



VISUALIZACIÓN AVANZADA DE DATOS DE OBSERVACIÓN MEDIOAMBIENTAL CON TECNOLOGÍAS SEMÁNTICAS

Trabajo Fin de Máster

**Máster Universitario de Investigación en
Ingeniería de Software y Sistemas Informáticos**

Universidad Nacional de Educación a Distancia

Itinerario: I - Ingeniería de Software

Código asignatura: 105128

Curso académico: 2016-2017

Convocatoria: Septiembre - Ordinaria

Autor: F. Rubén Arenas Hernán

**Directores: Dra. Elena Ruiz Larrocha
Dr. José Ramón Ríos Viqueira**

VISUALIZACIÓN AVANZADA DE DATOS DE OBSERVACIÓN MEDIOAMBIENTAL CON TECNOLOGÍAS SEMÁNTICAS

Trabajo Fin de Máster

**Máster Universitario de Investigación en
Ingeniería de Software y Sistemas Informáticos**

Universidad Nacional de Educación a Distancia

Itinerario: I - Ingeniería de Software

Código asignatura: 105128

Curso académico: 2016-2017

Convocatoria: Septiembre - Ordinaria

Autor: F. Rubén Arenas Hernán

Directores: Dra. Elena Ruiz Larrocha
Dr. José Ramón Ríos Viqueira

HOJA DE CALIFICACIONES

DECLARACIÓN JURADA DE AUTORÍA DEL TRABAJO CIENTÍFICO, PARA
LA DEFENSA DEL TRABAJO FIN DE MASTER

Fecha: 19/09/2017

Quién suscribe

Autor(a): F. Rubén Arenas Hernán

D.N.I./N.I.E./Pasaporte.: 53117926R

Hace constar que es la autor(a) del trabajo:

Visualización Avanzada de datos de observación medioambiental con tecnologías
semánticas

En tal sentido, manifiesto la originalidad de la conceptualización del trabajo, interpretación de datos y la elaboración de las conclusiones, dejando establecido que aquellos aportes intelectuales de otros autores, se han referenciado debidamente en el texto de dicho trabajo.

DECLARACIÓN:

Garantizo que el trabajo que remito es un documento original y no ha sido publicado, total ni parcialmente por otros autores, en soporte papel ni en formato digital.

Certifico que he contribuido directamente al contenido intelectual de este manuscrito, a la génesis y análisis de sus datos, por lo cual estoy en condiciones de hacerme públicamente responsable de él.

No he incurrido en fraude científico, plagio o vicios de autoría; en caso contrario, aceptaré las medidas disciplinarias sancionadoras que correspondan.

AUTORIZACIÓN DE PUBLICACIÓN CON FINES ACADÉMICOS

Autorizo a la Universidad Nacional de Educación a Distancia a difundir y utilizar, con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la memoria de este Trabajo Fin de Máster, como el código, la documentación y/o el prototipo desarrollado.

Firma del Autor



RESUMEN

El proyecto presentado en este documento se corresponde con una de las fases de implantación de tecnologías semánticas basadas en servicios web de recogida, almacenamiento y visualización de datos ambientales de carácter público de en dos entidades públicas de la comunidad autónoma de Galicia. Esta memoria contempla el trabajo de investigación, desarrollo e implantación de un cliente web SOS V2.0 puro y tareas de migración de adaptadores virtuales de fuentes de datos del servidor semántico SOS V1.0 al estándar SOS V2.0. Estas herramientas han sido desarrolladas para las agencias meteorológicas MeteoGalicia e INTECMAR, como medida de adaptación de sistemas de información a la inminente entrada en vigor de la directiva europea INSPIRE. El objetivo es estandarizar y ofrecer a la ciudadanía el acceso unificado a los datos medioambientales que han sido recogidos y/o generados mediante financiación pública.

PALABRAS CLAVE

SOS, SOS Api, OGC, Inspire, SDI, OpenData, O&M, SensorML, SWE, SWE Client, Sensor web, AOS, Integración semántica, datos ambientales, datos meteorológicos, Servicios web, USC, COGRADE, MeteoGalicia, INTECMAR

ÍNDICE DE CONTENIDOS

Capítulo 1. Introducción	15
1.1 Contextualización	16
1.2 Motivación	27
1.3 Objetivos del proyecto	30
1.4 Estructura del documento.....	31
Capítulo 2. Estado del arte	35
2.1 Arquitecturas para sistemas de integración de datos.....	35
2.2 Arquitectura en tres capas.....	38
2.3 Infraestructura de datos espaciales o IDE	40
2.4 Directiva INSPIRE	42
2.5 Open Geospatial Consortium	47
2.6 Open Data.....	51
2.7 Sensor Web Enablement (SWE).....	53
2.8 Sensor Observation Service o SOS.....	56
2.8.1 Metadatos SOS.....	57
2.8.2 Diferencias entre SOS V1.0 y SOS V2.0	58
2.8.3 Conceptos fundamentales de SOS 2.0	60
2.8.3.1 Offering.....	61
2.8.3.2 Procedure	62
2.8.3.3 Feature of Interest.....	62
2.8.3.4 Observed Property	63
2.8.3.5 Observation	63
2.8.3.6 Operaciones obligatorias de SOS	65
2.9 Arquitecturas de clientes SOS.....	69
2.9.1 Integración Directa con servidores SOS.....	70
2.9.2 Componente intermedio que realiza la lógica de negocio.....	71

2.10 52ºNorth.....	72
2.10.1 52ºNorth SOS 2.0	73
2.11 Herramientas empleadas	77
Capítulo 3. Gestión del proyecto	83
3.1 Análisis y gestión de riesgos.....	83
3.1.1 Riesgos de gestión del proyecto.....	86
3.1.2 Riesgos técnicos.....	90
3.1.3 Riesgos de personal	94
3.2 Planificación	97
3.2.1 Gestión del alcance (EDT)	98
3.2.2 Metodología de desarrollo	101
3.2.3 Metodología de diseño.....	104
3.2.4 Gestión del tiempo (Gantt).....	105
3.2.4.1 Calendario previsto.....	106
3.2.4.2 Calendario real	109
3.3 Gestión de costes.....	112
3.3.1 Costes de material.....	112
3.3.2 Costes de recursos humanos.....	113
3.3.3 Coste total	114
3.4 Gestión de la configuración.....	115
3.4.1 Control de código fuente	115
3.4.2 Control de la documentación	117
3.4.3 Gestión de los cambios	118
3.5 Sistema de seguimiento	120
Capítulo 4. Análisis del proyecto	123
4.1 Casos de uso.....	125
4.2 Catálogo de requisitos	131
4.2.1 Requisitos del proyecto	134
4.2.2 Requisitos funcionales	139

4.2.3 Requisitos de calidad	147
4.2.4 Requisitos de evolución	149
4.2.5 Requisitos de diseño	150
4.2.6 Requisitos de soporte	152
4.2.7 Requisitos de interfaz gráfica	154
4.2.8 Requisitos emergentes	158
4.3 Matriz de trazabilidad.....	161
Capítulo 5. Diseño del proyecto.....	163
5.1 Diseño de la Arquitectura del Sistema.....	164
5.2 Diseño de la Arquitectura de la Aplicación	166
5.2.1 Diagrama de secuencia	167
5.2.2 Diagrama de actividades	168
5.3 Diseño de la interfaz gráfica de usuario	171
Capítulo 6. Implementación del proyecto	177
6.1 Incremento 1	177
6.2 Incremento 2	185
6.3 Incremento 3	189
Capítulo 7. Pruebas	193
7.1 Verificación	194
7.2 Validación.....	198
Capítulo 8. Conclusiones.....	203
8.1 Conocimientos y habilidades adquiridas	205
8.2 Trabajo futuro.....	207
Capítulo 9. Referencias	209

LISTA NUMERADA DE FIGURAS

Fig. 1 Dispositivos de observación meteorológica	17
Fig. 2 Logotipo corporativo de MeteoGalicia.....	21
Fig. 3 Logotipo corporativo de INTECMAR.....	22
Fig. 4 Logotipo de La Universidad de Santiago de Compostela.....	22
Fig. 5 Logotipo de CiTIUS	23
Fig. 6 Logotipo de COGRADE	24
Fig. 7 Arquitectura del SOS semántico v1.0.....	26
Fig. 8 Arquitectura de componentes, interfaces y APIs del servicio 52ºN SOS....	28
Fig. 9 Esquema general de integración de datos.....	36
Fig. 10 Los tres tipos de sistemas de integración.....	37
Fig. 11 Arquitectura en tres capas	40
Fig. 12 Esquema representativo de las acciones y entes involucrados en las IDEs	42
Fig. 13 Logotipo INSPIRE	43
Fig. 14 Calendario directiva INSPIRE	44
Fig. 15 Arquitectura propuesta por INSPIRE para IDEs [28]	46
Fig. 16 Logotipo de OGC	51
Fig. 17 Resumen de SWE.....	54
Fig. 18 Elementos de SWE y acciones definidas entre los mismos.....	56
Fig. 19 Conceptos relativos a datos de observaciones en SOS	61
Fig. 20 Modelo de observaciones básico de O&M.....	64
Fig. 21 Arquitectura de cliente SOS de Integración Directa con servidores SOS..	70
Fig. 22 Arquitectura de cliente SOS con componente intermedio de lógica de negocio	71
Fig. 23 Diagrama del proceso de gestión de riesgos.....	83
Fig. 24 Estructura de Descomposición de Trabajo del proyecto	99
Fig. 25 Ciclo de vida iterativo incremental.....	103
Fig. 26 Diagrama de Gantt planificado.....	107
Fig. 27 Diagrama de recursos humanos planificado	108

Fig. 28 Diagrama de Gantt seguido.....	110
Fig. 29 Diagrama de recursos humanos seguido	111
Fig. 30 Herramienta GitLab empleada para el control de versiones de código ...	116
Fig. 31 Herramienta GitLab empleada para el control de versiones de la documentación.....	118
Fig. 32 Notación de los diagramas de casos de uso UML	126
Fig. 33 Diagrama de casos de uso del presente proyecto	127
Fig. 34 Arquitectura completa del sistema SOS V2.0 semántico	165
Fig. 35 Estructura jerárquica de los módulos de la aplicación	167
Fig. 36 Diagrama de Secuencia de petición de observaciones	168
Fig. 37 Diagrama de actividad para el proceso de descarga de datos de observaciones.....	169
Fig. 38 Diagrama de actividad para el proceso de obtención de datos de observaciones del servidor SOS V2.0.....	170
Fig. 39 Mockup de la interfaz inicial desde la que se construye la consulta de observaciones.....	172
Fig. 40 Mockup de la herramienta de visualización de datos de observación en tablas.....	173
Fig. 41 Mockup de la herramienta de visualización de datos de observación en gráficas	174
Fig. 42 Mockup de la herramienta de visualización de datos de observación en mapas	175
Fig. 43 Interfaz de formulario de consulta de datos de observación	179
Fig. 44 Ejemplo de dato de observación parseado y almacenado en caché del navegador	184
Fig. 45 Ejemplo de observaciones visualizadas con la herramienta de tablas.....	186
Fig. 46 Herramienta de visualización de datos mediante gráfica	187
Fig. 47 Ejemplo de gráfica con datos de series de profundidad para intervalo de tiempo de 1 año.....	188
Fig. 48 Menú desplegable de la herramienta de gráficas que permite descargar los datos generados	190
Fig. 49 Herramienta de visualización de datos de observaciones en mapa.....	191
Fig. 50 Página de inicio de Apache Tomcat	226
Fig. 51 Gestor de aplicaciones de Tomcat	227

Fig. 52 Ventana inicial del cliente SOS V2.0	230
Fig. 53 Estado del cliente SOS V2.0 una vez ha recibido la respuesta de GetCapabilities.....	231
Fig. 54 Estado del cliente SOS V2.0 esperando la selección de ofertas.	232
Fig. 55 Cliente SOS esperando la respuesta de la consulta GetFeatureOfInterest	233
Fig. 56 Cliente SOS con las FOIs asociadas a las Offerings seleccionadas.....	234
Fig. 57 Cliente SOS con FOIs seleccionadas.....	235
Fig. 58 Panel de detalles de FOI	235
Fig. 59 Panel de filtro temporal activo	236
Fig. 60 Cliente SOS V2.0 ejecutando la consulta GetObservation.....	237
Fig. 61 Datos crudos de observaciones en formato Tabla	238
Fig. 62 Herramienta de graficación de datos de observación sin aplicar ningún filtro	239
Fig. 63 Herramienta de gráficas esperando por la selección del parámetro a representar en el eje X	240
Fig. 64 Opciones de resolución para el muestreo de datos de profundidad en la herramienta de graficación	241
Fig. 65 Opciones de resolución introducidas para el muestreo de datos de profundidad en la herramienta de graficación.....	241
Fig. 66 gráfica generada según los filtros y resoluciones que el usuario ha introducido	242
Fig. 67 Herramienta de graficación con diferentes gráficas añadidas al diagrama	243
Fig. 68 Menú de exportación de datos procesados de la herramienta de gráficas	244
Fig. 69 Herramienta de representación de datos de observaciones en formato mapa	245

LISTA NUMERADA DE TABLAS

Tabla 1 Resumen de operaciones y bindings de 52ºNorth SOS	74
Tabla 2 Bindings empleados actualmente en 52ºNorth SOS server	75
Tabla 3 Elementos de la arquitectura de 52ºNorth SOS	76
Tabla 4 Guía para el cálculo de exposición a riesgos	85
Tabla 5 Plantilla empleada en el análisis y gestión de los riesgos	86
Tabla 6 Riesgo de gestión del proyecto RGP-1	87
Tabla 7 Riesgo de gestión del proyecto RGP-2	88
Tabla 8 Riesgo de gestión del proyecto RGP-3	89
Tabla 9 Riesgo de gestión del proyecto RGP-4	89
Tabla 10 Riesgo técnico RT-1	91
Tabla 11 Riesgo técnico RT-2	92
Tabla 12 Riesgo técnico RT-3	92
Tabla 13 Riesgo técnico RT-4	93
Tabla 14 Riesgo de personal RP-1.....	94
Tabla 15 Riesgo de personal RP-2.....	95
Tabla 16 Riesgo de personal RP-3.....	96
Tabla 17 Riesgo de personal RP-4.....	97
Tabla 18 Desglose de los costes de material	112
Tabla 19 Desglose de los costes de recursos humanos.....	113
Tabla 20 Desglose de los costes totales.....	114
Tabla 21 Detalles del caso de uso CU-01.....	128
Tabla 22 Detalles del caso de uso CU-02.....	130
Tabla 23 Detalles del caso de uso CU-03.....	130
Tabla 24 Identificadores de categorías de requisitos.....	132
Tabla 25 Plantilla de descripción de requisitos.....	133
Tabla 26 Requisito del proyecto RQ-PR-01	134
Tabla 27 Requisito del proyecto RQ-PR-02	135
Tabla 28 Requisito del proyecto RQ-PR-03	135
Tabla 29 Requisito del proyecto RQ-PR-04	136

Tabla 30 Requisito del proyecto RQ-PR-05	136
Tabla 31 Requisito del proyecto RQ-PR-06	137
Tabla 32 Requisito del proyecto RQ-PR-07	137
Tabla 33 Requisito del proyecto RQ-FN-01	139
Tabla 34 Requisito del proyecto RQ-FN-02	140
Tabla 35 Requisito del proyecto RQ-FN-03	140
Tabla 36 Requisito del proyecto RQ-FN-04	141
Tabla 37 Requisito del proyecto RQ-FN-05	141
Tabla 38 Requisito del proyecto RQ-FN-06	142
Tabla 39 Requisito del proyecto RQ-FN-07	143
Tabla 40 Requisito del proyecto RQ-FN-08	143
Tabla 41 Requisito del proyecto RQ-FN-09	144
Tabla 42 Requisito del proyecto RQ-FN-10	145
Tabla 43 Requisito del proyecto RQ-FN-11	145
Tabla 44 Requisito del proyecto RQ-FN-12	146
Tabla 45 Requisito del proyecto RQ-FN-13	146
Tabla 46 Requisito de calidad RQ-CA-01.....	147
Tabla 47 Requisito de calidad RQ-CA-02.....	148
Tabla 48 Requisito de calidad RQ-CA-03.....	148
Tabla 49 Requisito de calidad RQ-EV-01	149
Tabla 50 Requisito de calidad RQ-EV-02	150
Tabla 51 Requisito de diseño RQ-DI-01	150
Tabla 52 Requisito de diseño RQ-DI-02	151
Tabla 53 Requisito de diseño RQ-DI-03	152
Tabla 54 Requisito de soporte RQ-SO-01	152
Tabla 55 Requisito de soporte RQ-SO-02.....	153
Tabla 56 Requisito de soporte RQ-SO-03.....	154
Tabla 57 Requisito de interfaz gráfica RQ-IG-01	154
Tabla 58 Requisito de interfaz gráfica RQ-IG-02	155
Tabla 59 Requisito de interfaz gráfica RQ-IG-03	156
Tabla 60 Requisito de interfaz gráfica RQ-IG-04	157

Tabla 61 Requisito de interfaz gráfica RQ-IG-05	157
Tabla 62 Requisito de interfaz gráfica RQ-IG-06	158
Tabla 63 Requisito emergente RQ-EM-01	159
Tabla 64 Requisito emergente RQ-EM-02	160
Tabla 65 Requisito emergente RQ-EM-03	160
Tabla 66 Matriz de Trazabilidad	161
Tabla 67 Plantilla de descripción de pruebas	194
Tabla 68 Prueba P-01	195
Tabla 69 Prueba P-02	197
Tabla 70 Prueba P-03	198
Tabla 71 Tabla de descripción de comprobaciones	201
Tabla 72 Acta de la reunión nº1 - Tarea T02	248
Tabla 73 Acta de la reunión nº2 - T07.5	249
Tabla 74 Acta de la reunión nº3 - T08.6	251
Tabla 75 Acta de la reunión nº4 - T09.	252

Capítulo 1. Introducción

El presente documento se corresponde con la memoria descriptiva del Trabajo Final de la titulación Máster Universitario de Investigación en Ingeniería de Software y Sistemas Informáticos (ISSI), impartido por la Universidad Nacional de Educación a Distancia (UNED).

El Trabajo Fin de Máster es una materia de 15 créditos ECTS, de duración semestral y carácter obligatorio, encuadrada en el segundo semestre, del Máster. Esta asignatura está enfocada a la realización de un trabajo de iniciación a la investigación en alguno de los itinerarios del Máster. Comprende, por un lado, una recapitulación y aplicación de las materias previamente cursadas y, por otro, una primera aproximación al flujo de trabajo de las actividades investigadoras. Este estudio está enfocado en ejercitar las destrezas y competencias esenciales para el desarrollo de actividades investigadoras.

El trabajo presentado en este documento tiene la autoría de Rubén Arenas Hernán, estudiante del Máster ISSI e investigador en formación del Centro Singular de Investigación en TecnoloXías da Información (en adelante CiTIUS) y miembro del Grupo de investigación de Gráficos por Computador e Ingeniería de Datos (en adelante COGRADE) de la Universidad de Santiago de Compostela.

El desarrollo de este trabajo ha sido dirigido y supervisado por la Dra. Elena Ruiz Larrocha, profesora del Departamento de Ingeniería de Software y Sistemas Informáticos de la ETSI Informática de la UNED, y ha sido codirigido por el Dr. José Ramón Ríos Viqueira, profesor del Departamento de Electrónica y Computación de la Universidad de Santiago de Compostela (USC), miembro del personal investigador del CiTIUS y miembro del grupo COGRADE.

El trabajo presentado en este documento se encuadra dentro de un proyecto de investigación de mayor envergadura. Se corresponde con una de las fases de implantación de soluciones de explotación de tecnologías semánticas aplicadas a infraestructuras de datos geoespaciales basadas en estándares de servicios web. Estos

estándares definen el modelo de actuación en las tareas de recogida y distribución de datos de observaciones de sensores. Los datos manejados en el proyecto son específicos de la comunidad autónoma de Galicia y gozan de carácter público dado que su producción, almacenamiento y explotación están subvencionados por el gobierno regional de dicha comunidad autónoma.

El proyecto que se presenta en este documento ha sido desarrollado desde el CiTIUS por el grupo COGRADE para las organizaciones MeteoGalicia (Agencia Regional de Meteorología) e INTECMAR (Instituto Tecnológico para el Control del Medio Marino de Galicia). Este proyecto ha sido financiado por MeteoGalicia e INTECMAR mediante varios contratos de investigación en el marco del proyecto (RAIA) [1] [2].

1.1 Contextualización

La observación meteorológica y su principal aplicación, la predicción meteorológica, han experimentado una explosiva revolución en los últimos años. El proceso de predicción meteorológica se basa en el conocimiento de la información de la situación meteorológica en el momento de efectuar la predicción. A partir del conocimiento de la situación meteorológica se pueden aprender los mecanismos que actúan sobre la atmósfera y predecir los comportamientos de ésta en el futuro.

En décadas anteriores, esta extrapolación era efectuada manualmente por parte de los operadores predictores basándose en el estudio y conocimiento previo de los mecanismos atmosféricos. En un porcentaje elevado de ocasiones los resultados de dichas predicciones diferían completamente del estado meteorológico real debido al caos e incertidumbre que rigen la atmósfera. La evolución de este campo ha originado que en la actualidad esos métodos de predicción hayan sido sustituidos por las técnicas de modelización numérica. La aplicación de estos modelos matemáticos ha posibilitado que las predicciones puedan ser realizadas de forma automática y garantizando la validez de las predicciones, al menos para las setenta y dos horas posteriores a la última captura de datos.

