensores en tiempo real y también puede servir los resultados mediante un formato estándar.

La metodología presentada aborda dos problemas. El primer desafío es proporcionar un sistema capaz de analizar los datos de los sensores en tiempo real, con el que se soporten múltiples fuentes de datos del sensor, incluyendo todas las categorías dependiendo del formato (Sección 2.1). La metodología permite analizar cada una de las observaciones sin fallos. El segundo desafío es ofrecer una solución escalable e interoperable. La propuesta ofrece diferentes características que contribuyen a mejorar estos requisitos. Muestra de ello es que se pueden reutilizar los componentes diseñados para otros escenarios con otras propiedades. También se utiliza un patrón bróker para ofrecer múltiples conexiones con interfaces heterogéneas. Finalmente, para mejorar la interoperabilidad, se utilizan estándares para codificar los datos, además de proporcionar una variedad de protocolos.

La metodología presentada se aplica en primera instancia sobre los datos proporcionados por redes de sensores. Aunque el sistema puede funcionar con cualquier serie de valores (observaciones), es decir, es en sí mismo independiente de la naturaleza particular de los fenómenos observados. El tratamiento exacto dependerá de la clase de datos proporcionados para el análisis.

De esta manera, la metodología definida se puede aplicar igualmente a las fuentes proporcionadas por los teléfonos móviles, tabletas, wearables, etc. Tales dispositivos ofrecen cada vez más capacidades sensoriales, incluyendo entre ellos GPS, giroscopios, acelerómetros, cámaras y otros. Estos, llamados sensores ubicuos, pueden ser utilizados para proporcionar sistemas adaptables y sensibles al contexto, los cuales son ofrecidos mediante servicios que interactúan directamente con ellos.

Yendo un paso más allá, estos sensores pueden utilizarse para inferir las actividades de las personas, lo que lleva a un proceso de sensorización social, siendo las personas quienes generan enormes cantidades de datos y los comparten utilizando servicios Web 2.0 (por ejemplo, Twitter⁸ e Instagram⁹). Un posible uso de

⁸<http://www.twitter.com>

⁹<http://www.instagram.com/>

la metodología presentada podría ser el análisis de los datos proporcionados por estos servicios. En este caso, los sensores son los ciudadanos que proporcionan flujos de datos.

La metodología descrita puede considerarse una solución escalable, ya que todas las partes del sistema se han diseñado para facilitar su extensibilidad y su reutilización. De hecho los componentes siguen un diseño modular para mejorar la reutilización. Por ejemplo, el componente RMS sigue el concepto de Message Broker y se puede utilizar dependiendo de la situación específica. Los estándares utilizados y la variedad del protocolo también mejoran la conectividad entre los clientes. Esto ayuda a poder reutilizar el mismo cliente en diferentes escenarios.

En definitiva, se propone una plataforma de intermediación integrada y escalable que es capaz de integrar diferentes flujos de datos con alta frecuencia de actualización y también aplicar sobre ellos cualquier tipo de análisis. Este enfoque está destinado a conectarse con los entornos donde hay gran variedad de fuentes, especialmente en las zonas urbanas como las Smart Cities.

Como conclusiones para la metodología de análisis mediante geoprocesamientos estándares, se ha diseñado un sistema para la computación distribuida de cualquier tipo de modelo científico basado en estándares para facilitar la ejecución y el uso, además de posibilitar el poder compartir los modelos y los resultados de estos. Siguiendo principios de las SDI, los modelos se encapsulan como un servicio basado en el estándar de geoprocesamiento para mejorar la interoperabilidad y proporcionar acceso estándar. En particular, este servicio de geoprocesamiento implementa la interfaz estándar de OGC, llamada WPS.

El uso de interfaces basadas en estándares mediante servicios Web como WPS mejora la interoperabilidad, ya que pueden ser ejecutados a través de un servicio interconectable bien conocido, que le permite ser invocado y reutilizado en diferentes escenarios. Los procesos se ejecutan en el servidor de manera que no hay necesidad de instalar y almacenar datos localmente. Sin embargo, el uso de servicios estándares y de arquitecturas basadas en estándares, hace que la construcción de las aplicaciones clientes sea más compleja (Tamayo y otros, 2012b). Así pues, las aplicaciones generadas tienen que considerar un cliente de la forma que sea lo más amigable, para que los usuarios puedan interactuar de manera transparente con la funcionalidad ofrecida por el modelo. Estas aplicaciones cliente pueden ser para escritorio, Web o incluso una aplicación móvil.

Capítulo 6

Casos de uso

Tras definir completamente como se ha abordado cada una de las etapas del ciclo de vida de los datos de sensores, en este capítulo se presentan los diferentes casos de uso donde se ha aplicado cada una de ellas. De esta manera el capítulo se compone de las siguientes secciones. La primera sección (Sección 6.1) siguiendo el diseño de la plataforma *SEnviro* (Capítulo 2) se realiza un ejemplo para la monitorización meteorológica y de calidad del aire, siendo aplicada dentro del campus universitario de la Universitat Jaume I (UJI). La Sección 6.2 detalla un ejemplo de uso de la herramienta de publicación GSF. La sección tercera (Sección 6.3) muestra un caso de uso completo para la integración de datos de sensores de fuentes heterogéneas, con la publicación y acceso, además de un cliente móvil para su visualización. En la Sección 6.4 se muestra una implementación tecnológica para la metodología de análisis en tiempo real de grandes cantidades de datos. Para ello se utiliza un algoritmo para la detección de anomalías, con la finalidad de lanzar eventos cuando estas se produzcan. Finalmente, la Sección 6.5 se realiza un ejemplo para la predicción de incendios forestales, utilizando la metodología de análisis mediante geoprocesamiento estándar.

6.1. *SEnviro*: prueba de concepto en la Universitat Jaume I

Esta sección presenta una prueba de concepto para probar y validar la plataforma presentada en el Capítulo 2. En primer lugar, se describe el contexto donde la plataforma *SEnviro* se ha desplegado (Sección 6.1.1). En la Sección

6.1.2, se presenta un ejemplo de una red *SEnviro*. La tercera sección (Sección 6.1.3) realiza un estudio del consumo energético para el ejemplo de *SEnviro Thing* desarrollado. Por último, la Sección 6.1.4 muestra el cliente web implementado para visualizar y evaluar los datos obtenidos por la red *SEnviro*.

6.1.1. Contexto de la Universitat Jaume I

La prueba de concepto ha sido desplegada en el campus de la Universitat Jaume I¹. Dicho escenario puede ser considerado como una pequeña ciudad, ya que el campus de la Universidad tiene muchas partes comunes a una ciudad, pero a pequeña escala. Como un primer paso, se ha desplegado un *SEnviro Thing* para la monitorización medioambiental, donde se presenta un escenario con cinco *SEnviro Things*. La Figura 6.1 muestra la ubicación de cada *SEnviro Thing*.



Figura 6.1: Ubicaciones de cada *SEnviro Thing* dentro del campus de la Universitat Jaume I.

Cada *SEnviro Thing* se ha desarrollado con diferentes sensores, los cuales proporcionan las observaciones sobre algunos fenómenos básicos, pudiendo ser estos de carácter meteorológico o de calidad del aire.

¹<http://smart.uji.es/>

Por lo referente a los sensores meteorológicos, se incluyen los siguientes: temperatura, humedad, presión atmosférica, precipitación, además de la dirección y la velocidad del viento. Por los de calidad del aire, se ofrecen los siguientes fenómenos: partículas en suspensión, ruido, gases y luz. La razón porque no todos los *SEnviro Thing* tiene los mismos fenómenos, se debe a que ciertos fenómenos tales como temperatura, humedad, presión atmosférica, lluvia, velocidad o dirección del viento, no cambian significativamente dentro del área del campus. Para reducir el coste, se ha decidido usar sólo dos *SEnviro Things* en esta red que incluyan todos estos sensores.

La red *SEnviro* propuesta para este caso de uso sigue el paradigma de red ubicuo (Puccinelli y Haenggi, 2005), donde la red de sensores forma parte de Internet. A través de una aplicación Web, los usuarios tendrán acceso a la información proporcionada por los objetos inteligentes, ya sea directamente o a través de servidores intermedios. En el caso presentado, un servidor actúa generalmente como un sumidero en la red de objetos inteligentes para recopilar datos de cada uno de ellos.

La Universitat Jaume I cuenta con conexión Wi-Fi en todo el campus. De esta manera, cada *SEnviro Thing* está conectado con el punto de acceso inalámbrico (WAP, en sus siglas en inglés) más cercano, mediante el módulo Wi-Fi incluido y de este modo, envía las observaciones a un servidor central. Dicho servidor está abierto a Internet y es responsable de servir las observaciones a los clientes para su consumo. La configuración elegida para la red *SEnviro* es la conexión directa de cada *SEnviro Thing* a la red interna de la Universidad. De esta manera, la red tiene forma de estrella (Figura 6.2), con la particularidad de que tiene diferentes WAPs que serán utilizados por los nodos de *SEnviro* para establecer la conexión.

6.1.2. La plataforma *SEnviro* para la monitorización ambiental

En esta sección, se muestra un ejemplo de desarrollo tanto hardware como software (Capítulo 2), de la plataforma *SEnviro* para la monitorización de fenómenos ambientales.



Figura 6.2: Red *SEnviro* para el campus de la Universitat Jaume I.

Construyendo un *SEnviro Thing*

La Tabla 6.1 muestra una descripción general de los componentes de un sensor completo *SEnviro Thing* agrupados por las categorías detalladas en la Sección 2.3.2. El coste total, en el momento de la escritura, por cada *SEnviro Thing* es de 286.28 €. La Figura 6.3 muestra como se conectan los componentes que forman un *SEnviro Thing*. Como se comentó en el Capítulo 2, cada *SEnviro Thing* está compuesto por cinco partes (*Núcleo*, *Comunicación*, *Sensores*, *Fuente de alimentación* y las cajas de protección). A continuación se detalla cómo se compone cada una de estas categorías.

- **Núcleo.** Éste tiene cuatro elementos diferentes que son: el *Microcontrolador*, los *Conectores*, el *Reloj* y la *Memoria*. Se ha elegido como *Microcontrolador* el Arduino UNO². Como se comentó en la Sección 2.2.2, el Arduino UNO ofrece un bajo consumo de energía y un gran número de posibilidades mediante el uso de expansiones (shields). Estos dos aspectos son de vital importancia para este proyecto, ya que éste busca una plataforma energéticamente autónoma, además de querer ofrecer una alta gama de conexiones diferentes. El shield Grove ha sido utilizado como *Bus*. Éste tiene 16 conectores donde componentes (sensores) con el mismo conector Grove pueden ser fácilmente conectados. Como *Reloj*, se ha utilizado un módulo de Reloj en tiempo real (RTC, en sus siglas en inglés) para Grove³ y de este modo se puede guardar la hora actual en todo momento. Una marca de tiempo puede acompañar a cada observación cuando es recogida. *SEnviro Thing*

²<http://www.arduino.cc>

³http://www.seeedstudio.com/wiki/Grove_-_RTC

<i>Categoría</i>	<i>Componente</i>	<i>Descripción</i>	<i>Coste</i>
Núcleo	Microcontrolador	Arduino UNO	20.00 €
	Shield Grove	Base Shield V2	8.60 €
	Rejor	Real-Time Clock para Grove	5.60 €
	Módulo MicroSD	Módulo MicroSD para Arduino UNO	8.30 €
	Memoria MicroSD	MicroSD card 2Gb	6.00 €
	Conectores Screw (x3)	Screw Terminal para Grove	2.75 €
	Caja para Arduino	RETEX serie 102	10.65 €
	Caja para sensores	Caja impresa en impresora 3D	10.00 €
Comunicación	Módulo Wi-Fi	Módulo RN-XV WiFly	40.80 €
	Socket Bee	Bee Socket para Grove	6.72 €
Sensores	Temperatura y humedad	Sensores Grove de temperatura y humedad	12.60 €
	Ruido	Sensor Grove de ruido	4.95 €
	Luz	Sensor Grove de luz	2.90 €
	Partículas	Sensor Grove de partículas	14.45 €
	Barómetro	Sensor Grove de presión	16.35 €
	Gases	Sensor Grove MQ-9	7.95 €
	Pluviómetro y anemómetro	Weather Meters	72.00 €
Fuente de alimentación	Módulo de energía	Li-po Rider	9.50 €
	Batería	Polymer Lithium Ion Battery 2200mA 3.7V	9.66 €
	Panel solar	3W Solar Panel 138X160	15.00 €
	Pila de botón	CR1225 3V 12mm 47mA	1.50 €

Tabla 6.1: Lista de componentes que se incluye en el *SEnviro Thing* (los precios en el momento de la escritura).

utiliza dos tipos de *Memorias*. Por un lado, la memoria proporcionada por el Arduino UNO para guardar el estado actual del *SEnviro Thing*. Y por otro lado, almacenamiento externo a través de un módulo MicroSD⁴ utilizado para almacenar las observaciones que no han sido enviadas con éxito.

- *Sensores*. La Tabla 6.2 muestra toda la información sobre los *Sensores*

⁴http://www.dfrobot.com/wiki/index.php/MicroSD_card_module_for_Arduino



Figura 6.3: Montaje de un *SEnviro Thing*.

que han sido integrados para este caso de uso en particular, además de mostrar también las características de las medidas que se pueden obtener de ellos. Todos los *Sensores* elegidos son de bajo coste, ya que, diseñar una plataforma asequible es uno de los objetivos de este proyecto. A pesar de su bajo precio, la mayoría de ellos son sugeridos por la comunidad para ser utilizados para la monitorización en ambientes industriales, ya que ofrecen medidas absolutamente confiables. Además, todos los sensores presentados utilizan un conector de Grove para facilitar su integración.

- *Comunicación*. Un módulo Wi-Fi, llamado WiFly RN-XV⁵ se ha utilizado para desarrollar la parte de la *Comunicación*. Dicho módulo ofrece al *Núcleo* la capacidad de enviar y recibir paquetes a través de una red Wi-Fi. Cuenta con conector XBee y se conecta al Arduino mediante un socket XBee, el cual ofrece un conector Grove para su acople al *Microcontrolador*. Éste es conectado al shield Grove mediante el conector Transmisor-Receptor Asíncrono Universal (UART, en sus siglas en inglés). El socket Grove Bee⁶

⁵<http://rovingnetworks.com/products/RN171XV>

⁶http://www.seeedstudio.com/wiki/Grove_-_Bee_Socket

Tabla 6.2: Detalles de los Sensores incluidos.

Sensores	Fenómenos	Fabricante	Modelo	Interfaz de datos	Unidades	Rango	Precisión
DHT22 ^a	Temperatura Humedad	Seedstudio	SEN51035P	Analógico	Centígrados Ratio	[-40, 80] [5%, 99 %]	±0,5 Grados (C) ±2 RH
Bosch BMP085 ^b	Presión Temperatura	Seedstudio	SEN05291P	I2C	Hectopascal Centígrados	[300, 1100] [-40, 85]	±0,03 hPa ±2 gradosC
LDR GL5528 ^c	Intensidad de Luz	Seedstudio	SEN11302P	Analógico	Lux	[0,1024]	No especificado
LM2904 Amplifier ^d	Loudness	Seedstudio	SEN02281P	Analógico	Decibelios	[0,1024]	No especificado
PPD42NS ^e	Partículas	Seedstudio	SEN12291P	Digital	pcs/litros	[0,28000]	> 1 um
MQ-9 ^f	CO Combustible gas	Seedstudio	SEN04092P	Analógico	ppm ppm	[10, 1000] [100, 10000]	No especificado No especificado
Pluviómetro y anemómetro ^g	Velocidad viento Dirección viento	Sparkfun	SEN08942	Analógico (RJ11)	km/h Dirección (Grados)	No especificado [0,360]	No especificado No especificado
Pluviómetro					mm	No especificado	No especificado

^ahttp://www.seedstudio.com/wiki/Grove_-_Temperature_and_Humidity_Sensor_Pro

^bhttp://www.seedstudio.com/wiki/Grove_-_Barometer_Sensor

^chttp://www.seedstudio.com/wiki/Grove_-_Light_Sensor

^dhttp://www.seedstudio.com/wiki/Grove_-_Loudness_Sensor

^ehttp://www.seedstudio.com/wiki/Grove_-_Dust_sensor

^fhttp://www.seedstudio.com/wiki/Grove_-_Gas_Sensor

^ghttps://www.argentdata.com/files/80422_datasheet.pdf

permite al sistema poder reemplazar cómodamente el tipo de *Comunicación*, ya que el conector XBee es ampliamente aceptado como puerto para los módulos de comunicación. De esta manera, podemos expandir la plataforma con otras formas de comunicación, tales como servicios de datos móviles o Bluetooth.

- *Fuente de alimentación.* Se utiliza una batería de litio de 2200 mA^7 para ofrecer una plataforma energéticamente autónoma, la cual suministra una tensión de salida de 3.7 V y está encargada de almacenar la energía generada mediante la *placa solar de 3 W^8* que suministra una tensión de salida de 5.2 V . Estos dos elementos y el *Microcontrolador* se unen a una placa, llamada Lipo Rider V1.1⁹, que gestiona el flujo de energía entre los distintos componentes. Esta placa también tiene un puerto microUSB donde se puede cargar la batería de litio en caso de que la energía solar no sea suficiente. Además, no tiene que ser programada, ya que viene con un algoritmo para gestionar las fuentes de alimentación, aunque puede reemplazarse con uno personalizado si es necesario. Además, se ha utilizado una pila de botón (CR1225) para alimentar el *Reloj* de forma continua y poder guardar la hora.
- *Cajas.* Se han considerado dos cajas diferentes para proteger todas las partes del *SEnviro Thing*. Para proteger el *Núcleo*, la *Comunicación* y la *Fuente de alimentación*, se ha utilizado una caja aislante e impermeable llamada RETEX¹⁰. Además, se ha diseñado una caja de pagoda (Figura 6.4) para servir de unión a todas las partes de la plataforma y proteger los *Sensores* ante las condiciones climáticas desfavorables. También sirve para poner los *Sensores* en contacto con el medio ambiente y proporcionar mediciones confiables. La caja de pagoda ha sido impresa con una impresora 3D con material de ácido poliláctico (PLA, en sus siglas en inglés). Actualmente existen dos tipos de materiales para las impresoras 3D, el PLA y el Acrilonitrilo Butadieno Estireno (ABS, en sus siglas en inglés). Mientras que PLA es clasificado como industrialmente compostable, es altamente resistente a los rayos ultravioleta, mucho más que el ABS.

⁷http://www.seeedstudio.com/wiki/images/e/e5/PL805050P2200mAh_3.7V_1S1P_V_1.0.pdf

⁸http://www.seeedstudio.com/wiki/3W_Solar_Panel_138*160

⁹http://www.seeedstudio.com/wiki/Lipo_Rider_V1.1

¹⁰<http://www.retex.es/es/Cajas/Cajas-universales/Serie-102.axd>



Figura 6.4: Caja Pagoda.

Detalles del comportamiento del SEnviro Thing

Se ha desarrollado un programa para Arduino para ofrecer el comportamiento descrito en la Sección 2.3.3. El código de Arduino está escrito en un archivo llamado *sketch*, el cual es el programa principal. Dicho programa está dividido en dos partes diferentes: *setup* y *loop*. Estos dos métodos se corresponden con las dos etapas descritas en la Sección 2.3.3. El método *setup* (etapa *inicial*), se ejecuta como un inicializador cuando Arduino se inicia por primera vez. Cuando se termina dicha etapa, el método *loop* (etapa *repetitiva*) se ejecutará como un bucle infinito.

Arduino no proporciona un método para ofrecer la fecha y hora, por esa razón se utiliza un *Reloj*. En la fase inicial, el Arduino almacena la fecha y hora actual haciendo una petición a un servicio Web. Además de la fecha y hora, también se ofrece un servicio que se leerá de forma diaria para indicarle cuál es su comportamiento. Dicho comportamiento define cuáles serán los sensores activos y con qué frecuencia se tomarán las observaciones. Esta característica permite modificar el comportamiento de cada *SEnviro Thing*, cambiando simplemente el archivo de configuración que se encuentra alojado en un servidor centralizado. Los cambios serán efectivos en menos de 24 horas, sin la necesidad de desplegar

un nuevo *sketch* en el Arduino, con lo que no será necesario desmontar los *SEnviro Things* para el cambio de comportamiento. La respuesta ofrecida por el servidor, con los parámetros de configuración, contiene instrucciones simples, como por ejemplo, qué sensores consultar cada vez que el Arduino se despierte y su frecuencia de muestra. El formato de la respuesta es simple, ya que las capacidades de procesamiento del Arduino son limitadas. La respuesta se envía en formato de valores separados por comas (CSV, en sus siglas en inglés).

Como se ha dicho anteriormente, la etapa *repetitiva* ejecuta todas las instrucciones en un bucle infinito. Si el comportamiento del sistema indica un tiempo de espera entre observaciones, se puede añadir una espera (*delay*) justo antes de terminar una iteración para insertar una pausa de la longitud deseada. Sin embargo, en términos de consumo energético no es eficiente, ya que el *Microcontrolador* continua funcionando con alto consumo. Esa es la razón para poner el Arduino en un modo *sleep*, llamado *SLEEP_MODE_PWR_DOWN*, que permite un ahorro de energía mayor. Para despertar el *Microcontrolador*, se utiliza un mecanismo llamado Watch Dog Timer (WDT), con el cual se pueden obtener como máximo interrupciones cada 8 segundos. Si el sistema necesita tiempos de espera más largos, los ciclos de sueño pueden ser contados, por lo que dependiendo del número de ciclo el Arduino tiene que volver al método *sleep* o, por lo contrario, tiene que realizar la toma de observaciones.

El módulo WiFly RN-XV también puede ponerse en modo *sleep* mediante el envío de un comando específico y se puede despertar cuando recibe datos en el buffer de envío. Así, el Arduino pone el módulo Wi-Fi a dormir antes de entrar en su modo *sleep*. Entonces, cuando el Arduino tiene que recoger las observaciones de los sensores, envía un carácter particular al módulo Wi-Fi para despertarlo. Este carácter no interfiere en los envíos posteriores, ya que es descartado.

Cuando haya terminado el período de *sleep*, el sistema actualiza los valores de las observaciones tomadas desde cada uno de los sensores y los envía al servidor. Sin embargo, podría haber problemas de conectividad, de forma que se podrían perder las observaciones. Para evitar tales pérdidas, se ha implementado en el comportamiento del Arduino la capacidad de guardar las observaciones pendientes en una tarjeta MicroSD cuando el envío no tiene éxito. Además, cuando se restablece la conectividad, todas aquellas observaciones pendientes se envían junto con la fecha y la hora que fueron tomadas y a continuación, son borradas de la tarjeta MicroSD.

Se han utilizado diferentes bibliotecas de código abierto para lograr la funcionalidad deseada de cada uno de los componentes de hardware. Estas bibliotecas han sido desarrolladas por la comunidad y compartidas en Internet para facilitar su reutilización en otros proyectos con requerimientos similares, como el que se presenta. Además, se ha desarrollado una biblioteca propia para utilizar el módulo WiFly RN-XV de la forma descrita, ya que las existentes no se adaptaban a las necesidades del proyecto. Dicha librería se encuentra publicada para su libre utilización en un repositorio de Github¹¹.

6.1.3. Fuente de alimentación autónoma

Esta sección presenta cuál es el consumo de energía para el *SEnviro Thing* diseñado y también muestra diferentes pruebas de consumo de energía con el método WDT.

Consumo energético

En relación con el consumo de energía del Arduino UNO (*Núcleo*), la Tabla 6.3 muestra una comparación del consumo de energía en diferentes escenarios. El consumo puede variar considerablemente dependiendo de los periféricos conectados.

Modo	Consumo energético (mA)
Sleep	5-9
Normal	25-50
Alto rendimiento	300

Tabla 6.3: Consumo de energía del Arduino UNO.

El consumo de energía de los otros componentes diferentes al *Núcleo* se detallan en la Tabla 6.4. Tanto los consumos del *Reloj* como los del módulo de la MicroSD, tienen un menor consumo de energía que los sensores. El consumo total de energía de estos componentes es de 21.5 mA.

La Tabla 6.5 muestra el consumo de energía del módulo de *Comunicación* dependiendo de su nivel de actividad y estado. Se muestra bastante alto en

¹¹<https://github.com/alujans/WiFlyRNXV>

Componente	Consumo energético (mA)
Reloj	1.5
Módulo MicroSD	20

Tabla 6.4: El consumo de energía de los demás componentes diferentes al *Núcleo*.

términos de consumo de energía, ya que para enviar información requiere 185 mA. Únicamente consume 35 mA cuando no está transmitiendo datos y está preparado para recibir datos. Por lo contrario, es altamente eficaz y sólo consume 4 μ A cuando está en modo de reposo (*sleep*).

Modo	Consumo energético (mA)
Sleep	4×10^{-3}
RX activo	35
TX activo	185

Tabla 6.5: Consumo energético del módulo WiFly RN-XV.

El consumo de energía de los diferentes *Sensores* se muestra en la Tabla 6.6. Ésta muestra que los consumos de energía del sensor de partículas son significativamente mayores que los otros sensores. El consumo agregado es de 384 mA con unas condiciones normales de funcionamiento.

Componente	Consumo energético (mA)
Sensor de temperatura y humedad	1.5
Sensor de ruido	0.5
Sensor de luz	3
Sensor de partículas	90
Sensor de presión	89
Sensor de gases	150
Pluviómetro y anemómetro	50

Tabla 6.6: Consumo energético de los *Sensores*.

Pruebas de vida de la batería

Como se ha comentado, el *SEnviro Thing* ha sido diseñado para ser autónomo energéticamente. Para ello, se han incorporado los paneles solares y las baterías necesarias para proporcionar energía en todas las situaciones. Para conocer realmente cuál es el consumo de cada *SEnviro Thing*, ha sido necesario estudiar el consumo de forma práctica, ya que puede variar dependiendo de los periféricos conectados y no coincidir con el teórico. De esta forma, se han realizado pruebas para conocer el consumo real y verificar la autonomía, a la vez que probar el comportamiento de los paneles solares y baterías.

Los cálculos teóricos sugieren que con la batería de 2200 mA incluida, debería ser capaz de suministrar energía al sistema durante 4 horas y 30 minutos bajo un consumo total de 490.5 mA . Sin embargo, nuestra primera prueba de autonomía, reveló que cada batería sólo podría suministrar energía a un *SEnviro Thing* durante 3 horas sin usar ningún mecanismo de eficiencia. Como conclusión de esta primera prueba, es evidente que existe pérdida de energía, con lo cual las tasas de descarga reducen considerablemente la duración de las baterías. Esta prueba fue realizada sin los paneles solares, para comprobar la capacidad de la batería.

Además, se consideraba que era altamente ineficiente mantener los consumos normales durante los períodos que el Arduino no estaba recogiendo las observaciones. De este modo se decidió utilizar interrupciones WDT y el modo de suspensión en el Arduino y el módulo Wi-Fi (Sección 6.1.2). Con dicho cambio, se obtuvo: 1) un menor consumo energético, 2) el mantenimiento de una alimentación continua y constante y 3) el cumplimiento con las tareas que se espera que sean llevadas a cabo en cada *SEnviro Thing*.

Se realizaron nuevamente pruebas de autonomía con la configuración mejorada y la batería pudo suministrar energía a un *SEnviro Thing* durante 24 horas y 30 minutos. Con este nuevo enfoque, las baterías duraron 8 veces más que en el modo de funcionamiento normal. El consumo de energía obtenido es alrededor de 90 mA por hora (con las pérdidas de energía y las tasas de descarga).

Estos nuevos resultados son satisfactorios, ya que las horas obtenidas de funcionamiento son suficientes para mantener por la noche el *SEnviro Thing* en funcionamiento, sin que dependan de la energía suministrada por los paneles solares. Esto representa sólo la mitad del tiempo del máximo posible en un día,

por lo tanto, los paneles solares son capaces de recargar las baterías durante el día¹². Finalmente, se realizó una prueba final con una batería completamente cargada y un panel solar de 3W, obteniendo resultados realmente satisfactorios. El consumo de energía por la noche fue compensado por la energía suministrada desde el panel solar durante el día.

6.1.4. Un cliente Web capaz de conectarse a la API SensorThings

Como un primer prototipo, un cliente Web basado en HTML5, Javascript y hojas de estilo en cascada (CSS, en sus siglas en inglés), ha sido desarrollado (Figura 6.5). Este cliente es capaz de conectarse al servicio de OGC SensorThings API para obtener los datos de los diferentes *SEnviro Things* dentro de la Universidad. El objetivo principal del cliente desarrollado es visualizar los datos de los *Sensores* en tiempo real.

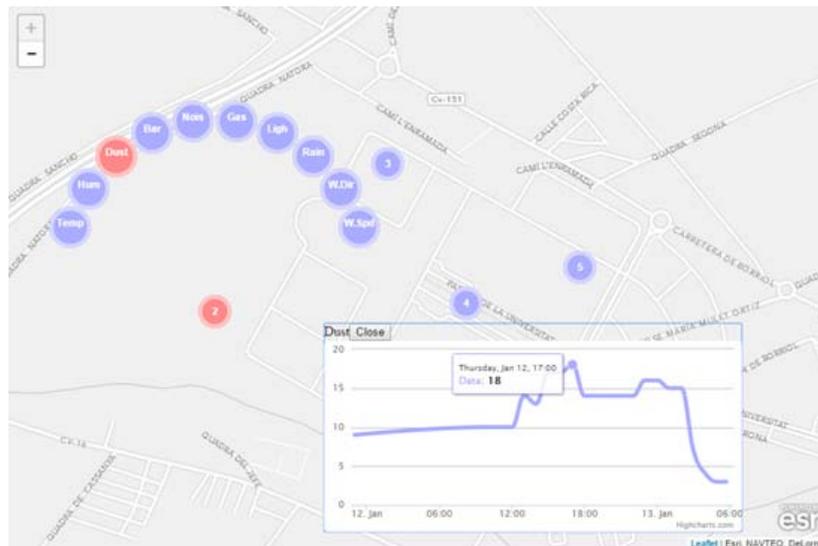


Figura 6.5: Cliente desarrollado para mostrar las observaciones de la red *SEnviro*.