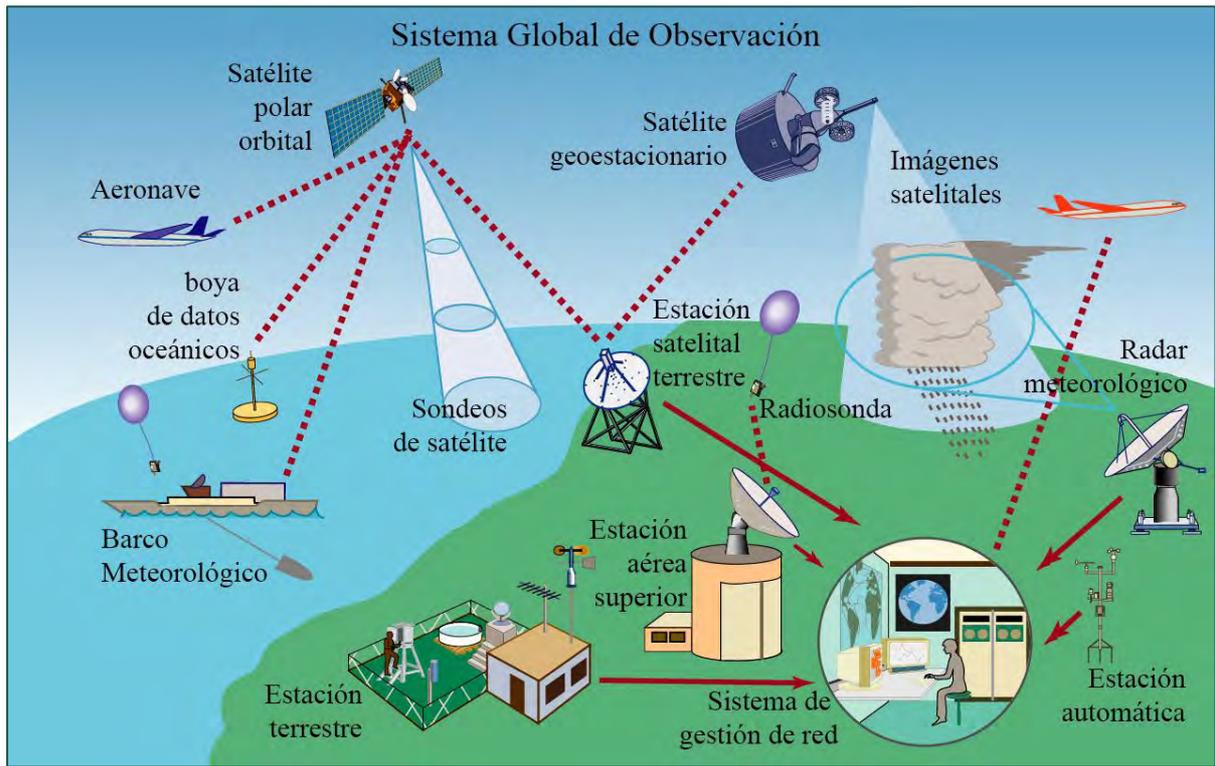


Fig. 1 Dispositivos de observación meteorológica

En términos generales, los datos de observación pueden ser producidos por expertos de un área específica de forma manual o automática; o generados semiautomáticamente por la conjunción de las medidas de dispositivos y procesos específicos de captación de datos. Esto último hecho provoca que sea necesario realizar mediciones de las magnitudes atmosféricas en los distintos niveles de la atmósfera; como pueden ser la presión atmosférica, la temperatura del aire, velocidad del viento, humedad relativa del aire, salinidad del agua, pH del terreno, etc.

Estas observaciones son obtenidas mediante instrumentos de naturaleza y métodos de funcionamiento completamente diferentes, ver Fig. 1, como pueden ser:

- Radiosondas, basadas en aerostatos no tripulados
- Estaciones de observación atmosférica, fijas o móviles compuestas por diferentes instrumentos de medida
- Estaciones de observación marina, fijas o móviles compuestas por distintas herramientas de medición

- Satélites orbitales
- Satélites geoestacionarios, como por ejemplo los satélites de la red Meteosat
- Radares meteorológicos
- Detectores de rayos

La constante observación de las diferentes propiedades meteorológicas genera un gran volumen continuo de datos de observación que deben ser almacenados, validados, manipulados y distribuidos eficaz y eficientemente. Por lo tanto, el diseño de la infraestructura de datos espaciales es crítico para permitir tanto el servicio adecuado de acceso y distribución de los mismos como la generación de predicciones meteorológicas en intervalos de tiempo reducidos [3].

Paralelamente, la evolución tecnológica de fabricación de satélites, radares meteorológicos y sensores específicos, junto con otros sistemas de teledetección ha posibilitado que la capacidad de observación de la situación meteorológica del planeta progrese enormemente. La reducción del coste de fabricación de los diferentes sensores está desencadenando un aumento de la utilización de estos dispositivos en escenarios que hasta el momento no se habían planteado sensorizar. Además la mejora de la potencia de procesamiento de estos sensores y el incremento de la eficiencia de medida y consumo energético se traducen en un incremento en la cantidad y calidad de datos generados y registrados por estos dispositivos de medida. Es por esto que las estimaciones indican que la cantidad de datos provenientes de sensores, captados y generados automáticamente, continúe aumentando cuantiosamente en el futuro.

En los últimos años ha surgido un interés general notable en el acceso y manipulación de datos ambientales de magnitudes del medio ambiente [4], como puede ser la humedad relativa de la atmósfera. Estos datos, que suelen ser agrupados según magnitudes, generalmente provienen de fuentes variadas. Éstos son demandados con el objetivo de resolver problemas específicos de campos concretos, como puede ser la realización de estudios de investigación en los que la información medioambiental juega un papel determinante en la generación de conclusiones y la toma de decisiones.

Por ejemplo, el estudio de la proliferación de determinadas bacterias hace necesario conocer y relacionar conjuntos de datos de determinadas magnitudes del entorno atmosférico; como pueden ser la humedad relativa, cantidad de precipitaciones del último mes, temperatura media registrada, etc. Para ello es vital, en primer lugar, conocer las fuentes de datos que son relevantes y útiles a partir de las cuales se puedan obtener los datos necesarios. En segundo lugar, es indispensable disponer de acceso a este conjunto de fuentes de datos meteorológicos para obtener todos los datos necesarios y poder analizar las consecuencias de los datos de las magnitudes físicas de la atmósfera en la proliferación de esa determinada bacteria.

Otro ejemplo del impacto del acceso a datos meteorológicos se corresponde con la investigación sobre el cambio climático y el calentamiento global. Investigadores de gran cantidad de organizaciones en todo el mundo dedican su actividad a estudiar este fenómeno analizando, entre otros, la información almacenada en ficheros históricos de temperaturas de las últimas décadas. Estas tareas implican una inversión de recursos importante que están provocando que dicha investigación no sea todo lo dinámica que requiere la situación. En el mejor de los casos la información se encuentra almacenada de forma digital, en repositorios indexados de máquinas o servidores dispares.

La aparición de la necesidad de adquisición de datos de múltiples fuentes heterogéneas de diferentes organizaciones ha definido un entorno que demanda soluciones técnicas apropiadas de almacenamiento, acceso y manipulación de los mismos. Actualmente la actividad de acceso y recolección de este tipo de datos implica el desarrollo tareas semiautomáticas específicas, que necesitan la supervisión de uno o varios técnicos, por lo que se necesita una inversión considerable de recursos para llevar a cabo satisfactoriamente dichas tareas.

Por otro lado, en el ámbito de la región de Galicia encontramos varias redes de estaciones meteorológicas de carácter público y con diferente cobertura. Éstas se definen como fuentes de datos de observación que ofrecen información actual o histórica del ámbito medioambiental. Así se distinguen redes en tres niveles diferentes, según su ámbito de operación:

- **Redes regionales:** Encontramos estaciones oceanográficas del Instituto Tecnológico para el Control del Medio Marino de Galicia (en adelante INTECMAR) repartidas por toda la costa gallega y estaciones meteorológicas de la Agencia Meteorológica Regional de Galicia (en adelante MeteoGalicia) situadas a lo largo y ancho del territorio de dicha comunidad autónoma.
- **Redes nacionales:** Existen estaciones meteorológicas de la Agencia Estatal de Meteorología (en adelante AEMET) que se localizan diseminados entre las cuatro provincias de la comunidad gallega.
- **Redes internacionales:** Encontramos estaciones meteorológicas de la Red Global de Climatología Histórica (GHCN), a las que se puede acceder a través de la Administración Nacional Oceánica y Atmosférica (NOAA) del Departamento de Comercio de los Estados Unidos.

Todas estas organizaciones poseen grandes cantidades de datos de observación, que en ocasiones se solapan entre sí, y que usualmente cuentan con sistemas de almacenamiento basados en tecnologías, herramientas y técnicas de naturaleza heterogénea entre sí. Esto supone un importante obstáculo que imposibilita completamente el acceso directo, homogéneo e integral a todos estos datos desde una única herramienta software.

Las dos agencias meteorológicas de la comunidad autónoma de Galicia dependientes del gobierno regional denominado Xunta de Galicia son las citadas MeteoGalicia e INTECMAR. MeteoGalicia, también conocido como Unidad de Observación y Predicción Meteorológica de Galicia es un organismo dependiente de la Consellería de Medio Ambiente, Territorio e Infraestructuras creada en el año 2000, fruto de un convenio entre la Universidad de Santiago de Compostela y la Xunta de Galicia. MeteoGalicia establece sus objetivos principales como:

- la realización de tareas de predicción meteorológica de Galicia, basada en la aplicación de modelos meteorológicos deterministas, modelos meteorológicos probabilísticos y predicción por conjuntos de los datos que son recibidos regularmente desde las estaciones de observación que conforman su propia red.

- la explotación y mantenimiento de la red de observación meteorológica y climatológica de la Xunta de Galicia
- cumplir todas las legislaciones y normativas vigentes sobre datos ambientales y predicción meteorológica [5].



Fig. 2 Logotipo corporativo de MeteoGalicia

Por otro lado, INTECMAR, acrónimo de Instituto Tecnológico para el Control del Medio Marino de Galicia constituye el instrumento de la comunidad autónoma de Galicia, para controlar la calidad del medio marino y la aplicación de las leyes en materia de control técnico de salud de pescados y mariscos.

Desde los comienzos de su actividad, INTECMAR estableció el siguiente listado como sus objetivos esenciales [6]:

- El estricto cumplimiento de la legislación vigente en los contextos de Europa, Estado español y comunidad autónoma en relación al control y la calidad de las áreas de producción de pescados y mariscos.
- Un conocimiento completo de las características de los medios:
 - Asegurando una calidad suficiente para la exploración de recursos.
 - Controlando las posibles fuentes de alteración ambiental que pueden llegar a perjudicar el nivel de calidad del agua.
- La promoción de nuevas estrategias para la explotación y comercialización de recursos para:
 - minimizar las pérdidas económicas resultantes de procesos naturales tóxicos, como la conocida marea roja.
 - minimizar las consecuencias provocadas por una pérdida de la calidad del agua.
- Garantizar una calidad óptima y asegurar la salud total de los recursos marinos con fines de explotación para:

- salvaguardar la salud pública.
- lograr cuotas altas de competitividad en los mercados nacionales e internacionales.



Fig. 3 Logotipo corporativo de INTECMAR

Tanto MeteoGalicia como INTECMAR se enfrentan continuamente a problemas que requieren de la actividad investigadora como vía de exploración de nuevas soluciones técnicas y nuevos métodos de trabajo. Estas necesidades son resueltas en gran medida por la Universidad de Santiago de Compostela. Concretamente el grupo de investigación COGRADE lleva desde hace varios años ayudando a resolver las necesidades técnicas de ambos organismos mediante su actividad investigadora. Fruto de la colaboración entre las tres organizaciones se estableció la necesidad de realizar un estudio de vigilancia tecnológica en torno a los datos medioambientales y la legislación que en 2007 fue aprobada por el parlamento Europeo.



Fig. 4 Logotipo de La Universidad de Santiago de Compostela

Un estudio de vigilancia tecnológica es un proceso organizado, selectivo y permanente, encargado de captar información del exterior y de la propia organización sobre ciencia

y tecnología, para clasificarla, analizarla y comunicarla, con el objetivo de convertirla en conocimiento útil para minimizar los riesgos asociados a las tareas de toma de decisiones y para poder actuar anticipadamente a los cambios [7].



Fig. 5 Logotipo de CiTIUS

Como resultado de dicho estudio de vigilancia tecnológica se tomó la decisión de realizar actividades de investigación en cuanto a manipulación, almacenamiento y distribución de los datos de observación de ambas organizaciones. Así uno de los objetivos establecidos para la actividad investigadora de COGRADE fue ofrecer soluciones técnicas novedosas que permitieran adaptar tanto las tareas, como los métodos y tecnologías que intervienen en los procesos de adquisición, almacenamiento y difusión de las grandes cantidades de datos medioambientales que tanto INTECMAR como MeteoGalicía capturan diariamente a través sus estaciones. Estas actividades investigadoras han estado centradas en una primera etapa en el dominio de los sistemas de información geográfica, para posteriormente centrarse en las infraestructuras de datos geoespaciales. En cuanto a esta segunda etapa hay que destacar el proyecto SOS V1.0 semántico [8][9] puesto que está relacionado directamente con el cometido realizado en este trabajo fin de máster.



Fig. 6 Logotipo de COGRADE

El proyecto SOS V1.0 semántico tuvo como objetivo el diseño e implementación de un servicio de publicación de datos siguiendo el estándar del Consorcio Geoespacial Abierto (OGC) denominado Servicio de Observación de Sensores (SOS), para servir los datos adquiridos y gestionados por MeteoGalicia e INTECMAR con respecto a:

- Datos obtenidos por el Radar de alta frecuencia (Radar HF)
- Datos de columna de agua obtenidos de sensores instalados en sondas CTD (conductividad, temperatura, profundidad)
- Datos de tormenta oceanográfica
- Datos de estaciones meteorológicas terrestres
- Datos de estaciones meteorológicas oceánicas
- Datos de radiosondas de muestreo
- Datos de satélite
- Datos de detección de rayos

La arquitectura del SOS V1.0 se detalla en la Fig. 7. En ella se puede observar que se ha implementado una arquitectura de infraestructuras de datos espaciales distribuida basada en tres capas de componentes software:

- La capa superior se corresponde con el mediador semántico, que es el elemento responsable de:
 - publicar las ofertas predefinidas en el modelo de integración. Este modelo se corresponde con un marco espacio-temporal que registra la solicitud de datos realizada por el usuario, para identificar los elementos que están involucrados en la misma.
 - dirigir las consultas al wrapper o adaptador que corresponda. Dependiendo del tipo de datos que el usuario solicita en cada petición, el mediador identifica la o las fuentes de datos de las cuales extraer la información y encauza la petición hacia dichas fuentes de datos.

El mediador es un componente software basado en el concepto de sistemas distribuidos para la integración de los datos. Este mediador implementa un

modelo de mediación semántica para la integración de fuentes de datos heterogéneas. La mediación semántica fue implementada empleando ontologías que relacionan los diferentes conceptos manejados en un modelo virtual de los datos.

- En la capa intermedia se implementaron unos wrappers o adaptadores específicos, desplegados para cada tipo de fuente de datos. Estos wrappers se encargan de conectar los datos de observaciones con el mediador del servicio. Los wrappers son independientes entre sí. Esto ofrece una gran ventaja, puesto que el sistema permite la escalabilidad a nivel de fuentes de datos. Esta arquitectura permite por tanto añadir fuentes de datos simplemente con definir nuevos wrappers para cada fuente y registrarlos en el mediador.
- En la parte inferior, se ubican las fuentes de datos de observaciones. En principio se contemplan almacenes basados en archivos Thredds y bases de datos relacionales, pero como hemos comentado, podría incluirse cualquier tipo de fuente de observación.

El SOS semántico V1.0 ha sido implementado basando su arquitectura en la herramienta desarrollada por la organización alemana 52°North para la implementación del estándar SOS V1.0, que será detallada en capítulos posteriores del presente documento.

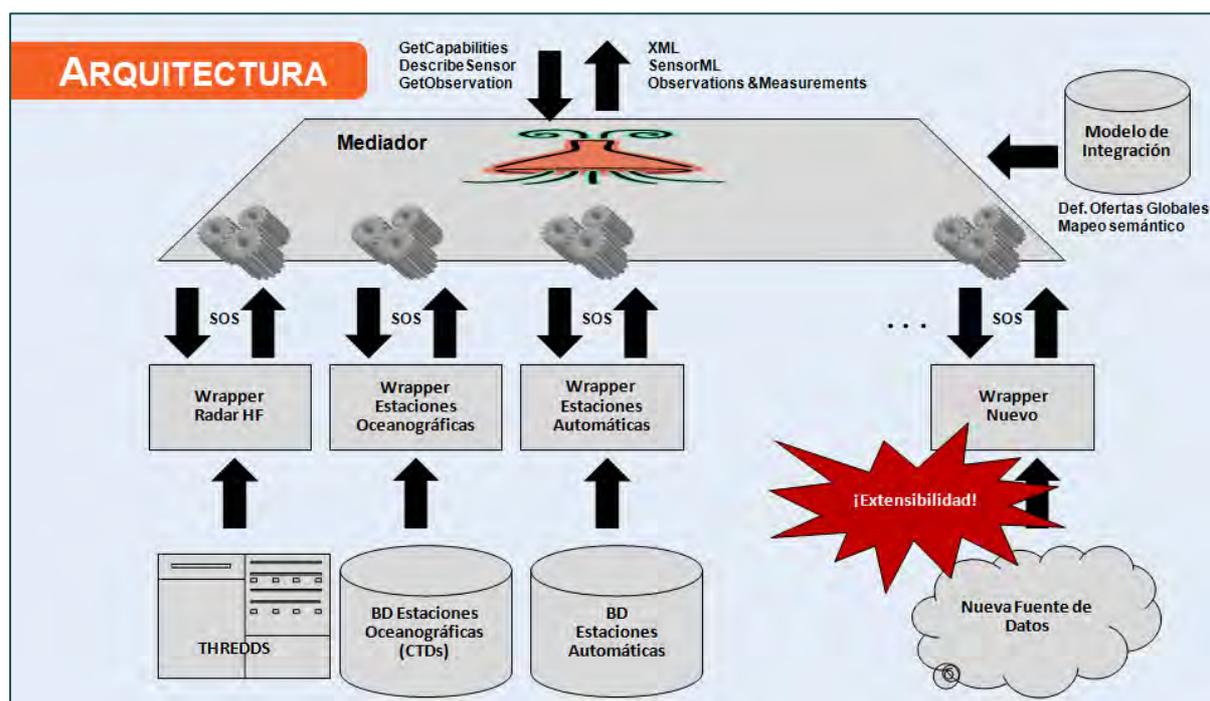


Fig. 7 Arquitectura del SOS semántico v1.0

1.2 Motivación

Actualmente el grupo COGRADE se encuentra en fases intermedias de un nuevo proyecto, realizando labores de migración y adaptación de la arquitectura diseñada e implementada en el proyecto SOS semántico V1.0 a la versión 2.0 del estándar SOS definido por el OGC. Esto es motivado por la próxima entrada en vigor de la directiva europea INSPIRE [10][11].

INSPIRE indica en su normativa la obligatoriedad de las administraciones públicas implicadas en medio ambiente, transportes, agricultura, economía, etc., a emplear infraestructuras de datos espaciales para proporcionar a los ciudadanos servicios de información espacial integrados. Estos servicios deben permitir identificar y acceder a información geográfica de un amplio rango de fuentes, desde el nivel local al global, en forma interoperable para diversos usos [12].

La normativa contempla la implantación de sus diferentes Anexos de forma progresiva y según un riguroso calendario, ver Fig. 14, de forma que todas aquellas administraciones públicas que no se hayan adaptado a la normativa en los plazos establecidos serán sancionadas, tal y como se detalla en los diferentes artículos de la directiva.

Un aspecto importante a tener en cuenta en cuanto a la interpretación de datos espaciales es la visualización y manipulación de dichos datos. En el ámbito atmosférico, los datos de observaciones generalmente son tratados como datos de series temporales que indican la variedad temporal de las magnitudes cuantificadas. En el caso concreto de los datos de observación de estaciones marítimas móviles de la agencia INTECMAR no tiene sentido este tratamiento. Esto es debido a que el proceso de medida se realiza empleando el uso de los parámetros de profundidad y debido además al modo de funcionamiento de la herramienta de medida, denominada CTD. Esta herramienta se encarga de realizar medidas de la conductividad y temperatura del agua a diferentes profundidades. OGC SOS V2.0 contempla el hecho de poseer datos de observación multidimensionales con parámetros asociados.

COGRADE ha detectado tras realizar las primeras pruebas del nuevo proyecto de migración al estándar OGC SOS V2.0 que la implementación de 52ºN del servicio presenta un perfil genérico en el que no se ha contemplado este tipo de información. Así mismo, los diferentes clientes desarrollados tanto por 52ºN como por otras organizaciones para la utilización del servicio SOS V2.0 manejan los datos de observación en todos sus casos como series temporales, ver Fig. 8. El acceso a los datos se realiza en todo caso empleando las operaciones definidas en la interfaz SOS pero siempre manipulados como series temporales. Esto es incompatible completamente con los datos de observación marítima ha motivado que desde COGRADE se plantee el desarrollo de un cliente SOS específico y más genérico que los actualmente desarrollados.

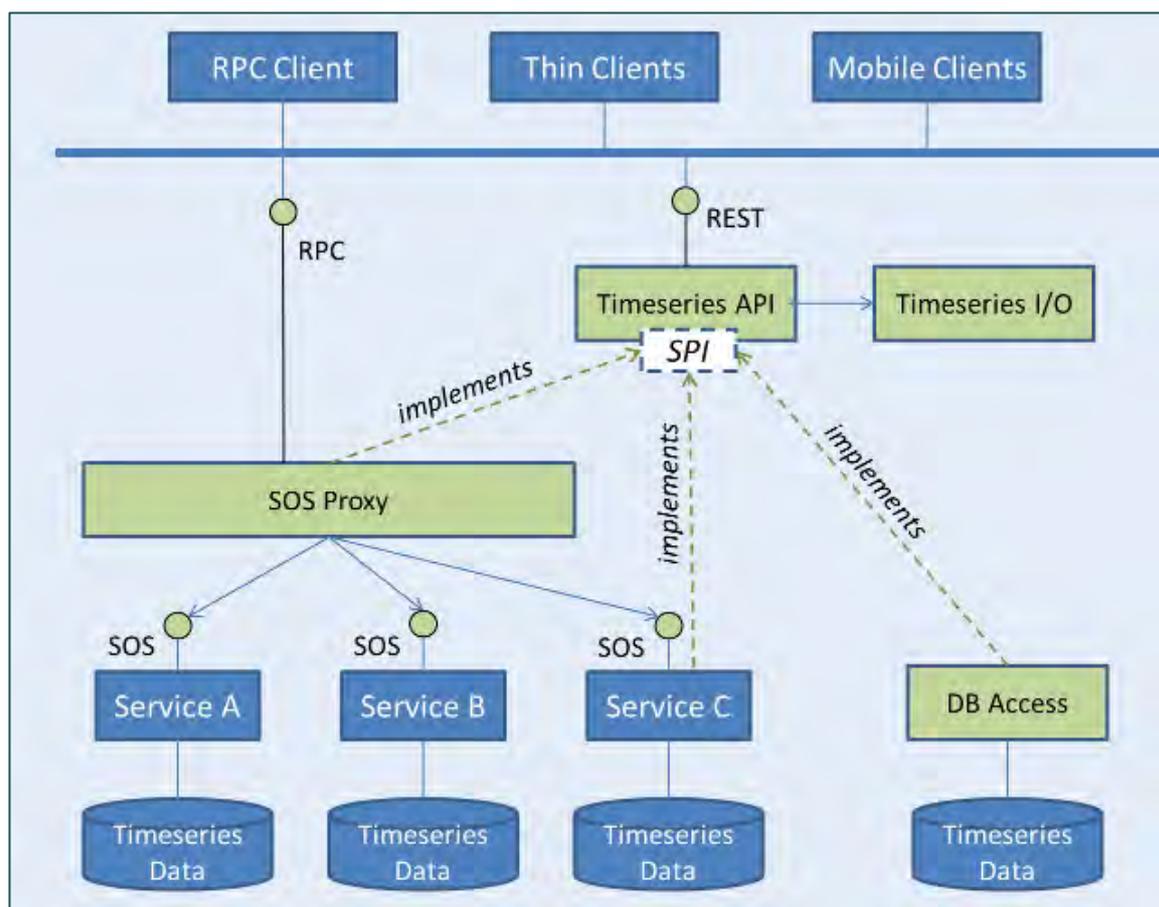


Fig. 8 Arquitectura de componentes, interfaces y APIs del servicio 52ºN SOS

Por otro lado, los datos de observación se analizan generalmente empleando métodos gráficos de representación como pueden ser:

- tablas
- gráficas (estáticas o dinámicas, 2D o 3D, de tarta, de caja, ojiva, etc)
- diagramas y esquemas
- mapas (interactivos, animados, 2D, 3D, etc)
- histogramas
- polígonos de frecuencias

Los datos de observación marítima de profundidad presentan una excepcionalidad con respecto a otros tipos de datos de observación meteorológica, ya que cada una de estas observaciones marítimas está asociada a la profundidad a la que ha sido recogida. De esta manera los datos de observaciones que se manejan en MeteoGalicia e Intecmar presentan dichos parámetros de profundidad que el SOS semántico V2.0 debe manipular. Las implementaciones genéricas de servidor SOS contempla la inclusión de información extra asociada a cada una de las observaciones bajo el concepto que OGC denomina parámetro. Dichos parámetros pueden ser variables en número para cada una de las observaciones; es decir, una observación puede tener desde 0 hasta n parámetros. Cada uno de estos parámetros estará formado por una magnitud y una unidad de medida. Teniendo esto en cuenta, COGRADE ha estudiado y detectado que ninguno de los clientes SOS V2.0 desarrollados hasta el momento contempla la posibilidad de manipular y representar observaciones con parámetros asociados. Por tanto, y para la correcta representación de los datos se deberán investigar y desarrollar mecanismos de visualización que permitan mostrar los parámetros de profundidad de los datos de observación oceanográfica.

Así mismo desde las agencias MeteoGalicia e INTECMAR se ha propuesto la implementación de filtros de resolución de los datos en las herramientas de representación de los datos de observación. El concepto de resolución amplía los horizontes de explotación de los datos de observación más allá de la representación básica que los clientes ligeros SOS 2.0 ofrecen actualmente. Esta característica provoca que el procesamiento de los datos de observaciones se realice de forma local, en el

propio navegador web donde se ejecuta el cliente. De esta forma el cliente se considerará un cliente pesado, puesto que conlleva carga de procesamiento de los datos. Aplicando estos filtros de resolución a los datos de observaciones para la representación de los datos de observación se generarán interpolaciones lineales o medias de las magnitudes según sea el valor de la resolución y la frecuencia de muestreo de las observaciones.

1.3 Objetivos del proyecto

Desde un punto de vista general, el principal objetivo de este trabajo es el desarrollo de un cliente web basada en el estándar SOS V2.0 para las agencias meteorológicas INTECMAR y MeteoGalicia. Dicha herramienta debe ser un demostrador que ponga en valor las capacidades y bondades de la tecnología de integración semántica de datos de fuentes heterogéneas de datos de observación desarrollada en el grupo de investigación COGRADE de la USC. Esta tecnología está basada en la implementación de la especificación V2.0 del estándar del servicio SOS desarrollada por la organización 52°North.

El cliente SOS V2.0 deberá contemplar la manipulación y visualización de datos de observación teniendo en cuenta los parámetros de profundidad que las estaciones meteorológicas marinas presentan. La visualización de los datos de observación deben ser manipulables a diferentes resoluciones, según los intereses y necesidades de los usuarios. Esto provoca que se delegue gran parte de las tareas de procesamiento de los datos en el lado de dicho cliente web, por lo que se considerará un cliente SOS pesado.

Además el cliente SOS V2.0 accederá a los datos a través de una arquitectura de conexión directa con servidor SOS empleando únicamente la interfaz SOS, a diferencia de los clientes SOS 2.0 actuales, que implementan la interfaz TimeSeries para la manipulación y representación de los datos de observación.

Por lo tanto el objetivo del presente proyecto se concreta en el siguiente listado, por lo que el cliente web a desarrollar debe:

- Definir y automatizar mediante interfaz web el proceso de petición y recogida de datos de observación de servidores SOS 2.0 de integración semántica de fuentes heterogéneas de datos.
- Codificar las peticiones web siguiendo el estándar SOS V2.0.
- Descodificar las respuestas del servidor semántico SOS V2.0.
- Procesar los datos de observaciones obtenidos sin emplear la interfaz de series temporales.
- Representar los datos de observaciones procesados en diferentes formatos visuales, como tablas, gráficas, diagramas y/o mapas.
- Permitir la manipulación de las representaciones gráficas mediante la introducción de resoluciones de muestreo.
- Permitir la descarga tanto de los datos crudos como de los datos procesados mediante resoluciones y en formatos estándares bien conocidos.

1.4 Estructura del documento

El presente documento constituye la memoria del Trabajo Fin de Máster que presenta los detalles de documentación del proyecto desarrollado, así como los resultados y las conclusiones extraídas del trabajo. Esta memoria aglutina toda la documentación requerida por los requisitos impuestos en el Reglamento para el Trabajo Fin de Máster del Máster Universitario de Investigación en Ingeniería de Software y Sistemas Informáticos [13].

Toda esta documentación ha sido organizada en los siguientes capítulos:

- **Introducción.** En esta sección se incluye la información general e introductoria del presente trabajo fin de máster. Se detallan tanto la motivación que han originado el desarrollo de este trabajo, como los objetivos a cumplir en el mismo.
- **Estado del arte.** En este apartado se reúnen un conjunto de descripciones detalladas del estado actual de las tecnologías, métodos y herramientas relacionadas directamente con el presente trabajo.
- **Gestión del proyecto.** Esta sección contiene toda la documentación relativa a:

- planificación,
- riesgos,
- costes,
- configuración,
- cambios
- y sistemas de seguimiento del proyecto.

Se describen por tanto todos los detalles de gestión así como los posibles cambios que hayan aparecido a lo largo del desarrollo.

- **Análisis del proyecto.** Este capítulo comprende la documentación recabada en la fase de captura de requisitos, junto a la información derivada del análisis de los mismos.
- **Diseño.** Esta sección reúne los:
 - diagramas,
 - figuras
 - y maquetas que definen los diferentes aspectos del proyecto.
- **Implementación.** Este apartado incluye una descripción pormenorizada de las diferentes etapas del proceso de implementación del proyecto.
- **Pruebas.** Dentro de este capítulo se incluye toda la documentación obtenida en la etapa de testeo de los diferentes desarrollos software creados con anterioridad.
- **Conclusiones.** Este apartado expone las conclusiones obtenidas durante todo el proceso de desarrollo del proyecto. Se sugieren también posibles cambios y mejoras susceptibles de ser incluidas en el proyecto. Además incluye la documentación del estado final del proyecto, así como las ampliaciones futuras del proyecto.
- **Bibliografía:** Este capítulo lista detalladamente la documentación consultada durante todas las fases de desarrollo del proyecto.
- **Apéndices:** Comprende los contenidos no incluidos en el cuerpo de la memoria referenciados en el texto. Contiene los siguientes apartados:
 - **Código del programa.** Esta sección se encarga de recoger secciones de código relevantes de ser detalladas, para la correcta comprensión de las diferentes etapas del ciclo de vida de desarrollo del presente trabajo.

- **Glosario, abreviaturas y acrónimos:** Esta sección se encarga de enumerar y describir el conjunto de términos técnicos, abreviaturas y acrónimos empleados a lo largo del presente documento.
- **Manuales:** Contiene el manual de instalación y el manual de usuario.
- **Acta de reuniones:** Esta sección aglutina los resúmenes obtenidos tras las reuniones entre las partes que han estado relacionadas con este proyecto.

Capítulo 2. Estado del arte

En esta sección discutiremos las diferentes disciplinas académicas, tecnologías y herramientas involucradas en el desarrollo de este proyecto. Una disciplina académica o campo de estudio es una rama del conocimiento. Estas ramas están definidas y reconocidas por:

- Las publicaciones académicas en donde se exponen los resultados de procesos de investigación.
- Círculos académicos, intelectuales o científicos a los cuales pertenecen los investigadores. Los campos de estudio tienen por lo general numerosas ramas o subdisciplinas y las líneas que las distinguen suelen ser arbitrarias y ambiguas, siendo en ocasiones compartidas entre diferentes campos

2.1 Arquitecturas para sistemas de integración de datos

Los sistemas de integración de datos se definen como soluciones estructuradas capaces de automatizar la tarea de extracción de datos heterogéneos almacenados en fuentes dispersas y la tarea de homogeneización de los datos extraídos. Por tanto el objetivo de los sistemas de integración de datos es la asistencia al usuario en el proceso de homogeneización de datos procedentes de fuentes dispersas y heterogéneas, y por lo tanto permitir al usuario abstraerse del problema del acceso distribuido de los datos. La Fig. 9 detalla el proceso gráficamente.

La utilización de datos heterogéneos implica procedimientos de homogeneización de los mismos para su adecuada explotación conjunta y para la transformación de dichos datos en información útil. Hace lustros este proceso se desarrollaba de forma semiautomática resultando en tareas arduas y premiosas. Por consiguiente el proceso

era muy costoso debido a las cantidades de datos que se manejaban. La necesidad de manejo de variadas fuentes obligó a pasar de un enfoque de trabajo basado en el acceso y análisis de los datos de una única fuente a gestionar datos de distinta índole y procedentes de fuentes heterogéneas y de diversa procedencia [14].

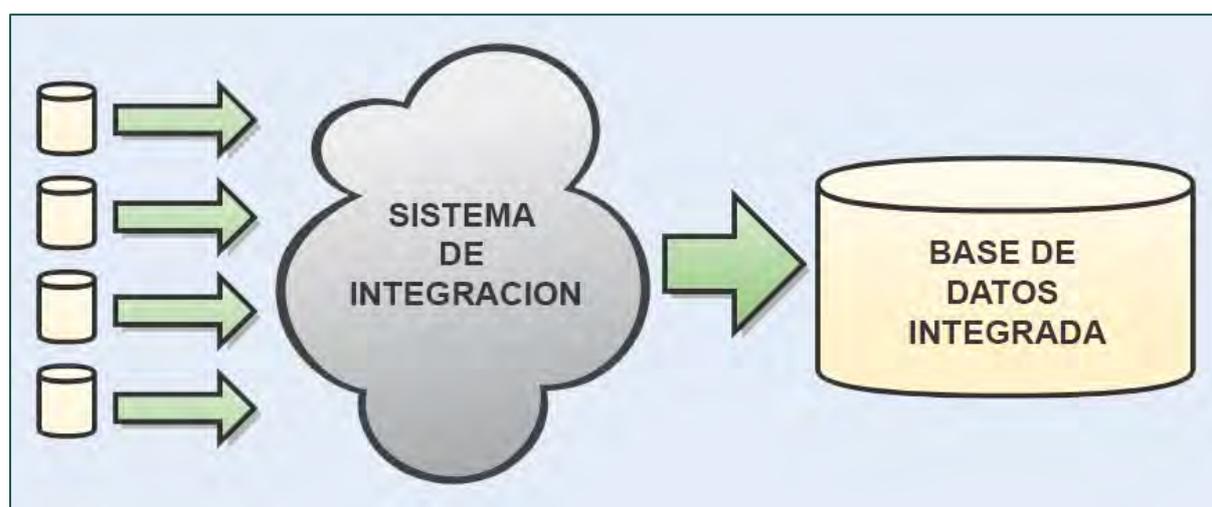


Fig. 9 Esquema general de integración de datos

Por lo tanto eran precisos nuevos enfoques que permitiesen procedimientos dinámicos para el acceso homogéneo a estas fuentes de datos. Por ello se comenzó hace más de una década a investigar en la creación de sistemas capaces de automatizar las tareas de extracción de datos dispersos heterogéneos y su posterior homogenización. El resultado es lo que se conoce como sistemas de integración de bases de datos heterogéneas o simplemente sistemas de integración de datos. En ellos surge el concepto de esquema global o virtual. La idea consiste en definir un esquema de datos que abarque todos los esquemas de las bases de datos a integrar. El usuario del sistema integración realiza consultas a este esquema global sin necesidad de conocer los esquemas de bases de datos subyacentes. El sistema se encarga de obtener los datos necesarios de las fuentes para responder a la consulta original. Este enfoque tiene la ventaja de una mayor usabilidad ya que permite al usuario abstraerse completamente del proceso integración de datos. En contrapartida hay que destacar que esos sistemas implican diseños e implementaciones más complejas y sofisticadas.

Los primeros sistemas en adoptar esta técnica fueron los denominados Data Warehouses. Éstos se basan en un enfoque centralizado, donde los datos son adquiridos y almacenados en un repositorio central. Este tipo de soluciones es utilizado mayoritariamente en entornos empresariales donde las fuentes de datos son bien conocidas y donde hay control exacto sobre los cambios en los datos [14].

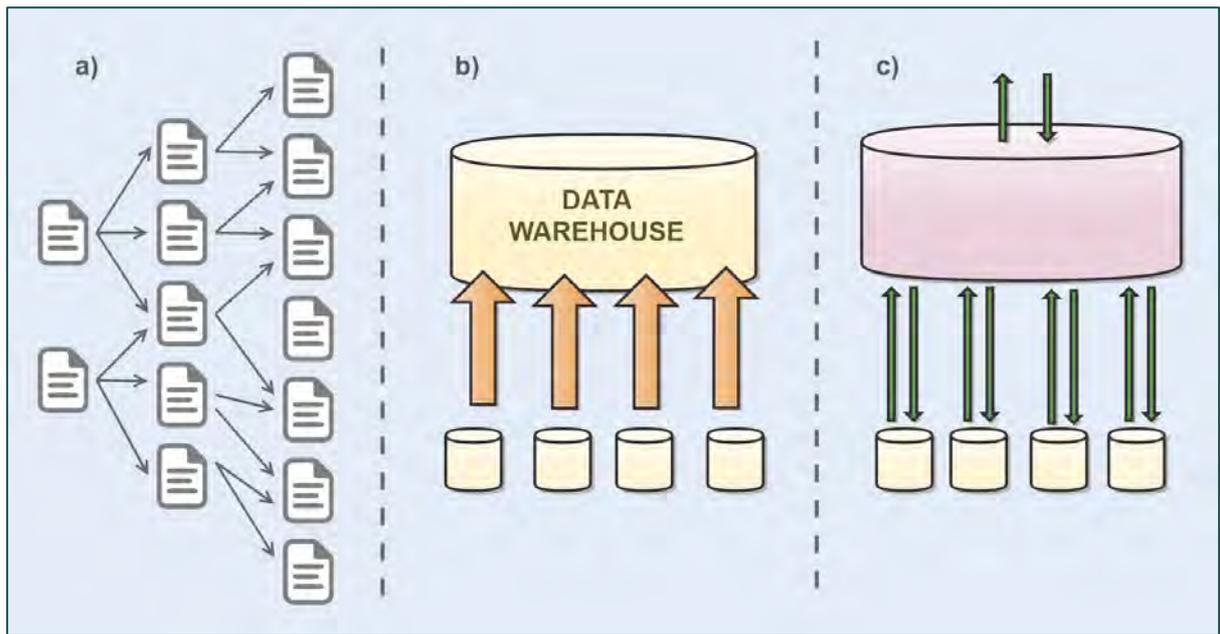


Fig. 10 Los tres tipos de sistemas de integración.

No obstante, el entorno de los datos de observación es diferente, ya que cuenta con fuentes de datos de libre acceso sobre las que no se puede ejercer control y cuyas actualizaciones son inciertas e impredecibles. Además el número de las fuentes de datos de observaciones está creciendo rápidamente debido a las mejoras técnicas y al descenso de los costes de los sensores. Esto ha provocado que en este entorno se exploren soluciones más flexibles frente a los cambios y más fáciles de actualizar ante nuevas fuentes de datos [14].

Como resultado de esta investigación se plantea un tipo de sistema opuesto a los Data Warehouses: los sistemas basados en enfoque federado distribuidos. La idea subyacente consiste en mantener los datos en todo momento en sus respectivas fuentes accediendo

a ellas a cuando las consultas lo requieran. De este modo se resuelta la falta de flexibilidad y adaptabilidad requerida en dominios abiertos y dinámicos. Hay que destacar que como contrapartida este tipo de sistemas son más costosos de desarrollar. De igual forma que los Data Warehouse se emplea el concepto de esquema global, con la salvedad de que no se emplea ninguna base de datos física ni tampoco las tareas ETL (extracción, transformación y carga). La Fig. 10 representa los tres tipos de sistemas de integración de datos que existen en la actualidad [14].

2.2 Arquitectura en tres capas

La programación por capas es el origen de la arquitectura de tres capas. La programación por capas es un modelo de desarrollo software en el que el objetivo primordial es el desacoplamiento de las partes que componen un sistema software. La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que surja algún cambio, solo afectará al nivel requerido sin tener que revisar el código fuente de otros módulos, dado que el acoplamiento se reduce al mínimo. Por tanto su objetivo principal es la separación en diferentes capas de aquellos elementos que son diferenciables desde un punto de vista lógico [15].

La arquitectura en capas se basa en los siguientes conceptos[16]:

- Cada capa debe establecerse en un sistema físicamente independiente, ofreciendo escalabilidad operativa de la arquitectura.
- Cada capa solo podrá intercambiar información (conjunto de peticiones y respuestas) con las capas situadas inmediatamente encima y debajo de ella.
- El contenido de cada capa puede ser intercambiable mientras se respete la definición concisa de la interfaz de comunicación (Application Program Interface, API) con las capas adyacentes, y siempre que no se demande a las otras capas ninguna información a mayores de la que recoja la propia API.

Las ventajas que presenta este tipo de arquitectura software son:

- Permite la división del trabajo entre diferentes equipos, cada uno especializado en un determinado nivel.
- Permite la sustitución de los componentes de una capa de modo transparente para el resto, simplemente es necesario respetar las interfaces.
- Facilita la escalabilidad funcional y operativa de las aplicaciones

Actualmente el diseño en tres capas es el más utilizado como evolución del diseño de dos capas propio de la arquitectura cliente-servidor, ver Fig. 11. En él se establece una separación entre la capa de presentación, la capa de negocio y la capa de datos.

La **capa de datos** como su nombre indica, se corresponde con la capa en la que residen los datos y es la encargada de la lógica algorítmica necesaria para acceder a ellos. Está formada por uno o más gestores de bases de datos o sistemas de ficheros, que realizan todo el almacenamiento y gestión de datos, reciben solicitudes de almacenamiento y recuperación de información. Se comunica únicamente con la capa de negocio.

En la **capa de negocio** residen los módulos responsables de la lógica de negocio, entendiendo lógica de negocio como el conjunto de algoritmos que implementan la funcionalidad del sistema. También se la conoce como capa de aplicación. Se comunica con la capa de presentación para recibir las solicitudes y presentar los resultados y con la capa de datos para realizar tareas de almacenamiento o consulta.