En primer lugar, cada *SEnviro Thing* se muestra en el mapa utilizando marcadores. Cuando un usuario hace clic en uno de ellos, nuevos marcadores aparecerán en un menú emergente. Cada nuevo marcador simboliza un fenómeno incluido

¹²Se considera el número de horas solares en España (mín. 9,5 horas solares y máx. 14,9 horas solares).

en ese *SEnviro Thing* correspondiente a un flujo de datos de OGC SensorThings API. El marcador seleccionado se muestra en rojo.

Cuando se hace clic en uno de ellos, el cliente muestra una ventana emergente descubriendo un gráfico con los últimos datos de las observaciones del sensor. El gráfico muestra interactivamente las observaciones. Es posible visualizar simultáneamente diferentes fenómenos e incluso de diferentes sensores, proporcionados por múltiples *SEnviro Things*. Esto ayuda a comparar los valores del mismo fenómeno dentro de la misma red.

Para implementar los requisitos descritos anteriormente, se ha buscado una solución que ofreciera flexibilidad, compatibilidad, así como cumplimiento de los estándares. Existen varios frameworks que facilitan la creación de clientes con mapas interactivos mediante HTML5. En particular se ha utilizado de los siguientes frameworks:

- Leaflet¹³ con la cartografía ESRI¹⁴, para poner los marcadores en el mapa. Leaflet demuestra ser rápido y eficaz. Además, se puede ejecutar en entornos restrictivos, tales como los teléfonos inteligentes. Leaflet es una librería de cartografía web de software libre.
- Otra librería utilizada es Bootstrap¹⁵. Ofrece la capacidad de crear páginas web responsive pudiendo adaptarse a las características del dispositivo. Además, se utiliza jQuery¹⁶ para manejar las ventanas emergentes y mostrar los gráficos.
- Finalmente, otra herramienta utilizada es Highchart JS¹⁷. Es una librería gráfica escrita en HTML5 y JavaScript. La librería proporciona una manera fácil e interactiva para la generación de gráficos en un entorno web. Highchart JS es gratuita para usos no comerciales.

6.2. Ejemplo de ejecución del GSF

En esta sección se propone un ejemplo de uso para la herramienta de publicación GSF, descrita en el Capítulo 3. Como se comentó, GSF es capaz de publicar

¹³Leaflet: An Open-Source JavaScript Library for Mobile-Friendly, <http://leafletjs.com>

¹⁴ESRI, <http://www.esri.com>

¹⁵Bootstrap, Twitter <http://getbootstrap.com>

¹⁶jQuery, jQuery Fundación <http://jquery.com>

¹⁷Highcharts JS, <http://www.highcharts.com>

contenido geoespacial en diferentes formatos, tanto ráster como vectorial. Aunque GSF está pensado para ser utilizado para la publicación de datos provenientes de sensores, también tiene la capacidad de publicar otros tipos de datos geoespaciales. En esta sección, se presenta un ejemplo de publicación de un dato vectorial (shapefile) que contiene las incendios declarados entre los años 2005 y 2010 en la provincia de Castellón (España) (Figura 6.6).

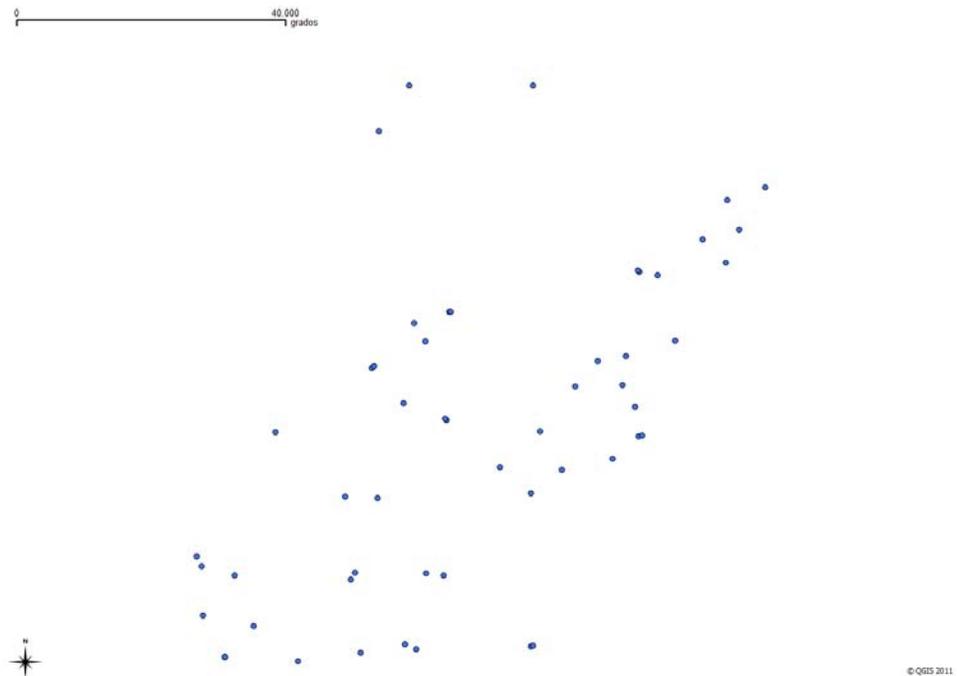


Figura 6.6: Representación de los incendios producidos en Castellón entre los años 2005 al 2010.

El objetivo de este procesamiento para la publicación, es ofrecer un mecanismo para que los datos estén disponibles en los diferentes servicios de las SDIs para mejorar su accesibilidad y que puedan ser reutilizados por otros usuarios. Además, con el fin de mejorar la visibilidad, los datos son publicados tanto en los servicios de visualización como de descarga. Para su descubrimiento se crea el metadato asociados al contenido publicado. También estos tienen que ser publicados en un catálogo de metadatos, el cual contendrá el enlace al contenido referenciado y las palabras clave que lo describen.

Actualmente, uno de los servidores de contenido geoespacial capaz de ofrecer

los servicios de visualización y descarga, es Geoserver¹⁸. Dicho servidor es uno de los más empleados en la actualidad y ha sido elegido para la publicación de los datos. Por otra parte, el catálogo de metadatos más importante es GeoNetwork¹⁹. Por lo que también se ha elegido dicha implementación para la publicación del metadatos.

El primer paso para configurar la herramienta GSF es definir el fichero de configuración SPP. El SPP es un documento XML que almacena toda la información para la ejecución de GSF. Por ejemplo, para cada tipo de contenido indica en qué servicios publicar. Además también almacena la dirección de los servidores a los que se publica la información, así como sus usuarios y contraseñas de acceso. El siguiente XML (Listado 6.1), muestra un ejemplo de SPP.

Listado 6.1: Ejemplo de Service Publication Profile.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServicePublicationProfile xmlns="http://geoinfo.uji.es/gsf" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://geoinfo.uji.es/gsf http://elcano.dlsi.uji.es/ldiaz/schemas/ServicePublicationProfile.xsd">
  <ServiceType name="Download" standard="WFS" version="1.0.0" url="http://elcano.dlsi.uji.es:8080/geoserver/">
    <Property name="supportedFormat">text/xml</Property>
    <Property name="supportedFormat">application/x-zipped-shp</Property>
    <Property name="supportedFormat">application/zip</Property>
    <Property name="supportedSchema">http://schemas.opengis.net/gml/3.0.0/base/feature.xsd</Property>
    <Property name="supportedSchema">http://schemas.opengis.net/gml/3.0.1/base/feature.xsd</Property>
    <Property name="supportedSchema">http://schemas.opengis.net/gml/3.1.0/base/feature.xsd</Property>
    <Property name="supportedSchema">http://schemas.opengis.net/gml/3.1.1/base/feature.xsd</Property>
    <Property name="supportedSchema">http://schemas.opengis.net/gml/3.2.1/base/feature.xsd</Property>
    <Property name="supportedSchema">http://schemas.opengis.net/gml/2.0.0/feature.xsd</Property>
    <Property name="supportedSchema">http://schemas.opengis.net/gml/2.1.1/feature.xsd</Property>
    <Property name="supportedSchema">http://schemas.opengis.net/gml/2.1.2/feature.xsd</Property>
    <Property name="supportedSchema">http://schemas.opengis.net/gml/2.1.2.1/feature.xsd</Property>
    <Property name="supportedEncoding">UTF-8</Property>
    <Property name="transactionalProtocol">Geoserver</Property>
    <Property name="user">admin</Property>
    <Property name="password">cronopio</Property>
  </ServiceType>
  <ServiceType name="Download" standard="SOS" url="http://elcano.dlsi.uji.es/52north/sos">
    <Property name="supportedSchema">http://schemas.opengis.net/om/1.0.0/om.xsd</Property>
    <Property name="supportedFormat">text/xml</Property>
    <Property name="supportedEncoding">UTF-8</Property>
    <Property name="transactionalProtocol">SOS-T</Property>
  </ServiceType>
  <ServiceType name="View" standard="WMS" url="http://elcano.dlsi.uji.es:8080/geoserver/">
    <Property name="supportedFormat">application/x-zipped-shp</Property>
    <Property name="supportedFormat">application/zip</Property>
```

¹⁸<http://geoserver.org>

¹⁹<http://geonetwork-opensource.org>

```

<Property name="supportedFormat">application/img</Property>
<Property name="supportedFormat">application/geotiff</Property>
<Property name="supportedFormat">application/dbase</Property>
<Property name="supportedFormat">application/remap</Property>
<Property name="supportedFormat">text/xml</Property>
<Property name="transactionalProtocol">Geoserver</Property>
<Property name="user">admin</Property>
<Property name="password">cronopio</Property>
</ServiceType>
<ServiceType name="Discovery" standard="CSW" url="http://elcano.dlsi.uji.es:8080/geonetwork/srv/en/csw">
  <Property name="supportedFormat">text/xml</Property>
  <Property name="supportedEncoding">UTF-8</Property>
  <Property name="transactionalProtocol">CSW-T</Property>
  <Property name="user">admin</Property>
  <Property name="password">cronopio</Property>
</ServiceType>
</ServicePublicationProfile>

```

Para la publicación del shapefile anunciado ha de definirse una petición para la operación *Execute* del GSF. En dicha petición se definirán las entradas del geoprocésamiento, donde se indica la ruta al contenido a publicar, la ruta al fichero de configuración y las palabras del contenido. A continuación se muestra un ejemplo de llamada *Execute* para el ejemplo propuesto (Listado 6.2).

Listado 6.2: Ejemplo de petición *Execute*.

```

<?xml version="1.0" encoding="UTF-8"?>
<wps:Execute xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:xlink="
  http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" service="WPS" version="
  1.0.0" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://geoserver.itc.nl:8080/wps/schemas/wps/1.0.0/
  wpsExecute_request.xsd">
  <ows:Identifier>org.n52.wps.server.algorithm.Publish</ows:Identifier>
  <wps>DataInputs>
    <wps:Input>
      <ows:Identifier>content</ows:Identifier>
      <wps:Reference mimeType="application/zip" xlink:href="http://elcano.dlsi.uji.es/data/incend_2005_4326.zip" />
    </wps:Input>
    <wps:Input>
      <ows:Identifier>ServicePublicationProfile</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://elcano.dlsi.uji.es/data/ServicePublicationProfile2.xml" />
    </wps:Input>
    <wps:Input>
      <ows:Identifier>keywords</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>a,b,c,d</wps:LiteralData>
      </wps>Data>
    </wps:Input>
  </wps>DataInputs>
</wps:Execute>

```

Después de la ejecución, se publican los datos a un servidor de datos geoespaciales, siendo el *getCapabilities* accesible a los servicios WMS y WFS. El siguiente XML (Listado 6.3) muestra información acerca del dato publicado. Tras su ejecución, vemos cómo se ha publicado el dato geoespacial en el servidor de datos Geoserver (Figura 6.7). Si se accede al dato publicado se obtiene la información al respecto (Figura 6.8).

Listado 6.3: Ejemplo de *getCapabilities*.

```

<Layer>
  <Name>eurogeoss:incend_2005_4326</Name>
  <Title>incend_2005_4326</Title>
  <Abstract/>
  <KeywordList/>
  <SRS>EPSG:4326</SRS>
  <!--
  WKT definition of this CRS:
  GEOGCS["WGS 84",
  DATUM["World Geodetic System 1984",
  SPHEROID["WGS 84", 6378137.0, 298.257223563, AUTHORITY["EPSG","7030"]],
  AUTHORITY["EPSG","6326"]],
  PRIMEM["Greenwich", 0.0, AUTHORITY["EPSG","8901"]],
  UNIT["degree", 0.017453292519943295],
  AXIS["Geodetic longitude", EAST],
  AXIS["Geodetic latitude", NORTH],
  AUTHORITY["EPSG","4326"]]
  -->
  <LatLonBoundingBox minx="-1.4795661150193993" miny="37.90006305852903" maxx="0.3527841387963444"
    maxy="40.68920370363359"/>
  <BoundingBox SRS="EPSG:4326" minx="-1.4795661150193993" miny="37.90006305852903" maxx="
    0.3527841387963444" maxy="40.68920370363359"/>
  <Style>
    <Name>polygon</Name>
    <Title>Default polygon style</Title>
    <Abstract>A sample style that just draws out a solid gray interior with a black 1px outline</Abstract>
    <LegendURL width="20" height="20">
      <Format>image/png</Format>
      <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple" xlink:href="http://elcano.dlsi.uji.
        es:8080/geoserver/wms?request=GetLegendGraphic&format=image %2Fpng&width=20&height=20&layer=
        incend_2005_4326"/>
    </LegendURL>
  </Style>
</Layer>

```

Tras la publicación del dato, se genera el metadato y se publica en GeoNetwork. La Figura 6.9 muestra de forma visual el metadato publicado.

Una ejecución completa de WPS, con la publicación de datos y generación y publicación de metadatos, tiene un coste de tiempo de 18 segundos. Este tiempo

Tipo	Espacio de trabajo	Almacén	Nombre de la capa	Habilitada?	SRS nativo
WFS	europaesa	zotolabad_1311756065566	incend_2005_4326	✓	EPSG:4326
WFS	sf	sf	roads	✓	EPSG:26713
WFS	sf	sf	bugsites	✓	EPSG:26713
WFS	sf	sf	archotes	✓	EPSG:26713
WFS	sf	sf	streams	✓	EPSG:26713
WFS	sf	sf	restricted	✓	EPSG:26713
WFS	sf	stodem	stodem	✓	EPSG:26713
WFS	topp	stades_shapefile	stades	✓	EPSG:4326
WFS	topp	tas_shape	tasmania_cities	✓	EPSG:4326
WFS	topp	tas_shape	tasmania_water_bodies	✓	EPSG:4326
WFS	topp	tas_shape	tasmania_roads	✓	EPSG:4326
WFS	topp	tas_shape	tasmania_state_boundaries	✓	EPSG:4326
WFS	nurc	img_sample2	PK50095	✓	EPSG:32633
WFS	nurc	worldImageSample	img_Sample	✓	EPSG:4326
WFS	nurc	mosaic	mosaic	✓	EPSG:4326
WFS	nurc	arcSndSample	Arc_Sample	✓	EPSG:4326
WFS	tiger	nyc	pork_landmarks	✓	EPSG:4326
WFS	tiger	nyc	tiger_roads	✓	EPSG:4326
WFS	tiger	nyc	gamt_polygon	✓	EPSG:4326
WFS	tiger	nyc	poi	✓	EPSG:4326

Figura 6.7: Captura de los datos publicados en Geoserver.

es causado en su medida por las plantillas XSLT ya que son un proceso costoso²⁰.

Otra experimentación realizada ha sido la publicación de datos provenientes de sensores, tanto para el registro de sensores como para insertar nuevas observaciones. En este caso la publicación se realiza sobre un servidor SOS, más concretamente se ha utilizado la implementación de 52North²¹ que sigue la especificación OGC SOS 2.0. Se ha publicado un sensor mediante la operación SensorRegister y el siguiente XML (Listado 6.4) muestra un ejemplo de registro de sensor. Para este caso, la publicación de sensores u observaciones tiene un coste temporal de 2 segundos en el mismo servidor.

Listado 6.4: Ejemplo de registro de sensor.

```
<?xml version="1.0" encoding="UTF-8"?>
<RegisterSensor xmlns="http://www.opengis.net/sos/1.0" xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc" xmlns:om="http://www.opengis.net/om/1.0" xmlns:ows="http://www.opengis.net/ows" xmlns:sml="http://www.opengis.net/sensorML/1.0.1" xmlns:swe="http://www.opengis.net/swe/1.0.1" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" service="SOS" version="1.0.0" xsi:schemaLocation="http://www.opengis.net/sos/1.0 http://schemas.opengis.net/sos/1.0.0/
```

²⁰Las características de la máquina son las siguientes: Intel(R) Xeon(R) CPU 5160 @ 3.00GHz, RAM Memory 16 GB DDR2 FB-DIMM Synchronous 667 MHz.

²¹<http://52north.org/>

Identificado como admin. Cerrar sesión

servidor

- Estado del servidor
- Información de contacto
- Configuración global
- Configuración de JAI
- Acercas de GeoServer

Servicios

- WFS
- WCS
- WFS
- WMS

Datos

- Espacios de trabajo
- Almacenes de datos
- Capas
- Grupos de capas
- Estilos

Security

- Users
- Data security
- Service security
- Catalog security

Demos

Previsualización de capas

eurogeoss:incend_2005_4326

Configure el recurso y la información de publicación para esta capa

Datos **Publicación**

Información básica del recurso

Nombre:

Título:

Resumen:

Palabras clave

Palabras clave actuales:

Nueva palabra clave:

Vínculos a metadatos

No hay vínculos de metadatos hasta el momento

Sistema de referencia de coordenadas

SRS nativo:

SRS declarado:

Gestión de SRS:

Encuadres

Encuadre nativo

Mín X	Mín Y	Máx X	Máx Y
-1.48	37.9	0.353	40.689

Encuadre Lat/Lon

Mín X	Mín Y	Máx X	Máx Y
-1.48	37.9	0.353	40.689

Detalles del Feature Type

Propiedad	Tipo	Ítulo permitido	Ocurrencias min/máx
the_geom	MultiPolygon	true	0/1
NUMPARTE	String	true	0/1
MUNICIPIOS	String	true	0/1
PROVINCIA	String	true	0/1
PARAJE	String	true	0/1
HOLA	String	true	0/1
FECHA	Date	true	0/1
CAUSA_CORT	String	true	0/1
SUP_RASA	Double	true	0/1
SUP_ARIOLA	Double	true	0/1
SUP_TOTAL	Double	true	0/1
SUPER_SHP	Double	true	0/1

Figura 6.8: Captura de la información del dato publicado en Geoserver.

```

sosRegisterSensor.xsd http://www.opengis.net/om/1.0 http://schemas.opengis.net/om/1.0.0/extensions/
observationSpecialization_override.xsd">
<SensorDescription>
  <sml:SensorML version="1.0.1">
    <sml:member>
      <sml:System>
        <sml:identification>
          <sml:IdentifierList>

```

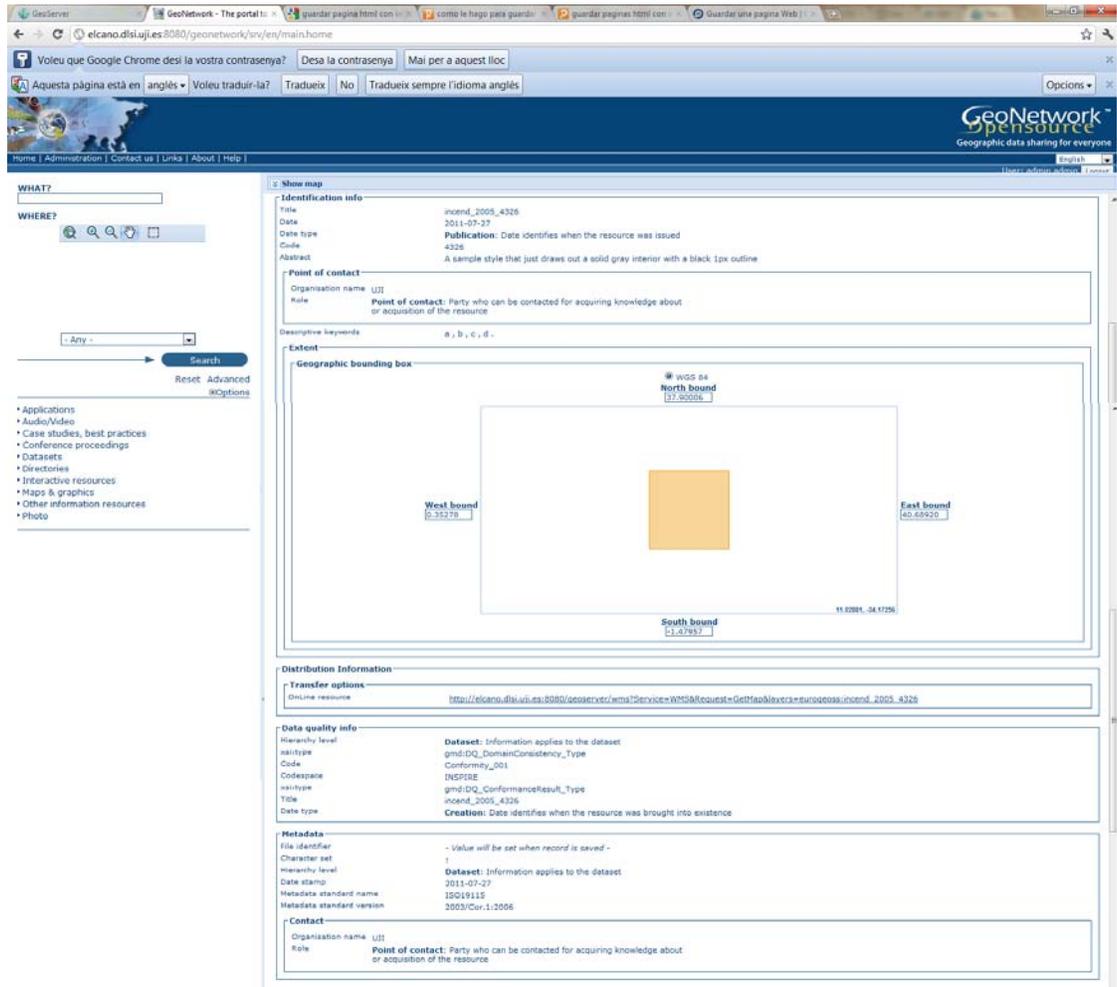


Figura 6.9: Captura de la publicación del metadato en GeoNetwork.

```

<sml:identifier>
  <sml:Term definition="urn:ogc:def:identifier:OGC:uniqueID">
    <sml:value>urn:ogc:object:feature:Sensor:IFGI:ifgi-sensor-1</sml:value>
  </sml:Term>
</sml:identifier>
</sml:IdentifierList>
</sml:identification>
<sml:position name="sensorPosition">
  <swe:Position referenceFrame="urn:ogc:def:crs:EPSG::4326">
    <swe:location>
      <swe:Vector gml:id="STATION.LOCATION">
        <swe:coordinate name="easting">
          <swe:Quantity>
            <swe:uom code="degree" />
            <swe:value>7.52</swe:value>
          </swe:Quantity>
        </swe:coordinate>
      </swe:Vector>
    </swe:location>
  </swe:Position>
</sml:position>

```

```

    </swe:coordinate>
    <swe:coordinate name="northing">
      <swe:Quantity>
        <swe:uom code="degree" />
        <swe:value>52.90</swe:value>
      </swe:Quantity>
    </swe:coordinate>
    <swe:coordinate name="altitude">
      <swe:Quantity>
        <swe:uom code="m" />
        <swe:value>52.0</swe:value>
      </swe:Quantity>
    </swe:coordinate>
  </swe:Vector>
</swe:location>
</swe:Position>
</sml:position>
<sml:inputs>
  <sml:InputList>
    <sml:input name="waterlevel">
      <swe:ObservableProperty definition="urn:ogc:def:phenomenon:OGC:1.0.30:waterlevel" />
    </sml:input>
  </sml:InputList>
</sml:inputs>
<sml:outputs>
  <sml:OutputList>
    <sml:output name="waterlevel">
      <swe:Quantity definition="urn:ogc:def:phenomenon:OGC:1.0.30:waterlevel">
        <gml:metaDataProperty>
          <offering>
            <id>GAUGE_HEIGHT</id>
            <name>gauge height in Muenster</name>
          </offering>
        </gml:metaDataProperty>
        <swe:uom code="cm" />
      </swe:Quantity>
    </sml:output>
  </sml:OutputList>
</sml:outputs>
<sml:components>
  <sml:ComponentList>
    <sml:component name="gaugeSensor">
      <sml:Component>
        <sml:identification>
          <sml:IdentifierList>
            <sml:identifier>
              <sml:Term definition="urn:ogc:def:identifier:OGC:uniqueID">
                <sml:value>urn:ogc:object:feature:Sensor:water_level_sensor</sml:value>
              </sml:Term>
            </sml:identifier>
          </sml:IdentifierList>
        </sml:identification>
      </sml:Component>
    </sml:component>
  </sml:ComponentList>
</sml:components>

```

```

    <sml:InputList>
      <sml:input name="gaugeHeight">
        <swe:ObservableProperty definition="urn:ogc:def:phenomenon:OGC:1.0.30:waterlevel" />
      </sml:input>
    </sml:InputList>
  </sml:inputs>
  <sml:outputs>
    <sml:OutputList>
      <sml:output name="gaugeHeight">
        <swe:Quantity definition="urn:ogc:def:phenomenon:OGC:1.0.30:waterlevel">
          <swe:uom code="cm" />
        </swe:Quantity>
      </sml:output>
    </sml:OutputList>
  </sml:outputs>
</sml:Component>
</sml:component>
</sml:ComponentList>
</sml:components>
</sml:System>
</sml:member>
</sml:SensorML>
</SensorDescription>
<ObservationTemplate>
  <om:Measurement>
    <om:samplingTime />
    <om:procedure />
    <om:observedProperty />
    <om:featureOfInterest />
    <om:result xsi:type="gml:MeasureType" uom="">0.0</om:result>
  </om:Measurement>
</ObservationTemplate>
</RegisterSensor>

```

6.3. Propuesta para el acceso a redes heterogéneas

Esta sección presenta un caso de estudio para probar algunas de las aportaciones detalladas en las secciones anteriores. En primer lugar, se describen las redes de sensores utilizadas como proveedores de datos (Sección 6.3.1). Dichas redes no siguen ningún estándar, por lo que necesitan un proceso de integración particular para cada una de ellas. La Sección 6.3.2 describe dicho pre-procesado. Tras este tratamiento, estas redes ya están disponibles para su acceso mediante la interfaz detallada en la Sección 4.3. La Sección 6.3.3 muestra los servicios para este caso de uso. Finalmente, se presenta una aplicación móvil capaz de utilizar dicha interfaz para visualizar las observaciones y datos de los propios sensores (Sección 6.3.4).

6.3.1. Fuentes de datos utilizadas

Se utilizan dos fuentes de datos diferentes, donde cada una de ellas ofrece las observaciones mediante un formato diferente. La primera de éstas es la red de vigilancia y control de la contaminación del aire de la Comunidad Valenciana²². Esta fuente de datos se considera como oficial ya que tiene un proceso de validación. La segunda, se trata de la red de nodos de sensorización de Meteoclimatic²³, que proporciona información meteorológica en tiempo real en toda la península Ibérica, Canarias y Baleares. Meteoclimatic es una red de voluntarios por lo que no tiene validación de sus observaciones.

Información oficial de sensores: red de calidad del aire de la Generalitat Valenciana

Se trata de una red de estaciones que realiza un seguimiento de los niveles de los distintos contaminantes y elementos meteorológicos en las zonas urbanas, rurales e industriales, extendiéndose dicho control a todo el territorio de la Comunidad Valenciana.

Los contaminantes medidos son los siguientes: dióxido de azufre (SO₂), monóxido de nitrógeno (NO), dióxido de nitrógeno (NO₂), óxidos de nitrógeno (NO_x), monóxido de carbono (CO), ozono (O₃), el benceno (C₆H₆), otros hidrocarburos tales como tolueno y xileno y las partículas de diámetro son inferiores de 10 micras (PM₁₀), de 2,5 micras (PM 2.5) y de 1 micra (PM₁). También se analizan los metales como el arsénico, el níquel, el cadmio, el plomo y los hidrocarburos aromáticos y policíclicos en fracciones de PM₁₀. Algunas estaciones también tienen sensores meteorológicos para diversos fenómenos, tales como la velocidad y dirección del viento, humedad relativa, radiación solar, presión atmosférica y precipitación.

En la actualidad, la red de calidad del aire tiene 61 estaciones en funcionamiento: 24 en Castellón, 24 en Valencia y 13 en Alicante (Figura 6.10). Estos datos se publican en un sitio web que se valida y actualiza cada hora con los datos que se han obtenido cada 2-3 horas antes.

²²<http://www.citma.gva.es>

²³www.meteoclimatic.net



Figura 6.10: Mapa de las 61 estaciones de calidad del aire en la red Valencia.

Información voluntaria de sensores: Meteoclimatic

Meteoclimatic es una gran red de estaciones meteorológicas no profesionales, que toman mediciones automáticas en tiempo real. La red Meteoclimatic se extiende a través de: la Península Ibérica, los dos archipiélagos, el sur de Francia y África (próximo al Estrecho de Gibraltar). Actualmente hay más de 1000 estaciones (Figura 6.11).

Meteoclimatic mide componentes meteorológicos tales como la temperatura, humedad, viento, presión, precipitación o radiación. La frecuencia de actualización es irregular y varía de cada uno de los nodos, pero la más habitual es cada 10 minutos.

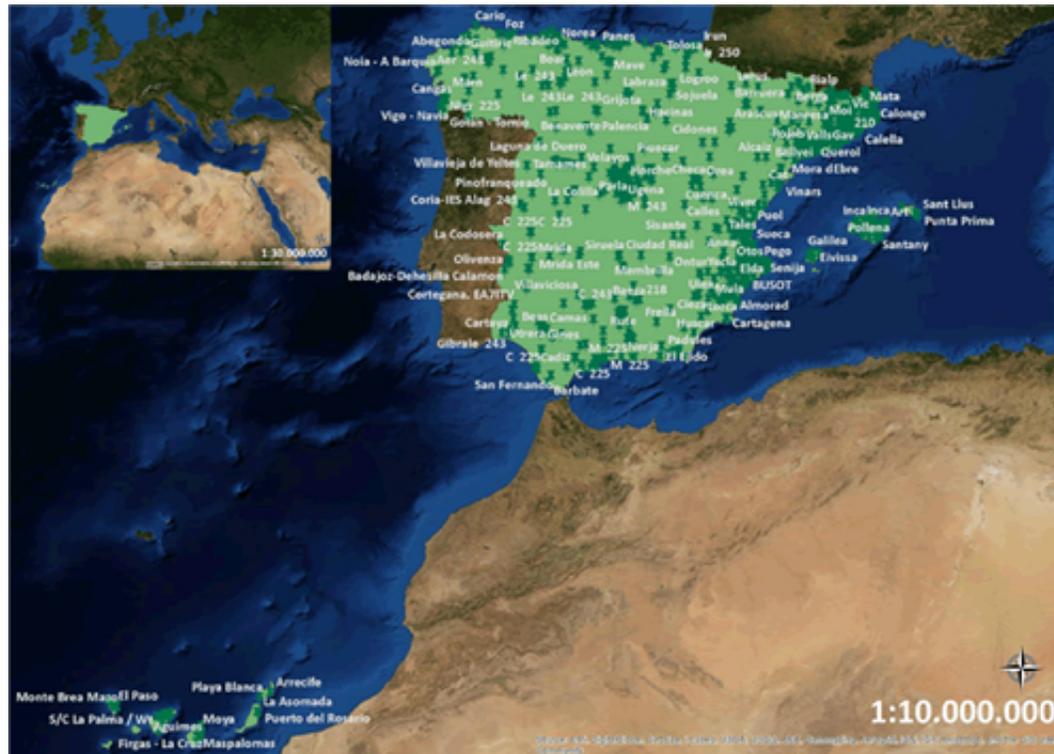


Figura 6.11: Mapa de las estaciones de la red Meteoroclimatic.

6.3.2. Pre-procesado de las fuentes

Como primer paso para empezar a trabajar con las fuentes detalladas en las secciones anteriores, se realizan varios pasos de pre-procesamiento, donde se preparan e integran las diferentes fuentes. Siguiendo la interfaz detallada en la Sección 4.3.1, se han preparado dos interfaces diferentes (REST y SOS). Por lo tanto, dos módulos diferentes se han diseñado para analizar e integrar las observaciones para cada una de estas dos interfaces (Figura 6.12). Estos módulos se ejecutan concurrentemente y usan dos submódulos comunes. Los números de las flechas en la figura (Figura 6.12) marcan la secuencia en que se ejecutan los distintos submódulos. Además, estos dos módulos se encuentran entre las capas de servicio y contenido de la arquitectura INSPIRE. A continuación se definen cada uno de ellos.

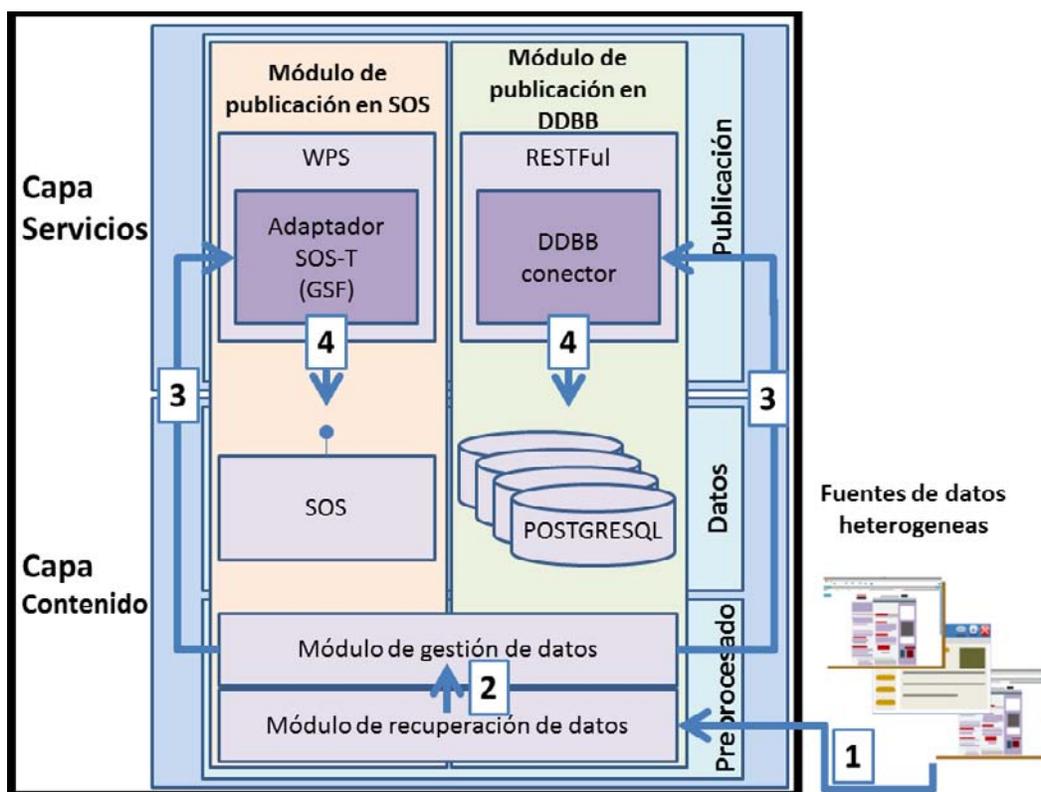


Figura 6.12: Módulo de publicación en BBDD y SOS.

Módulo de publicación en base de datos

Este módulo ha sido diseñado para la implementación de la interfaz de descarga RESTful. La tarea realizada por este módulo, es conectarse a las diferentes fuentes de datos, para obtener las observaciones correspondientes y guardarlas en una base de datos.

En la capa de contenido, este módulo ofrece dos submódulos para la recuperación de datos y otro para conectar a los distintos servicios de publicación.

El módulo de recuperación de datos es responsable de extraer las observaciones directamente de las estaciones de la red de sensores. De esta manera, este submódulo se conecta y procesa los datos para el almacenamiento, permitiendo que el sistema publique las observaciones a medida que se generen en las diferentes fuentes de datos.

Las fuentes detalladas anteriormente son publicadas en formato lenguaje de marcas de hipertexto (HTML, en sus siglas en inglés) y de Sindicación Realmente

Simple (RSS, en sus siglas en inglés), por lo que, este submódulo utiliza técnicas de *wrapping* web para extraer las observaciones de cada nodo. Este submódulo utiliza algunas librerías para la extracción. El submódulo adapta su ejecución a la frecuencia de publicación de cada fuente de sensores.

El submódulo de gestión de datos es responsable de la codificación de los sensores y las observaciones, además de publicarlas en la base de datos PostgreSQL²⁴ con PostGIS²⁵ y el servicio SOS.

En la capa de servicios, más concretamente en el servicio de publicación para la base de datos, el conector para la publicación es el responsable de la conexión con la base de datos, además de realizar las inserciones en ella.

Ya en la capa de contenidos, se almacena toda la información para cada sensor, como puede ser su ubicación, dirección o código, entre otros. Todas las observaciones de los fenómenos de cada sensor también se almacenan en la base de datos.

Módulo de publicación en SOS

A parte del módulo para la publicación en bases de datos (BBDD), se ha implementado un módulo para publicar los diferentes sensores y observaciones en un servidor de SOS. Este módulo integra tres submódulos diferentes que son: la recuperación de datos, la gestión de datos y el adaptador para el perfil transaccional del SOS (SOS-T) (Figura 6.12).

El módulo de recuperación es el mismo que se ha detallado para el módulo de publicación en BBDD. El módulo de gestión de datos es responsable de la codificación para las inserciones de los sensores y observaciones, para la publicación en el servicio SOS. El adaptador SOS-T es el responsable de establecer la conexión con el servicio de SOS y permitir la publicación de nuevos sensores y observaciones.

El adaptador de SOS-T incluye dos operaciones: *RegisterSensor* y *InsertObservation*. La operación *RegisterSensor* permite registrar un nuevo sensor al servicio. La operación de *InsertObservation* permite introducir nuevas observaciones sobre un sensor. Este submódulo utiliza la herramienta de publicación GSF (Capítulo 3).

Como ya se ha comentado, GSF permite la publicación de contenido geoespa-

²⁴<http://www.postgresql.org/>

²⁵<http://postgis.net/>

cial en los diferentes servicios GIS. En este caso se utiliza para la publicación de datos de sensores. Además, GSF puede generar y publicar metadatos para cada dato geoespacial que se publique.

De esta forma, el primer paso es publicar los datos de los sensores en el servidor SOS, siendo el segundo paso la generación de metadatos y su publicación en un catálogo. El beneficio de este proceso se encuentra en la posibilidad de descubrir el dato publicado. Todos los datos insertados se encuentran en un catálogo y pueden ser descubiertos, porque se describe con algunas palabras clave. Estas palabras claves están contenidas en los metadatos.

6.3.3. Servicios

En esta sección se presentan los servicios implementados para este caso de uso en particular. Estos se basan en las interfaces diseñadas en la Sección 4.3.1.

Dichos servicios son agrupados siguiendo la clasificación INSPIRE: descarga, visualización, publicación y procesado. La Figura 6.13 muestra cada uno de ellos. Como estos servicios e interfaces se detallaron en sus respectivos capítulos, no se detallarán nuevamente.

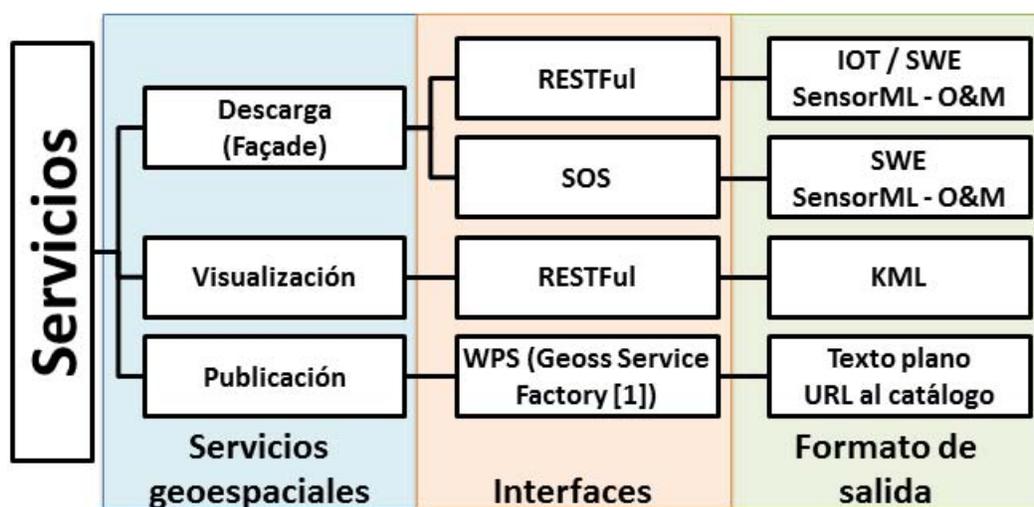


Figura 6.13: Esquema general con todos los servicios, interfaces y formatos de salida.

6.3.4. Cliente móvil

Para el uso y validación de los servicios anteriormente nombrados, se propone como prototipo inicial, una aplicación para el sistema operativo Windows Phone. La Figura 6.14 muestra los dos diferentes casos de uso de la aplicación móvil.

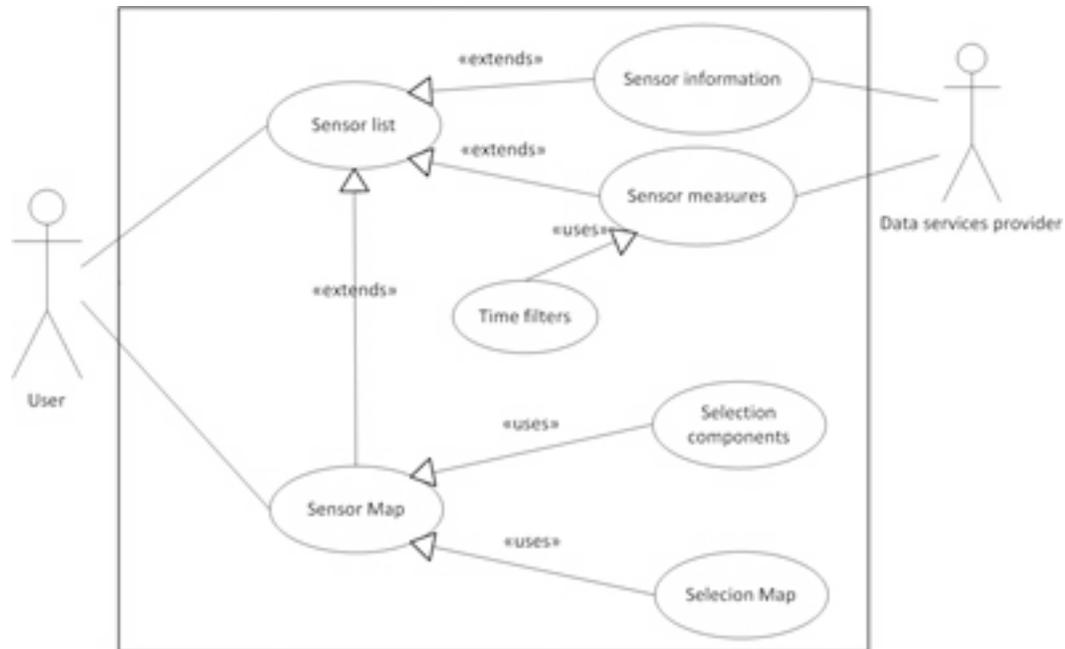


Figura 6.14: Casos de uso del cliente diseñado.

El primer uso verifica la capacidad de aplicación móvil para listar todos los nodos y la selección de uno en particular para consultar su información, así como las observaciones actuales y históricas. Se proporcionan filtros de tiempo para refinar las búsquedas. Además, la aplicación permite ubicar los nodos en un mapa y mostrar los últimos valores para ellos. Los usuarios también pueden seleccionar el contaminante y el elemento meteorológico para visualizar y seleccionar el tipo de mapa base.

La Figura 6.15 a) muestra una captura de pantalla de la aplicación móvil, donde se muestra la lista de todos los nodos de la red de calidad del aire de la región de Valencia. Para cada nodo, los usuarios pueden ver su nombre y la ciudad donde se encuentra. Si la última conexión con el nodo ha sido satisfactoria, se destaca con un círculo verde. Por el contrario, si la conexión ha fallado, se destaca con un círculo rojo.

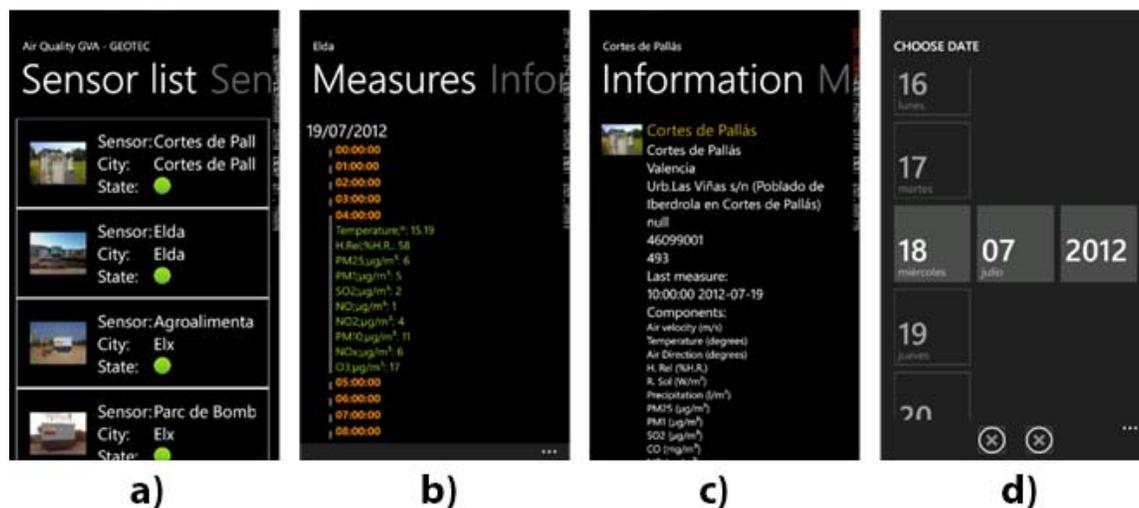


Figura 6.15: Diferentes pantallas. a) Lista de estaciones de calidad del aire. b) Información para una estación de calidad del aire. c) Observaciones de una estación de calidad del aire. d) Filtro temporal.

Cuando un usuario pulsa sobre un nodo de la lista, se visualiza una nueva pantalla con toda la información de los nodos (Figura 6.15 b)), como puede ser: el nombre del nodo, la ciudad, la dirección, el código de la estación, la altura, la última observación encontrada y fenómenos que monitoriza.

Si los usuarios hacen un arrastre horizontal, se pueden ver todas las observaciones del nodo seleccionado. Esta ventana muestra las observaciones actuales agrupadas por horas. Si se hace clic sobre el nodo aparece toda la información referente (Figura 6.15 c)). La aplicación también puede mostrar los días anteriores, utilizando el filtro temporal colocado en el menú en la parte inferior de la pantalla (Figura 6.15 d)).

El segundo uso verifica la capacidad de la aplicación móvil para mostrar todos los nodos en el mapa. La Figura 6.16 a) muestra cómo los nodos son ubicados con un marcador. Cuando un usuario hace clic en uno de ellos, aparece una ventana con la información de la última observación para dicho nodo, los fenómenos a mostrar se indican en la ventana de selección (Figura 6.16 b)); el mapa base predeterminado es el de Microsoft Bing, pero la aplicación es también capaz de utilizar los mapas base de ESRI y Google (Figura 6.16 c)), dicha opción puede ser seleccionada en el administrador de capas (Figura 6.16 d)).

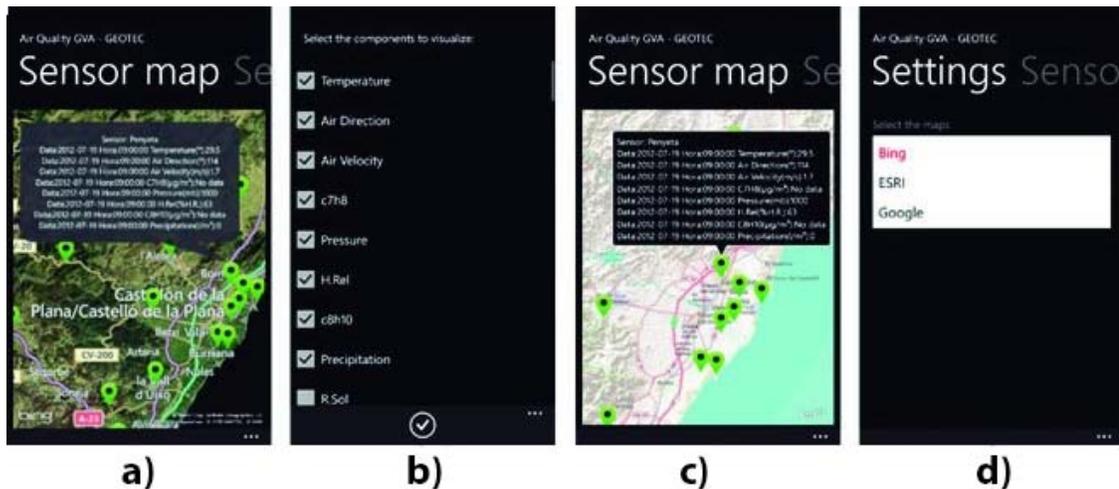


Figura 6.16: Diferentes pantallas. a) Últimas observaciones en mapas Bing. b) Selección de los fenómenos para mostrar. c) Últimas observaciones en mapas de ESRI. d) Selección de mapas.

6.4. Detección de anomalías sobre datos de sensores

En esta sección se detalla una implementación de la metodología de análisis de grandes cantidades de datos en tiempo real (Sección 5.3) para un caso de uso en particular, que es la detección de valores anómalos sobre observaciones de fenómenos de calidad del aire.

Para detectar valores anómalos se utiliza un algoritmo aplicado en el control de procesos industriales. Dicho algoritmo se detalla en la Sección 6.4.1. En la Sección 6.4.2 se presenta toda la implementación de la metodología para el caso de uso detallado.

6.4.1. Detección de eventos en tiempo real usando CUSUM

Con la finalidad de probar la metodología diseñada, se presenta un modelo de detección de anomalías sobre una serie de observaciones. Para ello se ha elegido un modelo llamado CUmulative SUM (CUSUM) (Page, 1954). CUSUM puede ser aplicado sobre cualquier serie de valores y detecta valores anómalos a medida que se producen. En esencia, este algoritmo compara dos instancias diferentes

de la misma función de probabilidad discreta para una serie de datos (Mesnil y P., 2009).

El algoritmo se desarrolló inicialmente para fines de control de procesos industriales. En los últimos años se ha utilizado con éxito en otras áreas. En la bibliografía existen varios trabajos que utilizan CUSUM, algunos de ellos se enumeran a continuación. Osanaiye y Talabi (1989), por ejemplo, utiliza el algoritmo CUSUM para detectar posibles brotes de epidemias. Grigg y otros (2003) analizó la mortalidad a 30 días para los pacientes después de una cirugía cardíaca.

CUSUM también es utilizado para mejorar la comunicación en WSNs. Jeske y otros (2009) desarrolló dos algoritmos de detección de cambio de punto. CUSUM además, ha sido utilizado en reconocimiento de patrones, específicamente en redes neuronales. Muestra de ellos es Guh y Hsieh (1999), donde se propone un modelo basado en red neuronal, que contiene varias redes de propagación posterior. Chuen-Sheng (1995) describe un enfoque alternativo para el control estadístico de procesos, utilizando la tecnología de redes neuronales y comparando su rendimiento con el de los esquemas combinados de Shewhart-CUSUM.

CUSUM ha ya sido utilizado con éxito para algunos problemas ambientales (Barratt y otros, 2007; Carslaw y otros, 2006; Chelani, 2011; Charles y Jeh-Nan, 2002). A raíz del éxito de estos trabajos, se ha decidido utilizar CUSUM como modelo para la detección de anomalías. Más concretamente se utiliza para la detección de anomalías sobre observaciones de calidad del aire.

CUSUM considera un conjunto de observaciones (x_i) recogidas entre $i = 1, \dots, n$ donde n es el número de puntos de datos. El algoritmo asume que estas observaciones se encuentran en control cuando el conjunto tiene una media (μ) y la desviación estándar (σ^2) para un período normal, siguiendo una distribución normal $N(\mu, \sigma^2)$. Cuando el proceso está en control, podemos obtener la suma acumulativa (S_i) de forma iterativa a través de la siguiente expresión:

$$S_i = S_{i-1}z_i \quad (6.1)$$

Donde $S_0=0$, z_i es la variable normal estándar, $z_i = \frac{x_i - \bar{x}}{s}$, \bar{x} es la media y s es la desviación estándar de la serie de tiempo. Además, el cambio en términos de aumento o disminución del proceso puede ser detectado, respectivamente con las siguientes expresiones (Lucas, 1982):

$$S_{H_i} = \text{MAX}[0, (z_i - k) + S_{H_{i-1}}] \quad (6.2)$$

$$S_{L_i} = \text{MIN}[0, (z_i + k) + S_{L_{i-1}}] \quad (6.3)$$

Donde el parámetro k es el valor de referencia para ser elegido apropiadamente. El parámetro, k , es la holgura admisible en el proceso y se establece generalmente a una mitad de la media que se desea detectar. Los límites de confianza (*umbral*) especificados para los gráficos de control CUSUM son $\pm h\sigma_x$, donde $h = 5$ y σ_x es la desviación estándar (Barratt y otros, 2007).

Cuando S_{H_i} o S_{L_i} superan el umbral, el algoritmo detecta un anomalía. Si el umbral es superior a S_{H_i} la anomalía será debido al aumento (*up-event*), y si es mayor que el umbral S_{L_i} , será debido a la disminución (*down-event*).

Dos características de CUSUM limitan la sensibilidad de los resultados. En primer lugar, la identificación de un proceso fuera de control se basa en la suposición de que las observaciones son estadísticamente independientes y siguen una distribución normal. En segundo lugar, las mediciones del fenómeno pueden estar afectadas por la estacionalidad o las tendencias a largo plazo. Esto puede acarrear que el umbral pueda estar fuera del ajuste correcto.

6.4.2. Propuesta de implementación

Tras detallar el modelo utilizado para la detección de anomalías, se muestra cómo se ha llevado a cabo. Los objetivos de este sistema son que tan pronto como se detecta la anomalía, se lance una notificación (evento) con el fin de (1) iniciar un proceso de toma de decisiones, e (2) informar sobre la anomalía que causó el evento, junto con información de contexto que rodea al sensor.

En esta sección se presenta dicha aplicación para detectar anomalías en los datos de sensores ambientales, como de calidad del aire. En esta sección se introduce sobre qué tipos de fuentes de datos se probará el sistema. Además muestra cómo se ha implementado la capa de servicios. Finalmente, se presenta un panel de control para probar la funcionalidad de la implementación realizada. La Figura 6.17 resume todas las tecnologías utilizadas para cada parte del sistema.

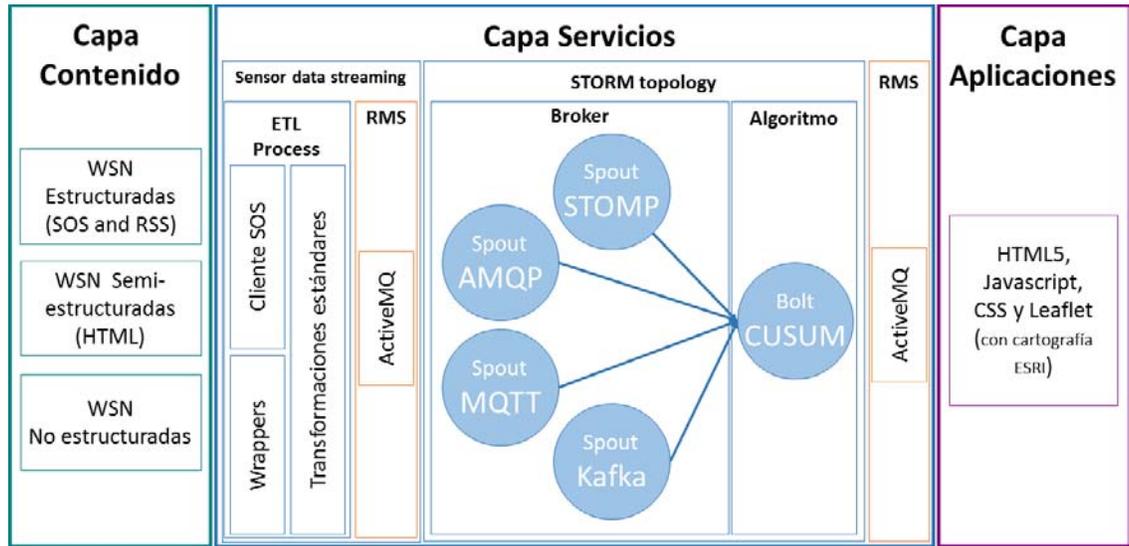


Figura 6.17: Resumen tecnológico de la implementación del sistema para detectar anomalías.

Capa contenido

El sistema propuesto contempla las tres categorías de fuentes de datos de WSN detalladas en el Capítulo 2, dependiendo del formato utilizado. Estos tipos son: WSNs estructuradas, semiestructuradas y no estructuradas. A continuación se propone un ejemplo para cada categoría:

- Estructuradas.** Esta categoría se divide en dos subcategorías diferentes. La primera de ellas, se caracteriza por incluir a las fuentes de datos de sensores proporcionadas mediante formatos estándares, como puede ser la especificación SOS de OGC. Un ejemplo de esta categoría es una fuente de datos basada en la implementación de 52North²⁶ para el estándar especificado. La segunda categoría reúne las fuentes de datos de sensores que siguen un formato estructurado, como puede ser RSS. Un ejemplo de estos es la red de estaciones meteorológica de Meteoclimatic (Sección 6.3.1) que ofrece las observaciones mediante el formato RSS.
- Semiestructuradas.** Se compone de redes de sensores servidas por formatos semiestructurados como es el HTML. La red de calidad del aire de la

²⁶<http://52north.org/communities/sensorweb/sos>

Comunidad Valenciana (Sección 6.3.1) es ofrecida como HTML. Esta fuente será la utilizada en primera estancia para este estudio.

- *No estructuradas*. Esta categoría engloba todas las fuentes de sensores que no siguen ningún formato obteniéndose directamente un valor, que es el de la observación. Un ejemplo de estas redes son los datos provenientes de la red *SEnviro* (Sección 6.1).

Para probar la metodología definida, se utiliza la red de sensores de calidad del aire de la Comunidad de Valencia. Esta fuente ya fue detallada en la Sección 6.3.1, por lo que no volverá a explicar.

Como se comentó anteriormente, para utilizar el algoritmo CUSUM se necesitan dos parámetros (umbral y k) por cada sensor diferente que el sistema es capaz de analizar. Dichos parámetros pueden ser obtenidos de datos históricos. Para ello, se ha utilizado un año de datos históricos. Para aplicar CUSUM y siguiendo los trabajos analizados en la bibliografía (Sección 6.4.1), se ha considerado que todos los fenómenos siguen una distribución normal.

Implementación de la capa de servicios

Esta sección muestra cómo se ha desarrollado la capa de servicios siguiendo el diseño detallado en las Secciones 4.3.3 y 5.3.3. Esta sección muestra cómo se ha creado el componente RMS. Seguidamente se presenta la implementación del SDS. Finalmente se detalla la tecnología utilizada para analizar los datos del sensor.

Servicio de mensajes real-time Para el componente RMS, se ha utilizado uno de los frameworks JMS que se describen en la Sección 4.2.3, llamado ActiveMQ. Con el objetivo de conectar con dicho servicio, se ofrece un cliente usando el protocolo STOMP, pudiendo ser utilizado tanto como *Productor* como *Consumidor*. Este cliente ofrece los diferentes modelos de conexión, *Punto a Punto* y *Publicación/Subscripción*. Será utilizado en los diferentes componentes posteriores con la finalidad de conectar con el RMS. Se utiliza únicamente una sola instancia de RMS, pero en el futuro se pueden añadir más instancias utilizando otros frameworks de los descritos en la Sección 4.2.2.

Sensor Streaming Data Como se detalló anteriormente (Sección 4.3.3), el SDS utiliza un proceso ETL para conectar y estandarizar las fuentes de datos de los sensores. La primera parte de este proceso es realizar las conexiones con las fuentes de datos de los sensores. Para ello, se establecen dos tipos de componentes dependiendo de la categoría de la fuente de datos. Para las fuentes de datos que siguen estándares como el SOS de OGC, se implementa un cliente SOS para realizar la conexión con el servidor SOS. Para lograr esto, se ha desarrollado la operación *GetObservationById* para recuperar la última observación de cada sensor. Para todas las demás fuentes, se requiere un componente específico para conectar con cada origen de datos. En estos casos, se crea un conector para cada fuente, con el fin de obtener las observaciones. Para este trabajo se utilizan los conectores detallados en la Sección 6.3.1.

Los conectores necesitan varios parámetros para establecer la conexión de forma correcta. Uno de ellos es la frecuencia de actualización y el enlace para conectar con las múltiples fuentes. Para almacenar dicha información, se utiliza un archivo XML. Este archivo contiene una sola entrada para definir una red de sensores. Además, también contiene los detalles sobre cada uno de los nodos sensoriales: un identificador, nombre y ciudad, estado o ubicación. Por cada nodo sensorial se incluye un elemento separado para cada sensor (fenómeno) que mide ese nodo. Cada uno de estos elementos contiene información detallada sobre el fenómeno: un identificador, propiedad observada, unidad de medida y frecuencia de actualización. También contiene los parámetros que se necesitan para ejecutar el modelo CUSUM (umbral y k) para ese fenómeno. Finalmente, cada fenómeno también contiene el enlace para conectar con la fuente de datos del sensor y el enlace para conectar con el RMS, además de acceder al flujo de observaciones para dicho fenómeno.

El SDS tiene como objetivo estandarizar las observaciones tanto en formato de codificación como en acceso. Para ello, las observaciones de fuentes de datos que no son estándares se les aplica una transformación. Cada observación será transformada utilizando el estándar O&M, utilizando para ello un analizador DOM. El O&M resultante es el detallado en la Sección 3.2.2.

Para conectar con el RMS y obtener el flujo de datos con las observaciones para cada sensor, una instancia del cliente RMS introducido se utiliza como *Productor*. Las observaciones producidas son enviadas a las correspondientes *Colas* (flujos de observaciones), habiendo una *Cola* diferente por nodo sensorial y fenómeno.

La Figura 4.9 muestra cómo se ha utilizado el RMS para dicha funcionalidad. En este caso, se aplica el modelo *Punto a Punto* del RMS, ya que asegura que el *Consumidor* de esas *Colas* obtendrá todos los *Mensajes* producidos. Se ha creado otro archivo XML para definir las configuraciones para conectarse con los flujos de datos de sensores. Este archivo ofrece la URL y el tipo de protocolo utilizado en cada flujo.

Topología para la detección de anomalías En esta sección se dará solución a dos partes distintas de la metodología definida, que son al bróker y el análisis. Para ello, se utiliza la herramienta descrita en la Sección 5.2.1 llamada Storm. Ésta resulta altamente confiable para la distribución y procesado de flujos de datos en tiempo real. El uso de Storm sobre el ambiente IoT es completamente novedoso, ya que no había estado aplicado anteriormente. Storm permite analizar varias secuencias (con diversos protocolos) y aplicar algoritmos de análisis, en este caso para la detección de anomalías. Como se comenta en la sección correspondiente, en Storm se definen topologías y éstas a su vez, describen dos componentes distintos, que son los *Spouts* y los *Bolts*. Los primeros se encargan de conectar con las fuentes de datos y los segundos se encargan de aplicar las operaciones sobre los datos leídos.

Para satisfacer el enfoque de bróker (Sección 5.3.1) se suministran varios conectores para diferentes protocolos, que servirán para conectar con las diferentes instancias del RMS. Estos son implementados como diferentes *Spouts* para cada uno de los protocolos de Message Broker soportados. Esto aumenta la interoperabilidad del sistema, ya que conecta con múltiples protocolos y permite utilizar diferentes instancias de RMS.

Para el bróker actual, se han implementado cuatro *Spouts* diferentes, de los cuales cada uno soporta un protocolo diferente. Se han seleccionado los cuatro protocolos más utilizados, siendo estos: Stomp, AMQP, MQTT y la conexión con Kafka. Cada *Spout* de la topología se desarrolla como un cliente *Consumidor*, con el fin de conectar cada uno con el diferente RMS disponible y así, obtener las observaciones de los sensores.

Storm ofrece la garantía de que todos los mensajes leídos serán procesados, pero no tiene en cuenta el orden en que los datos se emiten cuando son analizados en los *Bolts*. De hecho, las *tuplas* tienen un orden diferente cuando son procesados. Para el objetivo de este trabajo es necesario mantener el orden de

las observaciones de los sensores (*tuplas*). Para ello, se utiliza Trident ²⁷ pudiendo asegurar exactamente este hecho.

Trident es un framework con un nivel más alto de abstracción que Storm. Aunque una topología definida en Trident tiene diferentes operaciones, cuando se ejecuta la topología ésta es traducida a las tradicionales de Storm. Trident también utiliza *Spouts* para obtener las *tuplas* de las fuentes de datos. La diferencia recae en los Bolts que ofrecen operaciones más concretas, como funciones, filtros o agregaciones.

Para aplicar el modelo propuesto, se han diseñado los *Bolts* correspondientes para la funcionalidad definida. Dentro de los *Bolts* se puede aplicar cualquier algoritmo de análisis. Para el caso de uso actual, ha sido desarrollado un *Bolt* con el algoritmo CUSUM. Este *Bolt* es el encargado de aplicar CUSUM sobre la serie de observaciones proporcionadas por las diferentes *Spouts*. Para poder ejecutar el algoritmo CUSUM, en cada observación se ofrece la medida del fenómeno específico (como se describe en la Sección 4.3.3), creando *tuplas*, etiquetadas con identificadores únicos del sensor. El pseudocódigo (Algoritmo 1) de abajo resume como se ha aplicado CUSUM para la detección en tiempo real. Para la ejecución, se facilita una estructura de datos que acumula y almacena la suma para cada iteración. Finalmente, se comprueba si la suma acumulada para la nueva observación provoca un evento utilizando un umbral particular para cada fenómeno y nodo sensorial.

Cuando se detecta una anomalía mediante CUSUM, el *Bolt* usa el cliente para conectar con el Message Broker y enviar notificaciones a una instancia de RMS como *Proveedor*. En este caso, también se crea un *Cola* para cada nodo y fenómeno. El modelo utilizado para esta conexión es el *Publicación / Suscripción* (Figura 6.18).

Los eventos son codificados también en un formato estándar, llamado Common Alerting Protocol (CAP). CAP es un formato para codificar eventos diseñados para la distribución de las advertencias de forma pública. Fue estandarizado por la Organización para el Avance de Estándares de Información Estructurada (OASIS) en 2004, y utiliza XML como formato de codificación de datos. La Tabla 6.7 muestra el significado de cada elemento CAP.

Cada evento contiene un identificador del sensor (el elemento *Sender*) y el

²⁷<http://storm.apache.org/documentation/Trident-API-Overview.html>

Algorithm 1 Algoritmo CUSUM para tiempo real