La **capa de presentación** es la interfaz del sistema con el usuario. También se la conoce como capa de usuario o como interfaz gráfica y debe ser fácilmente entendible y utilizable por el usuario. Gracias a ella se pueden realizar peticiones y consultar respuestas. Esta capa es utilizada por el sistema para presentar la información resultante de una petición y se comunica exclusivamente con la capa de negocio.

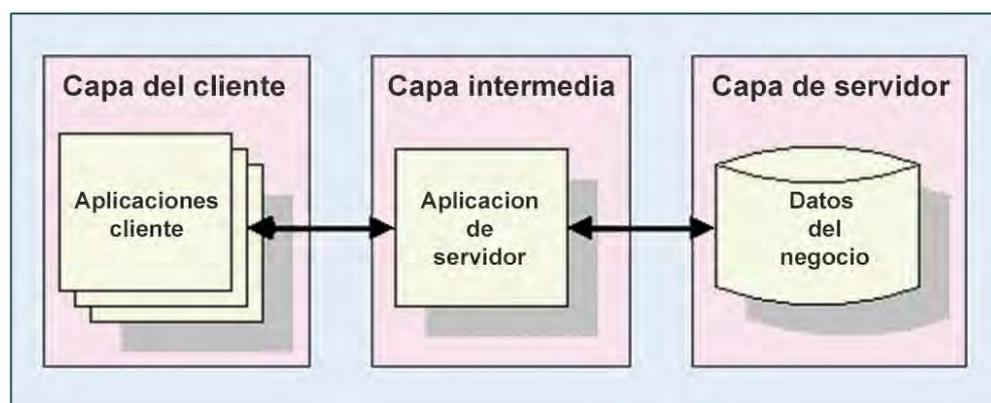


Fig. 11 Arquitectura en tres capas

2.3 Infraestructura de datos espaciales o IDE

Durante las últimas dos décadas han aumentado exponencialmente los esfuerzos y el interés en resolver de forma eficaz la adquisición y difusión de datos medio ambientales a través de interfaces abiertas en el dominio de las infraestructuras de datos espaciales (en adelante indistintamente IDEs o SDIs, del inglés Spatial Data Infrastructure)[17]. Este hecho es especialmente relevante para las organizaciones públicas debido a las normativas aprobadas por la Unión Europea la década pasada y para las que la fecha de formalización de carácter obligatorio se encuentra muy próximo a suceder.

Los avances tecnológicos y el hecho de que buena parte de las actividades humanas presenten un componente de localización, han originado que en la actualidad se disponga de un importante volumen de datos georreferenciados, que van desde los mapas topográficos a cualquier medida obtenida sobre, o ubicable en, el terreno. Las IDEs pretenden catalogar y poner al alcance del público en general toda esta información que, con frecuencia, es desconocida o no tiene canales adecuados de difusión[18]. En otras palabras, se trata de que el usuario emplee un navegador simple, denominado cliente ligero, con el objetivo de:

- buscar qué datos geográficos y qué servicios de datos hay disponibles en la Red
- seleccionar cuáles son de su interés
- visualizar los datos seleccionados

- invocar el servicio o servicios necesarios (servicios de visualización, de acceso a objetos, de nomenclátor, de transformación de coordenadas,...), de modo transparente y sin preocuparse de en qué nodo reside cada componente, obtener las respuestas deseadas y finalizar la sesión[16].

Para lograr estas tareas las IDEs se encargan de integrar:

- datos; almacenados de formas muy determinadas
- metadatos; que son los descriptores de los datos
- servicios web; que se corresponden con las funcionalidades accesibles mediante un navegador que una IDE ofrece al usuario para aplicar sobre los datos geográficos.
- aspectos organizativos. Estándares y normas que hacen que los sistemas puedan interoperar leyes, reglas y acuerdos entre los productores de datos geográficos, así como el personal humano y la estructura organizativa. Los organismos de estandarización más importantes son el OGC (Open Geospatial Consortium) y la ISO (Organización Internacional de Estandarización) [19][20].

Uno de los principales objetivos de las Infraestructuras de Datos Espaciales (IDEs) es facilitar la búsqueda y consulta de la información espacial existente. Para ello es necesaria la catalogación de dichos datos espaciales, y su posterior carga en un servicio de catálogo de metadatos.

La siguiente imagen muestra las diferentes actividades y roles que se definen generalmente en la mayoría de IDEs. Para cada una de estas tareas se incluyen de forma ilustrativa diferentes conjuntos de elementos tecnológicos ejemplificativos [21][16][22].

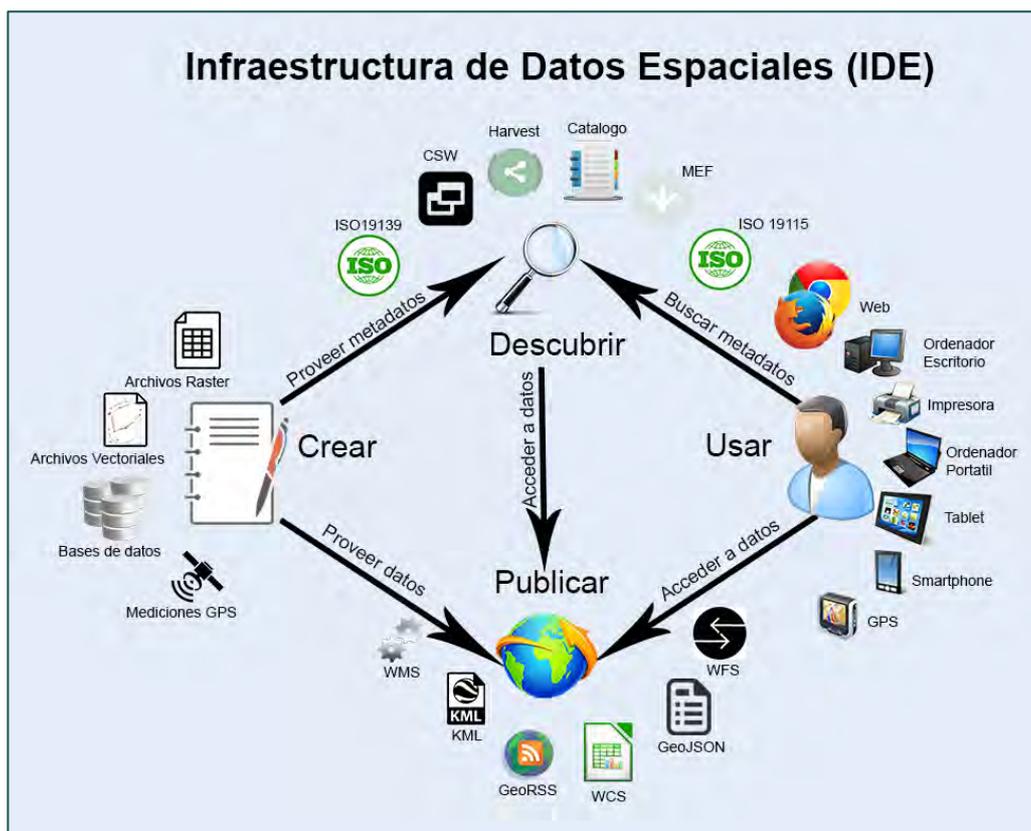


Fig. 12 Esquema representativo de las acciones y entes involucrados en las IDEs

2.4 Directiva INSPIRE

La Directiva 2007/2/CE del Parlamento y del Consejo de la Unión Europea, del 14 de marzo de 2007, insta a las normas generales para el establecimiento de una Infraestructura de Información Espacial en la Comunidad Europea (en adelante CE) basada en las Infraestructuras propias de los Estados miembros. Esta directiva se denomina comúnmente Directiva INSPIRE [23] [24], acrónimo de Infrastructure for Spatial Information in Europe. Se trata de una iniciativa surgida a partir del debate sobre datos geográficos en el seno de la comunidad medio ambiental. Ésta se percató de la existencia de múltiples inconvenientes en los datos de referencia, ya que eran demasiado heterogéneos, algunos inexistentes, y muy poco accesibles, lo que impedía formular políticas coherentes en los datos temáticos derivados. Entre los objetivos de INSPIRE se contempla la creación de una red de intercambio de información espacial y

medioambiental entre las diferentes organizaciones europeas del sector público para facilitar el libre acceso a dicha información por parte de ciudadanos y organizaciones empresariales. De esta forma, dicha Infraestructura se presenta como un instrumento clave para el propio diseño y coordinación del conjunto de políticas comunitarias. INSPIRE es por tanto una de las herramientas legislativas de referencia para los seguidores de datos abiertos de la Unión Europea (en adelante UE) y, en especial, para aquellos que no disponen en sus correspondientes países de repositorios activos equivalentes. Por tanto, INSPIRE plantea la puesta en común de una serie de conjuntos de datos espaciales (en adelante CDE) y servicios de datos, principalmente relacionados con aspectos medioambientales en el territorio de la Comunidad Europea. Esta apertura de datos es, hoy por hoy, la mejor garantía para la innovación, la producción de nuevos servicios, la transparencia pública, la interacción y la participación de los ciudadanos en el progreso de la Unión Europea [25].



Fig. 13 Logotipo INSPIRE

Con la aprobación de INSPIRE se inició un gran desafío para las administraciones públicas de los países miembros de la CE, entre ellos España. Dicho desafío conlleva la creación de una serie de escenarios colaborativos y de difusión para alcanzar los objetivos iniciales de INSPIRE, así como también una serie de actividades que están siendo desarrolladas progresivamente y que están permitiendo difundir los requerimientos técnicos y asignación de responsabilidades a los diferentes organismos públicos implicados. Esta normativa obliga a todas las administraciones públicas de los países miembros de la Unión Europea a publicar y facilitar a la ciudadanía el acceso a

los datos geospaciales a través de interfaces estándares bien conocidas, con fecha límite en el año 2019.

En líneas generales INSPIRE pretende conseguir la progresiva implantación de un sistema único de coordinación europea, capaz de delimitar unas reglas y estándares comunes durante los procesos de creación, gestión y difusión de los mencionados datos espaciales. Todo ello de acuerdo a unos criterios concretos de cantidad, calidad, accesibilidad, interoperabilidad, cooperación y capacidad para producir servicios de nueva generación.

Para ello, se establecen como pasos fundamentales la adecuada identificación, etiquetado y liberalización de los datos susceptibles de reutilización en los sectores público y privado, en especial de aquellos más estrechamente relacionados con la demografía, los transportes, el Medio Ambiente y las obras públicas[26][21].



Fig. 14 Calendario directiva INSPIRE

INSPIRE se basa en seis principios fundamentales[21][27]:

1. Los datos deben ser recogidos y mantenidos a nivel territorial, puesto que estas actividades se realizan de forma más efectiva en ese ámbito.
2. Debe permitirse la combinación de información espacial de diferentes fuentes de datos de la CE y debe permitirse compartirla entre diferentes usuarios y aplicaciones.
3. Debe ser posible que la información recogida en un nivel determinado sea compartida entre todos los demás niveles.
4. La Información Geográfica necesaria para el buen gobierno a todos los niveles debe ser abundante y accesible bajo condiciones que no restrinjan su uso extensivo.
5. Debe ser sencillo identificar la información geográfica accesible, que necesidades cubre cada una de ellas y bajo qué condiciones debe ser adquirida y utilizada.
6. Los datos geográficos deben poder ser fácilmente entendibles e interpretables pudiendo ser visualizados dentro de un contexto apropiado y seleccionados de una forma amigable para el usuario.

La Fig. 15 ofrece una descripción simplificada de los elementos clave de la arquitectura técnica de INSPIRE. Esta se basa en una arquitectura de tres capas.

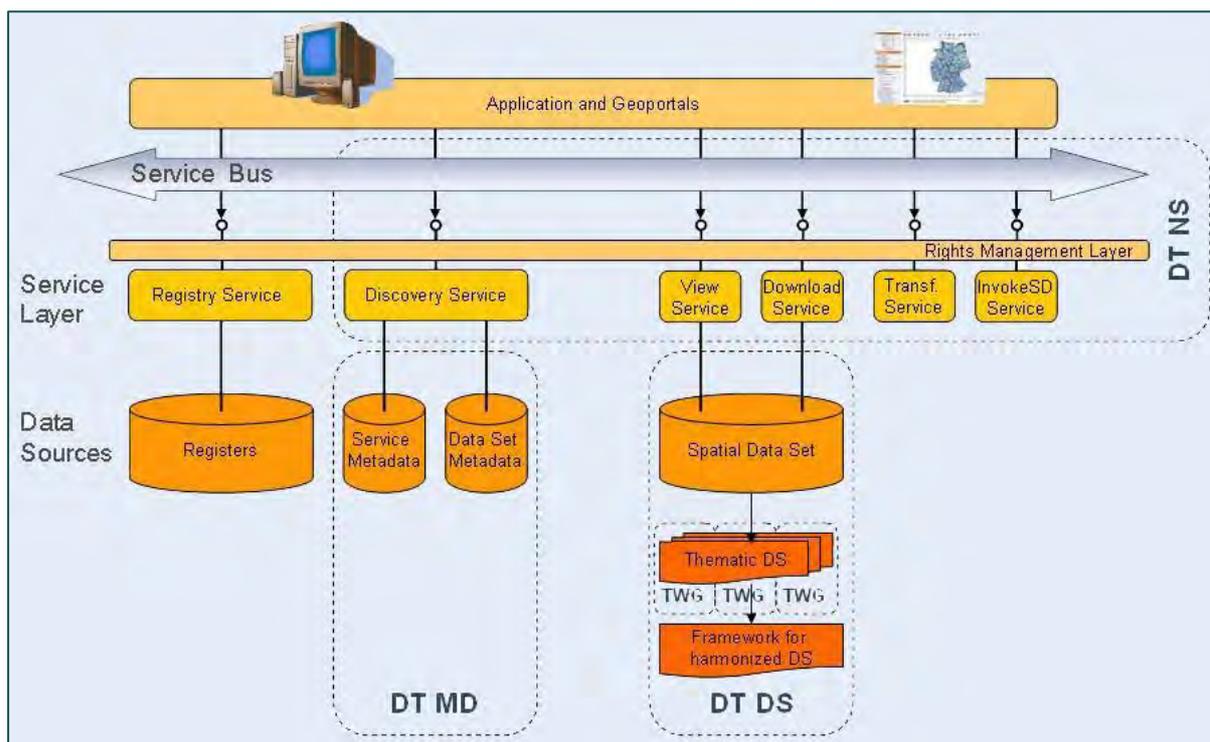


Fig. 15 Arquitectura propuesta por INSPIRE para IDEs [28]

Desde el punto de vista de la directiva INSPIRE, la arquitectura propuesta se descompone en:

- Capa de datos: puede estar gestionada mediante una base de datos o un sistema de ficheros.
- Capa de servicios: es la encargada de encapsular toda la lógica correspondiente a las capacidades funcionales del sistema. Esas capacidades se agrupan en servicios. A modo de ejemplo, el servicio de visualización será el que proveerá el conjunto de herramientas que permitirán a las aplicaciones visualizar las diferentes capas geográficas publicadas. Esta capa se muestra en forma de bus de servicios (repositorio) que se ofrece de forma pública a los diferentes clientes.
- Capa de aplicaciones: su propósito es recoger cualquier tipo de aplicación que explote alguno de los servicios provistos por la IDE, ya sea un cliente ligero o pesado. El presente trabajo se corresponde con el desarrollo de una herramienta encuadrada en esta capa, para la explotación del servicio SOS V2.0 semántico.

2.5 Open Geospatial Consortium

El OGC es una organización internacional sin ánimo de lucro comprometida con la creación de estándares abiertos de calidad en el ámbito geospacial. El Open Geospatial Consortium (OGC) que fue creado en 1994, agrupaba en 2009 a 372 organizaciones públicas y privadas. Actualmente el OGC está formado por 532 miembros. Los miembros del OGC componen un conjunto ecléctico, con orígenes dispares, como son organizaciones gubernamentales, organizaciones comerciales, ONG's, organizaciones académicas y organizaciones investigadoras. Dentro de OGC los miembros están clasificados según niveles, tal y como se detallan en el siguiente listado [29]:

- Miembros estratégicos. Formado 5 miembros:
 - Ordnance Survey
 - US Department of Homeland Security (DHS)
 - US Geological Survey (USGS)
 - US National Aeronautics and Space Administration (NASA)
 - US National Geospatial-Intelligence Agency (NGA)

- Miembros principales. Grupo integrado por 17 miembros:
 - Airbus Defence & Space
 - Bentley Systems, Inc.
 - Department of Science & Technology
 - Esri
 - Feng Chia University
 - GeoConnections - Natural Resources Canada
 - Geoscience Australia
 - Google
 - Intergraph Corporation
 - Oracle USA
 - US National Oceanic and Atmospheric Administration (NOAA)

- Miembros técnicos. Nivel compuesto por 63 miembros, entre los que destacan:
 - 1Spatial Group Ltd.
 - Abu Dhabi Systems & Information Centre
 - Australian Bureau of Meteorology

- Autodesk, Inc.
 - Defence Geospatial Information Working Group (DGIWG)
 - Department of Defense (Australia)
 - Eclipse Foundation, Inc.
 - EMSA (European Maritime Safety Agency)
 - Esri India
 - European Space Agency (ESA)
 - European Union Satellite Centre
 - Hitachi, Ltd., Central Research Laboratory
 - IBM
 - Instituto Nacional de Estadística y Geografía (INEGI)
 - Johns Hopkins University Applied Physics Laboratory
 - Korea Land & Geospatial InformatiX Corporation
 - Leica Geosystems AG
 - METEO-FRANCE
 - University of Nottingham
 - US Army Geospatial Center
-
- Miembros técnicos asociados. Nivel integrado por 2 miembros
 - Miembros asociados. Grupo formado por 132 miembros, entre los que destacan:
 - FedEx Corporation
 - Geodan Holding BV
 - Geological Survey of Finland
 - Geospatial Information Authority of Japan (GSI)
 - Great-Circle Technologies, Inc.
 - Hitachi, Ltd., Defense Systems Company
 - Indra Software Labs
 - Instituto Geográfico Nacional (IGN of Spain)
 - Instituto Nacional para la Evaluación de la Educación
 - Mitsubishi Electric Corporation
 - Thomson Reuters
 - U.S. Geospatial Intelligence Foundation (USGIF)
 - United Kingdom Hydrographic Office
 - US Air Force Weather Agency
 - US Dept. of Defense/DISA
 - Web3D Consortium, Inc.
-
- Miembros empresariales pequeños. Formado 46 miembros.
-

- GovFuture - Subnacional. Grupo integrado por 19 miembros
- Institutos de Investigación / Instituciones sin ánimo de lucro. Compuesto por 1 miembro.
- ONGs / Instituciones sin ánimo de lucro. Nivel compuesto por 61 miembros.
- GovFuture - local. Grupo que cuenta con 33 miembros.
- Universidad. Entre 106 miembros de este nivel destacan:
 - Centre for Geo-Information; Wageningen University
 - CyberGIS Center for Advanced Digital & Spatial Studies
 - Federation University Australia
 - FEUP (Faculdade de Engenharia da Universidade do Porto)
 - Georgia Tech
 - GIS Lab, Center for Space & Remote Sensing Research, National Central University
 - Harvard University
 - Johns Hopkins University
 - Luxembourg Institute of Science & Technology (LIST)
 - Marine Institute, School of Ocean Technology
 - MINES ParisTech
 - MIT
 - Stanford University
 - Technische Universität München - Runder Tisch GIS e.V.
 - Universidad Politecnica de Madrid (Grupo Mercator)
 - Universitat Autònoma de Barcelona (CREAF)
 - Universitat Jaume I
 - University of Muenster - Institute for Geoinformatics
 - University of Washington
 - University of Zaragoza
- Individual. Grupo formado por 37 miembros

Las raíces del OGC se encuentran en el software fuente libre GRASS y la fundación inicial data de 1992 bajo las siglas OGF, acrónimo de Open GIS Foundation. Posteriormente pasó a constituirse como Open GIS Consortium, para, finalmente adoptar la denominación actual[30].

Actualmente OGC se ha consolidado como una organización de estandarización no lucrativa, internacional que está liderando el desarrollo de estándares para los servicios web de información geográfica. A través de sus programas de cooperación OGC trabaja para crear los modelos, arquitecturas e interfaces de programación, para que los

sistemas de información geográfica (en adelante SIG) sean abiertos e interoperen entre sí. Su misión declarada es promover el desarrollo y uso de técnicas y estándares de Sistema Abiertos en el campo de la información geográfica.

Sus objetivos estratégicos son [16]:

- 1) Proporcionar a la comunidad, de modo libre, abierto y gratuito los estándares disponibles, beneficios tangibles a sus miembros, y ventajas mensurables a los usuarios.
- 2) Liderar a nivel mundial la creación y el establecimiento de estándares que permitan que los contenidos y los servicios geospaciales se integren sin costuras, es decir, se mimeticen, en los procesos de tratamiento de la información, ya sean públicos o privados, a través de la web espacial y la informática de empresa.
- 3) Facilitar la adopción en todo el mundo empresarial de arquitecturas espaciales de referencia abiertas.
- 4) Avanzar estándares para ayudar la formación de mercados nuevos e innovadores y nuevas aplicaciones de las tecnologías geospaciales.
- 5) Acelerar la asimilación en el mercado de las investigaciones sobre interoperabilidad a través de procesos colaborativos de participación en consorcios.

Las normas de OGC son utilizadas con fines de naturaleza completamente dispar en una amplia variedad de contextos que emplean datos georreferenciados, incluyendo el dominio de Medio Ambiente, Defensa, Salud, Agricultura, Meteorología o Desarrollo Sostenible entre otros. OGC busca, por un lado, la creación de acuerdos entre las diferentes empresas, organizaciones y organismos del sector que posibiliten la interoperación de sus sistemas de geoprocesamiento; y por otro, facilitar el intercambio de la información geográfica en beneficio de los usuarios.



Fig. 16 Logotipo de OGC

Los estándares OGC se desarrollan a través de procesos de consenso entre los diferentes miembros que lo componen. Dichos estándares son distribuidos con licencia de dominio público, por lo que se distribuyen libremente para que la ciudadanía y desarrolladores los utilicen con el objetivo de mejorar la calidad y facilidad de empleo de datos geospaciales. Estos estándares incluyen la especificación de servicios de descubrimiento de sensores, tareas específicas relacionadas con sensores, generación de alarmas y notificaciones derivadas de datos de sensores y, protocolos de acceso a datos reales de sensores.

OGC desarrolla sus funciones incluyendo los conceptos que se definen y encuadran dentro de Open Data.

2.6 Open Data

Open Data se corresponde con una filosofía y conjunto de prácticas que persiguen que determinados datos e informaciones sean accesibles y estén disponibles para todo el mundo, sin restricciones técnicas ni legales. El fin de la iniciativa Open Data es que la información pueda ser redistribuida y reutilizada tanto por ciudadanos como por organizaciones empresariales o investigadoras para conseguir un beneficio común a todas las partes[31]. Tener acceso a los datos de la Administración garantiza la transparencia porque se permite acceder a datos que proceden directamente de fuentes oficiales financiadas mediante fondos públicos. Además se fomenta la eficiencia y la igualdad de oportunidades, ya que los ciudadanos y las empresas pueden crear servicios que resuelvan sus necesidades en colaboración con la Administración y todo el mundo puede acceder a los datos en igualdad de condiciones [32].

La Reutilización de la Información del Sector Público (también conocida como RISIP [33][34]) es el objetivo principal de la iniciativa Open Data. RISIP consiste en ofrecer a la ciudadanía bajo determinados servicios la información generada por los diferentes estamentos del sector público. Generalmente la información se distribuye tal y como fue capturada, sin ningún tipo de refinamiento o filtrado y bajo formatos estándares abiertos, facilitando su acceso y permitiendo su reutilización tanto a particulares como a empresas para cualquier tipo de propósito. El sector público produce una gran variedad de información que es potencialmente reutilizable, como puede ser la información social, económica, geográfica, estadística, etc. Esta información es muy atractiva para su reutilización ya que es completa, fiable y de calidad. Open Data persigue conseguir que la reutilización y redistribución de la información pública repercuta en beneficio tanto de las administraciones públicas que divulgan sus datos, como de los ciudadanos y empresas que encuentran en estos datos la forma de cubrir de forma sencilla necesidades que hasta el momento de implantación de Open Data era complejo de alcanzar.

Desde las Administraciones Públicas se ha detectado la necesidad que ciertos grupos de personas y organizaciones tienen para acceder a determinada información manejada por las Administraciones y que no siempre han estado disponibles. Es por esto que las Administraciones Públicas han adoptado una actitud de progreso hacia un gobierno más abierto, basado en los valores de transparencia, participación, servicio y eficiencia. Además, Open Data pretende:

- promover la creación de valor mediante la reutilización de la información pública,
- simplificar y favorecer la ordenación interna de los sistemas de información dentro de las Administraciones,
- fomentar la interoperabilidad entre servicios del sector público.

Paralelamente, la liberación de datos de las Administraciones Públicas constituye también un interesante potencial económico ya que supone una base esencial para muchos productos de información digital. Es indispensable asegurar la reutilización eficaz de la información pública para conseguir aprovechar todo su potencial y

desarrollar nuevos productos, servicios y mercados que obtengan un mayor desarrollo económico y por ende, un incremento en la creación de puestos de trabajo.

El primer paso para la liberación de los datos es identificar las necesidades ciudadanas para que las Administraciones decidan qué conjuntos de datos deber ser puestos a su disposición y además identificar el formato más adecuado para dicho propósito, de forma que esos datos se puedan reaprovechar eficientemente[35]. El World Wide Web Consortium (cuyo acrónimo es W3C), que es la comunidad internacional encargada de desarrollar estándares para el crecimiento de la web, recomienda que los datos que se liberen sean útiles y estén en formatos reutilizables [36]. Cuanto mejor estructurados y enriquecidos estén los datos, más fácil resultará el proceso de reutilización y construcción de nuevas aplicaciones que realicen un tratamiento y/o análisis automatizado de dichos datos.

En este sentido, la actividad de OGC es intensa en cuanto a desarrollar y mantener la iniciativa Sensor Web Enablement (en adelante SWE)

2.7 Sensor Web Enablement (SWE)

SWE [37] es una iniciativa gestionada por OGC que se encarga de definir un conjunto de estándares que ofrece a la comunidad de desarrolladores permitir que los sensores, transductores y repositorios de datos de sensores sean detectables, accesibles y utilizables a través de la Web, sea cual sea su naturaleza, tecnología o tipología de construcción y manejo. SWE permite establecer una aproximación para el servicio y codificación de datos de sensores a través de la web bajo una arquitectura orientada a servicios (AOS, en inglés SOA).

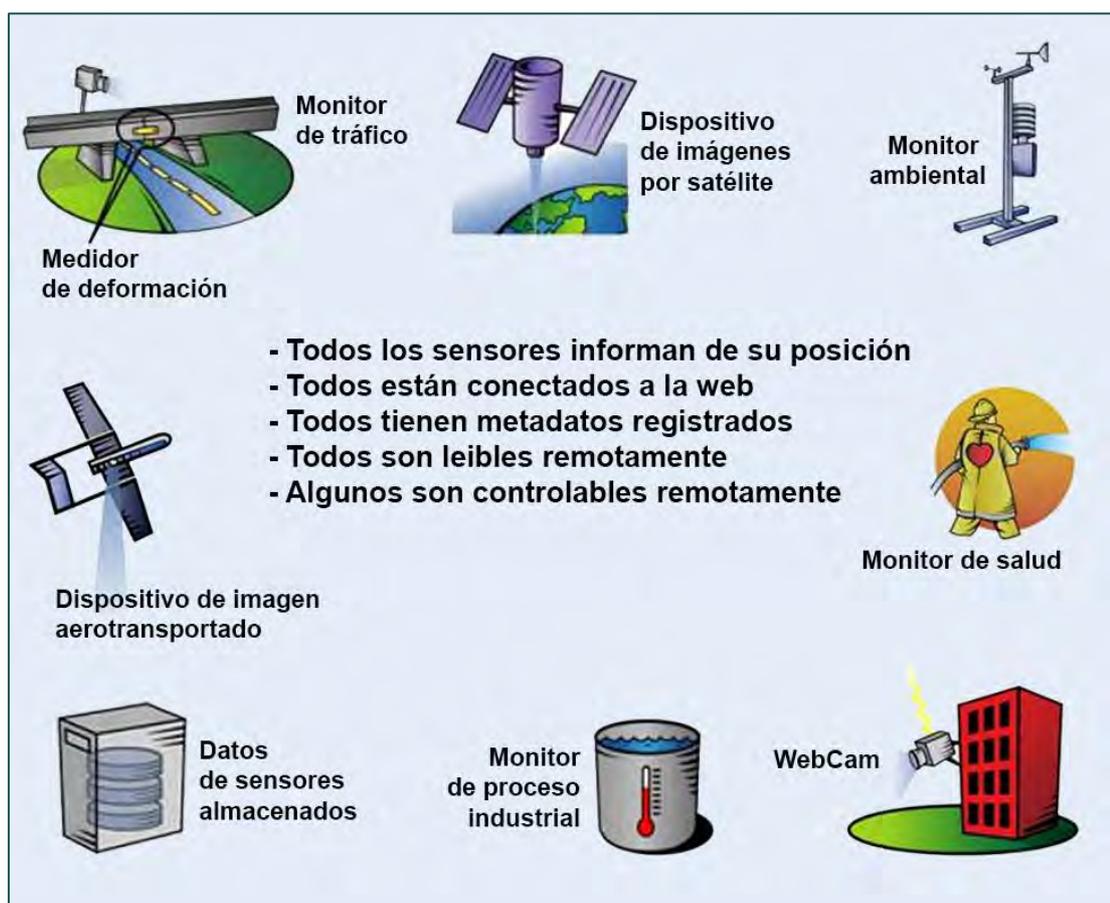


Fig. 17 Resumen de SWE

La suite SWE consta de varios estándares desarrollados que podemos clasificar en dos categorías [38] [39]:

- **Estándares XML de codificación**
 - Observaciones y Mediciones, en adelante **O&M**, siglas en inglés de Observations & Measurements, que detalla los modelos generales y codificaciones XML de observaciones y mediciones de sensores.
 - Lenguaje de modelado de sensores, en adelante **SensorML**, abreviatura de Sensor Modeling Language. Éste especifica los modelos generales y codificaciones XML de sensores.
 - Lenguaje de marcado de transductores, conocido como **TML**, siglas en inglés de Transducer Modeling Language.
- **Interfaces estándar de servicios web**

- Servicio de notificaciones web, conocido como **WNS**, siglas del inglés Web Notification Service. Es un servicio mediante el cual un cliente puede mantener una comunicación asíncrona, mediante el intercambio de mensajes, con uno o más servicios. Este servicio es útil cuando se requieren muchos servicios de colaboración para satisfacer una solicitud del cliente y/o cuando existan retrasos significativos para satisfacer la solicitud del usuario.
- Servicio de planificación de sensores, en adelante **SPS**, acrónimo del inglés Sensor Planning Service. Servicio mediante el cual un cliente puede determinar la viabilidad de una colección para un conjunto deseado de solicitudes de recogida de datos de uno o más sensores / plataformas móviles, o mediante el cual un cliente puede enviar un conjunto de peticiones de envío directamente a estos sensores / plataformas.
- Servicio de Observación de Sensores, en adelante **SOS**, abreviatura del inglés Sensor Observation Service. Es un servicio mediante el cual un cliente puede obtener observaciones de uno o más sensores/plataformas, permitiendo obtener datos de tipos mixtos de sensores /plataformas. Los clientes también pueden obtener información descriptiva de los sensores y plataformas asociados.
- Servicio de alertas de sensores, denominado por **SAS**, por sus siglas del inglés Sensor Alert Service. Es un servicio mediante el cual un cliente se suscribe a determinadas condiciones de alerta y/o autodefinidas y por el cual se realiza una notificación en caso de que coincida la condición. El SAS utiliza XMPP como protocolo de transporte.

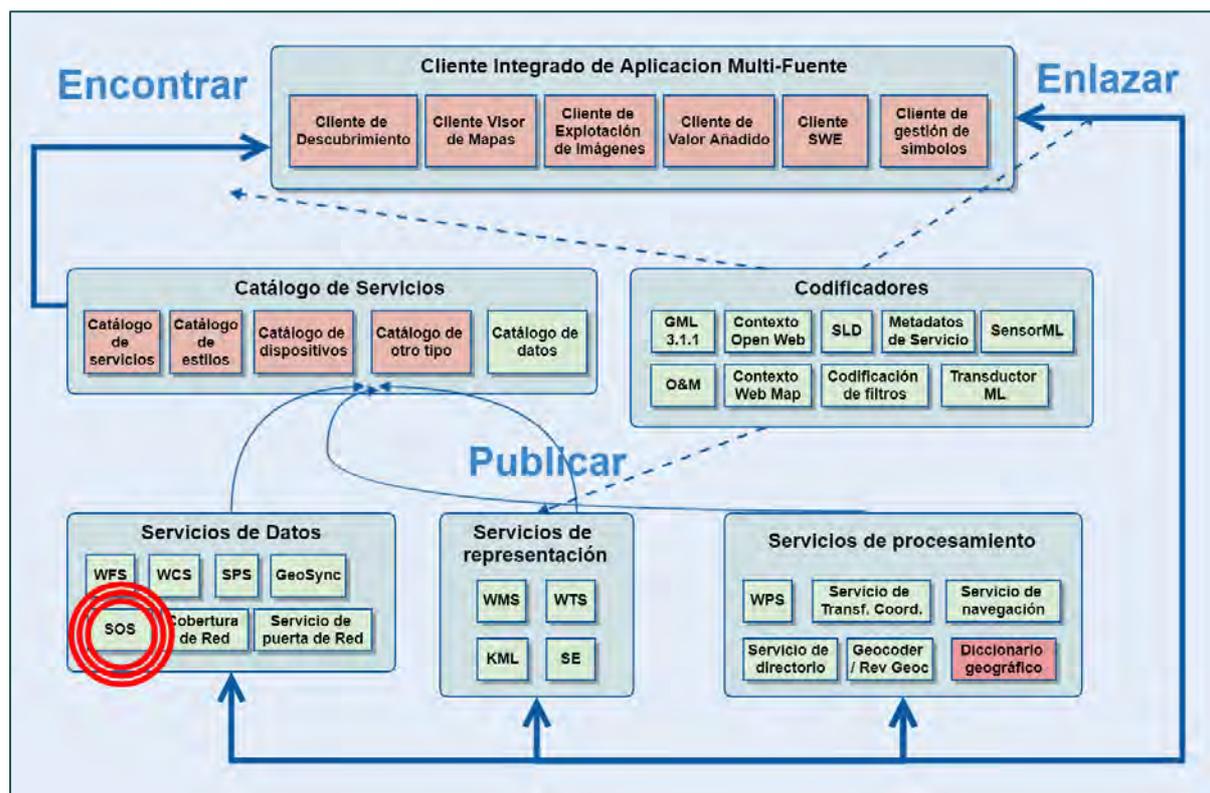


Fig. 18 Elementos de SWE y acciones definidas entre los mismos

2.8 Sensor Observation Service o SOS

Del listado anterior de estándares de la suite SWE, hay que destacar el Servicio de Observación de Sensores (SOS) [40] ya que se ha posicionado desde hace varios años como el estándar a emplear en el desarrollo de sistemas de recuperación de información geoespacial de sensores. El estándar SOS está considerado como el estándar idóneo para el servicio de publicación y recuperación de datos georreferenciados compatibles con la directiva INSPIRE, ya que cumple estrictamente los requerimientos que la normativa impone [39].

El estándar OGC SOS precisa un interfaz estandarizado y un conjunto de operaciones que permiten acceder a datos de observaciones de sensores y sistemas de sensores. SOS permite el acceso a sensores y sistemas de sensores a través de la definición de tareas, acciones y flujos de trabajo específicos bajo una arquitectura cliente/servidor. El

estándar SOS define una interfaz de servicio Web que permite consultar observaciones y metadatos de sensores, así como representaciones de características observadas. Además, SOS define medios para registrar nuevos sensores y eliminar los existentes. SOS también establece claramente las operaciones que permiten la inserción de nuevas observaciones para cada uno de los sensor.

El estándar SOS define un modelo común para sensores y sistemas de sensores y que se pueden utilizar sin conocimientos previos de esquemas de aplicación de los dominios específicos. Así, SOS ofrece acceso estandarizado a colecciones de datos de observación generados por diferentes procesos, siendo coherente con todos los sistemas actuales de sensores, incluyendo sistemas remotos, in-situ, fijos y sensores móviles. SOS proporciona los resultados de consultas siguiendo la codificación del formato estándar de O&M para modelizar observaciones de sensores y siguiendo la especificación SensorML para modelizar sensores y sistemas de sensores.

2.8.1 Metadatos SOS

El concepto de metadatos ha evolucionado con el tiempo. De forma general los metadatos se pueden definir como los datos que describen a los datos (ISO 19115), lo que supone el análogo digital a toda la información secundaria (leyenda, fecha, autor, etcétera) que rodea a un documento papel[16]. De esta forma los metadatos deben ser considerados como los datos que describen cualquier tipo de recurso, ya sea una organización, un ordenador, un libro, una aplicación, etc.

Concretando el concepto y a la luz del paradigma IDE en el que se sustituyen los datos por los servicios como concepto central a partir del que se concibe todo un sistema basado en Arquitecturas Orientadas a Servicios, los metadatos se corresponden con la información que describe a los datos y servicios[16].

Teniendo en cuenta esta definición de metadatos, podemos indicar que SOS maneja un conjunto interesante de los mismos que se encargan de definir tanto al servicio SOS con sus propiedades y anotaciones, como a todos y cada uno de los sensores que están registrados en el servicio. En esta misma sección se detallaran las dos operaciones que

define el interfaz SOS para el acceso a estos metadatos: la operación GetCapabilities y la operación DescribeSensor.

2.8.2 Diferencias entre SOS V1.0 y SOS V2.0

Actualmente existen dos definiciones del estándar SOS [41], desarrolladas y depuradas por los miembros del OGC. A lo largo de la última década se han desarrollado la versión 1.0.0 cuyo título es OpenGIS Sensor Observation Service [42] y la versión 2.0, adoptada en 2012, que tiene como título OGC® Sensor Observation Service Interface Standard [43].

SOS 2.0 difiere y complementa a SOS 1.0 en los siguientes aspectos:

- SOS 2.0 incluye una reestructuración modular del documento definitorio del estándar.
- SOS 2.0 contempla una forma nueva y sencilla de realizar la vinculación de los datos mediante pares clave-valor.
- SOS 2.0 define una nueva vinculación de SOAP.
- SOS 2.0 incluye un rediseño del concepto de Offering de observación.
- SOS 2.0 se basa en el modelo de servicio común SWE de OGC.

SOS 2.0 es altamente modular y sigue el patrón de diseño de núcleo/extensión (core/extensión) de OGC. La especificación OGC SOS 2.0 está formada por estos cuatro módulos o extensiones [44]:

- **Extensión del núcleo (Core Extension)**

Este módulo engloba las operaciones centrales e imprescindibles del servicio. Éstas son:

- **GetCapabilities**, que se corresponde con la operación que permite solicitar una descripción automática y completa del servicio.
- **GetObservation**, que se define como la operación empleada para solicitar los datos de sensores puros codificados en O&M 2.0

- **DescribeSensor**, que es empleada para solicitar información sobre un determinado sensor. La respuesta es codificada en un documento de instancia de SensorML 1.0.1.

- **Extensión mejorada (Enhanced Extension)**

Este módulo contempla estas dos operaciones:

- **GetFeatureOfInterest**. Esta operación se emplea en la tarea de solicitud de la representación codificada GML 3.2.1 de la Feature of Interest objetivo de la observación.
- **GetObservationById**, Esta operación permite solicitar los datos puros de los sensores mediante un determinado identificador de observación.

- **Extensión transaccional (Transactional Extension)**

En este módulo se definen las siguientes cuatro operaciones:

- **InsertSensor**. Esta operación se emplea para la tarea de publicación de nuevos sensores dentro del servicio.
- **UpdateSensorDescription**. Esta operación se define con el objetivo de actualizar la descripción de un sensor determinado.
- **DeleteSensor**. Esta operación permite la eliminación del servicio de un determinado sensor y toda la información relacionada con el mismo.
- **InsertObservation**. Esta operación se emplea para la publicación de observaciones para sensores registrados.

- **Extensión de manejo de resultados (Result Handling Extension)**

Este módulo incluye las siguientes cuatro operaciones:

- **InsertResultTemplate**. Esta operación es empleada para insertar una plantilla de resultados en un servidor SOS que describe la estructura de los valores de una solicitud InsertResult de GetResult.
- **InsertResult**. para cargar valores sin procesar según la estructura y la codificación definidas en la solicitud InsertResultTemplate

- **GetResultTemplate.** Esta operación se usa para obtener la estructura de resultados y la codificación de constelaciones de parámetros específicos
- **GetResult.** Esta operación es utilizada para obtener los datos sin procesar para las constelaciones de parámetros específicos

El estándar SOS define un subconjunto del global de operaciones como operaciones imprescindibles que todo SOS debe implementar. Este conjunto de operaciones imprescindibles se describen minuciosamente en los siguientes apartados de este mismo capítulo.

Se prevé que en el futuro sea probable la definición de nuevas extensiones sobre el marco original que define SOS 2.0, según aparezcan nuevas necesidades en el dominio y por lo tanto sea conveniente la realización de las adaptaciones oportunas del estándar a estas necesidades. Hay que destacar por tanto la idea de que la versión 2.0 de SOS no es una definición cerrada y definitiva del estándar para el tratamiento de los datos de sensores, por lo que se permite a cualquier equipo de desarrollo ampliar y añadir nuevos módulos bajo su propio criterio[45].

2.8.3 Conceptos fundamentales de SOS 2.0

A continuación se realiza una descripción pormenorizada de los conceptos fundamentales que intervienen en la definición del estándar SOS V2.0. Es recomendable revisar el documento original redactado y distribuido por el OGC en relación a las especificaciones particularizadas del estándar SOS para interiorizar y profundizar sobre todos y cada uno de los aspectos del estándar [46][47] [48].