```

1: procedure CUSUM EN TIEMPO REAL
2:   quantityOfSensors = cantidad de sensores compatibles en el sistema
3:    $SUM_{Previous_{alto}}[numPhen] = 0$  (inicializar el vector para cada sensor)
4:    $SUM_{Previous_{baja}}[numPhen] = 0$  (inicializar el vector para cada sensor)
5:   while NewObservation  $\neq false$  do
6:     ObservationValue = valor de la nueva observación
7:     idSensor = identificador del sensor
8:      $StandardNormalVariate = \frac{ObservationValue - \mu[idSensor]}{\sigma[idSensor]}$ 
9:      $SUM_{current_{alto}}[idSensor] = MAX[0, (StandardNormalVariate -$ 
     $k[idSensor]) + SUM_{previous_{alto}}[idSensor]]$ 
10:     $SUM_{current_{baja}}[idSensor] = MIN[0, (StandardNormalVariate +$ 
     $k[idSensor]) + SUM_{previous_{baja}}[idSensor]]$ 
11:    if  $SUM_{current_{alto}}[idSensor] \geq umbral[idSensor]$  then
12:      enviar Evento por arriba
13:       $SUM_{current_{alto}}[idSensor] = 0$ 
14:       $SUM_{current_{baja}}[idSensor] = 0$ 
15:    end if
16:    if  $SUM_{current_{baja}}[idPhen] \geq umbral[idSensor]$  then
17:      enviar Evento por abajo
18:       $SUM_{current_{baja}}[idSensor] = 0$ 
19:       $SUM_{current_{alto}}[idSensor] = 0$ 
20:    end if
21:     $SUM_{previous_{alto}}[idSensor] = SUM_{current_{alto}}[idSensor]$ 
22:     $SUM_{previous_{baja}}[idSensor] = SUM_{current_{baja}}[idSensor]$ 
23:  end while
24: end procedure

```

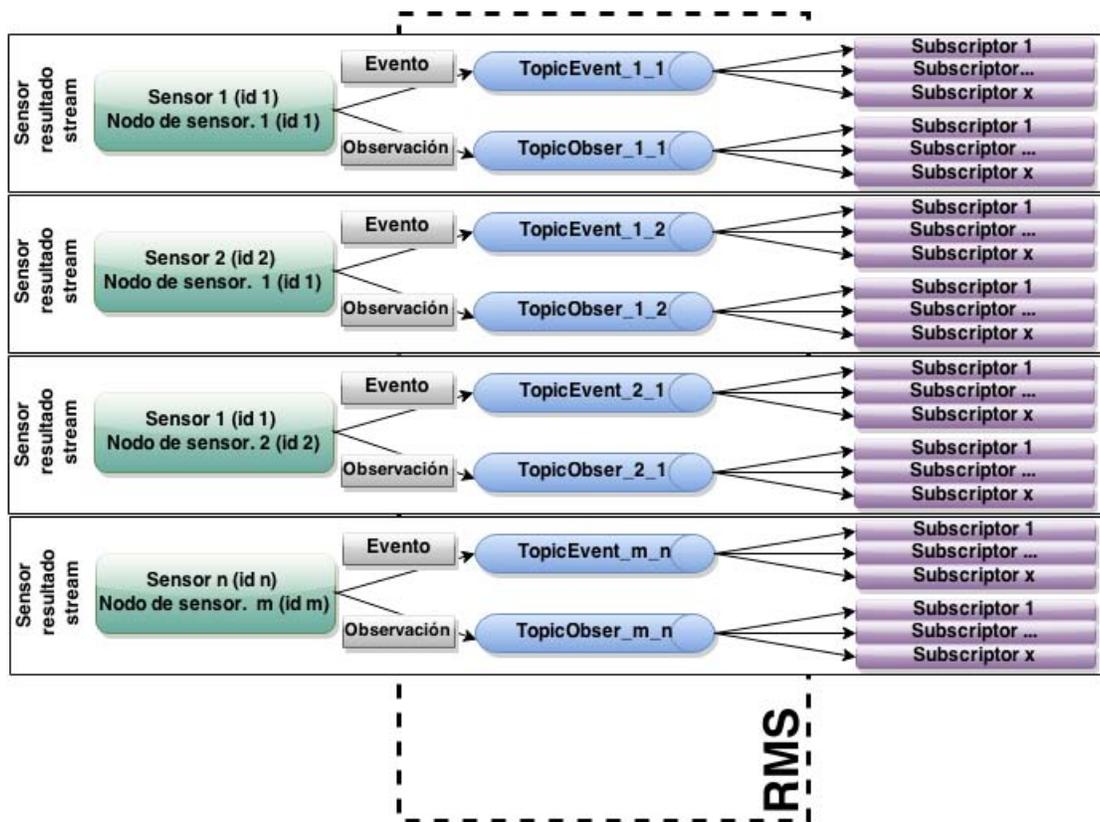


Figura 6.18: La figura muestra cómo se utiliza RMS para las salidas, tanto para eventos como para observaciones.

identificador de la observación en particular que ha provocado el evento (elemento *Identifier*). El elemento *msgType* indica que el evento identificado se refiere a que si la anomalía ha sido por superar el umbral o por descenso.

Un segundo *Bolt* se utiliza para proporcionar acceso a las últimas observaciones analizadas. Este *Bolt* es necesario para ofrecer las observaciones analizadas de una forma homogénea y que además, puedan ser consumidas por las aplicaciones finales.

Los dos Bolts utilizan el cliente RMS como *Productores* para conectar con el RMS. También, las funcionalidades de ambos se realizan con el modelo *Publicación/Subscripción*, para que los diferentes *Consumidores* (aplicaciones finales) se puedan conectar a las *Colas*.

<i>Elemento</i>	<i>Significado</i>
Alert	Es el contenedor para todas los elementos del formato
Identifier	El identificador del mensaje de alerta
Sender	El identificador del remitente del mensaje de alerta (identificador del sensor)
Sent	La hora y la fecha de origen del mensaje de alerta
Status	El elemento que denota el manejo apropiado del mensaje de alerta
MsgType	El elemento que denota la naturaleza del mensaje de alerta
Scope	El elemento que denota el alcance previsto del mensaje de alerta

Tabla 6.7: Descripción de todos los elementos utilizados en el CAP.

Panel de control para la visualización

Se ofrece un panel de control mediante un cliente Web para visualizar los datos proporcionados por la capa de servicios. Dicho cliente, visualizará la información obtenida a partir de los resultados del análisis anterior mostrando las anomalías detectadas en un contexto espacio-temporal.

El cliente muestra todos los nodos de sensorización de la red en un mapa usando marcadores (Figura 6.19 a)). Dentro de los marcadores aparece la cantidad de eventos que se han detectado para este nodo en particular. Si en este nodo se produce un evento, el marcador se pinta en rojo y en el otro caso, el marcador será azul. El panel de control no diferencia si el evento es por exceso o por reducción. Se aplica una agrupación en los marcadores cuando se realiza zoom. Los marcadores resultantes de cada agrupación contendrán el número total de eventos obtenidos en cada uno de los marcadores que lo forman (Figura 6.19 b)). El color del marcador resultante de la agrupación será de color rojo si algún sensor que lo compone identificó algún evento.

Si un usuario selecciona un marcador simple (sin aplicar clustering), nuevos marcadores aparecerán como un menú flotante (Figura 6.20 a)). Cada nuevo marcador simboliza un sensor (fenómeno) que esté asociado al nodo seleccionado, donde si se ha notificado un nuevo evento, el marcador aparecerá en rojo. Cuando se haga clic en uno de los marcadores de los fenómenos, aparecerá un gráfico como ventana emergente. Dicho gráfico muestra las últimas observaciones (Figura 6.20 b)). Las observaciones en el gráfico se obtienen de la capa de servicios.

El gráfico se actualiza dinámicamente con las observaciones más recientes.

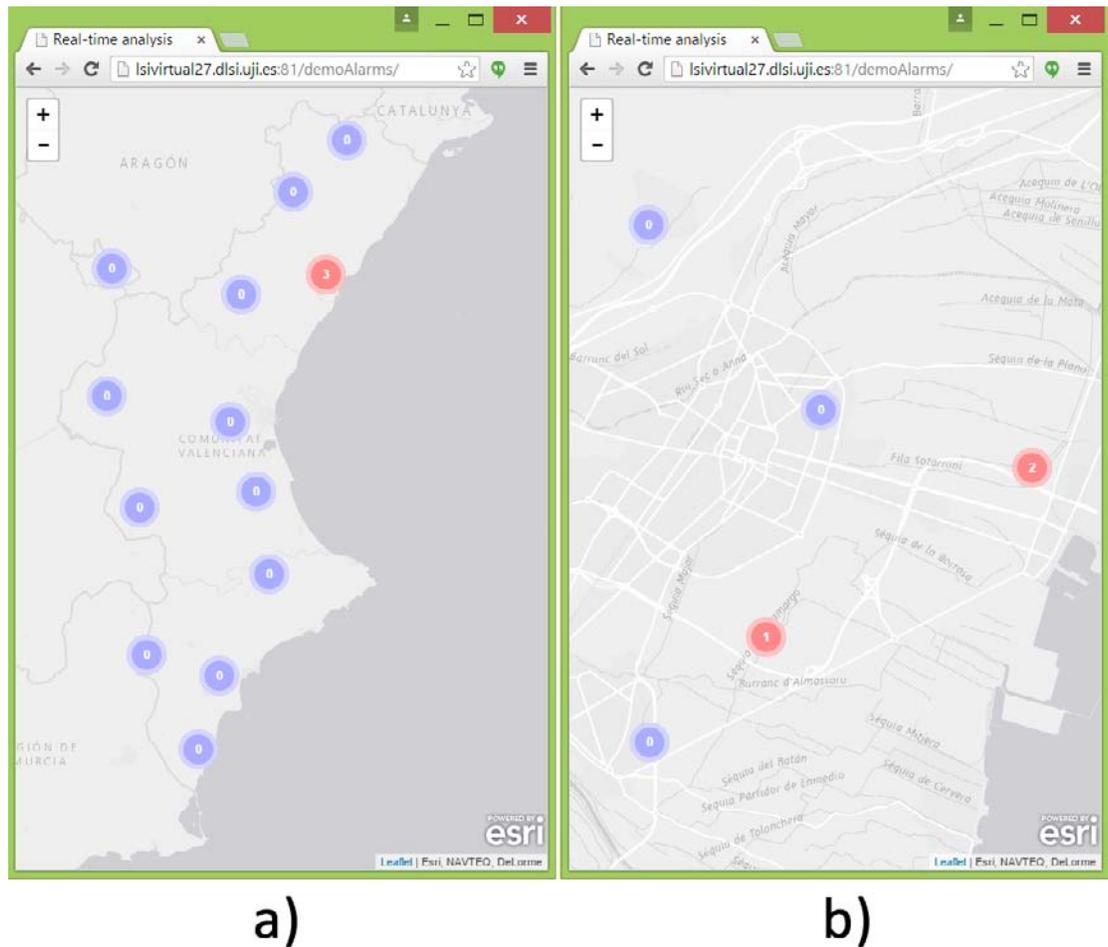
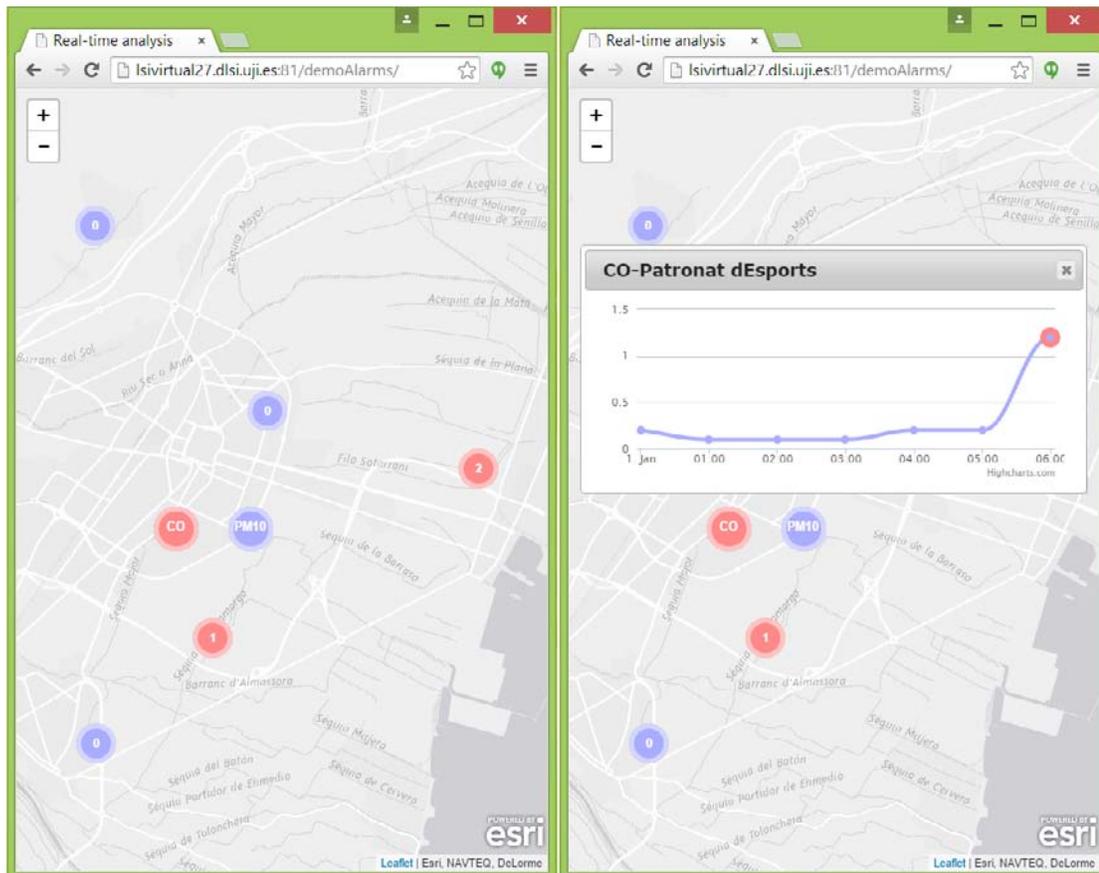


Figura 6.19: Diferentes capturas del cliente. a) Muestra los nodos de sensores (o agrupación) como marcadores, indicando los eventos en el marcador. b) Muestra cómo se aplica la agrupación teniendo en cuenta el número de eventos.

Dentro de la gráfica, los eventos se destacan con un rombo rojo. El gráfico se actualiza cuando se produce una nueva observación. El usuario puede ver gráficos distintos simultáneamente, incluso diferentes fenómenos de distintos nodos. De esta manera, se pueden comparar los valores de un mismo fenómeno dentro de la misma red de sensores. Para la creación del panel de control se utilizan las mismas herramientas que en el cliente, detallado en la Sección 6.1.4. De esta forma se ha reutilizado parte del trabajo hecho.



a)

b)

Figura 6.20: Diferentes capturas del cliente. a) Muestra un menú para visualizar los fenómenos que componen cada nodo. b) Un gráfico para visualizar las últimas observaciones y los eventos lanzados

6.5. Predicción del riesgo de incendios

En esta sección se detalla un ejemplo de implementación siguiendo la metodología definida para el análisis mediante geoprocesamientos estándares (Sección 5.4). Para ello, se presenta un caso de uso para la predicción de riesgo de incendios, donde se utilizan unos algoritmos matemáticos llamados procesos puntuales espaciales. A continuación se expone una introducción al problema.

Hoy en día, los incendios forestales son uno de los principales problemas para la conservación de los bosques mediterráneos y para la prevención de la

desertificación en estos (Palahi y otros, 2008; Shakesby, 2011). La destrucción de los bosques produce un impacto perjudicial, por ejemplo, en años recientes en algunas áreas boscosas de España, se han perdido grandes cantidades de hectáreas debido a los incendios forestales (incendios forestales en España en 2005: 25.492 incendios y 18.8697 hectáreas quemadas) (de Medio Ambiente, 2006). Si se analizan las áreas de bosques quemados en España, podemos tener una visión general de los incendios forestales (de Medio Ambiente, 2006) y, cómo los incendios destruyen los bosques y su biodiversidad (Bengtsson y otros, 2000), además de la pérdida de la conservación del suelo (Palahi y otros, 2008; Shakesby, 2011) y peor aún la pérdida de la vida. Esta situación conduce a los bosques a una evolución de la zona quemada, iniciando un proceso de pérdida de suelo y de desertificación.

La reducción de los incendios forestales requiere un plan de gestión forestal para conservar los bosques, reducir el impacto de los incendios y facilitar su extinción (Velez y otros, 2000). La utilización de modelos de incendios forestales son un punto de partida para el diseño de estas acciones preventivas (Gonzalez-Olabarria y Pukkala, 2011; Juan y otros, 2012), seguido de los medios para compartir este tipo de análisis con el fin de comparar los resultados y tomar acciones (Granell y otros, 2010).

La preservación de las áreas boscosas es muy importante en el Mediterráneo, ya que son una fuente importante de recursos de conservación de la biodiversidad y del suelo para el correcto funcionamiento del ciclo de vida que afecta directamente a la economía global (Palahi y otros, 2008). Los incendios naturales son causados por factores tales como las tormentas y las estaciones secas, entre otros, pero estos no afectan el equilibrio ecológico del planeta ya que son parte del ciclo de vida natural de un bosque y provocan la regeneración total o parcial de especies (Palahi y otros, 2008; Velez y otros, 2000). Sin embargo, los incendios naturales son sólo una pequeña porción de todos los incendios que se producen en las regiones mediterráneas. Los incendios forestales en su mayoría son causados por factores humanos (de Medio Ambiente, 2006; Bianchini, 2006), estos pueden reducirse mediante los mencionados modelos de incendios y teniendo conciencia del riesgo de incendio existente en un área determinada.

Actualmente, se encuentran diferentes trabajos de investigación dedicados al estudio de los incendios forestales, ya que es un punto de interés muy candente en la comunidad científica y tiene un impacto directo en los intereses económicos

y sociales de las regiones afectadas. De forma breve, estos trabajos pueden dividirse en dos ramas diferentes (Bianchini, 2006):

- *Simulación de fuego.* Utilizado para predecir la evolución de un incendio en tiempo real. Éste está basado en las condiciones físicas, en factores meteorológicos y en factores humanos. Esto ayuda a mejorar la eficacia y reducir el riesgo de acciones de extinción de incendios forestales.
- *Análisis de riesgo de incendio.* Proporciona mapas de análisis de riesgos para cada región. Éste también se basa en las condiciones físicas, factores meteorológicos y humanos. Este enfoque es más útil para acciones de prevención de incendios forestales. El caso de uso que se presenta sigue esta línea de investigación, donde se propone un método novedoso para el análisis de riesgo de incendios forestales y para ser accesible de una manera interoperable.

En la literatura se encuentran diferentes trabajos relacionados. Por lo referente a los modelos que simulan el fuego, existen modelos de este tipo que intentan predecir el comportamiento del fuego considerándolo continuo y uniforme (Rothermel, 1983; Velez y otros, 2000), teniendo en cuenta algunos parámetros como la pendiente de terreno, la dirección y fuerza del viento o la combustibilidad de la zona del incendio (Molina-Terrén y otros, 2006; Arca y otros, 2007). Este enfoque requiere un punto de ignición, además de que la bondad de la salida depende de la resolución de la entrada de datos tanto en espacio como en tiempo. Sin embargo se requieren muchos puntos de incendios para cubrir un área, convirtiéndose en una tarea costosa (Gonzalez-Olabarria y Pukkala, 2011). Este enfoque tiene en cuenta las condiciones climáticas y la combustibilidad del suelo (Chuvieco y Salas, 1996). Este tipo de modelos se pueden encontrar por ejemplo en el sistema europeo de información de incendios forestales (San-Miguel-Ayanz y otros, 2003). La disponibilidad de bases de datos históricas de incendios hace posible estimar y utilizar modelos para generar mapas sobre los riesgos de incendios (González y otros, 2006). El análisis estadístico de la predicción del fuego es útil en la prevención de incendios forestales, para planificar acciones preventivas y reducir al mínimo las consecuencias de los incendios forestales (Bianchini, 2006).

Muchos estudios han trabajado en la predicción del riesgo de incendios, por ejemplo (Gregori y A., 2006). Algunos de los parámetros de entrada en estos

modelos son: la pendiente del terreno, densidad de árboles, composición por especies y uso del suelo. Estas variables tienen una relación directa con los incendios forestales (Van Wagner, 1977; González y Pukkala, 2007). Otro factor son las variables meteorológicas tales como la temperatura y la velocidad del viento, ya que parecen influir directamente en el riesgo de incendios forestales y en su comportamiento (Vázquez, 2001). El uso de las variables de temperatura o humedad propone un nuevo campo a explorar en el futuro en los estudios de procesos puntuales.

La estacionalidad del fuego está determinada por dos factores. Uno de ellos son las causas naturales (Teensma, 1987) y el otro son las causas humanas (Lloret y otros, 2003). En el caso de causas naturales, el riesgo de incendios forestales depende de tres factores tales como:

- *Condiciones climáticas* (Sakr y otros, 2011; Andrews, 1986).
- *Tipo y estructura de la vegetación* (Mermoz y otros, 2005) que en algunos casos favorece y en otros casos disminuye su progresión (Nunes y otros, 2005).
- *Condiciones socioeconómicas* (Chas, 2007; Marey y otros, 2006).

Serra y otros (2013) define modelos ambientales para la predicción del riesgo de incendios. En este trabajo se presenta un modelo que realiza un análisis del riesgo de incendios teniendo en cuenta el tipo y la estructura del terreno, además de otros factores tales como la pendiente, la orientación, el uso del suelo y la elevación, que también son utilizados por el modelo como covariables separadas o todas juntas. Se puede trabajar con datos diarios, mensuales o anuales.

6.5.1. Materiales: datos para la predicción de riesgo de incendios forestales

Para la ejecución de los modelos se han definido siete parámetros de entrada, que incluyen: el área (polígono a analizar), localización de los incendios, pendiente, uso del suelo, orientación, elevación y el número de modelo a ejecutar (de los disponibles).

Los datos históricos sobre los incendios forestales han sido recogidos por la oficina de prevención de incendios forestales de la Generalitat Valenciana, España.

Los agentes ambientales han realizado una tarea de recolección de todos los incendios, registrando su situación, el perímetro y las condiciones meteorológicas. Se utilizan los centroides de los incendios en vez del perímetro.

Algunos de los parámetros utilizados son obtenidos mediante la capa vectorial Corine Land Cover (CLC) 2006 (Heymann y otros, 1994) del Instituto Geográfico Nacional (IGN). Este modelo digital del terreno tiene una resolución espacial de 200 metros y se ha utilizado para obtener la elevación, pendiente y orientación del terreno.

El área está representada como un conjunto de puntos que definen la superficie al ser analizada. Los parámetros de los incendios forestales se definen para cada punto en el mapa.

Respecto a las variables de uso del suelo, se han empleado los datos del programa de coordinación de la información sobre el medio ambiente (CORINE, en sus siglas en inglés). El programa CORINE se inició en 1985 por la Comisión Europea y fue adoptado por la Agencia Europea de Medio Ambiente (AEMA) en 1994 (Agency, 2007; Heymann y otros, 1994). Se ha optado por estos datos por poder usar la misma fuente de datos para toda la Unión Europea en caso de querer replicar los resultados en otras localizaciones a la presentada.

Se han reclasificado los usos del suelo en diez categorías: coníferas, bosques densos, pastos, frutales, zonas artificiales no agrícolas, zonas agrícolas, zonas forestales de transición, matorrales, zonas naturales de pastos, bosques mixtos, zonas urbanas, playas, arenales, rocas, zonas quemadas y aguas continentales.

Para poder modelar la dependencia de los modelos puntuales con las covariables espaciales se requieren varios requisitos. Primero, las covariables deben ser cuantitativas y observables en cada localización de la zona a estudiar (por ejemplo: pendiente, orientación y sombreado). Estas covariables pueden ser valores continuos o categorías (como por ejemplo el uso del suelo). Segundo, los valores de las covariables deben estar disponibles para cada punto del mapa. Y tercero, los valores en otros puntos de la zona a estudiar tienen que estar disponibles.

6.5.2. Modelos para la predicción de riesgo de incendios forestales

Como ya se ha comentado, como caso de uso a la metodología definida en la Sección 5.4, se presenta un análisis para la predicción de riesgo de incendio, en

el cual se utilizan procesos puntuales espaciales. La sección tiene como objetivo realizar una introducción a los procesos puntuales. En ella se definen cada uno de los tipos de modelos utilizados para la predicción del riesgo de incendio.

Procesos puntuales espaciales

El objetivo de estos procesos es utilizar la metodología estadística desarrollada en procesos espaciales para analizar y modelar el comportamiento de cualquier incendio. Si a cada incendio se le asocian sus coordenadas espaciales de longitud y latitud del centroide de la zona quemada, junto con otras variables como la pendiente, contorno de la zona a analizar, la orientación, la elevación o usos del suelo, controlando la temporalidad de los incendios, se puede identificar el conjunto de incendios forestales mediante un proceso estocástico espacio-temporal. De este modo, teniendo como parámetros de entrada el área y los incendios declarados, se asegura que el modelo sea multi-escala, siendo útil para predecir el riesgo en cualquier escala de resolución.

Los procesos puntuales espaciales se definen como un modelo estocástico que genera un número infinito de eventos $x_i, i = 1 \dots n$ en un conjunto X y un espacio D . Si estos eventos tienen asociadas medidas o marcas, tenemos un proceso puntual marcado. A continuación se definen desde el más simple, como los *Poisson*, al más complejo, como *Area-Interaction*.

Aleatoriedad espacial completa La aleatoriedad espacial completa (CSR, en sus siglas en inglés), es el “ruido blanco” generado por un proceso puntual espacial, pues caracteriza la ausencia de estructura de los datos. Por tanto, habitualmente es la hipótesis nula de un test estadístico para determinar cuándo hay estructura espacial en un diseño puntual dado. Un proceso completamente aleatorio se identifica con un proceso homogéneo de *Poisson* y está definido por los siguientes postulados:

1. El número de eventos en una región plana A , con área $|A|$ sigue una distribución de *Poisson* con media $\lambda |A|$, donde λ es la intensidad del proceso, es decir, el número esperado de eventos por unidad de área.
2. Dados n eventos $\{x_i\}_{i=1}^n$ en la región A , los x_i son una muestra aleatoria de la distribución uniforme en A , por tanto, no hay interacción entre eventos.

Por el segundo postulado, la densidad condicional de una n -upla ordenada (x_1, x_2, \dots, x_n) dado $N(A) = n$, es:

$$f(x_1, x_2, \dots, x_n) = 1/|A|^n \quad (6.4)$$

Así, teniendo en cuenta ambos postulados, la densidad conjunta de n y (x_1, x_2, \dots, x_n) , viene dada por:

$$f((s_1, s_2, \dots, s_n), n) = \lambda^n \ln^{-\lambda|A|} / n! \quad (6.5)$$

El principal interés de la CSR es que representa un modelo ideal que, a pesar de no alcanzarse en la práctica, resulta muy útil como primera aproximación de algunos patrones observados. En general, el análisis de un diseño puntual observado se inicia con un test de CSR. Esto ocurre por dos razones fundamentales. En primer lugar, un diseño que verifica CSR, apenas necesita más análisis estadístico formal. Y en segundo lugar, además del interés intrínseco que tiene rechazar el CSR, los tests se utilizan como método de exploración de los datos para conocer si se está desviando de la hipótesis nula.

Propiedades de los procesos puntuales Considerando $N(A)$ la variable aleatoria con la que se representa el número de eventos en una región plana A .

$$N(A) = \#(x_i \in A) \quad (6.6)$$

Algunas propiedades de interés con las que se pueden verificar los procesos puntuales espaciales son las siguientes:

- El proceso es estacionario, si para cualquier entero k y regiones $A_i, 1, \dots, k$ la distribución conjunta de $N(A_1), N(A_2), \dots, N(A_k)$ es invariante por translaciones de los A_i por cualquier conjunto x .
- El proceso es isotrópico, si para cualquier entero k y regiones $A_i, 1, \dots, k$, la distribución conjunta de $N(A_1), N(A_2), \dots, N(A_k)$ es invariante por rotación de la unión de los A_i .
- El proceso es ordenable cuando no hay eventos coincidentes, es decir:

$$\lim_{dx \rightarrow 0} \frac{P(N(dx) > 1)}{|dx|} \quad (6.7)$$

- El proceso es ordenable de segundo orden cuando para cualquier par de eventos x y y se verifica:

$$\lim_{|dx|, |dy| \rightarrow \{0,0\}} \frac{P(N(dx) > 1)P(N(dy) > 1)}{|dx| |dy|} \quad (6.8)$$

Características de primer y segundo orden en procesos univariantes A continuación se definen en procesos puntuales, estructuras análogas a la media y la covarianza para procesos con valores reales.

Las características de primer y segundo orden de la variable aleatoria $N(B)$, para $B \subseteq S$, están descritas por la medida de intensidad y el momento factorial de orden dos, definidos como:

Definición

La medida de la intensidad μ en R^d viene dada por:

$$\mu(B) = E[N(B)], B \subseteq R^d \quad (6.9)$$

El momento factorial de orden 2, $\alpha^{(2)}$ en $R^d \times R^d$ viene dado por:

$$\alpha^{(2)}(C) = E \left[\sum_{\xi \neq \eta \in X} I((\xi, \eta) \in C) \right], C \subseteq R^d \times R^d \quad (6.10)$$

Si la medida de intensidad μ se puede escribir como:

$$\mu(B) = \int_B \lambda(\xi) d\xi \quad (6.11)$$

Donde λ es una función no negativa, entonces λ será la función de intensidad. Sea dx un entorno infinitesimal del punto x .

Definición

La función de intensidad de primer orden de un proceso puntual espacial es:

$$\lambda(x) = \lim_{\|dx\| \rightarrow 0} \left\{ \frac{E[N(dx)]}{|dx|} \right\} \quad (6.12)$$

Definición

La función de intensidad de segundo orden de un proceso puntual espacial es:

$$\lambda_2(x, y) = \lim_{|dx|, |dy| \rightarrow 0} \left\{ \frac{E[N(dx)N(dy)]}{|dx| |dy|} \right\} \quad (6.13)$$

Definición

La función de covarianza de un proceso puntual espacial es:

$$\gamma(x, y) = \lambda_2(x, y) - \lambda(x)\lambda(y) \quad (6.14)$$

Para un proceso puntual estacionario e isotrópico, se obtiene:

1. $\lambda(x) = \lambda = E[N(A)]/|A|$ (*constante para toda A*)
2. $\lambda_2(x, y) = \lambda_2(\|x - y\|)$
3. $\gamma(u) = \lambda_2(u) - \lambda^2$

Definición

La intensidad condicional de un proceso puntual espacial es:

$$\lambda_c(x/y) = \lim_{|dx| \rightarrow 0} \left\{ \frac{E[N(dx)/\text{evento en } y]}{|dx|} \right\} \quad (6.15)$$

Lema

Para un proceso estacionario isotrópico y ordenable, se verifica:

$$\lambda_c(x/y) = \lambda_2(\|x - y\|)/\lambda \quad (6.16)$$

Baddeley (2008) estudiaron los procesos estacionarios reponderados de orden 2, que verifican $\lambda_2(x/y)/\lambda(x)\lambda(y) = \rho(t)$, dependiendo únicamente de $t = \|x - y\|$, donde $\lambda(\cdot)$ no puede tomar valores muy próximos a 0. La estacionalidad reponderada en procesos puntuales es análoga suponiendo que en un proceso espacial con valores reales, la media varía en el espacio mientras que las variaciones lo hacen entorno a la media local donde son estacionarias.

A partir de ahora, salvo indicación específica, se asume estacionalidad, isotropía y ordenabilidad. Bajo estas condiciones, λ tiene fácil interpretación como por ejemplo el número esperado de eventos por unidad de área. La intensidad de segundo orden es más difícil de interpretar. Consideramos una nueva definición:

Definición

La función de segundo momento reducida de un proceso puntual estacionario e isotrópico viene dada por:

$$K(t) = \lambda^{-1} E[N_0(t)] \quad (6.17)$$

Donde $N_0(t)$ denota el número de eventos situados a una distancia menor que t de un evento arbitrario. Se entiende por arbitrario un evento elegido al azar.

Procesos Poisson

Este tipo de procesos en su estado básico son detallados en Baddeley y Turner (2005). En los procesos puntuales espaciales se necesita alguna notación adicional, donde dx denota una pequeña región que contiene un punto x . La función de intensidad (de primer orden) de un proceso puntual espacial viene dada por:

$$\lambda(x) = \lim_{|dx| \rightarrow 0} \left\{ \frac{E[N(dx)]}{|dx|} \right\}, \quad (6.18)$$

Donde E es definida por la Ecuación 6.21 y la función de intensidad de segundo orden tiene la forma:

$$\lambda_2(x, y) = \lim_{\substack{|dx| \rightarrow 0 \\ |dy| \rightarrow 0}} \left\{ \frac{E[N(dx)N(dy)]}{|dx||dy|} \right\}. \quad (6.19)$$

Las propiedades de segundo orden proporcionan información sobre la interacción entre puntos en un patrón de puntos espaciales y establece un medio para probar la hipótesis nula sobre la CSR en la distribución. Son conocidos para proporcionar de la mejor forma posible la representación estadística de la información distributiva, inherente en los patrones de puntos y en la explicación de la correlación entre ellos (Illian y otros, 2008). Por tanto, basándose en esta premisa se puede decidir si un patrón de un punto espacial sigue de una forma agrupada o dispersa un patrón de distribución o si no la sigue como tal.

El proceso es estacionario e isotrópico si sus propiedades estadísticas no cambian bajo traslación y rotación, respectivamente. Si se supone que el proceso es estacionario e isotrópico, la función de intensidad se reduce a una constante λ , igual que el número esperado de eventos por unidad de área. Además, la intensidad del segundo momento se reduce a una función de distancia $\lambda_2(x, y) = \lambda_2(r)$, donde $r = \|x - y\|$ es la distancia entre x y y . En este contexto, la función de un proceso puntual espacial estacionario e isotrópico viene dada por:

$$K(r) = 2\pi\lambda^{-2} \int_0^r \lambda_2(s) s ds. \quad (6.20)$$

La importancia de la Ecuación 6.20 es que puede expresarse como una expectativa de una cantidad observable, la cual sugiere cómo se pueden estimar los datos. De hecho,

$$\lambda K(r) = E(\text{Núm. de eventos dentro de una distancia } r \text{ de un evento arbitrario}) \quad (6.21)$$

Donde $E(\cdot)$ es el operador de expectativa. Para un modelo completamente al azar, los eventos se colocan independientemente, por tanto, $K(r) = \pi r^2$. Esto le da un punto de referencia contra el que evaluar las salidas de CSR. Para los patrones con agregación, $K(r)$ es relativamente grande en distancias pequeñas, ya que cada evento normalmente forma parte de un grupo de eventos mutuamente cercanos. Contrariamente, para los patrones regulares $K(r)$ es relativamente pequeño en distancias próximas, ya que cada evento tiende a estar rodeado de espacio vacío.

Otra característica útil es que $K(r)$ es aleatoriamente invariable, es decir, significa que la retención o supresión de eventos de acuerdo a una serie de ensayos de Bernoulli independientes (Baddeley, 2008).

El proceso de *Poisson* homogéneo es el proceso puntual más simple que no representa ningún proceso subyacente, lo que resulta en la aleatoriedad completa en la distribución de incendios forestales. La función K propuesta por Ripley (1976) de un proceso *Poisson* homogéneo se define como:

$$K(r) = \frac{1}{A} \sum_{i=1} \sum_{j \neq 1} \frac{\omega_{ij} I(d_{ij} \leq r)}{\lambda^2} \quad (6.22)$$

En esta función, A indica el área de la parcela; λ es la densidad de un incendio; ω_{ij} es un término de corrección de borde; d_{ij} representa la distancia entre dos puntos; y I es una función de índice donde $I = 1$, si $d_{ij} \leq r$, y $I = 0$ (Ripley, 1976). La densidad de incendio λ , es el parámetro a estimar en este modelo.

El proceso de *Poisson* no homogéneo puede utilizarse para modelar la asociación heterogénea en incendios forestales. En este modelo, las relaciones entre la densidad y la heterogeneidad se incluyen mediante una función de intensidad

espacial heterogénea, $\lambda(s)$ (Illian y otros, 2008). La función K del proceso de *Poisson* no homogéneo se define como:

$$K_{inh}(r) = \frac{1}{A} \sum_{i=1} \sum_{j \neq 1} \frac{\omega_{ij} I(d_{ij} \leq r)}{\lambda(s_i) \lambda(s_j)} \quad (6.23)$$

Donde A , λ , ω_{ij} , d_{ij} y I son las mismas que en el proceso de *Poisson* homogéneo; y $\lambda(s_i)$ y $\lambda(s_j)$ son los valores de la función de la intensidad en los puntos s_i y s_j , respectivamente.

En concreto, la función de intensidad, $\lambda(s)$, se modela como una regresión polinómica logarítmica:

$$\lambda(s) = \exp(\beta^T X(s)) \quad (6.24)$$

Donde $X(s)$ es un vector de variables y β^T es un vector de parámetros de regresión. En este estudio se utilizaron dos tipos diferentes de regresión polinómica logarítmica: la regresión lineal logarítmica con covariables y la regresión cuadrática con las coordenadas de los incendios forestales.

Procesos Area-Interaction

El interés en modelar patrones agregados se debe a que estos se centran en las interacciones de orden superior, donde un modelo útil es el denominado *Area-Interaction*. Los procesos *Area-Interaction* (Widom, 1970; Baddeley y Van Lieshout, 1995) con radio r , parámetro de intensidad κ y parámetro de interacción γ son un proceso puntual con densidad de probabilidad:

$$f(s_1, \dots, s_n) = \alpha \kappa^{n(s)} \gamma^{(-A(s))} \quad (6.25)$$

donde s_1, \dots, s_n representan los puntos del patrón, $n(s)$ es el número de puntos en el patrón y $A(s)$ es el área de la región formada por la unión de los discos de radio r centrada en los puntos s_1, \dots, s_n . α es una constante de normalización.

El parámetro interacción γ puede ser cualquier número positivo. Si $\gamma = 1$ entonces el modelo se reduce a un proceso de *Poisson* con intensidad κ . Si $\gamma < 1$ entonces el proceso es regular, mientras que si $\gamma > 1$ el proceso está agrupado. Por tanto, un proceso de *Area-Interaction* puede utilizarse para modelar cualquier agrupación o puntos que siguen un patrón. Dos puntos interaccionan si la distancia entre ellos es menor que $2 * r$.

Aquí, se ha parametrizado el modelo de una forma diferente a Baddeley y Turner (2005), que es más fácil de interpretar. En forma canónica de libre de escala, la densidad de probabilidad se reescribe como:

$$f(s_1 \dots s_n) = \alpha \beta^{n(s)} \eta^{-C(s)} \quad (6.26)$$

Donde β es el nuevo parámetro de intensidad, η es el nuevo parámetro de interacción y $C(s) = B(s) - n(s)$ es la interacción potencial.

Aquí,

$$B(s) = A(s) / (\pi \cdot r^2) \quad (6.27)$$

es el área normalizada (de modo que los discos tienen la unidad del área). En esta fórmula, cada punto aislado del patrón contribuye con un factor β a la densidad de probabilidad (así que la primera tendencia de orden es β). La cantidad $C(s)$ es una verdadera interacción potencial, en el sentido que $C(s) = 0$ si el punto del patrón x no contiene todos los puntos continuos que se encuentren cerca (más próximos que $2r$).

Los parámetros anteriores κ y γ de la forma estándar están relacionados con los nuevos parámetros β y η , de forma canónica de libre de escala, por:

$$\beta = \kappa \gamma^{(-\pi \cdot r^2)} = \kappa / \eta \quad (6.28)$$

y

$$\eta = \gamma^{(-\pi \cdot r^2)} \quad (6.29)$$

siendo κ y γ positivos y finitos (Baddeley y Turner, 2005).

En la forma canónica de libre de escala, el parámetro η puede tomar cualquier valor no negativo. El nuevo valor $\eta = 1$ corresponde a un proceso de *Poisson*, con intensidad β . Si $\eta < 1$, siendo así en un el proceso regular, mientras que si $\eta > 1$ el proceso está agrupado. El valor $\eta = 0$ corresponde a un núcleo duro con un radio r (distancia de interacción $2r$).

El proceso de *Area-Interaction* no homogéneo es similar, excepto que la contribución de cada punto individual s_i es una función $\beta(s_i)$ de ubicación, en lugar de una beta constante.

6.5.3. Modelos definidos

Con los dos modelos detallados en las secciones anteriores, se han definido diferentes tipos de estos para la modelización de los riesgos de incendio. En cada caso, se han introducido diversas covariables tales como elevación, pendiente, usos del suelo y su unión. En total se han definido nueve modelos (Tabla 6.8). Para la ejecución de dichos modelos se ha utilizado el software libre R ²⁸. Éste es un lenguaje y entorno para el análisis estadístico y gráfico. Tiene un lenguaje propio que se puede ejecutar tanto en modo consola proporcionada por el entorno de R , como mediante scripts.

<i>Número de modelo</i>	<i>Tipo de modelo</i>	<i>Covariables</i>
1	Poisson	Con polinomio X Y
2	Poisson	Con polinomio X Y 2
3	Poisson	Con la covariable elevación
4	Poisson	Con la covariable usos del suelo
5	Poisson	Con todas las covariables
6	Area-Interaction	Con polinomio X Y
7	Area-Interaction	Con la covariable elevación
8	Area-Interaction	Con la covariable usos del suelo
9	Area-Interaction	Con todas las covariables

Tabla 6.8: Correspondencia de números a métodos.

6.5.4. Ejecución de los modelos

Como se define en la Sección 5.4 se ha utilizado un geoprocamiento siguiendo la especificación de OGC WPS (Schut, 2008). Se ha elegido la implementación de 52North²⁹ para dicha especificación. Un WPS es la infraestructura para poder ejecutar cualquier tipo de algoritmo. En este caso, se ha aplicado un algoritmo para la predicción de riesgo de incendios, el cual debe ejecutarse en un entorno controlado pero con la encapsulación en un WPS, pasando a poder ser ejecutado de forma distribuida.

²⁸<http://www.r-project.org>

²⁹<http://52north.org/>

Uno de los principales objetivos de este trabajo es facilitar la reutilización de los modelos científicos definidos. Estos algoritmos, que se ejecutan localmente en un determinado espacio de trabajo, son inaccesibles para un usuario no especializado. Durante esta investigación, se han envuelto y ofrecido como un servicio Web basado en estándar, con el objetivo de facilitar su accesibilidad y reutilización.

El primer paso, en un WPS es la definición de las entradas que va a recibir cuando se invoque a través de la petición *Execute*. Las entradas del sistema están definidas en un fichero XML. Además de definir las entradas, también se muestran las salidas.

Para las entradas del modelo se ha buscado un formato geográfico que sea capaz de almacenar cómodamente la información y que además, se pueda trabajar fácilmente con él. Para ello, se ha utilizado el formato shapefile³⁰. De esta forma cada una de las entradas será codificada mediante este tipo de formato.

Son siete las entradas que se han definido para el sistema (Figura 6.21). Seis de ellas describen la información geoespacial que va a necesitar el sistema: contorno, centroides de los incendios, pendiente, orientación, elevación y uso del suelo. La entrada restante es el número del modelo a ejecutar. Ésta consiste en una cadena con el número del algoritmo a ejecutar, pudiendo concatenar algoritmos con espacios. La definición de las entradas quedaría como se muestra en el Listado 6.5.

Listado 6.5: Definición de las entradas.