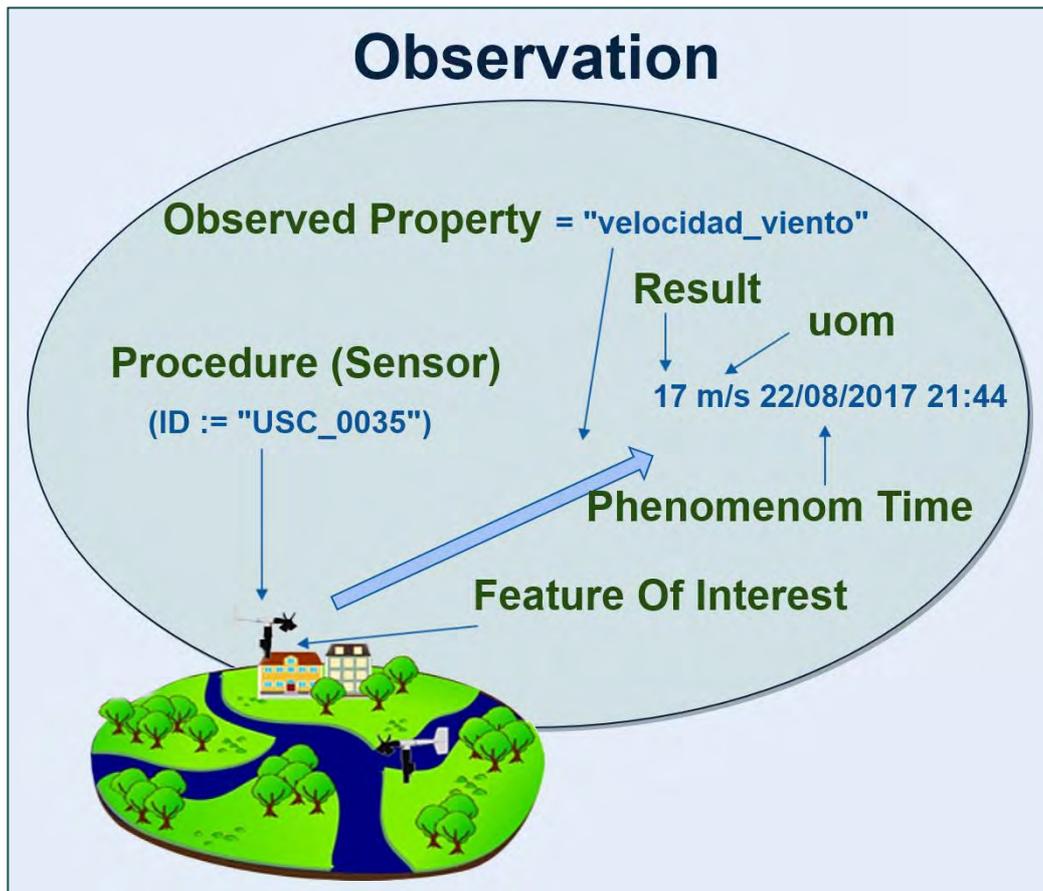


Fig. 19 Conceptos relativos a datos de observaciones en SOS

2.8.3.1 Offering

Los datos ofrecidos por un servicio SOS se agrupan en diversas Offerings u ofertas. Se pueden considerar las diferentes Offerings como secciones o apartados que clasifican los diversos datos según su naturaleza u origen. Cada Offering relaciona los datos de un mismo sensor o de una misma red de sensores, descritos como un Procedure [48].

Por ejemplo, un servicio SOS de meteorología podría presentar sus datos agrupados en las siguientes Offerings:

- datos de radar,
- medidas de estaciones meteorológicas terrestres,
- medidas de estaciones meteorológicas marinas,
- imágenes de satélite,
- etc.

2.8.3.2 Procedure

Un Procedure describe un sensor, una colección de sensores, o un proceso que genera un conjunto de observaciones. Cada Procedure ofrece:

- metadatos sobre las entradas y salidas del sensor,
- datos de procesado y calibración,
- información de contacto,
- la disponibilidad de los datos mediante las extensiones espacial y temporal,
- etc.

Generalmente los Procedures se detallan mediante el formato estándar que se define en el lenguaje SensorML. Los Procedures se pueden interpretar como fichas de metadatos acerca del sensor o conjunto de sensores, o procesos encargados de la obtención de los datos que proporciona el servicio.

Cada una de las Offerings está relacionada con una única Procedure, mientras que una Procedure puede ser utilizada en diferentes Offerings. Por ejemplo, una Procedure podría corresponderse con una “Red de Estaciones Meteorológicas”, y ésta misma red de estaciones ser utilizada en múltiples Offerings[48].

2.8.3.3 Feature of Interest

Cada una de las observaciones de un servicio SOS debe estar siempre asociadas a una Feature Of Interest (en adelante FoI), que usualmente define el lugar donde el fenómeno observado tuvo lugar. Por ejemplo, para imágenes satelitales, la FoI

correspondiente a cada una de estas observaciones. Podría identificarse como el polígono que determina la región de la superficie terrestre fotografiada. Otro ejemplo es una observación de temperatura, donde la FoI podría ser la ubicación del termómetro, siendo dicha ubicación un punto concreto del espacio representado con algún tipo de sistema, ya sean coordenadas espaciales o cualquier otro tipo.

Las FoIs pueden considerarse como el conjunto de lugares donde se realizan las diferentes mediciones de los datos[48].

2.8.3.4 Observed Property

Una Observed Property indica la propiedad, parámetro o magnitud medida en una determinada observación, como por ejemplo la Temperatura del agua, Reflectividad de la atmósfera, Número de vehículos, Dirección del viento, Salinidad del agua, Humedad relativa media, etc.

Una Observed Property puede ser de cuatro tipos:

- un valor numérico de una magnitud física, es decir, una cantidad y una unidad de medida, como por ejemplo 21°C.
- un valor lógico, que necesariamente tomará el valor verdadero o el valor falso, como por ejemplo, Activo = verdadero
- categórico, que necesariamente será un valor de entre un conjunto finito de valores, como por ejemplo, soleado / nublado / lluvioso.
- Descriptivo, que se corresponde con un texto representativo de esa Observed Property[48].

2.8.3.5 Observation

El concepto sobre el que se fundamenta todo el estándar SOS se denomina Observation, y se corresponde con un evento o acción cuyo resultado es una representación del valor de un sensor. Las Observations u observaciones están definidas mediante los siguientes conceptos:

- Phenomenon Time, que indica el momento en el que se realizó la actividad de medida.
- FeatureOfInterest, que se corresponde con la entidad que realiza la medición.
- ObservedProperty, que se corresponde con la característica o propiedad que se ha medido.
- Procedure, que indica el procedimiento de cómo se ha realizado la medición.

Por tanto, una Observation es el valor que toma una Observed Property en un determinado momento (Phenomenon Time) y un determinado lugar (Feature Of Interest). Por ejemplo: “La humedad relativa en Santiago de Compostela el 11/01/2017 a las 13:21 es del 66%” [48].

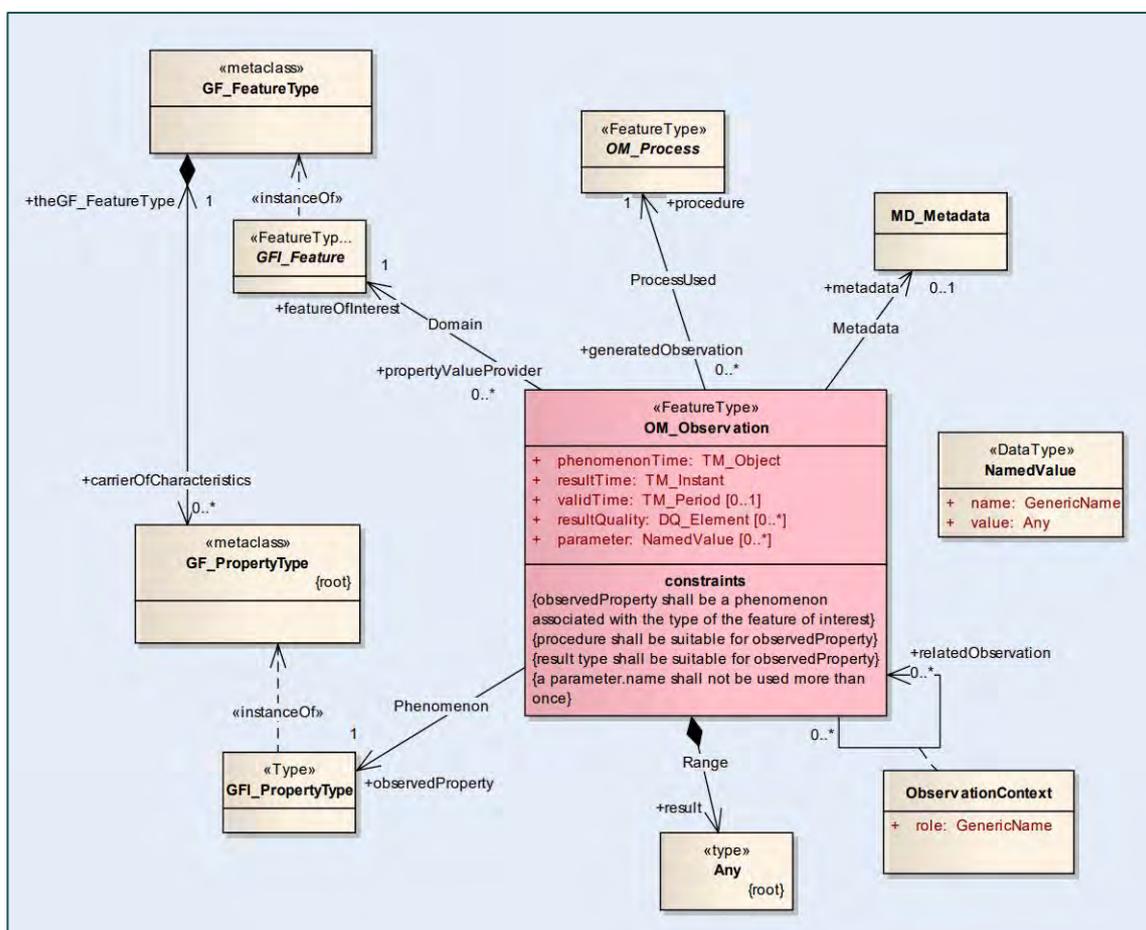


Fig. 20 Modelo de observaciones básico de O&M

2.8.3.6 Operaciones obligatorias de SOS

Las operaciones se corresponden con las peticiones o consultas de demanda de datos que un usuario puede realizar sobre un servicio SOS determinado. Todas y cada una de las peticiones SOS deben incluir los siguientes parámetros:

- Service: SOS
- Version: 2.0.0
- Request: el nombre de la petición, por ejemplo GetFeatureOfInterest
-

Como hemos indicado en apartados anteriores de este capítulo, el núcleo SOS o Core SOS establece como requisito obligatorio la implementación de tres operaciones. Son las siguientes:

GetCapabilities

GetCapabilities se corresponde con la operación que permite solicitar al servicio una descripción automática y completa del mismo. Puesto que la respuesta de un GetCapabilities es bastante extensa, el servicio contempla añadir filtros en la petición. Podemos incluir un parámetro sections para recuperar sólo parte del documento.

En concreto, la sección contents describe el servicio como una colección de Offerings.

Éste documento de Capabilities (sección contents) es el punto de entrada para descubrir cómo está estructurado un determinado servicio SOS, así como los datos que el servicio puede ofrecer. El documento contiene muchos identificadores de los distintos elementos (procedures, properties) pero no sus detalles, que deberán obtenerse mediante otras peticiones como DescribeSensor o GetFeatureOfInterest.

A continuación se incluye un ejemplo de petición GetCapabilities en formato JSON:

```
POST http://sensors.fonts.cat/sos/json
```

```
Content-Type: application/json
```

```
Content:
```

```
{
  "service": "SOS",
  "version": "2.0.0",
  "request": "GetCapabilities",
  "sections": ["Contents"]
}
```

DescribeSensor

La petición DescribeSensor acepta como parámetro un identificador de Procedure, y devuelve un documento SensorML que contiene metadatos acerca del (los) sensor(es) o proceso(s) que genera(n) las observaciones.

Esta petición se utiliza para ampliar detalles que no se ofrecen a través del GetCapabilities, en especial la descripción de las Observable Properties (sus nombres y unidades de medida).

Los contenidos más relevantes de este documento son:

- El identificador de la Procedure, un nombre corto y un nombre más largo,
- Una colección de palabras clave (útiles para servicios de búsqueda en catálogos de metadatos),
- Información de contacto,
- El período de tiempo de validez (redundante con la respuesta de Capabilities),
- El BBOX observado (redundante con la respuesta de Capabilities),
- La colección de Features of Interest (sus identificadores - nueva información que no se encuentra en el GetCapabilities),
- La colección de Offerings (sus identificadores) que se basan en esta Procedure,
- Una lista de salidas (Outputs): Una colección de ObservableProperties y su descripción: IDs, nombres, tipos y unidades de medida.

Ejemplo de petición DescribeSensor en formato JSON:

```
POST http://sensors.fonts.cat/sos/json
Content-Type: application/json
Contenido:
{
  "service": "SOS",
  "version": "2.0.0",
  "request": "DescribeSensor",
  "procedure": "http://sensors.portdebarcelona.cat/def/weather/procedure",
  "procedureDescriptionFormat": "http://www.opengis.net/sensorML/1.0.1"
}
```

GetObservation

Esta operación es la que permite hacer peticiones sobre determinados conjuntos de observaciones. Una petición GetObservation acepta los siguientes parámetros:

- Un offering,
- Una colección de FeatureOfInterest,
- Una colección de ObservedProperties,
- Filtros espaciales y/o temporales.

El filtrado es especialmente interesante, puesto que pueden restringirse las búsquedas de datos a, por ejemplo, un período de tiempo o un área geográfica concreta.

Ejemplo de petición GetObservation en formato JSON:

```
POST http://sensors.fonts.cat/sos/json
Content-Type: application/json
Contenido:
```

```
{
  "service": "SOS",
  "version": "2.0.0",
  "request": "GetObservation",
  "offering": "http://sensors.portdebarcelona.cat/def/weather/offerings#10m",
  "featureOfInterest": ["http://sensors.portdebarcelona.cat/def/weather/features#P3"],
  "observedProperty": ["http://sensors.portdebarcelona.cat/def/weather/properties#31"],
  "temporalFilter": [{
    "equals": {
      "ref": "om:resultTime",
      "value": "latest"
    }
  }]
}
```

La respuesta es una colección de observaciones, donde cada observacion consta de:

- El identificador del Offering del que procede,
- El identificador del Procedure que la generó,
- La Feature of Interest a la que se refieren. Con: descripción completa, ID, nombre y geometría,
- El identificador de la Property que se ha observado,
- PhenomenonTime que es el instante cuándo ha sucedido lo que se ha medido, y resultTime que es el momento cuándo se ha medido el dato,
- Y el resultado, que consta de un valor y una unidad de medida (uom).

Así, la respuesta completa es minuciosa, conteniendo centenares o miles de repeticiones sucesivas de algunos de los elementos descriptivos en el mismo documento de respuesta. Esto es así puesto que en la parte del cliente se realizan parseos de los datos atendiendo a la estructura que el servidor SOS contempla en sus respuestas.

GetFeatureOfInterest

La operación `GetFeatureOfInterest` acepta una `procedure` como parámetro, y devuelve todas las `Features of Interest` relacionadas con dicho `procedure`. De hecho, las `Features of Interest` están vinculadas a cada una de las `Observation`, pero esta operación nos devuelve una suerte de inventario de todos sus posibles valores. Es útil para obtener los detalles de las diversas localizaciones, como sus nombres y geometrías. Así que generalmente se utiliza ésta operación para poder dibujar un mapa o un selector de `Features` por nombre.

Ejemplo de petición `GetFeatureOfInterest` en formato JSON:

```
POST http://sensors.fonts.cat/sos/json
Content-Type: application/json
Contenido:
{
  "service": "SOS",
  "version": "2.0.0",
  "request": "GetFeatureOfInterest",
  "procedure": "http://sensors.portdebarcelona.cat/def/weather/procedure"
}
```

2.9 Arquitecturas de clientes SOS

En la actualidad encontramos diferentes clientes web de acceso a datos de observación ofrecidos por servidores que cumplen la normativa europea INSPIRE bajo el estándar SOS. La gran mayoría de estos desarrollos se basan en el acceso a los datos de observaciones siguiendo la definición de operaciones que la versión 1.0 del estándar SOS contempla y combinando dicha interfaz con otros interfaces estándares. Estos clientes difieren entre sí en varios aspectos, siendo el diseño de la arquitectura uno de los puntos críticos. Así encontramos ejemplos de clientes basados en dos conceptos arquitectónicos diferentes:

2.9.1 Integración Directa con servidores SOS

Esta arquitectura es la más sencilla, pero no por ello la menos eficiente. En ella el cliente y el servidor SOS se comunican directamente, sin artefactos o componentes intermedios. Los clientes utilizan las operaciones dedicadas del SOS para recuperar los datos relevantes del sensor o conjunto de sensores de la estación meteorológica. Esto implica que cualquier lógica para las interacciones, así como la codificación y decodificación de peticiones SOS y las respuestas correspondientes a dichas peticiones deben ser ejecutadas directamente por los propios clientes web, en la propia máquina del usuario demandante de datos de observación. Con este tipo de arquitectura se alcanza una notable descarga de trabajo del servidor delegando tareas a la parte cliente. Esta arquitectura es conveniente que sea llevada a cabo en sistemas donde el volumen de acceso a los datos es elevado y por lo tanto crítico para el rendimiento del servidor. Es importante resaltar que la plataforma de destino de dichos clientes es un punto crítico a la hora de seleccionar este modelo de arquitectura como solución técnica, puesto que las prestaciones de las máquinas destino no se conocen de antemano, pudiendo limitar de forma crítica el volumen de datos que el cliente puede soportar.



Fig. 21 Arquitectura de cliente SOS de Integración Directa con servidores SOS

2.9.2 Componente intermedio que realiza la lógica de negocio

En contraste con la comunicación directa, existen desarrollos que emplean un componente intermedio adicional para realizar tareas de lógica de negocio intermedias arbitrarias. Este componente intermedio funciona con hardware de alto rendimiento y encapsula la comunicación con el servidor SOS. Además, puede ofrecer otras funcionalidades, como almacenar metadatos de sensor en caché, realizar tareas de generalización (véase más adelante) o ofrecer una interfaz ligera (por ejemplo, REST / JSON) a los clientes. Siguiendo este patrón de arquitectura, tenemos la ventaja de que las tareas de negocio complejas, como la decodificación de grandes conjuntos de datos XML o el almacenamiento en caché de metadatos, se desplazan de los clientes a un componente del servidor. Sin embargo, el lado del cliente no está autorizado a agregar interactivamente otro servidor SOS y visualizar sus datos, ya que este conocimiento tiene que ser añadido al componente intermedio en sí.



Fig. 22 Arquitectura de cliente SOS con componente intermedio de lógica de negocio

La gran mayoría de clientes SOS existentes en la actualidad se basan en el segundo modelo de arquitectura, por lo que la integración de los datos no puede ser realizada en

ninguno de los casos en la parte del cliente. Además estos clientes han sido desarrollados para la versión 1.0 del estándar SOS.

2.10 52°North

La iniciativa de software de código abierto 52°North se define como una red internacional abierta de expertos de los ámbitos de investigación, industria y administración pública. Su principal objetivo es fomentar la innovación en el campo de la geoinformática a través de procesos colaborativos de I+D. Las comunidades de I+D involucradas en 52°North se encargan de desarrollar nuevos conceptos y tecnologías, como por ejemplo, herramientas software para la gestión de datos de sensores en tiempo real, integrando tecnologías de geoprocésamiento en IDEs. Además los miembros de 52°North se encargan de evaluar nuevas tendencias en diferentes campos, como el Internet de las Cosas, la Web Semántica o los Datos Abiertos Vinculados, para encontrar formas de potenciar su utilización en la práctica. Todos los miembros de 52°North tienen un registro largo y excepcional en el dominio de la geoinformática y contribuyen activamente al desarrollo de estándares internacionales, como por ejemplo, en W3C, ISO, OGC o INSPIRE.

Todo el software desarrollado dentro de este proceso de desarrollo colaborativo se publica bajo una licencia de código abierto. 52°North es una entidad de confianza y bien establecida a lo largo de los años en el ámbito de la geoinformática. Su software es ampliamente utilizado en entornos operativos de tecnologías de la información, laboratorios de investigación y educación.

52°North fue fundada en 2004 por el Instituto de Geoinformática de la Universidad de Muenster (IFGI) como una iniciativa informal centrada en el software geoespacial libre y de código abierto. 52°North fundamenta su trabajo en la idea de mejorar la cooperación existente entre organizaciones, llegando incluso a plantearse la asociación con organizaciones de investigación e industria mediante el desarrollo de un modelo de negocio y un modelo de organización apropiados a cada circunstancia. El elemento

central de este modelo de negocio es publicar los resultados del proceso de desarrollo de software colaborativo bajo licencias de código abierto [49][50].

2.10.1 52°North SOS 2.0

52°North SOS es probablemente la implementación más relevante del principal estándar OGC sobre sensores. 52°North SOS está formado por un conjunto de módulos o extensiones que implementan de forma rigurosa el estándar SOS de la organización OGC para servicio de datos de observaciones de sensores cuya última versión (4.0), desarrollada en 2012, implementa la versión más reciente del estándar SOS, la versión 2.0 definida por el OGC. Nos encontramos por tanto, ante una herramienta servidor de datos de observaciones SOS. Desgraciadamente esta implementación del Servidor SOS no tiene asociada una interfaz de usuario cliente a su altura. 52°North SOS cuenta únicamente con una sencilla consola ejemplificadora del servicio que puede ser empleada para ejecutar consultas en formato texto y para mostrar la respuesta, también en formato texto, obtenida directamente del servidor [51][52].

52°North SOS devuelve las observaciones de las respuestas codificadas en lenguaje XML bajo el estándar O&M de OGC. Además la información sobre los sensores que devuelve el servidor 52°North SOS está codificada en lenguaje XML bajo el modelo SensorML. 52°North SOS ofrece una amplia gama de capacidades interoperables para descubrir, unir y consultar sensores individuales, plataformas de sensores o constelaciones de sensores en red en entornos en tiempo real, de ficheros o simulados.

En la documentación oficial [44] podemos encontrar la Tabla 1 más detalladamente. Aquí se incluyen únicamente las operaciones relevantes a nuestro trabajo siguiendo esta nomenclatura:

- = implementado
- = no implementado
- = no especificado

Extensión	Operación	Descripción	KVP	SOAP	POX	REST	JSON	EXI
Núcleo (core)	GetCapabilities	solicita una autodescripción del servicio SOS	•	•	•	Ver doc	•	•
Núcleo (core)	GetObservations	solicita los datos de sensores puros	•	•	•	Ver doc	•	•
Operaciones mejoradas (Enhanced Operations)	GetFeaturesOfInterest	solicita las features del SOS	•	•	•	Ver doc	•	•

Tabla 1 Resumen de operaciones y bindings de 52ºNorth SOS

Un binding se corresponde con el método por el cual se comunican clientes y servidores SOS 2.0. Se corresponde con el formato en que se desarrolla la comunicación, haciendo referencia a protocolos, tipos de operación y lenguaje empleado. Así los bindings de SOS se detallan en el siguiente listado:

Nombre	Descripción
KVP	Codificación de solicitudes mediante Par Clave Valor a través de HTTP GET
SOAP	Codificación basada en SOAP de todas las extensiones vía HTTP POST
POX	Codificación basada en XML de todas las extensiones vía HTTP POST (sin encapsulado SOAP)
REST	Binding restful para todas las operaciones útiles de recursos que están disponibles en SOS

JSON	<p>Codificación basada en JSON de todas las extensiones vía HTTP POST.</p> <p>La implementación de JSON se basa tanto como sea posible en las definiciones abstractas de las especificaciones, pero aún no se ha especificado y no todas las funciones son compatibles.</p>
EXI	<p>Codificación basada en Efficient XML Interchange (EXI) Format 1.0 de todas las extensiones vía HTTP POST y peticiones codificadas en POX.</p> <p>Las peticiones SOAP actualmente no son compatibles a través del binding EXI</p>

Tabla 2 Bindings empleados actualmente en 52°North SOS server

La muestra detalladamente cómo es la arquitectura en 3 capas que actualmente emplea la implementación 4.X de 52°North SOS. Se detalla en ella como se emplean artefactos propios de la implementación en java.

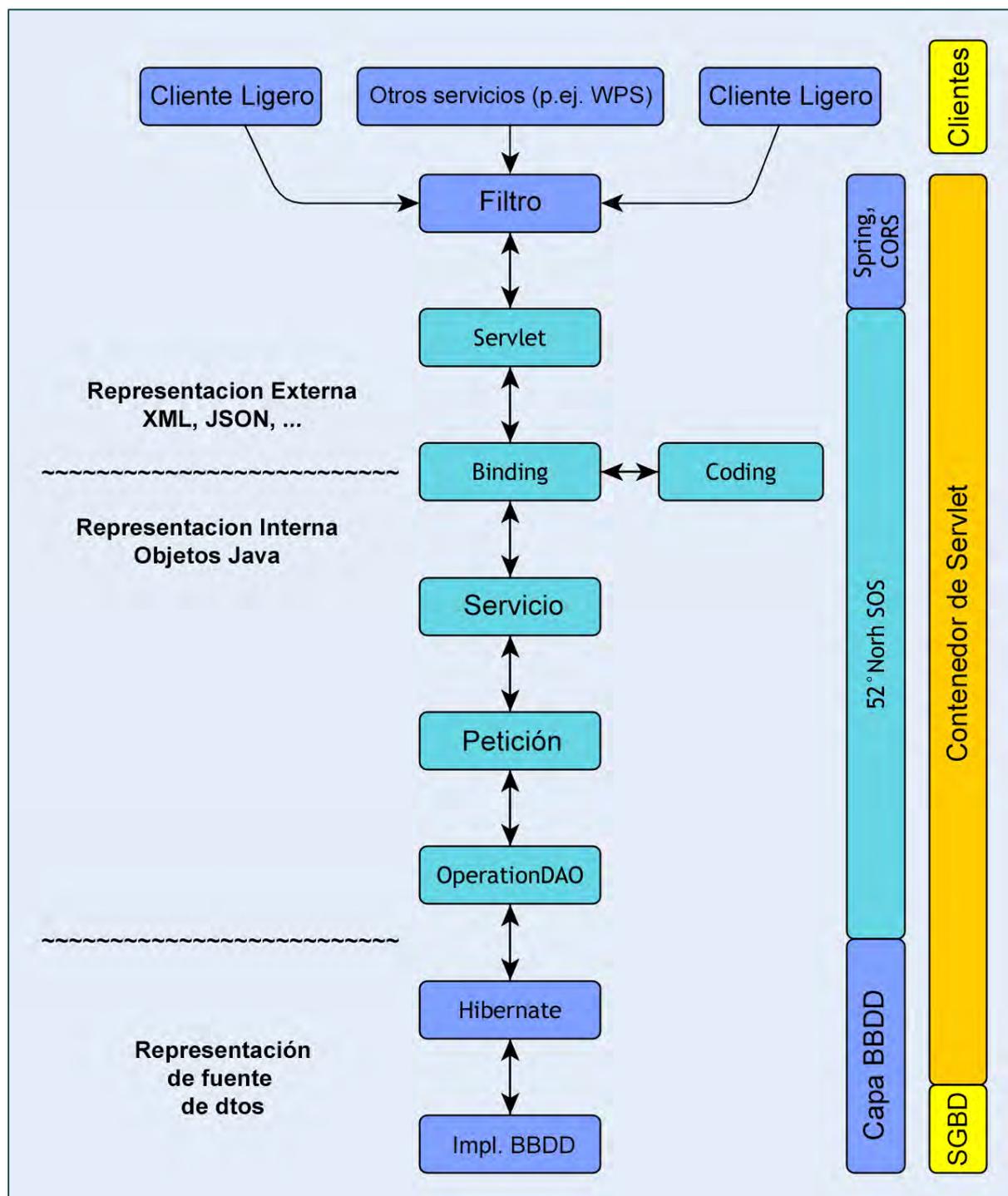


Tabla 3 Elementos de la arquitectura de 52ºNorth SOS

2.11 Herramientas empleadas

Como herramienta software base para la implementación de este proyecto se ha empleado 52º North SOS Server versión 4.3.6 [53] modificada por el grupo COGRADE. Los requisitos técnicos de dicho paquete son los siguientes:

- **Java Runtime Environment (JRE) 7.0** o superior
- **Servidor de aplicaciones** compatible con Java Servlet-API 2.5
- **Sistema de gestión de bases de datos**
- **Apache Maven 3**

Además del paquete 52º North SOS V2 semántico y sus dependencias se han empleado las siguientes herramientas, paquetes de software, plugins de extensión de funcionalidad y/o librerías:

- **NetBeans 8.1.** Es un entorno de desarrollo integrado libre, construido principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE es una herramienta libre y gratuita que no presenta restricciones de utilización.
- **Apache Maven 3.** Es una herramienta para la gestión y construcción de proyectos de software escritos en lenguaje Java. Maven utiliza un Project Object Model (POM) para describir el proyecto de software a construir, sus dependencias de otros módulos y componentes externos, y el orden de construcción de los elementos. Viene con objetivos predefinidos para realizar ciertas tareas claramente definidas, como la compilación del código y su empaquetado.
- **Apache Tomcat V8.0.27.0.** Es una herramienta que funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Oracle Corporation y se integra perfectamente con el IDE de desarrollo Netbeans.

- **JRebel.** Es una extensión del IDE de desarrollo Netbeans. JRebel realiza un seguimiento rápido del desarrollo de aplicaciones Java al omitir los pasos de compilación y despliegue. Estas tareas normalmente consumen muchos recursos temporales durante el ciclo de vida de desarrollo de aplicaciones Java. JRebel permite que los desarrolladores aumenten su productividad ya que éstos pueden chequear los cambios del código fuente en tiempo real.
- **Javascript.** Aabreviado comúnmente como JS, es un lenguaje de programación interpretado, con sintaxis muy parecida a Java. Sus características más importantes son:
 - orientado a objetos
 - basado en prototipos
 - imperativo
 - débilmente tipado
 - dinámico
 - empleado en desarrollo de clientes web
 - implementado como parte de navegadores web
- **Librerías javascript:**
 - **Jquery v2.1.4.** jQuery es una biblioteca multiplataforma de JavaScript, creada inicialmente por John Resig, que permite:
 - simplificar la manera de interactuar con los documentos HTML,
 - manipular el árbol DOM,
 - manejar eventos,
 - desarrollar animaciones
 - agregar interacción con la técnica AJAX a páginas web.
 - **GeoExt v1.1** Es un kit de herramientas JavaScript enfocadas para el desarrollo de aplicaciones basadas en mapas web enriquecidos. GeoExt

permite reunir el conocimiento geoespacial que ofrece OpenLayers con las interfaces de usuario que permite desarrollar Ext.js para ayudar a los desarrolladores a crear aplicaciones web de información geográfica. Emplea JavaScript como lenguaje de programación.

- **Sencha Ext.js v3.3.1** que aprovecha las características de HTML5 de los navegadores modernos. Ext.js cuenta con más de cien diferentes componentes de interfaz de usuario de alto rendimiento
- **OpenLayers v2.10** es una librería que permite incluir fácilmente mapas dinámicos en cualquier sitio web. OpenLayers es una librería Javascript, de uso totalmente libre bajo licencia BSD, para acceder, manipular y mostrar mapas en páginas web. Proporciona una API que permite la creación de clientes web para acceder y manipular información geográfica procedente de fuentes muy variadas.

OpenLayers plugin permite añadir un importante número de servicios, como pueden ser:

- Google Maps (Physical, Streets, Hybrid, Satelite)
 - OpenStreetMap
 - Bing (Road, Aerial, Aerial with labels)
 - Apple iphoto map
-
- **HighCharts.js v4.1.7** Es una librería que permite crear gráficas interactivas fácilmente para sitios web. Highcharts es la API de gráficos más simple y a la vez más flexible del mercado.
 - **Stupid jQuery table.** Esta es una librería jQuery muy sencilla que permite la creación y ordenación de tablas.
 - **Simple Datetimepicker.** Es una librería jQuery de fácil empleo de calendarios, para escoger fechas o períodos de fechas en herramientas web.

- **MagicSuggest.js.** Es una librería bootstrap que permite la selección múltiple en elementos de interfaz gráfica combobox.
- **Bootstrap 3.** Es un framework o conjunto de herramientas de código abierto empleadas para el diseño de sitios y aplicaciones web. Contiene, entre otros:
 - plantillas de diseño con tipografía,
 - formularios,
 - botones,
 - cuadros,
 - menús de navegación
 - otros elementos de diseño basado en HTML y CSS
 - extensiones de JavaScript opcionales
- **AJAX**, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para el desarrollo de aplicaciones web interactivas. Estas aplicaciones web se ejecutan en el lado del cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.
- **Microsoft SQL Server 2014.** Es un sistema de gestión de bases de datos de modelo relacional. El lenguaje que emplea es Transact-SQL (TSQL), una implementación del estándar ANSI del lenguaje SQL, utilizado para manipular y recuperar datos, crear tablas y definir relaciones entre ellas.
- **Firefox Developer Edition.** Es una edición especial del navegador web Mozilla Firefox que permite crear y depurar experiencias web con poderosas herramientas de desarrollo de código abierto. Algunas de sus herramientas:
 - Herramienta Inspector que permite revisar y perfeccionar el código para crear diseños perfectos.

- Herramienta Consola que se corresponde con una herramienta destinada a encontrar errores de seguridad, redes CSS y JavaScript.
 - Herramienta Depurador de JavaScript compatible con multiples infraestructuras
 - Herramienta Red que permite monitoriza las solicitudes de redes que pueden ralentizar o bloquear tu sitio.
 - Editor de estilos, empleado para editar y administrar todas tus hojas de estilo CSS en tu navegador.
 - Herramienta Borrador que permite editar, escribir y ejecutar código JavaScript en tiempo real.
- **Firebug**, es una extensión que se integra con Firefox para ofrecer a los desarrolladores una gran cantidad de herramientas a su alcance mientras se navega. Firebug permite editar, depurar y monitorear CSS, HTML y JavaScript en cualquier página web
 - **Firefox Clear Caché**. Es una extensión del navegador Mozilla Firefox que permite gestionar de forma manual la liberación de la memoria caché del navegador. De esta forma, se permite la liberación bajo demanda.
 - **OpenOffice**. Se corresponde con una suite ofimática libre, de código abierto y distribución gratuita, de la que hemos utilizado : o su procesador de textos, llamado OpenOffice Writer o su hoja de cálculo, llamada OpenOffice Calc o su programa de presentación, llamado OpenOffice Impress o su editor de gráficos vectoriales, llamada OpenOffice Draw
 - **Gliffy Diagrams**. Es una herramienta de diagramación que simplifica la creación de diagramas complejos, dotando un aspecto profesional a los resultados. Ofrece soporte para diagramas de flujo, diagramas UML o diagramas de red, entre otros.
 - **GanttProject**. Es una herramienta de escritorio multiplataforma para la programación y gestión de proyectos, compatible con la mayoría de programas de gestión de proyectos del mercado. Es gratuita, libre y su código es opensource.
-

- **Balsamiq**, nos permite crear, editar y exportar dibujos de alta calidad, facilitando la tarea de diseño de interfaces gráficas de aplicaciones mediante wireframes y maquetas. Esta herramienta es utilizada para producir nuevas ideas de experiencia de usuario con un coste temporal pequeño.
- **Notepad++**, es un editor de código fuente que soporta un gran conjunto de lenguajes de programación. Funciona en entorno MS Windows y se distribuye bajo los términos de la licencia GNU GPL.
- **GitLab**. Es una herramienta de muy fácil manejo para el control de versiones y revisión de código fuente. GitLab es un proyecto de código libre basado en Git que se puede instalar en servidores propios permitiendo manejar repositorios privados, sin coste alguno.
- **Assembla**. Es una empresa que provee herramientas de colaboración y de seguimiento de errores y tareas basadas en la nube para organizar y administrar proyectos software.
- **Adobe Photoshop CS5**, es un editor de gráficos rasterizados desarrollado por Adobe Systems principalmente usado para el retoque de fotografías y gráficos. Se ha empleado para corroborar la información relativa a las propiedades de las imágenes capturadas.
- **Gmail**. Es un servicio de correo electrónico proporcionado por Google que ha servido como principal herramienta de comunicación con los directores del proyecto y los clientes.
- **Google Drive**. Es un servicio de alojamiento de archivos multiplataforma en la nube. El servicio permite a los usuarios almacenar y sincronizar archivos en línea y entre ordenadores, para así compartir archivos y carpetas con otros.

Capítulo 3. Gestión del proyecto

El objetivo de la gestión del proyecto consiste en cumplir el alcance del proyecto teniendo en cuenta las restricciones temporales y económicas. Para ello se aplican una serie de técnicas para supervisar el proyecto durante todo su ciclo de vida.

3.1 Análisis y gestión de riesgos

Los riesgos son eventos que pueden producirse con cierta probabilidad y que provocan consecuencias positivas o negativas en el desarrollo preestablecido del proyecto. Los riesgos involucran por un lado un componente de incertidumbre sobre lo que pueda suceder en el futuro, y por otro lado un componente de pérdida o ganancia donde el riesgo que se corre se convierte en realidad. Es imprescindible la realización de un correcto análisis y gestión de esta serie de riesgos para esquivar los eventos que pongan en riesgo el éxito del proyecto y beneficiarnos de las oportunidades de los riesgos positivos. Para ello es imprescindible cuantificar el grado de incertidumbre y de pérdida o ganancia asociada con cada uno de los riesgos [54].



Fig. 23 Diagrama del proceso de gestión de riesgos

Los riesgos afectan a futuros acontecimientos, por lo que la salud del proyecto se verá determinada por la intensidad en que el equipo de desarrollo se pregunte qué riesgos podrían hacer que el proyecto fracasara. Dado que el riesgo implica cambio, el equipo debe preguntarse cómo afectan estos cambios en los requisitos del cliente, en las tecnologías de desarrollo, al cumplimiento de la planificación y al éxito en general. Por lo tanto el riesgo enfrenta al equipo de desarrollo a elecciones como por ejemplo qué métodos y herramientas deben emplear, gente debería estar implicada, cuánta importancia hay que darle a la calidad.

Para cubrir todas estas necesidades surgen el análisis y la gestión del riesgo. Éstas se definen como una serie de pasos que ayudan al equipo del software a comprender y gestionar la incertidumbre. Estas tareas deben ser desarrolladas por todos los que están involucrados en el proceso del software. Los pasos del análisis y gestión del riesgo se identifican en la Fig. 23, y se corresponden con:

- **Identificación de riesgos**, que da como resultado un listado de riesgos potenciales.
- **Análisis de riesgos**, que genera un listado de priorización de riesgos.
- **Planificación de riesgos**, que resulta en un conjunto de planes de prevención y contingencia
- **Supervisión de riesgos**, que da como resultado una valoración de riesgos.

Las primeras dos de estas etapas consisten en la identificación y análisis de los riesgos. En estas etapas es importante definir la probabilidad que tiene cada riesgo identificado de llegar a producirse, así como el impacto que generaría en caso de que llegara a producirse. A partir de estas dos variables se realiza el cálculo de la exposición a cada riesgo a partir de los valores de la Tabla 4.

A continuación se determina la etapa de planificación generando los planes de prevención y contingencia. Para ello, se ha creado un plan de prevención (acciones para evitar que el riesgo ocurra) para cada riesgo y además se ha creado también un plan de contingencia (acciones a realizar en caso de que se produzca el riesgo para los riesgos con una exposición alta) [55].

		Probabilidad		
		Alta	Media	Baja
Impacto	Alto	Alta	Alta	Media
	Medio	Alta	Media	Baja
	Bajo	Media	Baja	Baja

Tabla 4 Guía para el cálculo de exposición a riesgos

Para la representación en la presente memoria de la gestión de los riesgos se ha empleado una plantilla resumen de cada uno de los riesgos, tal y como se detalla en la Tabla 5. En ella se incluyen un identificador del riesgo, que está compuesto por unas siglas que hacen referencia al tipo de riesgo, seguidas de un dígito único. Además en esta plantilla se contempla una descripción del nombre del riesgo identificado, así como una descripción detallada del mismo. Además se incluye el impacto del riesgo y su probabilidad de aparición, así como su exposición asociada. En caso de crearse planes de prevención y contingencia, también se incluyen en cada tabla.

Hay que aclarar que la gestión de riesgos desarrollada en este proyecto se ha realizado únicamente sobre los riesgos que puedan producir efectos negativos en el correcto desarrollo y consecución del proyecto.

ID	RXX-Y
Nombre	
Descripción	
Probabilidad de suceso	

Impacto	
Exposición	
Plan de prevención	
Plan de contingencia	

Tabla 5 Plantilla empleada en el análisis y gestión de los riesgos

3.1.1 Riesgos de gestión del proyecto

Este conjunto de riesgos ponen en peligro el plan inicial de desarrollo del proyecto. Si estos riesgos surgen y se cumplen, el proyecto requerirá de una inversión mayor tanto en recursos financieros como en esfuerzo y dedicación. Los riesgos contemplados en esta categoría están relacionados con [56]:

- Presupuesto. Los riesgos de esta categoría implicarán la ampliación del presupuesto originalmente contemplado.
- Planificación. Si surgen riesgos relacionados con la planificación significa que será necesario invertir más tiempo para conseguir los resultados de proyecto inicialmente planteados.
- Personal. Estos riesgos se traducen en una necesidad de ampliación o formación del equipo involucrado en el proyecto.
- Recursos. Si aparecen riesgos relacionados con los recursos, éstos se pueden transformar en la necesidad de mejorar el equipo de instrumentación de desarrollo.

A continuación, se describen los riesgos de gestión del proyecto que se han identificado, ordenados según su nivel de exposición. Por lo tanto se agrupan de esta forma los riesgos asociados a la inadecuada aplicación de conocimientos, habilidades,

herramientas y técnicas en el desarrollo de las actividades del proyecto para satisfacer los requisitos del mismo.

ID RGP-1	
Nombre	Retraso respecto a la planificación inicial
Descripción	Debido a la inexperiencia en proyectos de este calibre, o por la aparición de algún otro riesgo, es posible que se produzca un retraso con respecto a la planificación temporal realizada en la fase inicial.
Probabilidad de suceso	Media
Impacto	Alto
Exposición	Alta
Plan de prevención	Se realizarán reuniones cada semana con alguno de los directores del proyecto, que ya tienen experiencia en proyectos de este tipo, en las que se podrá controlar el avance del proyecto.
Plan de contingencia	Realizar horas extras o eliminar algún requisito opcional, según el nivel de retraso

Tabla 6 Riesgo de gestión del proyecto RGP-1

ID RGP-2	
Nombre	Cambios en los requisitos
Descripción	Durante el desarrollo del proyecto pueden surgir nuevos requisitos haciendo necesaria la implementación de nuevas funcionalidades.
Probabilidad de suceso	Media
Impacto	Medio
Exposición	Media
Plan de prevención	Se realizarán reuniones para revisar el trabajo realizado e identificar los cambios a realizar cuanto antes.

Tabla 7 Riesgo de gestión del proyecto RGP-2

ID RGP-3	
Nombre	No satisfacción de todos los requerimientos solicitados por los clientes
Descripción	Puede ocurrir que, la definición ambigua de algún requisito en las reuniones con los clientes pueda llevar a confusión en los mismos o a que quede alguna funcionalidad sin ser establecida de modo preciso, encontrándonos en fases avanzadas del proyecto con funcionalidades no implementadas que el cliente desea.

Probabilidad de suceso	Baja
Impacto	Alto
Exposición	Media
Plan de prevención	Establecer correctamente la definición de todos los requisitos del proyecto en las primeras reuniones con el cliente.

Tabla 8 Riesgo de gestión del proyecto RGP-3

ID RGP-4	
Nombre	Aspectos desconocidos del proyecto necesitan más tiempo del estimado
Descripción	Las áreas desconocidas del proyecto implican más tiempo del estimado inicialmente en cuanto a las fases de diseño y de implementación.
Probabilidad de suceso	Baja
Impacto	Alto
Exposición	Media
Plan de prevención	Se estimará con más precisión y en relación a actividades similares desarrolladas en otros proyectos tanto la fase de diseño como la de implementación.