```
<DataInputs>
  <Input minOccurs="1" maxOccurs="1">
    <ows:Identifier>Contorn</ows:Identifier>
    <ows:Title>Contorn</ows:Title>
    <ows:Abstract>Contorn</ows:Abstract>
    <ComplexData>
      <Default>
        <Format>
          <MimeType>application/x-zipped-shp</MimeType>
          <Schema/>
        </Format>
      </Default>
      <Supported>
        <Format>
          <MimeType>text/plain</MimeType>
          <Schema/>
        </Format>
      </Supported>
    </ComplexData>
  </Input>
</DataInputs>
```

³⁰<http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>

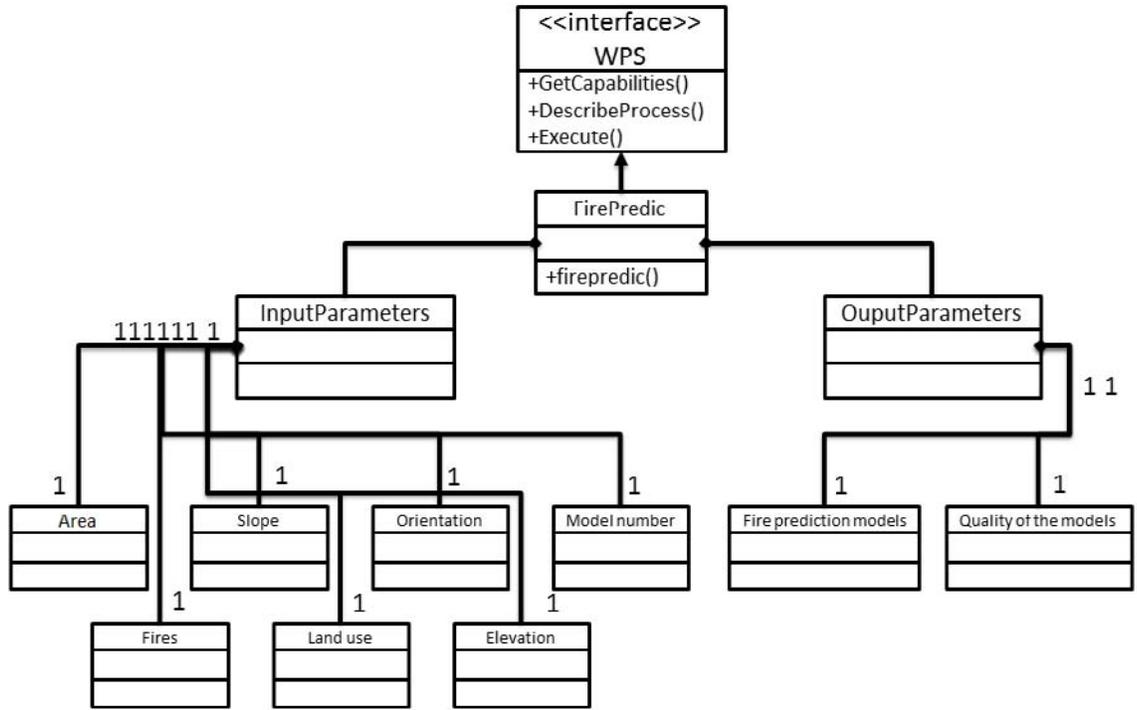


Figura 6.21: Parámetros de entrada y salida de los modelos.

```

</Supported>
</ComplexData>
</Input>
<Input minOccurs="1" maxOccurs="1">
  <ows:Identifier>IncendCentroid</ows:Identifier>
  <ows:Title>Incend Centroid</ows:Title>
  <ows:Abstract>Incend Centroid</ows:Abstract>
  <ComplexData>
    <Default>
      <Format>
        <MimeType>application/x-zipped-shp</MimeType>
        <Schema/>
      </Format>
    </Default>
    <Supported>
      <Format>
        <MimeType>text/plain</MimeType>
        <Schema/>
      </Format>
    </Supported>
  </ComplexData>
</Input>

```

```

</ComplexData>
</Input>
<Input minOccurs="1" maxOccurs="1">
  <ows:Identifier>Pending</ows:Identifier>
  <ows:Title>Pending</ows:Title>
  <ows:Abstract>Pending</ows:Abstract>
  <ComplexData>
    <Default>
      <Format>
        <MimeType>application/x-zipped-shp</MimeType>
        <Schema/>
      </Format>
    </Default>
    <Supported>
      <Format>
        <MimeType>text/plain</MimeType>
        <Schema/>
      </Format>
    </Supported>
  </ComplexData>
</Input>
<Input minOccurs="1" maxOccurs="1">
  <ows:Identifier>Orientation</ows:Identifier>
  <ows:Title>Orientation</ows:Title>
  <ows:Abstract>Orientation</ows:Abstract>
  <ComplexData>
    <Default>
      <Format>
        <MimeType>application/x-zipped-shp</MimeType>
        <Schema/>
      </Format>
    </Default>
    <Supported>
      <Format>
        <MimeType>text/plain</MimeType>
        <Schema/>
      </Format>
    </Supported>
  </ComplexData>
</Input>
<Input minOccurs="1" maxOccurs="1">
  <ows:Identifier>Use</ows:Identifier>
  <ows:Title>Use</ows:Title>
  <ows:Abstract>Use</ows:Abstract>
  <ComplexData>
    <Default>
      <Format>
        <MimeType>application/x-zipped-shp</MimeType>
        <Schema/>
      </Format>
    </Default>
    <Supported>
      <Format>

```

```

        <MimeType>text/plain</MimeType>
    </Format>
</Supported>
</ComplexData>
</Input>
<Input minOccurs="1" maxOccurs="1">
    <ows:Identifier>Elevation</ows:Identifier>
    <ows:Title>Elevation</ows:Title>
    <ows:Abstract>Elevation</ows:Abstract>
    <ComplexData>
        <Default>
            <Format>
                <MimeType>application/x-zipped-shp</MimeType>
                <Schema/>
            </Format>
        </Default>
        <Supported>
            <Format>
                <MimeType>text/plain</MimeType>
                <Schema/>
            </Format>
        </Supported>
    </ComplexData>
</Input>
<Input minOccurs="1" maxOccurs="1">
    <ows:Identifier>NumAlgo</ows:Identifier>
    <ows:Title></ows:Title>
    <ows:Abstract></ows:Abstract>
    <LiteralData>
        <ows:DataType ows:reference="xs:string"></ows:DataType>
        <ows:AllowedValues>
            <ows:Value></ows:Value>
        </ows:AllowedValues>
    </LiteralData>
</Input>
</DataInputs>

```

Por lo que se refiere a las salidas, el modelo genera un fichero de texto con una malla en la que cada línea contiene un punto con las coordenadas X e Y, además de la predicción de riesgo. La solución adoptada ha sido crear como salida un fichero KML ³¹. Éste contendrá un *Placemark* por cada punto y, dentro del mismo se definirá el punto y un *ExtendedData* con la predicción de los modelos ejecutados. Las salidas se definirían como muestra el Listado 6.6

Listado 6.6: Definición de las salidas.

```

<ProcessOutputs>
    <Output>

```

³¹<https://developers.google.com/kml/documentation/kmlreference>

```

<ows:Identifier>km1</ows:Identifier>
<ows:Title></ows:Title>
<ows:Abstract></ows:Abstract>
<LiteralOutput>
  <ows:DataType ows:reference="xs:string"/>
</LiteralOutput>
</Output>
<Output>
  <ows:Identifier>graphics</ows:Identifier>
  <ows:Title></ows:Title>
  <ows:Abstract></ows:Abstract>
  <LiteralOutput>
    <ows:DataType ows:reference="xs:string"/>
  </LiteralOutput>
</Output>
</ProcessOutputs>

```

A continuación se describen los pasos para la ejecución de los modelos definidos (Figura 6.22). En el primer paso, el usuario invoca el modelo mediante el envío de una petición *Execute* del WPS, donde todos los parámetros son necesarios. Además de verificar que todos los parámetros son correctos.

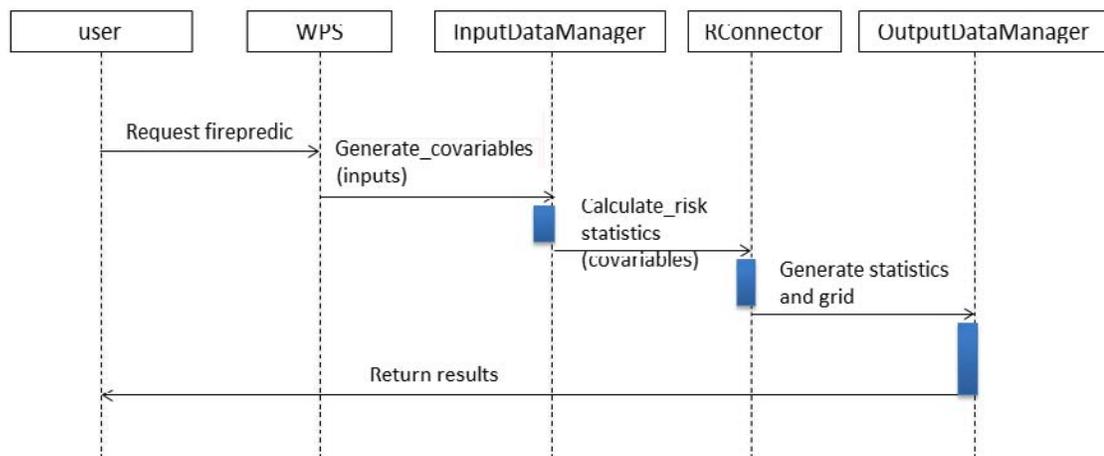


Figura 6.22: Diagrama de secuencia para la ejecución del riesgo de incendio.

El segundo paso lanza una rutina de eliminación de carpetas temporales de otras ejecuciones anteriores. Se ha previsto esta rutina para tener siempre limpio el servidor, aunque se eliminen los ficheros que no son necesarios, las salidas continúan en la carpeta temporal.

El tercer paso, es la descompresión de los shapefiles y su tratamiento para adaptar las entradas al proceso de *R*, siendo éste un proceso algo costoso ya que la información geoespacial suele tener mucho tamaño.

El cuarto paso es llamar al proceso de R para su ejecución. Esto es posible realizarlo en varios sistemas operativos, tanto en Windows como en Unix, siempre que tengan las librerías necesarias. Como ya se ha comentado anteriormente, los modelos de predicción de incendios están implementados en R , más concretamente tienen que ejecutarse en la versión 2.9.2, por lo que se ha previsto de un conector de R , capaz de ejecutar cualquier proceso en dicho lenguaje.

En el quinto paso, una vez terminado el proceso de R , se realiza la lectura de los ficheros de texto generados para cada modelo y se genera el KML para la salida. Además, también se obtiene la cadena con las rutas de las gráficas. Finalmente, se devuelve el flujo al WPS indicándole las salidas para que éste las muestre.

6.5.5. Un ejemplo de ejecución

En esta sección, se muestra la experimentación para probar el trabajo llevado a cabo. Se ha elegido la zona geográfica de Castellón de la Plana (España) como zona de pruebas. Se ha elegido esta área por dos razones. La primera de ellas, se debe a que la provincia de Castellón no es muy grande, adaptándose así a las exigencias del coste computacional y el tiempo. Y, en segundo lugar, porque está disponible la información histórica de incendios declarados.

Para este experimento se han ejecutado los nueve modelos definidos (Tabla 6.8). Una cuestión importante es la distancia entre los puntos en el mapa; se ha definido que cada punto está separado por una distancia de 200 metros con la misma distancia de la trama de resolución del modelo del terreno digital. La Figura 6.23 muestra la representación de los parámetros del sistema.

Los gráficos siguientes (Figura 6.24) son los resultados obtenidos al ejecutar el WPS para los nueve modelos.

En el eje X de las gráficas, el valor de r , muestra la distancia entre puntos o incendios. El eje Y muestra el valor de la Función-L. Se utiliza la Función-L en lugar de la K , para mejorar la visualización de las gráficas del modelo y comprobar la predicción de una forma más fácil. En Juan y otros (2012) se define como:

$$L(r) = \sqrt{K(r)/\pi} \quad (6.30)$$

Todos los gráficos muestran cuatro líneas, una superior (verde), una inferior

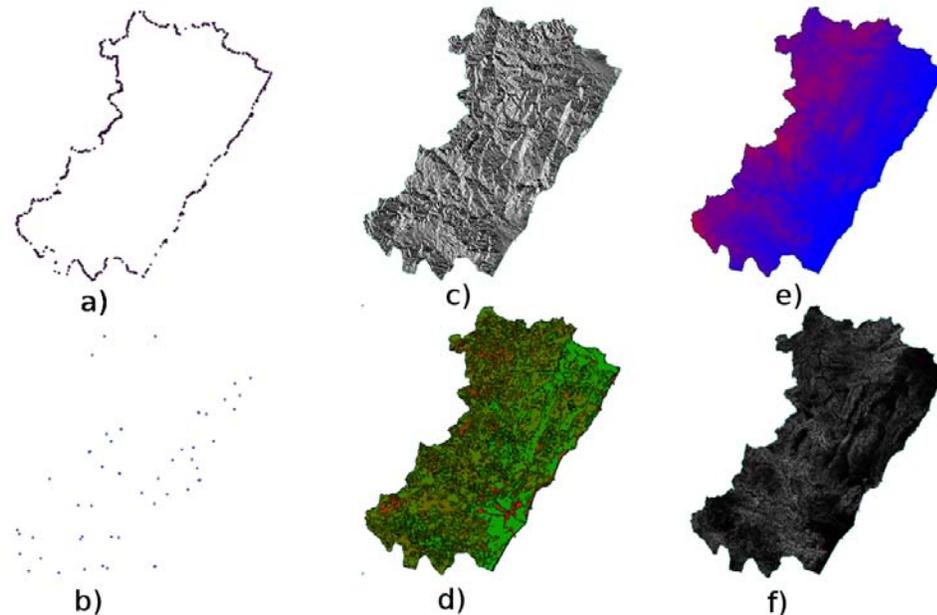


Figura 6.23: Representación visual de los parámetros de entrada que son, a) el contorno, b) el incendio, c) la orientación, d) el uso de la tierra, e) la elevación y f) la pendiente.

(azul) que son los límites del modelo, una interior (roja) que es la distribución media teórica de las simulaciones, el valor teórico del modelo utilizado en cada caso y la última línea que está por encima de todas las demás, siendo ésta la distribución empírica (negra) y mostrando los valores reales obtenidos para dicho modelo.

Para saber qué modelo ofrece una mejor predicción del riesgo de incendio hay que fijarse en la línea de color negro. Esta línea tiene que situarse dentro de la zona con sombreado gris. Si la línea empírica está por encima del límite superior significa que el caso está agrupado y, si la línea empírica está por debajo de la línea inferior significa que el proceso puntual debe ser modelado como inhibición.

Los gráficos anteriores muestran que el método de *Area-Interaction* es el mejor, porque la línea de distribución empírica siempre se mantiene dentro de las líneas del recinto. Los cuatro métodos son buenos para el riesgo de incendios forestales, pero tal vez el mejor de los cuatro es el modelo con todas las covariables.

El mapa con el riesgo de incendio también es devuelto por el servicio en forma de una capa geoespacial codificada en formato KML. Este mapa se muestra en la

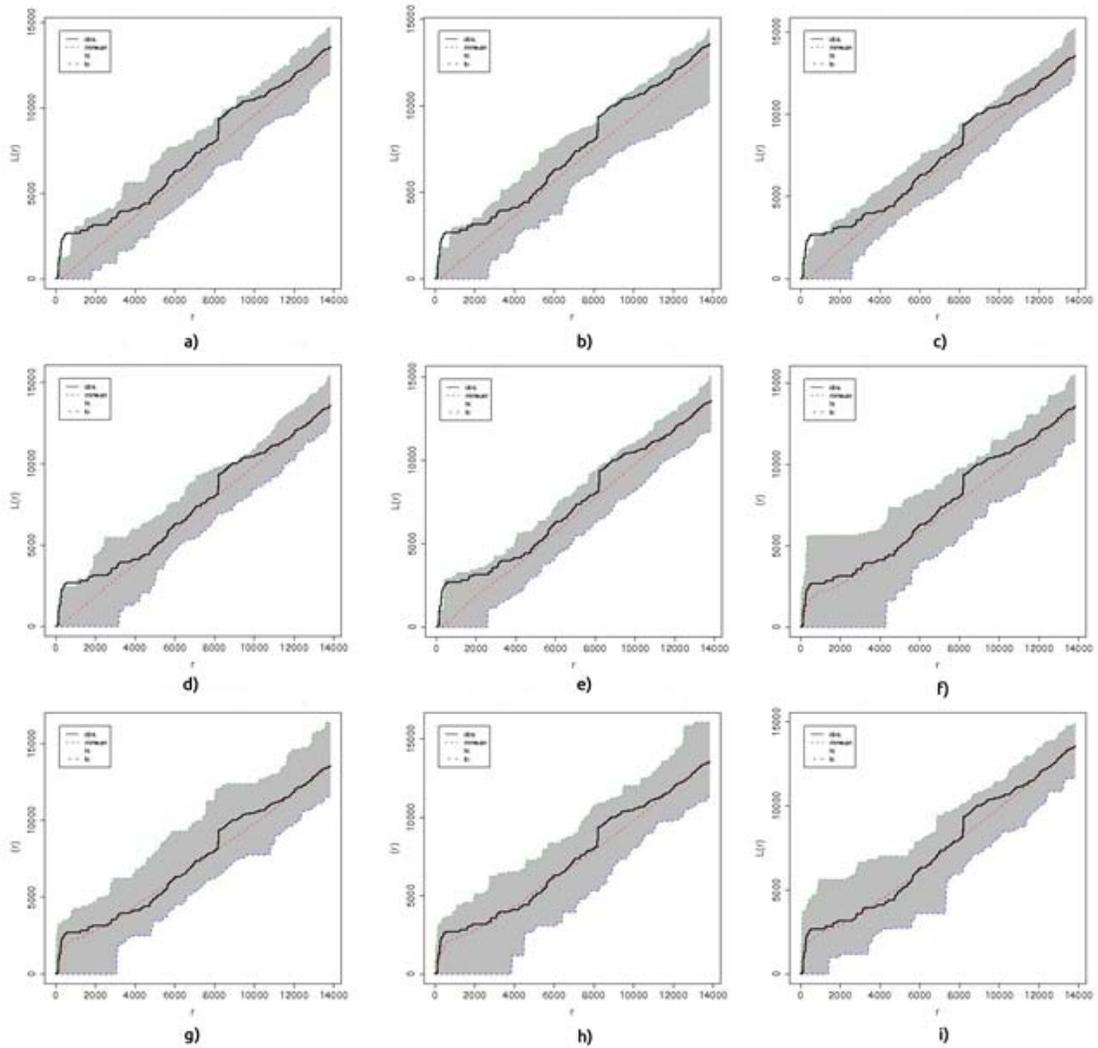


Figura 6.24: Gráficas de todos los modelos. a) Gráfico del modelo de *Poisson* polinómicas con x, y . b) Gráficos de modelo de *Poisson* con $x, y, 2$. c) Gráficos de modelo de *Poisson* con covariable elevación. d) Gráfico del modelo de *Poisson* con el uso del suelo de covarianza. e) Gráfico del modelo de *Poisson* con todas las covariables. f) Gráfico del modelo de *Area-Interaction* con x, y polinómicas. g) Gráfico del modelo de *Area-Interaction* con covariable elevación. h) Gráfico del modelo de *Area-Interaction* con el uso del suelo de covarianza. i) Gráfico del modelo de *Area-Interaction* con todas las covariables.

Figura 6.25. Cada punto en el mapa tiene sus resultados de predicción asociadas. Cada punto tiene el resultado de todos los modelos ejecutados, almacenándose

mediante el atributo ExtendedData.



Figura 6.25: La representación visual de los contenidos de un punto KML.

Los resultados se dividen en 20 clases, utilizando el cuartil divisional, para poder ver la predicción para cada modelo de una forma visual. Las siguientes figuras (Figura 6.26) muestran las predicciones de riesgo de incendio obtenidas para cada modelo definido. La tabla (Tabla 6.9) muestra los resultados en tiempo computacional para cada uno de los modelos definidos.

Núm. de modelo	Tipo de modelo	Covariables	Tiempo
1	Poisson	Con polinomio X Y	42 s
2	Poisson	Con polinomio X Y 2	42 s
3	Poisson	Con la covariable elevación	1 m 31 s
4	Poisson	Con la covariable usos del suelo	1 m 29 s
5	Poisson	Con todas las covariables	1 m 31 s
6	Area-Interaction	Con polinomio X Y	4h 51 m 55 s
7	Area-Interaction	Con la covariable elevación	2h 13 m 30 s
8	Area-Interaction	Con la covariable usos del suelo	2h 20 m 15 s
9	Area-Interaction	Con todas las covariables	2h 20 m 38 s

Tabla 6.9: Tiempos de ejecución para cada modelo.

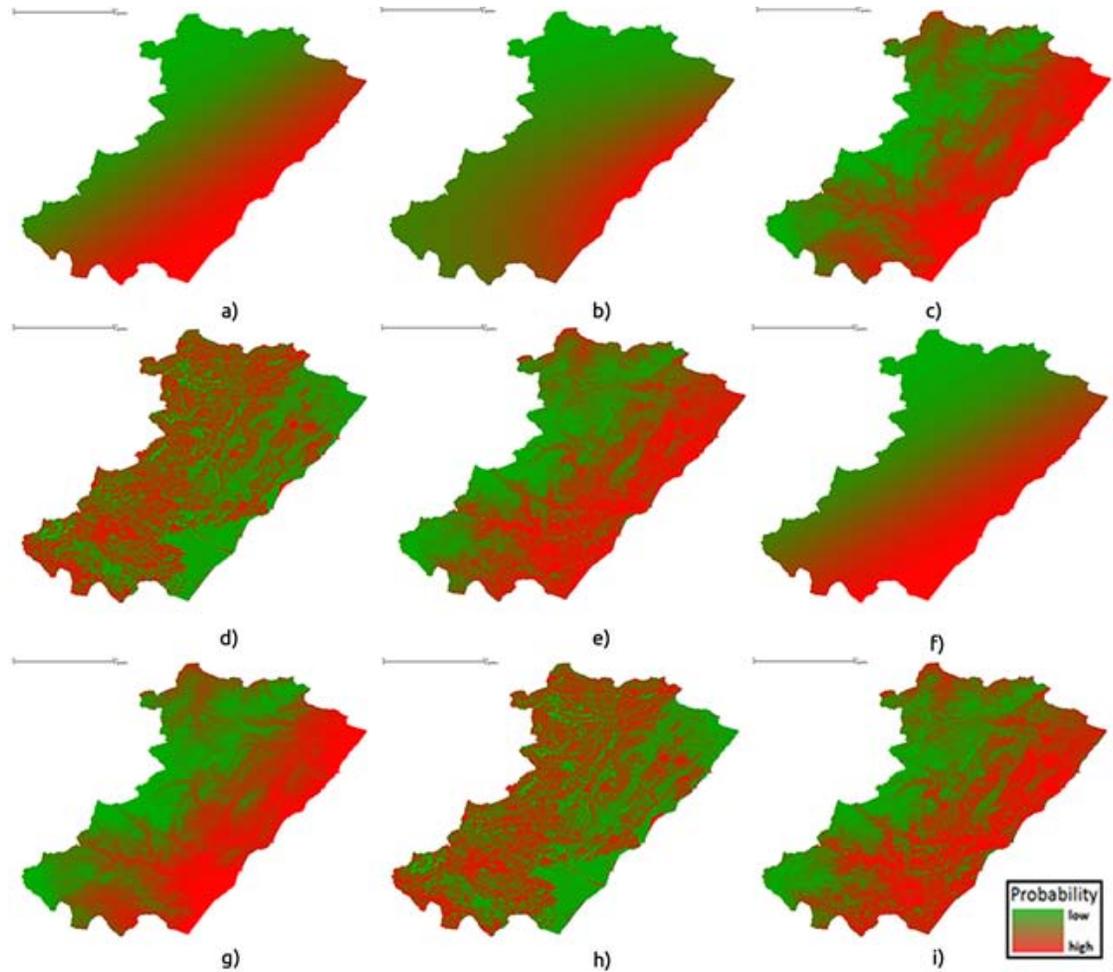


Figura 6.26: Mapa con los riesgos de incendio. a) Mapa del modelo de *Poisson* polinómicas con x, y . b) Mapa de modelo de *Poisson* con $x, y, 2$. c) Mapa de modelo de *Poisson* con covariable elevación. d) Mapa del modelo de *Poisson* con el uso del suelo de covarianza. e) Mapa del modelo de *Poisson* con todas las covariables. f) Mapa del modelo de *Area-Interaction* con con x, y . g) Mapa del modelo de *Area-Interaction* con covariable elevación. h) Mapa del modelo de *Area-Interaction* con el uso del suelo de covarianza. i) Mapa del modelo de *Area-Interaction* con todas las covariables.

Capítulo 7

Conclusiones y Trabajo Futuro

Este capítulo resume las contribuciones alcanzadas por el trabajo presentado a lo largo de esta tesis. En primer lugar, se presentan las conclusiones (Sección 7.1), indicando las principales aportaciones para cada una de las etapas del ciclo de vida definido y, en segundo lugar, en la Sección 7.2 se discutirán las posibles líneas de trabajo futuro.

7.1. Conclusiones generales

En esta sección se detallan cuáles son las conclusiones y contribuciones generales a lo largo del trabajo realizado, dentro la línea de investigación presentada. Las contribuciones se presentaran siguiendo la estructura marcada por el ciclo de vida de los datos de sensores.

7.1.1. Adquisición

Para cumplir la etapa de adquisición, se ha presentado una propuesta de plataforma de sensorización llamada *SEnviro*. Esta plataforma cumple con los tres objetivos principales para los que había sido diseñada, que son: utilizar hardware libre; ofrecer la interoperabilidad a través de estándares; y ser capaz de trabajar con cualquier tipo de sensores para adaptarse a diferentes uso.

La plataforma desarrollada utiliza hardware libre, lo cual proporciona muchos beneficios. Ofrece la posibilidad de facilitar el acceso a una gran comunidad de desarrolladores y facilita su propio uso. Además, permite su expansión debido a que hay muchos componentes compatibles, los cuales tienen los esquemas

totalmente disponibles. Otra ventaja de utilizar hardware libre es el bajo coste de estos componentes, ya que hay muchos fabricantes que proporcionan el mismo componente.

Para dotar de servicios interoperables a la plataforma, se ha utilizado un candidato a estándar llamado OGC SensorThings API. Esta API, ofrece un acceso fácil y ligero a los datos de los sensores utilizando los paradigmas de la IoT y WoT. La ventaja de ofrecer una interfaz estándar es la posibilidad de reutilizar los clientes.

SEnviro ofrece una conexión fácil tanto en términos de software como de hardware. A nivel de hardware, ofrece conexiones plug&play usando el shield Grove. A nivel de software, *SEnviro* desarrolla un *Core*, en el que los sensores pueden ser añadidos con poco esfuerzo.

La plataforma *SEnviro* ha sido diseñada para integrarse en el IoT y WoT. No sólo ofrece una interfaz CRUD, también el hardware ha sido diseñado teniendo en cuenta los dos paradigmas. Cada *SEnviro Thing* puede ser considerado como un objeto inteligente ya que está permanentemente conectado a Internet mediante el protocolo IP utilizando el módulo Wi-Fi incluido.

Otro de los retos de *SEnviro* fue ofrecer una solución energéticamente autónoma. Para ello se añade una fuente de alimentación con una batería y una placa solar. De esta forma, *SEnviro* puede ser desplegado en cualquier ambiente exterior. Otro punto a destacar, es la capacidad de *SEnviro* para modificar su comportamiento sin tener que reprogramar el microcontrolador. Esto evita la sustitución de los sensores cuando el *SEnviro Thing* está instalado y no necesita sustituirlo para el cambio de sus ajustes.

Con el fin de validar la arquitectura propuesta, se presenta una red de sensores para la monitorización ambiental mediante el despliegue de múltiples *SEnviro Things*. Se ha elegido el campus de una Universidad como el contexto para esta validación. Para ello, una red con cinco *SEnviro Thing* se ha diseñado exitosamente. Por otra parte, se ofrece un cliente web capaz de consumir los datos siguiendo las interfaces ofrecidas por la plataforma.

7.1.2. Publicación

Una vez obtenidas las observaciones, la siguiente etapa es su publicación. La publicación es una de las etapas más críticas, ya que si ésta no se produce de una forma correcta, el ciclo de vida del contenido se ve interrumpido, imposibilitando

el acceso a los datos.

Para llenar este paso, se ha desarrollado un procesamiento automático para la publicación de contenido geoespacial, entre los cuales se contemplan los datos generados por los sensores. La publicación se realiza mediante un procesamiento estándar y el contenido se publica en los diferentes servicios geoespaciales. Este procesamiento se amplía para la generación y publicación automática de metadatos.

Además, se adapta para ofrecer la capacidad de publicar datos de sensores, como los propios sensores y las observaciones producidas, en un servicio SOS. Para la creación del procesamiento, se ha utilizado la especificación WPS y la implementación de 52North.

El componente ofrece la mejora de la disponibilidad de datos en el entorno de la SDI, ya que mejora el mantenimiento de datos y metadatos. La generación y publicación de metadatos facilita el saber dónde se encuentran los datos, además de facilitar su descubrimiento. Se ha propuesto una posible solución al problema de la disponibilidad de datos y metadatos geoespaciales y, lo más importante, se ofrece un punto de acceso a los datos publicados, mejorando la visibilidad y el descubrimiento de los datos, así como su acceso.

7.1.3. Acceso

Tras la publicación de los datos se presenta la etapa de acceso. Hoy en día, hay muchas fuentes de datos procedentes de diferentes redes de sensores. Uno de los actuales problemas es la gran variedad de formatos y protocolos utilizados para servir datos de sensores.

Un reto importante es aumentar la interoperabilidad de estas fuentes de datos. Se propone una arquitectura orientada a servicios para implementar una aplicación, la cual organiza un flujo de trabajo para la gestión de los formatos de datos heterogéneos. Como primera aportación para el acceso, se propone un sistema que tiene como objetivo la integración de fuentes de datos no estructurados, además de ofrecerlos de manera manejable, estructurada y estándar. Se propone una capa de servicio con un diseño mediante el patrón Fachada, que actúa como un *middleware* entre el cliente y los servicios para aumentar la interoperabilidad, ayudando a los clientes a obtener la información fácilmente.

Se proponen interfaces SOS y RESTful para mejorar la interoperabilidad entre

las diferentes fuentes de datos. El servicio REST es capaz de representar cada sensor por un URI siguiendo los paradigmas de IoT y WoT. Sin embargo, también mantiene el enfoque de los estándares SWE para los datos de sensores (SensorML y O&M). Los beneficios de esta interfaz son mejorar el rendimiento en ambientes con características reducidas, como pueden ser teléfonos móviles (Aijaz y otros, 2009). Se ha creado una aplicación móvil para que los usuarios puedan interactuar con la funcionalidad ofrecida, además de poder acceder y visualizar los datos que ofrecen las fuentes de los sensores elegidas.

La segunda aportación para el acceso es la utilización del estándar candidato OGC SensorThings API, el cual también está basado en los paradigmas de IoT y WoT y, como todos los estándares, ofrece los beneficios de interoperabilidad y reusabilidad.

La última aportación para el acceso ofrece un mecanismo, SDS, para el consumo de cualquier tipo de mensaje, en este caso observaciones de sensores, para fuentes de datos que tengan una alta tasa de refresco. Dicho mecanismo es muy útil para el análisis en tiempo real, ya que garantiza la transición de los mensajes entre productor y consumidor de forma instantánea. Además, ofrece multitud de protocolos de conexión, aumentando así la interoperabilidad del sistema en el que se utilizará.

Otras de las características de este mecanismo son ofrecer una comunicación síncrona y asíncrona; permitir la transformación del formato de datos; ser flexible en el uso entre aplicaciones; procesar paralelamente la comunicación de los mensajes; ofrecer varios niveles de calidad de servicio; ser capaz de entregar datos en tiempo real a través de multicast. Este mecanismo ha sido utilizado en la metodología para el análisis de grandes cantidades de datos en tiempo real.

7.1.4. Análisis

En la última etapa, el análisis, se han presentado dos metodologías capaces de analizar y aplicar cualquier tipo de modelo sobre los datos de sensores.

En la primera de ellas se ha desarrollado una metodología general para analizar los flujos de datos de las redes de sensores en tiempo real. La metodología incluye puntos de acceso a las diferentes fuentes de datos de sensores ofreciendo múltiples protocolos. Además, compatible con una amplia gama de herramientas para conectar con fuentes de datos independientemente de los tipos de datos

que ofrezcan. La metodología tiene un enfoque de bróker, con el que se permite conectar con diferentes interfaces en tiempo real, de modo que el sistema es capaz de conectarse, analizar grandes cantidades de observaciones y ofrecerlas en un formato estándar.

Esta metodología aborda dos retos diferentes. El primer desafío es proporcionar un sistema capaz de analizar datos de sensores en tiempo real. Para ello, se soportan múltiples fuentes de sensores, incluyendo todas las aplicaciones sobre IoT y permitiendo analizar cada una de las observaciones sin pérdidas. El segundo reto es ofrecer una solución escalable e interoperable. De esta forma, se ofrecen diferentes características para contribuir a mejorar estos principios. Para este propósito, los componentes diseñados se pueden reutilizar para otros escenarios con otras propiedades. También un patrón bróker se utiliza para conectar con múltiples fuentes heterogéneas. Por último, para mejorar la interoperabilidad se utilizan estándares para codificar las observaciones de sensores y se ofrece gran variedad de protocolos para servirlos.

Para probar esta metodología, se presenta una prueba de concepto para la detección de anomalías en entornos de monitorización ambiental. En esta prueba, se utilizan nuevas tecnologías para el procesamiento de flujos de datos en tiempo real, como Storm y JMS. Aunque la fuente de datos utilizada tiene bajas tasas de actualización, la metodología podría ser escalada en fuentes con mayores tasas de refresco o añadir múltiples fuentes para que fueran analizadas. El algoritmo CUSUM ha demostrado ser útil en el estudio de las anomalías en series de observaciones de calidad del aire. Sin embargo, CUSUM presenta algunas limitaciones que deben tenerse en cuenta, como por ejemplo que se considere que las series de observaciones de los fenómenos analizados siguen una distribución normal, o que no tengan cambios de tendencias debido a la estacionalidad.

La segunda metodología de análisis que se ha presentado, utiliza un servicio de geoprocesamiento especificado mediante estándares. Se ha utilizado la especificación de OGC WPS, con la implementación de 52North. El uso de estándares mejora la interoperabilidad porque proporciona un punto de ejecución estándar compatible con clientes existentes de WPS.

Uno de los objetivos del uso de este tipo de geoprocesamientos, es la posibilidad de compartir los modelos científicos y sus resultados, ya que al respetar los estándares y ofrecer un diseño distribuido facilita su reusabilidad.

En particular, esta metodología permite envolver los modelos científicos mediante una interfaz WPS estándar. El uso de interfaces basadas en estándares, como por ejemplo WPS, mejora la interoperabilidad. De esta forma, la funcionalidad está disponible a través de un servicio con una interfaz ampliamente conocida, que permite que se invoque y se reutilice en diferentes escenarios. Las arquitecturas orientadas a servicios como WPS, ofrecen la posibilidad de trasladar las tareas más complejas a la parte servidora, con lo que ayuda a crear aplicaciones menos costosas en la parte de los usuarios.

Como caso de uso a esta metodología, se ha aplicado a un modelo para la predicción de los posibles riesgos de incendios forestales. Dichos modelos, anteriormente eran ejecutados en un ambiente de escritorio, utilizando el software *R*. Mediante la metodología definida, se ha posibilitado el uso de este modelo en un entorno distribuido, sin la necesidad de estar acoplado a un entorno particular.

Tras la ejecución de los modelos definidos, los modelos *Area-Interaction* proporcionan una buena aproximación en la mayoría de los casos, a diferencia de los otros modelos, incluyendo el *Poisson* con todas las covariables. Con respecto a los resultados obtenidos, se concluye que, a pesar de la variabilidad encontrada entre covariables, el modelo que mejor se adapta al comportamiento de los incendios para la mayoría de las situaciones es el *Area-Interaction*. El mejor modelo es el *Area-Interaction* con todas las covariables.

El tiempo de ejecución del *Area-Interaction* es mayor que los modelos de *Poisson*; esto es porque los modelos de *Area-Interaction* necesitan más operaciones que los modelos *Poisson* porque la estructura es más compleja, los modelos de *Area-Interaction* dependen de la cantidad de parámetros que describen el agrupamiento y las posibilidades de inhibición.

Estos modelos tienen un coste temporal con medidas en horas. Esto no es causado por la distribución de las capacidades de las interfaces del servicio utilizado. Además, la integración y la preparación para la ejecución de los modelos son realmente rápidas, incluso cuando se utilizan entradas con datos muy grandes (100 MB) para este tipo de computación. El problema radica en los propios modelos en *R*. Más características a tener en cuenta para gestionar los costes y mejorar el rendimiento son las características del servidor utilizado¹ y de la red.

¹Características del equipo utilizado: procesador Intel (r) Xeon (r) CPU 5160 @ 3.00GHz MHz memoria RAM 16 GB FB-DIMM DDR2 667 síncronica.

Mediante el uso de la metodología de procesos puntuales descrita se obtienen ciertas limitaciones computacionales. En una pequeña región se pueden utilizar uno o diez mil puntos, haciendo que el proceso sea más o menos rápido. Por lo que el error asociado será diferente, ya que depende del número de puntos de la región y si los datos son buenos.

7.2. Líneas futuras

Desafortunadamente, el tiempo del actual trabajo está limitado y el objetivo preferente ha sido el de desarrollar una visión general de cómo tratar cada una de las etapas definidas para el ciclo del contenido de sensores. De esta forma, no ha sido objetivo del actual trabajo concretar en una etapa en particular, por lo que existen todavía una gran variedad de puntos y direcciones que son posibles para explotar y resolver en el ámbito del ciclo de vida planteado. Al igual que en las conclusiones, el trabajo futuro se detallará por cada una de la etapas definidas.

En la primera de ellas, la etapa de adquisición, se propone como un objetivo futuro el ofrecer un cliente web con el que ser capaz de ajustar el comportamiento de cada *SEnviro Thing*. Actualmente, los ajustes en los que se basa el comportamiento de cada *Thing* son proporcionados por medio de un simple archivo de texto en el lado del servidor. Otra línea de desarrollo del proyecto sería agregar diferentes alternativas de conectividad, también basadas en IP, como pueden ser datos móviles o Bluetooth. Una tercera y última mejora sería aplicar también la metodología de análisis en tiempo real definida en el Capítulo 5 sobre los datos proporcionados por la red *SEnviro*.

En la etapa de publicación se plantean como futuras líneas las siguientes propuestas. La primera de ellas es aumentar la cantidad de tipos de contenido soportados al igual que el número de los servicios donde publicar. Referente a la creación de metadatos, se pretenden aumentar los elementos del metadato mediante el uso de ontologías, ayudando éstas a proporcionar un proceso que puede ofrecer las palabras clave de forma automática mediante el propio contenido de los datos. De esta manera, el usuario no tendrá que indicar las palabras clave, ya este proceso les permitirá obtenerlas automáticamente. Otra mejora será la creación de un cliente web que facilitará la ejecución del WPS, tanto en un ambiente de escritorio como móvil.

En la siguiente etapa de acceso se adoptarán y ampliarán las interfaces descri-

tas con más características para definir tanto los sensores como las observaciones, de forma que se adapten en mayor medida a los estándares definidos por OGC. Además, también se propone la adaptación a diferentes fuentes de datos con diferentes formatos a los estudiados.

La última etapa, el análisis. Referente a la primera metodología diseñada, se pretende que sea aplicada a las fuentes de datos generados por los sensores previstos en nuevos dispositivos, como teléfonos inteligentes o "weareble", tales como: acelerómetro, giroscopio o pulsómetro, entre otros. También, que se aplique dicha metodología sobre los datos publicados en la Web 2.0, con el fin de inferir las actividades de las personas en el mismo momento en que se producen los eventos (anomalías).

El uso de algoritmos alternativos para la detección de anomalías, por ejemplo el desarrollado por Chelani (2011), queda por investigar. Otros tipos de algoritmos específicos para cada fenómeno también serán presentes en futuros trabajos.

Las posibles mejoras sobre la segunda metodología presentada, el análisis mediante geoprocésamiento estándar, son diversas. La primera de ellas está relacionada con la interacción del usuario con el geoprocésamiento. Su objetivo es ayudar a los usuarios a ejecutar los modelos, facilitar las entradas y salidas de los parámetros necesarios para la ejecución del modelo, así como elegir qué tipo de modelo ejecutar de los predefinidos. Para ello, se implementará una aplicación cliente para teléfonos móviles. Tamayo y otros (2012a) describe cómo generar fácilmente un cliente WPS para dispositivos móviles Android.

Referente a las mejoras sobre los modelos detallados, se considera el hecho de mejorar los modelos definidos para incorporar nuevos avances, considerando procesos puntuales con componentes espacio-temporales. Un beneficio directo de este enfoque es el modelado y la evaluación de una zona, considerando la temporalidad como característica, siendo algo que no se ha considerado dentro del alcance del actual trabajo. En esta área, actualmente se está investigando la integración de covariables en tiempo real, tales como datos meteorológicos. De esta forma podría ser aplicado sobre los datos provenientes de la red *SEnviro* utilizando los servicios para servir datos en tiempo real diseñados en el Capítulo 4.

Capítulo 7

Conclusions and Future Work

This chapter summarizes the contributions achieved from the work presented throughout this thesis. First, the conclusions (Section 7.1) are presented, indicating the main contributions for each stage of the defined life cycle and, secondly, in Section 7.2 the future research are discussed.

7.1. General conclusions

This section details the overall conclusions and contributions throughout the presented work. The contributions will be presented following the structure marked by the life cycle of the sensor data.

7.1.1. Acquisition

To accomplish the acquisition stage, a proposal for sensing platform called *SEnviro* is developed. This platform meets the three main objectives for which it was designed they are the use of open hardware; providing interoperability using standards; and being able to work with any type of sensors to adapt different usages.

The developed platform uses open hardware, which provides many benefits. It offers the possibility of providing access to a large community of developers and facilitate component usage. It also allows expansion because there are many compatible components, which are fully available schemes. Another advantage of using open hardware is the low cost of these components, as there are many manufacturers that provide the same component.

To provide an interoperable platform, a candidate OGC standard called SensorThings API is used. This API, provides easy access to sensor data using the paradigms of IoT and WoT. The advantage of offering a standard interface is the ability to reuse clients. *SEnviro* provides an easy connection both in terms of software and hardware. At the hardware level, it provides plug&play connections using the shield Grove. At the software level, *SEnviro* develops a *core*, in which the sensors can be added with little effort.

The *SEnviro* platform has been designed to be integrated into the IoT and WoT paradigms. It offers a CRUD interface, also the hardware has been designed keeping in mind the two paradigms. Each *SEnviro Thing* can be considered as a smart object, it is permanently connected to the Internet using the IP protocol using the included Wi-Fi module.

Another challenge of *SEnviro* was to offer an autonomous energy solution. To meet this goal, *SEnviro* it has a power supply with a battery and a solar panel. Thus *SEnviro* can be installed in an outdoor environment without any restrictions. Another highlight is the ability to change its behavior without having to reprogram the microcontroller. This offers the option of being able to change the settings without removing the *SEnviro Thing*.

In order to validate the proposed architecture, a sensors network for environmental monitoring using multiple *SEnviro Things* is presented. A university campus for this validation is chosen. To do this, a sensor network with five *SEnviro Thing* is designed. Moreover, a web client to consume sensor data is provided.

7.1.2. Publication

After obtaining the observations, the next step is publication. Publication is one of the most critical stages, because if it is not carried out properly the content life cycle is disrupted, preventing access to data.

To complete this step, an automated process to publish geospatial content, including sensors data is developed. In order to publish in different geospatial services a standard geo process is used. This process is extended for metadata generation and publication.

In addition, it adapts to provide the ability to publish sensor data, such as the sensors themselves and observations produced in a SOS service. To achieve this, the WPS specification with 52North implementation is used.

The component addresses the improvement of data availability in the environment of the SDI as well as its maintenance, the particular generation and publication of metadata facilitates where data is found, because it will generate and publish metadata which defines and facilitates its location. There has been a possible solution to the problem of availability of data and metadata, and most importantly the data and metadata are linked using the link, so that every time data and/or metadata are published, and improve not only the visibility and discovery of the data but also its access.

7.1.3. Access

Afterwards, the publication of the data access stage it is presented. Nowadays, there are many data sources coming from different sensor networks. The main concern about the huge amount of data provided by these sensor networks is the heterogeneity of the data.

An important challenge is to increase the interoperability of these data sources. In this paper a service-oriented architecture is proposed to implement an application, which orchestrates a workflow for the management of heterogeneous data formats provided by sensor network. A system that aims at the integration of unstructured data sources with different characters is proposed, and which offers them in a structured and interoperable way. In addition, the service layer in our application is enhanced with a module that implements the design pattern façade, and acts as a “middleware” between the client and the services to increase interoperability by implementing several interfaces and helping users to obtain information more easily.

SOS and RESTful interfaces are proposed to improve interoperability between different data sources. A RESTful service is able to represent each sensor by a URI following the paradigms of IoT and WoT. Nevertheless, the standard GIS for the sensorial data (SensorML and O&M) are also maintained. The benefits of this interface are to improve performance in environments with reduced features, such as mobile phones (Aijaz y otros, 2009). A mobile client application for users to interact with the offered functionality and to be able to access and visualize data from the selected sensor network is implemented.

The second contribution for this stage is the standard candidate OGC SensorThings API, which is also based on the paradigms of IoT and WoT and, like all

standards, offers the benefits of interoperability and reusability.

The latest contribution in this stage is to provide a mechanism, called SDS, for the consumption of any type of message, in this case sensor data sources with a high refresh rate. This mechanism is useful for analysis in real time, as it ensures the transition between producer and consumer messages instantly. It offers many connection protocols, thus increasing system interoperability.

Other features of this mechanism are to provide a synchronous and asynchronous communication; allow data format transformation; be flexible in use between applications; parallel processing communication messages; offer several levels of service quality; be able to deliver real-time data via multicast. This mechanism has been used in the methodology for analyzing large amounts of data in real time.

7.1.4. Analysis

In the last stage, analysis, two methodologies capable of analyzing and applying any type of model sensor data are designed. In the first of them a general methodology for analyzing data streams in real time is developed. The methodology includes access points to different sensor data sources offering multiple protocols. It supports a rich range of tools to connect with sensor data sources regardless of the particular data types that they offer. This brokering approach also permits connecting with different real-time interfaces, so that the system can analyze large amounts of sensor data in real-time and can serve the results as a standard format.

The presented methodology addresses two challenges. The first challenge of the presented work was to provide a system able to analyze data from sensors in real-time. Multiple sensor data sources are supported, including all IoT applications. Our methodology allows for the analysis of each of the observations without faults. Although, the data source used has low refresh rates, it could scale to higher refresh rates or add multiple data sources into the same system. The second challenge is to offer a scalable and interoperable solution. The methodology offers different features to contribute to improve these requirements. For this purpose, the designed components can be reused for other scenarios with other properties. Also a broker pattern is used to offer multiple connections with heterogeneous interfaces. Finally, to enhance the interoperability, standards are used to encode the data and a variety of protocols are provided. To test and illustrate this methodology, a proof of concept implementation that detects anomalies from environmental

monitoring stations is presented. In this proof, new technologies are used such as the stream processing framework Storm, and JMS. CUSUM algorithm has proven to be useful in looking at anomalies in the series of observations of air quality and weather. However, CUSUM presents some limitations that must be taken into account, such as the consideration that all of the series follow a normal distribution and a series of observations cannot have trend changes.

The second methodology to analyse sensor data uses a standard geoprocessing service. Also, the WPS specification with the 52North is used. The use of standard-based interfaces for web services such as WPS improves interoperability. Now the functionality is available through a well-known interfaced service, which allows it to be invoked and reused in different scenarios. Processes are run at the server so there is no need to install and store anything locally. However, the use of standard-based services and service oriented architectures to build applications make the application more complex. Therefore, the applications have to consider the implementation of user-friendly client sides so users can interact transparently with the functionality.

This methodology is used to apply a model to predict the possible risks of forest fires. These models were previously executed in a desktop environment with the *R* software. Using the methodology defined, is possible to use this model in a distributed environment, without the need to be coupled to a particular environment.

After executing the defined models, *Area-Interaction* models provide a good approximation in most cases, including *Poisson* with all covariates. Regarding the results, the best model is the *Area-Interaction* with all covariates.

The execution time of the *Area-Interaction* is greater than the *Poisson* models; this is because the *Area-Interaction* models need more operations than *Poisson* models because their structure is more complex. *Area-Interaction* models depend on the quantity of parameters that describe the clustering and the inhibition possibilities.

These models have a temporal cost measure in hours. This is not caused by the distribution of the modelling capacities of the standard-based service interfaces or protocols, but the causes of the slow performance is due to statistical models when they execute complex operations for many points. Furthermore, the integration and execution of the -package to collect and prepare data to be processed is really fast, even when data inputs are very large (100 MB) for this type of computation. More

temporal features to take into account to manage costs and improve performance are the computer and network characteristics ¹.

7.2. Future research

Unfortunately, the current work has limited time and the priority objective has been to develop an overview of how to propose a solution for each stage defined for the life cycle of the sensor data. Thus, this don't have the aim to work only in a specific stage, so there are a lot of points and directions to continue and resolve in this field. As in other sections, the future work will be detailed for each of the steps defined.

In the first stage, acquisition, is proposed as a future goal to offer a web client to be able to adjust the behavior of each *SEnviro* Thing. Currently, the settings with the behavior of each *SEnviro* Thing are provided by a simple text file on the server side. Another future research of development of the project can be to provide different connectivity alternatives, also based on IP, such as mobile data or Bluetooth. A third and final improvement would be to apply the methodology of real-time analysis defined in Chapter 5 on the data provided by the *SEnviro* network.

In the publication stage the following objectives are proposed as future work. The first is to increase the number of content types supported and the number of services to publish. Regarding the creation of metadata, they are intended to increase the elements of metadata using ontologies. In this way, the user does not have to specify keywords, but this process will allow for obtaining these keywords automatically. Another improvement is the creation of a web client that will facilitate the WPS execution, in an environment of desktop and mobile.

In the next stage of access will be adopted and expanded the described interface with more features to define both sensors and observations, always using standards defined by OGC. Furthermore, different data sources with multiple formats will be studied in order to be included in the study.

The last stage, the analysis. Regarding the first methodology, the proposed real-time analysis has in mind to be applied to new generation data sources provided by smart phones or wearables, such as: accelerometer, gyroscopes, heart rate

¹Features of the computer used: Processor Intel(R) Xeon(R) CPU 5160 @ 3.00 GHz Memory RAM 16 GB DDR2 FB-DIMM synchronous 667 MHz.

monitor and so on. And also over the published data into Web 2.0. In order to infer peoples activities in the same moment that the anomalies are produced.

The use of alternative algorithms for anomaly detection, such as developed by Chelani (2011), remains to be investigated. Other types of specific algorithms for each event will also be present in future work.

Possible improvements on the second methodology presented, analysis by standard geoprocessing, are diverse. The first one is related to the interaction of the user with the model. Our aim is to assist users in working more interactively, by being able to select and refine the input parameters such as the model type to be used (Poisson or Area-Interaction) or how many covariates will be needed. To do this, a client application for mobile phones will be implemented. Tamayo y otros (2012a) describes how to easily generate a WPS client for Android mobile devices.

Regarding improvements on detailed models, considering space-time point process models. A direct benefit of this approach is modelling and evaluating the corresponding space-time interaction, which is something that it have not considered in the scope of this thesis. In this area, currently new investigations to integrate real-time factors such as climate data will be included in the future. Thus it could be applied to data from the *SEnviro* using network services to serve real-time data designed in Chapter 4.

Apéndice A

Publicaciones

Revistas con impacto:

1. **Sergio Trilles**, Óscar Belmonte, Sven Schade, Joaquín Huerta. *A methodology to analyze sensor data streams in real time - with a proof of concept implementation for anomaly detection from environmental data*. SI: Big Data: Dealing with INFFUS. Information Fusion. (Enviado) **(Q1)**
2. **Sergio Trilles**, Alejandro Luján, Óscar Belmonte, Raúl Montoliu, Joaquín Torres-Sospedra, Joaquín Huerta. *SEnviro: A Sensorized Platform Proposal Using Open Hardware and Open Standards*. Sensors 2015, 15(3), 5555-5582. [dx.doi.org/10.3390/s150305555](https://doi.org/10.3390/s150305555). **(Q1)**
3. **Sergio Trilles**, Óscar Belmonte, Laura Díaz, Joaquín Huerta. *Mobile Access to Sensor Networks by Using GIS Standards and RESTful Services*. IEEE Sensors, 14 (12): 4143 – 4153, 2014, ISSN 1530-437X. [dx.doi.org/10.1109/JSEN.2014.2339931](https://doi.org/10.1109/JSEN.2014.2339931). **(Q1)**
4. **Sergio Trilles**, Pablo Juan, Laura Díaz, Pau Aragó, Joaquín Huerta. *Integration of Environmental Models in Spatial Data Infrastructures: A Use Case in Wildfire Risk Prediction*. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 6: 1-11, 2013, ISSN 1939-1404, [dx.doi.org/10.1109/JSTARS.2012.2236538](https://doi.org/10.1109/JSTARS.2012.2236538). **(Q1)**

Revistas:

1. **Sergio Trilles**, Laura Díaz, José Gil, Joaquin Huerta. *Assisted Generation and Publication of Geospatial data and metadata. International Journal of Spatial Data Infrastructure Research*. (Escribiendo)
2. **Sergio Trilles**, Pablo Juan, Laura Díaz, Joaquín Huerta. *Integración de modelos medioambientales en infraestructuras de datos espaciales: un caso de uso en predicción de incendios Revista Catalana de Geografía*. February 2012. ISSN: 1988-2459.

Congresos:

1. **Sergio Trilles**, S. Schade, Ó. Belmonte Fernández y J. Huerta Guijarro. *Real-Time Anomaly Detection from Environmental Data Streams*. En: 18th International Conference on Geographic Information Science (AGILE´2015). ISSN 1863-2246. [dx.doi.org/10.1007/978-3-319-16787-9_8](https://doi.org/10.1007/978-3-319-16787-9_8).
2. **Sergio Trilles**, Sven Castelyn, Carlos Granell, Ignacio Guerrero, Joaquín Huerta, Gloria Bordogna, Monica Pepe, Lorenzo Busetto, Mirco Boschetti. *ERMES: Servicios de seguimiento del cultivo de arroz basados en una IDE*. V Jornadas Ibéricas de las Infraestructuras de Datos Espaciales (JIIDE 2014). Lisboa, November 2014.
3. **Sergio Trilles**, Óscar Belmonte, Laura Díaz, Joaquín Huerta. *Agile access to sensor network*. In: Joaquín Huerta, Sven Schade, Carlos Granell. *Connecting a Digital Europe through Location and Place*. Selected best short papers and posters of the AGILE 2014 Conference, 03-06 June 2014, Castellón, Spain. Castellón, Spain, June 2014. AGILE Digital Editions, 2014, ISBN 978-90-816960-4-3.
4. Sven Casteleyn, Carlos Granell, **Sergio Trilles**, Joaquin Huerta, Mirco Boschetti, Lorenzo Busetto, Monica Pepe, Dimitrios Katsantonis, Roberto Con-falonieri, Francesco Holecz, Javier García Haro, Ioannis Gitas. *How Earth Observation, Crop Modeling and ICT tools can help rice cultivation: the ERMES project*. In: Joaquín Huerta, Sven Schade, Carlos Granell. *Connecting a Digital Europe through Location and Place*. Selected best short papers and posters of the AGILE 2014 Conference, 03-06 June 2014, Castellón,

Spain. Castellón, Spain, June 2014. AGILE Digital Editions, 2014, ISBN 978-90-816960-4-3.

5. **Sergio Trilles**, Óscar Belmonte, Joaquín Huerta. *Mobile access to sensor network. A use case on wildfire monitoring*. DCAI'14. Salamanca, June 2014
6. **Sergio Trilles**, Laura Díaz, Joaquín Huerta. *Acceso ágil a redes de sensores*. IV Jornadas Ibéricas de las Infraestructuras de Datos Espaciales (JIIDE 2013). Toledo, November 2013.
7. Pau Aragó, P. Salvador, P. Juan, C. Díaz-Avalos, L. Serra, M. Sáez, D. Varga, **Sergio Trilles**, J. Mateu. *Aplicación de modelos de procesos puntuales para la caracterización espacio-temporal del régimen de incendios en el este de España*. VI Congreso Forestal Español, Vitoria, Junio 2013. ISBN: 978-84-937964-9-5
8. Irene Garcia Martí, Luis E. Rodríguez, Mauricia Benedito, **Sergio Trilles**, Arturo Beltran, Laura Díaz and Joaquín Huerta. *Mobile Application for Noise Pollution Monitoring through Gamification Techniques*. Lecture Notes in Computer Science, 2012, Volume 7522/2012, Entertainment Computing – ICEC 2012, pp. 562-571.
9. Irene Garcia, Luis E. Rodríguez, Mauricia Benedito, **Sergio Trilles**, Arturo Beltran, Laura Díaz, Joaquín Huerta. *Aplicación móvil para la monitorización de la contaminación acústica en entornos urbanos a través de técnicas de Gamification*. III Jornadas Ibéricas de las Infraestructuras de Datos Espaciales (JIIDE 2012). Madrid, October 2012
10. **Sergio Trilles**, Laura Díaz, Joaquín Huerta. *Monitorización de datos de calidad de aire*. III Jornadas Ibéricas de las Infraestructuras de Datos Espaciales (JIIDE 2012). Madrid, October 2012.
11. **Sergio Trilles**, Laura Díaz, José Gil, Joaquín Huerta. *Assisted generation and publication of geospatial metadata*. Proceedings of the 15th AGILE International Conference on Geographic Information Science (AGILE 2012). Avignon, France. April 2012. ISBN: 978-90-816960-0-5.

12. **Sergio Trilles**, José Gil, Laura Díaz, Joaquín Huerta. *Un geoprocesamiento para la publicación de metadatos geográficos*. JIIDE 2011. Barcelona, November 2011
13. **Sergio Trilles**, Pablo Juan, Laura Díaz, Joaquín Huerta. *Integración de modelos medioambientales en infraestructuras de datos espaciales: un caso de uso en predicción de incendios*. JIIDE 2011. Barcelona, November 2011.

Estancias:

1. Realizada en el Institute for Environment and Sustainability (IES) del Joint Research Centre (JRC) - Comisión Europea (Ispra, Italia), bajo la supervisión de Dr. Sven Schade desde el 1 de Septiembre del 2014 al 31 de Diciembre del 2014.

Bibliografía

- ABRAHAM, K. y PANDIAN, S. (2013). «A Low-Cost Mobile Urban Environmental Monitoring System». En: *Intelligent Systems Modelling Simulation (ISMS), 2013 4th International Conference on*, pp. 659–664. ISSN 2166-0662.
- AGENCY, EUROPEAN ENVIRONMENT (2007). «Technical report No 17/2007. CLC2006 technical guidelines.» *Informe tico*.
- AIJAZ, FAHAD; ALI, SYED ZAHID; CHAUDHARY, MUZZAMIL AZIZ y WALKE, BERNHARD (2009). «Enabling high performance mobile web services provisioning». En: *Vehicular Technology Conference Fall (VTC 2009-Fall), 2009 IEEE 70th*, pp. 1–6. IEEE.
- AKIDAU, TYLER; BALIKOV, ALEX; BEKIROGLU, KAYA; CHERNYAK, SLAVA; HABERMAN, JOSH; LAX, REUVEN; MCVEETY, SAM; MILLS, DANIEL; NORDSTROM, PAUL y WHITTLE, SAM (2013). «MillWheel: Fault-Tolerant Stream Processing at Internet Scale». En: *Very Large Data Bases*, pp. 734–746.
- ALESHEIKH, A. A.; OSKOU EI, A. K.; ATABI, F. y HELALI, H. (2005). «Providing interoperability for air quality in-situ sensors observations using GML technology». *International Journal of Environmental Science & Technology*, **2(2)**, pp. 133–140. ISSN 1735-1472.
- ALLIANCE, ZIGBEE (2008). «ZigBee Home Automation Public Application Profile, revision 15». *Informe tico*.
- ANDREWS, P. (1986). «BEHAVE : Fire Behavior Prediction and Fuel Modeling System - BURN Subsystem, Part 1». *The Bark Beetles, Fuels, and Fire Bibliography*.

- ARCA, BACHISIO; DUCE, PIERPAOLO; LACONI, MAURIZIO; PELLIZZARO, GRAZIA; SALIS, MICHELE y SPANO, D (2007). «Evaluation of FARSITE simulator in Mediterranean maquis». *International Journal of Wildland Fire*, **16(5)**, pp. 563–572.
- ARNABOLDI, VALERIO; CONTI, MARCO; DELMASTRO, FRANCA; MINUTIELLO, GIOVANNI y RICCI, LAURA (2013). «Sensor Mobile Enablement (SME): A light-weight standard for opportunistic sensing services». En: *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on*, pp. 236–241. IEEE.
- BADDELEY, A. (2008). «Analysing Spatial Point Patterns in R». *Perth, CSIRO and University of Western Australia*.
- BADDELEY, A. y TURNER, R. (2005). «Spatstat: An R package for analyzing spatial point patterns». *Journal of statistical software*, **12**, pp. 1–42.
- BADDELEY, A.J. y VAN LIESHOUT, M.N.M. (1995). «Area-interaction point processes». *Annals of the Institute of Statistical Mathematics*, **47(4)**, pp. 601–619.
- BAHGA, ARSHDEEP y MADISETTI, VIJAY (2014). *Internet of Things: A Hands-On Approach*. VPT.
- BAKER, ED (2014). «Open source data logger for low-cost environmental monitoring.». doi: 10.3897/BDJ.2.e1059.
- BARRATT, B.; ATKINSON, R.; ANDERSON, H.R.; BEEVERS, S.; KELLY, F.; MUDWAY, I. y WILKINSON, P. (2007). «Investigation into the use of the CUSUM technique in identifying changes in mean air pollution levels following introduction of a traffic management scheme». *Atmospheric Environment*, **41(8)**, pp. 1784 – 1791. ISSN 1352-2310.
- BEARD, KATE (1996). «A structure for organizing metadata collection». En: *Third International Conference/Workshop on Integrating GIS and Environmental Modelling*, .
- BELTRAN, ARTURO; ABARGUES, CARLOS; GRANELL, CARLOS; HUERTA, JOAQUIN y DIAZ, LAURA (2009). *Description and publication of geospatial information*. INTECH Open Access Publisher.

- BELTRÁN, ARTURO y MARTÍN, CHRISTIAN (2010). «Generando descripciones de recursos para gvSIG desde geocrawler». En: *6th gvSIG User Conference (gvSIG 2010)*. Valencia, Spain, .
- BENGTSSON, J.; NILSSON, S. G.; FRANC, A. y MENOZZI, P. (2000). «Biodiversity, disturbances, ecosystem function and management of European forests». *Forest Ecology and Management*, **132(1)**, pp. 39–50. ISSN 0378-1127. doi: 10.1016/S0378-1127(00)00378-9.
- BERNARD, LARS; KANELLOPOULOS, IOANNIS; ANNONI, ALESSANDRO y SMITS, PAUL (2005). «The European geoportal—one step towards the establishment of a European Spatial Data Infrastructure». *Computers Environment and Urban Systems*, **29(1)**, pp. 15–31. ISSN 01989715.
- BIANCHINI, G. (2006). *Wildland Fire Prediction based on Statistical Analysis of Multiple Solutions*. Tesis doctoral, Universitat Autònoma de Barcelona (Spain).
- BISHOP, I.D.; ESCOBAR, F.J.; KARUPPANNAN, S.; SUWARNARAT, K.; WILLIAMSON, I.P.; YATES, P.M. y YAQUB, H.W. (2000). «Spatial data infrastructures for cities in developing countries: Lessons from the Bangkok experience». *Cities*, **17(2)**, pp. 85–96. doi: 10.1016/S0264-2751(00)00004-4.
- BOTTS, M. y ROBIN, A.. (2007). «OpenGIS Sensor Model Language». *Informe tico*, Open Geospatial Consortium (OGC).
- BOTTS, MIKE; PERCIVALL, GEORGE; REED, CARL y DAVIDSON, JOHN (2008). «OGC® Sensor Web Enablement: Overview and High Level Architecture». En: Silvia Nittel; Alexandros Labrinidis y Anthony Stefanidis (Eds.), *GeoSensor Networks*, volumen 4540 de *Lecture Notes in Computer Science*, pp. 175–190. Springer Berlin Heidelberg. ISBN 978-3-540-79995-5. doi: 10.1007/978-3-540-79996-2_10.
- BRÖRING, ARNE; ECHTERHOFF, JOHANNES; JIRKA, SIMON; SIMONIS, INGO; EVERDING, THOMAS; STASCH, CHRISTOPH; LIANG, STEVE y LEMMENS, ROB (2011). «New Generation Sensor Web Enablement». *Sensors*, **11(12)**, pp. 2652–2699. ISSN 1424-8220.
- BRÖRING, ARNE; REMKE, ALBERT y LASNIA, DAMIAN (2012). *SenseBox - A Generic Sensor Platform for the Web of Things*. volumen 104 de *Lecture Notes of*

- the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 186–196. Springer Berlin Heidelberg. ISBN 978-3-642-30972-4.
- BULTERMAN, DICK CA (2004). «Is it time for a moratorium on metadata?» *IEEE Multimedia*, **11(4)**, pp. 10–17.
- BURATTI, CHIARA; CONTI, ANDREA; DARDARI, DAVIDE y VERDONE, ROBERTO (2009). «An Overview on Wireless Sensor Networks Technology and Evolution». *Sensors*, **9(9)**, pp. 6869–6896. ISSN 1424-8220. doi: 10.3390/s90906869.
- BUSCHMANN, F.; MEUNIER, R.; ROHNERT, H. y SOMMERLAD, P. (1996). «Pattern-oriented software architecture, volume 1: A system of patterns».
- CAMPBELL, T. (2004). «Fostering a Culture of Metadata Production». *GSDI10: Tenth International Conference for Spatial Data Infrastructure*.
- CANNATA, M; MOLINARI, ME; XUAN LUAN, T; HUNG LONG, N y otros (2012). «Web Processing Services for Shallow Landslide». *International Journal of Geoinformatics*, **8(1)**.
- CARSLAW, D; ROPKINS, K y BELL, MC. (2006). «Change-point detection of gaseous and particulate traffic-related pollutants at a roadside location». *Environ Sci Technol.*, **40(22)**, pp. 6912–6918.
- CETINTEMEL, UGUR; ABADI, DANIEL; AHMAD, YANIF; BALAKRISHNAN, HARI; BALAZINSKA, M; CHERNIACK, M; HWANG, J; LINDNER, W; MADDEN, S; MASKEY, A y otros (2006). «The Aurora and Borealis Stream Processing Engines». *Data Stream Management: Processing High-Speed Data Streams*, Springer-Verlag.
- CHARLES, J.C. y JEH-NAN, P. (2002). «Evaluating environmental performance using statistical process control techniques». *European Journal of Operational Research*, **139(1)**, pp. 68 – 83. ISSN 0377-2217.
- CHAS, M.L (2007). «Forest fires in Galicia (Spain): Threats and challenges for the future». *Journal of Forest Economics*, **13(1)**, pp. 1–5. ISSN 1104-6899.
- CHELANI, A. (2011). «Change detection using CUSUM and modified CUSUM method in air pollutant concentrations at traffic site in Delhi». *Stochastic Environmental Research and Risk Assessment*, **25(6)**, pp. 827–834. ISSN 1436-3240.

- CHEN, NENGCHENG; DI, LIPING; YU, GENONG y MIN, MIN (2009). «A flexible geospatial sensor observation service for diverse sensor data based on Web service». {*ISPRS*} *Journal of Photogrammetry and Remote Sensing*, **64(2)**, pp. 234 – 242. ISSN 0924-2716. doi: <http://dx.doi.org/10.1016/j.isprsjprs.2008.12.001>.
- CHUEN-SHENG, C. (1995). «A multi-layer neural network model for detecting changes in the process mean». *Computers & Industrial Engineering*, **28(1)**, pp. 51–61. ISSN 0360-8352.
- CHUVIECO, E. y SALAS, J. (1996). «Mapping the spatial distribution of forest fire danger using GIS». *International journal of geographical information systems*, **10(3)**, pp. 333–345.
- COX, S. (2007). «Observations and Measurements Part 1 - Observation Schema». *Informe tico*, Open Geospatial Consortium (OGC).
- CRAGLIA, MASSIMO; KANELLOPOULOS, IOANNIS y SMITS, PAUL (2007). «Metadata: where we are now, and where we should be going». En: *Proceedings of 10th AGILE International Conference on Geographic Information Science*, .
- CROCKFORD, DOUGLAS (2006). «The application/json Media Type for JavaScript Object Notation (JSON)». *RFC 4627*, IETF.
- CURRIER, BARTON J. ET AL., S. (2004). «Quality assurance for digital learning objects repositories: issues for the metadata creation process». *Research in Learning Technology*.
- CURRY, EDWARD (2004). «Message-Oriented Middleware». En: Qusay H Mahmoud (Ed.), *Middleware for Communications*, caplo 1, pp. 1–28. John Wiley and Sons, Chichester, England. ISBN 978-0-470-86206-3. doi: 10.1002/0470862084.ch1.
- DARBEE, P. (2005). «INSTEON: The Details». *Informe tico*.
- DAVIDOVIC, NIKOLA; MEDVEDEVA, ALEKSANDRA; RANČIĆ, DEJAN y STOIMENOV, LEONID (2013). «ArdSense: extending mobile phone sensing capabilities using open source hardware for new Citizens as Sensors based applications». En: *16th AGILE Conference on Geographic Information Science, Leuven, Belgium*, .

- DE FRANCISCI MORALES, GIANMARCO (2013). «SAMOA: A platform for mining big data streams». En: *Proceedings of the 22nd international conference on World Wide Web companion*, pp. 777–778. International World Wide Web Conferences Steering Committee.
- DE LA BEAUJARDIERE, JEFF (Ed.) (2006). *OpenGIS Web Map Service Implementation Specification, Version 1.3.0*. Open Geospatial Consortium Inc..
- DE MEDIO AMBIENTE, M (2006). «Los incendios forestales en España. decenio 1996-2005». *Ministerio de Medio Ambiente, Tech. Rep.*
- DEAN, JEFFREY y GHEMAWAT, SANJAY (2008). «MapReduce: simplified data processing on large clusters». *Communications of the ACM*, **51(1)**, pp. 107–113.
- DEMUTH, DUSTIN (2012). «A Web of Things integrated Sensor Platform for Precision Agriculture». *Institute for Geoinformatics*.
- DÍAZ, L.; GRANELL, C.; GOULD, M. y HUERTA, J. (2011). «Managing user-generated information in geospatial cyberinfrastructures». *Future Generation Computer Systems*, **27(3)**, pp. 304–314. ISSN 0167-739X. doi: 10.1016/j.future.2010.09.002.
- DÍAZ, LAURA y SCHADE, SVEN (2011). «GEOSS service factory: Assisted publication of geospatial content». En: *Advancing Geoinformation Science for a Changing World*, pp. 423–442. Springer.
- DINES, E.; AL-MAJEED, H.; FERNANDO, A.; ABDALLA, M. y GOHIL, J. (2012). «A new WSN paradigm for environmental monitoring and data collection». En: *Telecommunication Networks and Applications Conference (ATNAC), 2012 Australasian*, pp. 1–6. doi: 10.1109/ATNAC.2012.6398057.
- DIRECTIVE, INSPIRE (2007). «Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE)». *Published in the official Journal on the 25th April*.
- DUBOIS, GREGOIRE; SCHULZ, M; SKØIEN, JON; BASTIN, LUCY y PEEDELL, STEPHEN (2013). «eHabitat, a multi-purpose Web Processing Service for ecological modeling». *Environmental Modelling & Software*, **41**, pp. 123–133.

- ERL, THOMAS (2004). *Service-oriented architecture: a field guide to integrating XML and web services*. Prentice Hall PTR.
- FALUDI, ROBERT (2010). *Building Wireless Sensor Networks: With ZigBee, XBee, Arduino, and Processing*. O Reilly Media, Inc., 1st edici2010. ISBN 0596807732, 9780596807733.
- FERDOUSH, SHEIKH y LI, XINRONG (2014). «Wireless Sensor Network System Design Using Raspberry Pi and Arduino for Environmental Monitoring Applications». *Procedia Computer Science*, **34(0)**, pp. 103–110. ISSN 1877-0509.
- FIELDING, ROY T. y TAYLOR, RICHARD N. (2002). «Principled Design of the Modern Web Architecture». *ACM Trans. Internet Technol.*, **2(2)**, pp. 115–150. ISSN 1533-5399. doi: 10.1145/514183.514185.
- FIELDING, ROY THOMAS (2000). *Architectural styles and the design of network-based software architectures*. Tesis doctoral, University of California, Irvine.
- GADDAM, A; AL-HROOBY, M y ESMAEL, WF (2014). «Designing a Wireless Sensors Network for Monitoring and Predicting Droughts».
- GAMMA, ERICH; HELM, RICHARD; JOHNSON, RALPH y VLISSIDES, JOHN (1994). *Design patterns: elements of reusable object-oriented software*. Pearson Education.
- GARLAND, M.; LE GRAND, S.; NICKOLLS, J.; ANDERSON, J.; HARDWICK, J.; MORTON, S.; PHILLIPS, E.; ZHANG, YAO y VOLKOV, V. (2008). «Parallel Computing Experiences with CUDA». *Micro, IEEE*, **28(4)**, pp. 13–27. ISSN 0272-1732. doi: 10.1109/MM.2008.57.
- GONZÁLEZ, J.R.; PALAHÍ, M.; TRASOBARES, A. y PUKKALA, T. (2006). «A fire probability model for forest stands in Catalonia (north-east Spain)». *Annals of Forest Science*, **63(2)**, p. 8.
- GONZÁLEZ, J.R. y PUKKALA, T. (2007). «Characterization of forest fires in Catalonia (north-east Spain)». *European Journal of Forest Research*, **126**, pp. 421–429. ISSN 1612-4669, 1612-4677. doi: 10.1007/s10342-006-0164-0.

- GONZALEZ-OLABARRIA, J.R. y PUKKALA, T. (2011). «Integrating fire risk considerations in landscape-level forest planning». *Forest Ecology and Management*, **261(2)**, pp. 278–287. ISSN 0378-1127. doi: 10.1016/j.foreco.2010.10.017.
- GOODCHILD, MICHAEL F. (2007). «Citizens as sensors: the world of volunteered geography». *GeoJournal*, **69(4)**, pp. 211–221. ISSN 0343-2521. doi: 10.1007/s10708-007-9111-y.
- GRANELL, C.; DÍAZ, L. y GOULD, M. (2010). «Service-oriented applications for environmental models: Reusable geospatial services». *Environmental Modelling and Software*, **25(2)**, pp. 182–198. doi: 10.1016/j.envsoft.2009.08.005.
- GREGORI, JUAN P. MATEU J. SAURA F., P. y A., CALLE (2006). «Modelos espacio-temporales para la evaluación y caracterización de índices de riesgo de incendios forestales en la Comunidad Valenciana». *Davalos-Fletcher*. ISBN: 84-689-7396-3.
- GRIGG, O.; FAREWELL, V. y SPIEGELHALTER, D. (2003). «Use of risk-adjusted CUSUM and RSPRTcharts for monitoring in medical contexts». *Statistical methods in medical research*, **12(2)**, pp. 147–170.
- GROUP, ANZLIC METADATA WORKING y otros (1996). «ANZLIC Guidelines: Core Metadata Elements Version 1 Report: ANZLIC Working Group on Metadata».
- GUALTIERI, G. y TARTAGLIA, M. (1998). «Predicting urban traffic air pollution: A gis framework». *Transportation Research Part D Transport and Environment*, **3(5)**, pp. 329–336. ISSN 13619209.
- GUBBI, JAYAVARDHANA; BUYYA, RAJKUMAR; MARUSIC, SLAVEN y PALANISWAMI, MARIMUTHU (2013). «Internet of Things (IoT): A vision, architectural elements, and future directions». *Future Generation Computer Systems*, **29(7)**, pp. 1645 – 1660. ISSN 0167-739X. doi: 10.1016/j.future.2013.01.010.
- GUH, R. y HSIEH, Y. (1999). «A neural network based model for abnormal pattern recognition of control charts». *Computers & Industrial Engineering*, **36(1)**, pp. 97–108. ISSN 0360-8352.
- GUINARD, DOMINIQUE; TRIFA, VLAD; MATTERN, FRIEDEMANN y WILDE, ERIK (2011). *From the Internet of Things to the Web of Things: Resource-oriented*

- Architecture and Best Practices*. caplo chapter 5, pp. 97–129. Springer Berlin Heidelberg. ISBN 978-3-642-19156-5.
- GUINARD, DOMINIQUE; TRIFA, VLAD; PHAM, THOMAS y LIECHTI, OLIVIER (2009). «Towards Physical Mashups in the Web of Things». En: *Proceedings of the 6th International Conference on Networked Sensing Systems, INSS'09*, pp. 196–199. IEEE Press, Piscataway, NJ, USA. ISBN 978-1-4244-6313-8.
- HALLIDAY, L (2004). «JORUM Scoping and Technical Appraisal Study, Volume VII: Digital Rights Management».
- HAPNER, M.; BURRIDGE, R.; SHARMA, R.; FIALLI, J. y STOUT, K. (2002). «Java message service». *Sun Microsystems Inc., Santa Clara, CA*.
- HAVENS, STEVE (2007). «OpenGIS Transducer Markup Language (TML) Implementation Specification 1.0. 0». *Wayland, MA, USA: OGC*.
- HEYMANN, Y.; STEENMANS, CH.; CROISSILLE, G. y BOSSARD, M. (1994). «Corine land cover - Technical guide.» *Informe tico*.
- ILLIAN, J.; PENTTINEN, A.; STOYAN, H. y STOYAN, D. (2008). *Statistical Analysis and Modelling of Spatial Point Patterns*. Wiley-Interscience, 1 edici2008. ISBN 0470014911.
- JESKE, D.; MONTES DE OCA, V.; BISCHOFF, W. y MARVASTI, M. (2009). «Cusum techniques for timeslot sequences with applications to network surveillance». *Computational Statistics Data Analysis*, **53(12)**, pp. 4332–4344. ISSN 01679473.
- JI, ZHANLIN; GANCHEV, IVAN; ODRAMA, MAIRTIN; ZHAO, LI y ZHANG, XUEJI (2014). «A Cloud-Based Car Parking Middleware for IoT-Based Smart Cities: Design and Implementation». *Sensors*, **14(12)**, pp. 22372–22393. ISSN 1424-8220. doi: 10.3390/s141222372.
- JORUM (2004). «The JISC Online Repository for [learning and teaching] Materials: JORUM Scoping and Technical Appraisal Study». *Volume V: Metadata*.
- JUAN, P.; MATEU, J. y SAEZ, M. (2012). «Pinpointing spatio-temporal interactions in wildfire patterns.» *Stochastic Environmental Research and Risk Assessment*..

- KIEHLE, C. (2006). «Business logic for geoprocessing of distributed geodata». *Computers & Geosciences*, **32(10)**, pp. 1746 – 1757. doi: 10.1016/j.cageo.2006.04.002.
- KILDOW, M. (1996). «The value of Metadata (An NSDI report). US Fisheries and Wildlife Services». *Web document*. <http://www.r1.fws.gov/metadata/meta.html>.
- KORTUEM, G.; KAWSAR, F.; FITTON, D. y SUNDRAMOORTHY, V. (2010). «Smart objects as building blocks for the Internet of things». *Internet Computing, IEEE*, **14(1)**, pp. 44–51. ISSN 1089-7801. doi: 10.1109/MIC.2009.143.
- KUMAR, SIMPAL (2014). *Real Time Data Analysis for Water Distribution Network using Storm*. Tesis doctoral, University of Fribourg.
- LAUTENBACHER, CONRAD C (2006). «The global earth observation system of systems: science serving society». *Space Policy*, **22(1)**, pp. 8–11.
- LI, SHENG-TUN y SHUE, LI-YEN (2004). «Data mining to aid policy making in air pollution management». *Expert Systems with Applications*, **27(3)**, pp. 331 – 340. ISSN 0957-4174.
- LIU, DONGGANG y NING, PENG (2003). «Establishing Pairwise Keys in Distributed Sensor Networks». En: *Proceedings of the 10th ACM Conference on Computer and Communications Security, CCS '03*, pp. 52–61. ACM, New York, NY, USA. ISBN 1-58113-738-9. doi: 10.1145/948109.948119.
- LLORET, F.; CALVO, E.; PONS, X. y DÍAZ-DELGADO, R. (2003). «Wildfires and landscape patterns in the Eastern Iberian Peninsula». *Ecological Research*, **17(8)**, pp. 745–759.
- LUCAS, J.M. (1982). «Combined Shewhart-CUSUM quality control schemes». *Journal of Quality Technology*, **14(2)**.
- MANOVICH, L. (2012). «Trending: The Promises and the Challenges of Big Social Data.» En: Matthew K. Gold (Ed.), *Debates in the Digital Humanities*, pp. 460–75. U of Minnesota P, Minneapolis.
- MANSO, MA; NOGUERAS-ISO, J; BERNABE, MA y ZARAZAGA-SORIA, FJ (2004). «Automatic metadata extraction from geographic information». En: *7th Confe-*

- rence on Geographic Information Science (AGILE 2004), Heraklion, Greece, pp. 379–385.
- MAREY, M.F.; RODRÍGUEZ, V. y CRECENTE, R. (2006). «Using GIS to measure changes in the temporal and spatial dynamics of forestland: experiences from north-west Spain». *Forestry*, **79(4)**, pp. 409–423.
- MASSER, I.; RAJABIFARD, A. y WILLIAMSON, I. (2008). «Spatially enabling governments through SDI implementation». *International Journal of Geographical Information Science*, **22(1)**, pp. 5–20. doi: 10.1080/13658810601177751.
- MASSER, IAN (2005). *Redlands*: ESRI Press. ISBN 978-1589481220.
- MAYER-SCHÖNBERGER, VIKTOR y CUKIER, KENNETH (2013). *Big data: A revolution that will transform how we live, work, and think*. Houghton Mifflin Harcourt.
- MCAFEE, ANDREW y BRYNJOLFSSON, ERIK (2012). «Big data: the management revolution.» *Harvard business review*, (**90**), pp. 60–6.
- MCINERNEY, D.; BASTIN, L.; DIAZ, L.; FIGUEIREDO, C.; BARREDO, J.I. y SANMIGUEL AYANZ, J. (2012). «Developing a Forest Data Portal to Support Multi-Scale Decision Making». *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, **5(1)**, pp. 1–8. doi: 10.1109/JSTARS.2012.2194136.
- MEIN, GUNNAR; PAL, SHANKAR; DHONDU, GOVINDA; ANAND, THULUSALAMATOM KRISHNAMURTHI; STOJANOVIC, ALEXANDER; AL-GHOSEIN, MOHSEN y OEUVRAY, PAUL M (2002). «Simple object access protocol». US Patent 6,457,066.
- MENDEZ, D.; PEREZ, A.J.; LABRADOR, M.A. y MARRON, J.J. (2011). «P-Sense: A participatory sensing system for air pollution monitoring and control». En: *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, pp. 344–347.
- MERMOZ, M.; KITZBERGER, T. y VELEN, T. T. (2005). «Landscape influences on occurrence and spread of wildfires in patagonian forests and shrublands». *Ecology*, **86**, pp. 2705–2715. ISSN 0012-9658. doi: 10.1890/04-1850.

- MESNIL, B. y P., PETITGAS (2009). «Detection of changes in time-series of indicators using CUSUM control charts». *Aquatic Living Resources*, **22(02)**, pp. 187 – 192.
- MICHAELIS, CHRISTOPHER D. y AMES, DANIEL P. (2009). «Evaluation and Implementation of the OGC Web Processing Service for Use in Client-Side GIS». *Geoinformatica*, **13(1)**, pp. 109–120. doi: 10.1007/s10707-008-0048-1.
- MINELLI, MICHAEL; CHAMBERS, MICHELE y DHIRAJ, AMBIGA (2012). *Big data, big analytics: emerging business intelligence and analytic trends for today's businesses*. John Wiley & Sons.
- MIZOUNI, RABEB; SERHANI, M. ADEL; DSSOULI, RACHIDA; BENHARREF, ABDELGHANI y TALEB, IKBAL (2011). «Performance Evaluation of Mobile Web Services». pp. 184–191. IEEE. ISBN 978-1-4577-1532-7.
- MOLINA-TERRÉN, D.; MARTÍNEZ-LÓPEZ, E. y GARCIA-MARCO, D. (2006). «Farsite simulations for cost-efficient wildland fire planning: Case studies in Spain». *Forest Ecology and Management*, **234(1)**, pp. 217–217. doi: 10.1016/j.foreco.2006.08.244.
- MYKKÄNEN, JUHA A y TUOMAINEN, MIKA P (2008). «An evaluation and selection framework for interoperability standards». *Information and Software Technology*, **50(3)**, pp. 176–197.
- NANDAN, NAVEEN (2013). «Live Analytics on High Velocity Sensor Data Streams using Event-based Systems». *Journal of Industrial and Intelligent Information Vol*, **1(1)**.
- NATIVI, S.; CRAGLIA, M. y PEARLMAN, J. (2012). «The brokering approach for multidisciplinary interoperability: A position paper». *International Journal of Spatial Data Infrastructures Research*, **7**, pp. 1–15.
- NEBERT, WHITESIDE A., D. (2004). «OpenGIS catalogue services specification (version 2.0)». *OpenGIS Project Document 04-021r2*, Open GIS Consortium Inc.
- NOGUERAS-ISO, JAVIER; ZARAZAGA-SORIA, F JAVIER y MURO-MEDRANO, PEDRO R (2005). «Geographic information metadata for spatial data infrastructures». *Resources, Interoperability and Information Retrieval*.

- NUNES, M. C.S.; VASCONCELOS, M. J.; PEREIRA, J. M.C.; DASGUPTA, N.; ALLDREDGE, R. J. y REGO, F. C. (2005). «Land Cover Type and Fire in Portugal: Do Fires Burn Land Cover Selectively?» *Landscape Ecology*, **20**, pp. 661–673. ISSN 0921-2973, 1572-9761. doi: 10.1007/s10980-005-0070-8.
- ONEILL, ROBERT V.; HUNSAKER, CAROLYN T.; JONES, K. BRUCE; RIITTERS, KURT H.; WICKHAM, JAMES D.; SCHWARTZ, PAUL M.; GOODMAN, IRIS A.; JACKSON, BARBARA L. y BAILLARGEON, WILLIAM S (1997). «Monitoring environmental quality at the landscape scale». *BioScience*, **47(8)**, pp. 44–51. ISSN 513-519. doi: 10.1109/MIC.2009.143.
- OSANAIYE, P. y TALABI, C. (1989). «On some non-manufacturing applications of counted data cumulative sum (CUSUM) control chart schemes». *The Statistician*, pp. 251–257.
- PAGE, E.S. (1954). «Continuous Inspection Schemes». *Biometrika*, **41(1/2)**, pp. 100–115.
- PALAHÍ, M.; MAVSAR, R.; GRACIA, C. y BIROT, Y. (2008). «Mediterranean forests under focus». *International Forestry Review*, **10(4)**, pp. 676–688. doi: 10.1505/ifor.10.4.676.
- PAUTASSO, CESARE y WILDE, ERIK (2010). «RESTful web services, principles, patterns, emerging technologies». p. 1359. ACM Press. ISBN 9781605587998.
- PEBESMA, EDZER; CORNFORD, DAN; DUBOIS, GREGOIRE; HEUVELINK, GERRARD B.M.; HRISTOPULOS, DIONISIS; PILZ, JÜRGEN; STÖHLKER, ULRICH; MORIN, GARY y SKØIEN, JON O. (2011). «INTAMAP: The design and implementation of an interoperable automated interpolation web service». *Computers & Geosciences*, **37(3)**, pp. 343–352. ISSN 00983004.
- PERCIVAL, G. (2008). «Open Geospatial Consortium Inc. Date: 2008-11-11. Reference number: OGC 08-062r4. Version: 2.0». *Informe tico*, OGC Reference Model.
- POORAZIZI, EBRAHIM; ALESHEIKH, ALI ASGHAR y AWT-TAG (2011). «Monitoring Real-Time Environmental Information Using Web 2.0 and GIServices Technology». *International Journal of Civil Engineering*, **9**.

- POORAZIZI, MOHAMMAD EBRAHIM; LIANG, STEVE H. L. y HUNTER, ANDREW J. S. (2012). «Testing of sensor observation services, a performance evaluation». pp. 32–38. ACM Press. ISBN 9781450317016.
- PUCCINELLI, D. y HAENGGI, M. (2005). «Wireless sensor networks: applications and challenges of ubiquitous sensing». *Circuits and Systems Magazine, IEEE*, **5(3)**, pp. 19–31. ISSN 1531-636X. doi: 10.1109/MCAS.2005.1507522.
- RAJABIFARD, A.; FEENEY, M-E. F. y WILLIAMSON, I. P. (2002). «Future directions for SDI development». *International Journal of Applied Earth Observation and Geoinformation*, **4(1)**, pp. 11–22. doi: 10.1016/S0303-2434(02)00002-8.
- RIPLEY, B.D. (1976). «The second-order analysis of stationary point processes.» *Journal of Applied Probability*, pp. 13, 256–266.
- RODRIGUES, JOEL J. P. C. y NEVES, PAULO A. C. S. (2010). «A survey on IP-based wireless sensor network solutions». *International Journal of Communication Systems*, **23(8)**, pp. 963–981. ISSN 1099-1131. doi: 10.1002/dac.1099.
- ROTHERMEL, R.C. (1983). *How to predict the spread and intensity of forest and range fires*. Intermountain Forest and Range Experiment Station Ogden.
- ROUACHED, MOHSEN; BACCAR, SANA y ABID, MOHAMED (2012). «RESTful Sensor Web Enablement Services for Wireless Sensor Networks». pp. 65–72. IEEE. ISBN 978-1-4673-3053-4.
- SAKR, G. E.; ELHAJJ, I. H. y MITRI, G. (2011). «Efficient forest fire occurrence prediction for developing countries using two weather parameters». *Engineering Applications of Artificial Intelligence*, **24**, pp. 888–894. ISSN 09521976. doi: 10.1016/j.engappai.2011.02.017.
- SAN-MIGUEL-AYANZ, J.; BARBOSA, P.; SCHMUCK, G.; LIBERTÁ, G. y MEYER-ROUX, J. (2003). «AGILE 2003: 6th AGILE Conference on Geographic Information Science». PPUR presses polytechniques, Lyon, France. ISBN 9782880745417.
- SCHMIDT, DOUGLAS C; STAL, MICHAEL; ROHNERT, HANS y BUSCHMANN, FRANK (2013). *Pattern-Oriented Software Architecture, Patterns for Concurrent and Networked Objects*. volumen 2. John Wiley & Sons.

- SCHUT, P. (2008). «Opengis web processing service version 1.0.0.» *Informe tico*, Open Geospatial Consortium (OGC).
- SERRA, LAURA; JUAN, PABLO; VARGA, DIEGO; MATEU, JORGE y SAEZ, MARC (2013). «Spatial pattern modelling of wildfires in Catalonia, Spain 2004–2008». *Environmental Modelling & Software*, **40**, pp. 235–244.
- SHAKESBY, R.A. (2011). «Post-wildfire soil erosion in the Mediterranean: Review and future research directions». *Earth-Science Reviews*, **105(3-4)**, pp. 71–100. ISSN 0012-8252. doi: 16/j.earscirev.2011.01.001.
- SHETH, AMIT; HENSON, CORY y SAHOO, SATYA S. (2008). «Semantic Sensor Web». *IEEE Internet Computing*, **12(4)**, pp. 78–83. ISSN 1089-7801.
- SILVA DE LA CRUZ, ANAYANSI DA (2013). «Una aproximación MDA para la conversión entre servicios web SOAP y RESTful».
- SIMONCELLI, D.; DUSI, M.; GRINGOLI, R. y NICCOLINI, S. (2013). «Stream-monitoring with Blockmon: Convergence of Network Measurements and Data Analytics Platforms». *SIGCOMM Comput. Commun. Rev.*, **43(2)**, pp. 29–36. ISSN 0146-4833.
- SIMONIS, I (2007). «OGC Implementation Specification 07-014r3: OpenGIS Sensor Planning Service». *Open Geospatial Consortium: Wayland, MA, USA*.
- SIMONIS, INGO (2006). «OGC Best Practices 06-028r3: OGC Sensor Alert Service Candidate Implementation Specification». *Open Geospatial Consortium, Tech. Rep.*
- SIRISH, CHANDRASEKARAN; OWEN, C; AMOL, D; WEI, H; SAILESH, K; SAMUEL, M; VIJAYSHANKAR, R y FREDERICK, R (2003). «TelegraphCQ: Continuous dataflow processing for an uncertain world». *CIDR'03*.
- SITARAM, D.; SRINIVASARAGHAVAN, H.; AGARWAL, K.; AGRAWAL, A.; JOSHI, N. y RAY, D. (2013). «Pipelining Acoustic Model Training for Speech Recognition Using Storm». En: *Computational Intelligence, Modelling and Simulation (CIMSIm), 2013 Fifth International Conference on*, pp. 219–224.

- SOHRABY, KAZEM; MINOLI, DANIEL y ZNATI, TAIEB (2007). *Wireless Sensor Networks: Technology, Protocols, and Applications*. Wiley-Interscience. ISBN 0471743003.
- SUNDERRAJAN, A.; AYDT, H. y KNOLL, A. (2014). «Real Time Load Prediction and Outliers Detection Using STORM». En: *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems, DEBS '14*, pp. 294–297. ACM, New York, NY, USA. ISBN 978-1-4503-2737-4.
- TAMAYO, A.; GRANELL, C.; DIAZ, L. y HUERTA, J. (2012a). «Building Compact Standard-Based Geoprocessing Mobile Clients». *Submitted to AGILE 2012*.
- TAMAYO, ALAIN; GRANELL, CARLOS; DÍAZ, LAURA y HUERTA, JOAQUÍN (2012b). «Building standards-based geoprocessing mobile clients». En: *Proceedings of the 15th AGILE international conference on geographic information science (AGILE 2012)*. Avignon, France, .
- TAMAYO, ALAIN; VICIANO, PABLO; GRANELL, CARLOS y HUERTA, JOAQUIN (2011). *Empirical Study of Sensor Observation Services Server Instances*. volumen 1 de *Lecture Notes in Geoinformation and Cartography*, caplo chapter 10, pp. 185–209. Springer Berlin Heidelberg. ISBN 978-3-642-19788-8.
- TAUSSI, MATTI (2007). «Automatic production of metadata out of geographic datasets». *University of Technology*.
- TEENSMA, P.D.A. (1987). *Fire history and fire regimes of the Central Western Cascades of Oregon*. Tesis doctoral, University of Oregon, Eugene.
- THIEBES, BENNI; BELL, RAINER; GLADE, THOMAS; JÄGER, STEFAN; ANDERSON, MALCOLM y HOLCOMBE, LIZ (2013). «A WebGIS decision-support system for slope stability based on limit-equilibrium modelling». *Engineering Geology*, **158**, pp. 109–118.
- TOSHNIWAL, A.; TANEJA, S.; SHUKLA, A.; RAMASAMY, K.; PATEL, J.M.; KULKARNI, S.; JACKSON, J.; GADE, K.; FU, M.; DONHAM, J.; BHAGAT, N.; MITTAL, S. y RYABOY, D. (2014). «Storm@Twitter». En: *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, SIGMOD '14*, pp. 147–156. ACM, New York, NY, USA.

- UCKELMANN, DIETER; HARRISON, MARK y MICHAELLES, FLORIAN (2011). *An Architectural Approach Towards the Future Internet of Things*. pp. 1–24. Springer Berlin Heidelberg. ISBN 978-3-642-19156-5.
- VAN WAGNER, C.E. (1977). «Conditions for the start and spread of crown fire». *Canadian Journal of Forest Research*, **7(1)**, pp. 23–34.
- VÁZQUEZ, A. (2001). «Spatial distribution of forest fires in Sierra de Gredos (Central Spain)». *Forest Ecology and Management*, **147(1)**, pp. 55–65. doi: 10.1016/S0378-1127(00)00436-9.
- VELEZ, RICARDO MUNOZ y otros (2000). *La defensa contra incendios forestales: fundamentos y experiencias*. McGraw-Hill Madrid.
- VIQUEIRA, JOSÉ R. R.; VARELA, JOSÉ; TRIÑANES, JOAQUÍN y COTOS, JOSÉ M. (2010). *A Sensor Observation Service Based on OGC Specifications for a Meteorological SDI in Galicia*. volumen 6413 de *Lecture Notes in Computer Science*, caplo chapter 6, pp. 43–52. Springer Berlin Heidelberg. ISBN 978-3-642-16384-5.
- VOIGT, THIEMO; RITTER, HARTMUT y SCHILLER, JOCHEN (2003). «Solar-Aware Routing in Wireless Sensor Networks». En: Marco Conti; Silvia Giordano; Enrico Gregori y Stephan Olariu (Eds.), *Personal Wireless Communications*, volumen 2775 de *Lecture Notes in Computer Science*, pp. 847–852. Springer Berlin Heidelberg. ISBN 978-3-540-20123-6. doi: 10.1007/978-3-540-39867-7_79.
- VRETANOS, PANAGIOTIS A (2005). «Web feature service implementation specification». *Open Geospatial Consortium Specification*, pp. 04–094.
- WHITE, MICHAEL A; HOFFMAN, FORREST; HARGROVE, WILLIAM W y NEMANI, RAMAKRISHNA R (2005). «A global framework for monitoring phenological responses to climate change». *Geophysical Research Letters*, **32(4)**.
- WIDOM, B. (1970). «New Model for the Study of Liquid?Vapor Phase Transitions». *The Journal of Chemical Physics*, **52**, p. 1670. ISSN 00219606. doi: 10.1063/1.1673203.
- YANG, C.; LI, W.; XIE, J. y ZHOU, B (2008). «Distributed geospatial information processing: sharing distributed geospatial resources to support Di-

- gital Earth». *International Journal of Digital Earth*, **1(3)**, pp. 259–278. doi: 10.1080/17538940802037954.
- YAZAR, DOGAN y DUNKELS, ADAM (2009). «Efficient Application Integration in IP-based Sensor Networks». En: *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, BuildSys '09, pp. 43–48. ACM, New York, NY, USA. ISBN 978-1-60558-824-7. doi: 10.1145/1810279.1810289.
- YUTAN, D.; JUN, L.; FANG, L. y LUYING, C. (2014). «A Real-Time Anomalies Detection System Based on Streaming Technology». En: *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2014 Sixth International Conference on*, volumen 2, pp. 275–279.
- Z-WAVE (2007). «Z-Wave Protocol Overview, revision 4». *Informe tico*.
- ZARAZAGA-SORIA, F JAVIER; LACASTA, JAVIER; NOGUERAS-ISO, JAVIER; TORRES, M PILAR y MURO-MEDRANO, PR (2003). «A Java tool for creating ISO/FGDC geographic metadata». *Geodaten-und Geodienste-Infrastrukturen-von der Forschung zur praktischen Anwendung. Beiträge zu den Münsteraner GI-Tagen*, **26**, p. 27.
- ZORZI, M.; GLUHAK, A.; LANGE, S. y BASSI, A. (2010). «From today's INTRAnet of things to a future INTERnet of things: a wireless- and mobility-related view». *Wireless Communications, IEEE*, **17(6)**, pp. 44–51. ISSN 1536-1284. doi: 10.1109/MWC.2010.5675777.