Tabla 9 Riesgo de gestión del proyecto RGP-4

3.1.2 Riesgos técnicos

Los riesgos contemplados en esta categoría ponen en peligro la calidad del proyecto resultante. Si estos riesgos aparecen durante el desarrollo, el proyecto será más complejo de lo estimado inicialmente. Estos riesgos están relacionados con:

- Requisitos. Harán que se necesiten más condiciones o mejor definidas.
- Diseño. Estos riesgos implican una arquitectura de mayor calidad.
- Implementación. Los riesgos provocarán que se necesite más tiempo o personal para completar el proyecto.
- Interfaz. Los riesgos relacionados con la interfaz obligarán a realizar estudios de usabilidad e interacción más completos.
- Verificación. Si aparecen riesgos relacionados con la verificación será necesario realizar baterías de pruebas más exhaustivas.
- Mantenimiento. Si surgen riesgos relacionados con el mantenimiento esto tendrá que traducirse en más carga de trabajo una vez se haya realizado la entrega final del producto software.
- Incertidumbre técnica: Los riesgos provocarán que se necesite mayor conocimiento sobre los diferentes ámbitos técnicos involucrados en el proyecto.

Tecnologías desconocidas: se necesita mayor conocimiento sobre cómo realizar el proyecto de forma óptima. A continuación, se describen los riesgos técnicos que se han identificado, ordenados según su nivel de exposición. La gestión de riesgos de este proyecto se realizará únicamente sobre los que puedan producir efectos negativos.

ID	RT-1
Nombre	Baja eficiencia del cliente SOS
Descripción	Debido a que no se tienen datos sobre la eficiencia del conjunto de operaciones que se van a utilizar en el contexto de trabajo, es posible que no se ejecuten con la

	eficiencia esperada.
Probabilidad de suceso	Media
Impacto	Alto
Exposición	Alta
Plan de prevención	Se implementaran en primer lugar las operaciones que afecten en mayor medida al rendimiento del sistema, y posteriormente el trabajo se centrará en la mejora de la eficiencia del código de las mismas.

Tabla 10 Riesgo técnico RT-1

ID RT-2	
Nombre	Error de funcionamiento en el equipo de desarrollo.
Descripción	La maquina desarrollo presenta un funcionamiento incorrecto que imposibilita el desarrollo normal del proyecto.
Probabilidad de suceso	Baja
Impacto	Alto
Exposición	Media

Plan de prevención	Se realizarán tareas de mantenimiento de la máquina de desarrollo y se mantendrá una copia actualizada de los ficheros de desarrollo en repositorios privados en la nube.
---------------------------	---

Tabla 11 Riesgo técnico RT-2

ID RT-3	
Nombre	Error de consumo de recursos
Descripción	La herramienta desarrollada demanda demasiados datos del servidor o consume demasiados recursos
Probabilidad de suceso	Media
Impacto	Alto
Exposición	Alta
Plan de prevención	La herramienta desarrollada debe presentar filtros de recursos consumidos, así como filtros de demanda de datos del servidor, ajustables a la máquina de ejecución de la herramienta.

Tabla 12 Riesgo técnico RT-3

ID	
RT-4	
Nombre	Maquina de desarrollo inutilizada
Descripción	La máquina de desarrollo presenta un funcionamiento incorrecto que imposibilita el desarrollo normal del proyecto. Se necesitará invertir tiempo en montar y preparar otra máquina de desarrollo con todas las herramientas operativas y funcionales.
Probabilidad de suceso	Baja
Impacto	Alto
Exposición	Medio
Plan de prevención	Se crearán imágenes de disco backup realizadas una vez que todas las herramientas necesarias para el desarrollo del proyecto estén instaladas. De esta forma se podrá cargar dicha imagen en una nueva máquina en poco tiempo, y por lo tanto se reducirán notablemente los tiempos de retraso. También se crearán copias regulares de los ficheros de desarrollo y documentación en repositorios basados en sistemas de control de versiones

Tabla 13 Riesgo técnico RT-4

3.1.3 Riesgos de personal

A continuación, se describen los riesgos asociados a recursos humanos que se han identificado, ordenados según su nivel de exposición. La gestión de riesgos de este proyecto se realizará únicamente sobre los que puedan producir efectos negativos.

ID		RP-1	
Nombre	Limitada disponibilidad del investigador/desarrollador		
Descripción	Puede ocurrir que el investigador/desarrollador no tenga total disponibilidad		
Probabilidad de suceso	Alta		
Impacto	Alto		
Exposición	Alta		
Plan de prevención	Aplicar el máximo de horas disponibles al desarrollo		
Plan correctivo	Aumentar la carga de trabajo de desarrollo los fines de semana y festivos.		

Tabla 14 Riesgo de personal RP-1

ID	RP-2
Nombre	El investigador necesita un tiempo extra para aprender un framework, metodología, lenguaje o plugin de programación nuevo.
Descripción	Puede ocurrir que durante el desarrollo del proyecto sea necesario invertir tiempo extra en la adquisición de conocimientos de herramientas o tecnologías que hasta el momento no conocía el investigador.
Probabilidad de suceso	Media
Impacto	Alto
Exposición	Alta
Plan de prevención	Dedicar horas dentro de planificación a tareas formativas
Plan correctivo	Realizar el comienzo del desarrollo por las tareas que se han identificado con una dificultad asociada de grado menor, para tomar contacto con las tecnologías y modelos de desarrollo.

Tabla 15 Riesgo de personal RP-2

ID		RP-3
Nombre	El investigador trabaja más lento de lo esperado	
Descripción	Puede darse el caso que durante el desarrollo del proyecto el investigador trabaje más lento de lo inicialmente estimado, debido a diferentes razones, como inexperiencia, falta de motivación, revisión excesiva de tareas finalizadas, etc	
Probabilidad de suceso	Baja	
Impacto	Alto	
Exposición	Media	
Plan de prevención	Desarrollar un plan de seguimiento del modo de trabajo del investigador, identificando los tiempos de cada tareas y determinando las posibles causas del retraso.	
Plan correctivo	Una vez determinadas las posibles causas que provocan la falta de dinamismo en la consecución de las tareas se desarrollarán planes específicos atendiendo a la naturaleza de dichas causas.	

Tabla 16 Riesgo de personal RP-3

ID		RP-4	
Nombre	El investigador ha sufrido accidentes, problemas de salud o personales que introducen retrasos en la planificación		
Descripción	Puede darse el caso que durante el desarrollo del proyecto el investigador sufra accidentes, problemas de salud, o personales que provocan que su dedicación no sea la originalmente estimada.		
Probabilidad de suceso	Baja		
Impacto	Alto		
Exposición	Media		
Plan de prevención	La estimación de tiempos inicial estará realizada a la alta, pudiendo contemplar un margen de tiempo que funcione como amortiguador del riesgo.		

Tabla 17 Riesgo de personal RP-4

3.2 Planificación

Se denomina planificación de proyectos de software al proceso de gestión para la creación de un sistema o herramienta software. El objetivo primordial de la planificación de proyectos es proporcionar un marco de referencia del trabajo a desarrollar que permita realizar estimaciones precisas, razonables y lógicas sobre la inversión y asignación de los diferentes recursos y su planificación temporal [57]. Además las tareas de planificación deben permitir la adaptación y actualización del plan de proyecto conforme se progresa en las diferentes etapas del mismo.

Por lo tanto, en primer lugar se deben realizar estimaciones sobre esfuerzos, tiempos, personal y todos los recursos involucrados en el desarrollo del proyecto. Una vez terminada la actividad de estimación, se procede a la propia tarea de planificación. Para ello se establecerá un plan de proyecto que defina cometidos y fechas clave, identificando responsables por tareas y especificando dependencias entre tareas [58].

3.2.1 Gestión del alcance (EDT)

La EDT o Estructura de Descomposición de Trabajo es una representación visual de la descomposición de actividades que deberán ser llevadas a cabo a lo largo de todo el proyecto software. Es muy similar a la estructuración de ideas que se realizan en los mapas mentales, ya que representa de forma directa y natural la agrupación y relación entre las diferentes actividades.

El objetivo de una EDT o WBS (por sus siglas en inglés: Work Breakdown Structure) es organizar y definir el alcance total del proyecto. Su representación de las tareas hace discernir fácilmente los elementos finales, también llamados “Paquetes de trabajo” [59].

La EDT en ningún caso será contemplada como una lista de tareas a desarrollar en el proyecto ni tampoco será considerado un organigrama, ya que no presenta una ordenación secuencial de las actividades.

La EDT es útil para ser empleada como un elemento a partir del cual se basen las fases posteriores del desarrollo del proyecto software. Además la EDT es una buena herramienta para realizar estimaciones y asignación de responsabilidades de las tareas, así como un sistema de referencia para futuros proyectos. La EDT también facilita la realización de un control de los cambios que sufra el proyecto y facilita la medición de los diferentes aspectos del mismo.

En nuestro caso la EDT, representada en forma de mapa conceptual, ha servido a los diferentes grupos de tareas como apoyo para realizar la descomposición de tareas total de la planificación, que se presentará en el siguiente punto de este mismo capítulo.

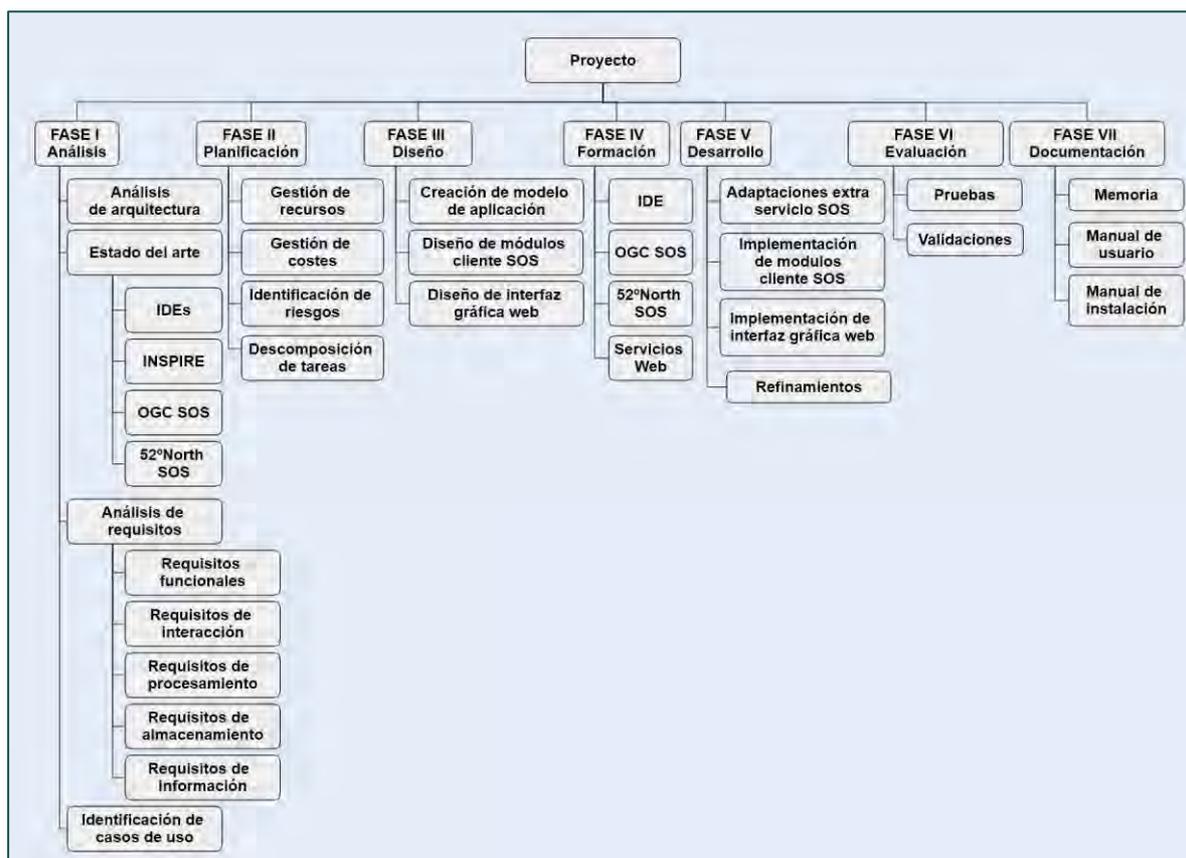


Fig. 24 Estructura de Descomposición de Trabajo del proyecto

En la EDT de la Fig. 24 se puede observar de forma clara que el esfuerzo del proyecto se descompondrá en 7 grandes fases, cada uno de los cuales se describirá a continuación:

- **Fase de Análisis:** Como primera fase del proyecto se realiza un análisis de la arquitectura necesaria para su desarrollo. Se realiza también un estudio de las artes implicadas en el proyecto, así como un análisis de los requisitos del sistema teniendo en cuenta los riesgos que se identifiquen para el mismo. Una vez obtenidos los requisitos del proyecto se elaborarán los casos de uso de la aplicación.
- **Fase de Planificación:** Se realiza una descomposición de tareas identificadas para cumplir con los requisitos identificados. Durante esta etapa además se deben gestionar los recursos disponibles, realizar la gestión del alcance del proyecto, de sus costes asociados, del tiempo de realización del mismo, de su configuración y de los riesgos identificados.

- **Fase de Diseño:** A partir de los requisitos identificados y de la composición establecida para la arquitectura del sistema, se realiza un diseño de los módulos y funcionalidades que compondrán los mismos. Para ello se estudian las diferentes operaciones que el cliente web podrá soportar y bajo qué lenguajes. Además se detallará el modo de operación de los diferentes módulos, teniendo en cuenta que tipo de información pueden emplear conjuntamente y delimitando las tareas de cada uno de ellos. Así mismo se diseñan los algoritmos que posteriormente se implementarán en la fase de procesamiento de datos de observaciones.

Para todos estos propósitos es una práctica habitual hacer uso del lenguaje unificado de modelado o UML que permite construir un diseño independientemente del lenguaje, librerías, artefactos, arquetipos y cualquiera de las técnicas de programación que se decidan emplear para la etapa de implementación.

En esta etapa también se diseña el sistema de centralización de los módulos y la interfaz gráfica de usuario de la aplicación.

- **Fase de Formación:** En esta etapa el investigador debe familiarizarse y adquirir conocimientos en las diferentes tecnologías elegidas para el desarrollo del proyecto. Como paso previo se ha invertido un porcentaje considerable del presente TFM en estudiar tanto la implementación previa de SOS V1.0 Semántico del grupo COGRADE, como la implementación estándar del SOS V2.0 de 52ºNorth, así como los clientes web que en la actualidad existen para este servicio. Así mismo habrá que profundizar y extender los conocimientos tanto de la filosofía de desarrollo del paradigma de Arquitecturas Orientada a Servicios, como de las diferentes tecnologías existentes en cuanto a desarrollo web.
- **Fase de Desarrollo:** En este bloque de trabajo se deberá llevar a cabo la implementación de las funcionalidades identificadas siguiendo el diseño elaborado en etapas anteriores y aplicando el conocimiento adquirido en la etapa de formación. Por tanto se impone el desarrollo de los módulos que componen el sistema junto con las operaciones empleadas en los mismos. También se implementará el sistema de construcción de peticiones y análisis de respuestas; y la interfaz gráfica de usuario.
- **Fase de Evaluación:** Se realizan pruebas unitarias sobre las diferentes operaciones y módulos implementados para garantizar que éstas llevan a cabo su cometido satisfactoriamente de forma independiente. Paralelamente se ejecutan pruebas de integración de dichas operaciones en sus módulos

correspondientes, y de estos módulos con el sistema central y con la interfaz gráfica web. Como último paso, se llevan a cabo pruebas de eficiencia que aseguren que la ejecución de los módulos se lleva a cabo correctamente.

- **Fase de Documentación:** En este bloque de trabajo se llevará a cabo la elaboración de la documentación generada durante el desarrollo del proyecto. Se elaborará tanto la documentación entregada a los clientes como la presente memoria en donde se explica el transcurso del trabajo efectuado durante la elaboración del proyecto. Adicionalmente se generará un manual de usuario y de instalación detallando los métodos de utilización de la aplicación web desarrollada y el proceso y requisitos de instalación de la misma.

3.2.2 Metodología de desarrollo

En este apartado se describe la metodología que se ha empleado en el desarrollo del proyecto, es decir, las etapas necesarias para llevar a cabo el proyecto desde su inicio hasta su finalización. A este conjunto de etapas o fases se les denomina en ingeniería de software como ciclo de vida.

Las diferentes metodologías existentes de ciclos de vida pueden agruparse en dos categorías: clásicas y ágiles. En las metodologías clásicas, también llamadas pesadas, se dedica gran esfuerzo en documentar y controlar cada una de sus fases, que tienden a manejar duraciones considerablemente altas, y donde las entregas al cliente son escasas o prácticamente nulas hasta el momento final. Estas metodologías pesadas tratan de determinar con precisión todos y cada uno de los requisitos del producto y determinar el tiempo necesario para cada fase durante el desarrollo inicial del proyecto [55] [59].

En contraposición a las metodologías pesadas se encuentran las metodologías ágiles. Éstas están basadas en un proceso de desarrollo iterativo, en el cual los requisitos van evolucionando según avanza el proyecto y los procesos por los que se pasa en cada iteración son simples, obteniendo al final de cada fase una versión funcional con algún nuevo requisito añadido que puede ser testeado tanto por el equipo de desarrollo como por los clientes. Por tanto, son más flexibles que las metodologías clásicas y prefieren la comunicación cara a cara antes que la documentación exhaustiva, lo que disminuye la carga de control y documentación de cada fase. Así mismo permite realizar

correcciones del proyecto sobre la marcha, ya que es posible detectar requisitos mal capturados o implementados de forma dinámica.

En este proyecto se deben tener en cuenta varios aspectos que establecerán las pautas de la elección del ciclo de vida adecuado para el desarrollo y posterior aceptación del TFM. Las reuniones periódicas con las partes involucradas en este proyecto serán un punto determinante a la hora de realizar un buen seguimiento en el desarrollo de la aplicación. Puesto que debemos implementar funcionalidades con cierta complejidad, los resultados se presentarán en diferentes tipos de reuniones durante el transcurso del desarrollo del software. De estas reuniones podremos extraer nuevos requisitos y actualizaciones del proyecto, y se tomarán en conjunto las decisiones oportunas con el objetivo de definir una aplicación más robusta y eficiente.

La metodología de desarrollo escogida y que se va a seguir durante el desarrollo de este proyecto se corresponde con el ciclo de vida iterativo incremental. El modelo iterativo incremental es un modelo de tipo evolutivo que se basa en la realización de un conjunto de ciclos de vida en cascada realimentados, aplicados repetidamente con una filosofía iterativa. Por cada una de las iteraciones en cascada se ofrece una versión con funcionalidad reducida y completamente operativa del software. De esta forma utilizaremos una metodología ágil evolutiva, donde los requisitos centrales son conocidos de antemano, aunque no estén bien definidos a nivel de detalle, que se irán refinando y completando en las diferentes etapas incrementales.

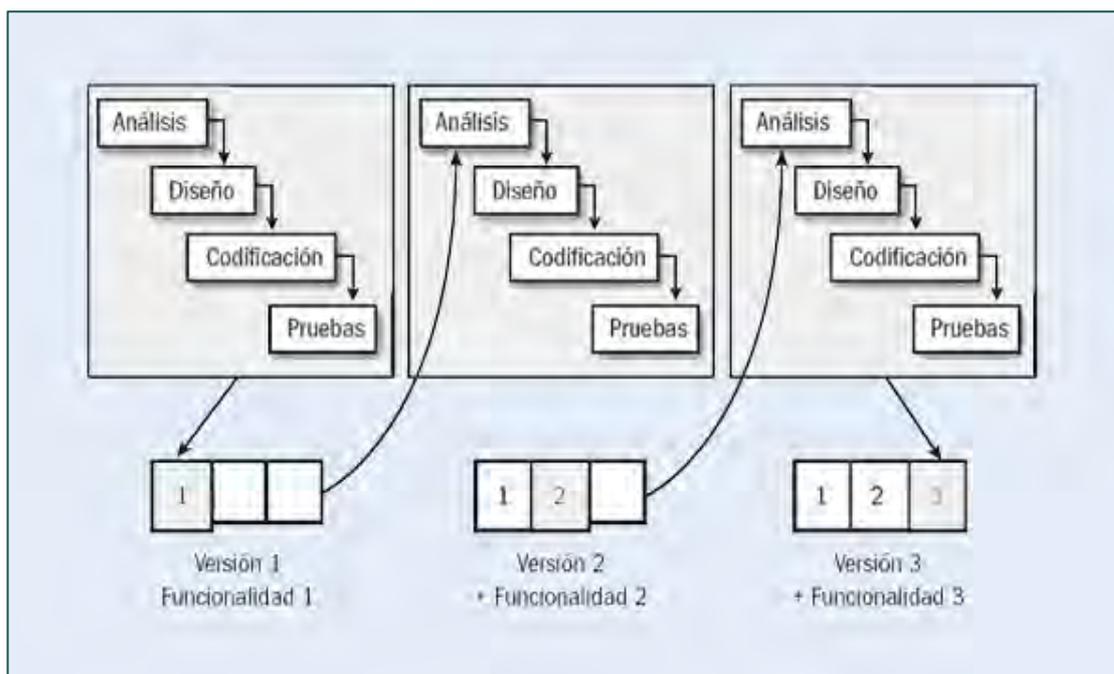


Fig. 25 Ciclo de vida iterativo incremental

En la Fig. 25 se muestra un esquema del ciclo de vida elegido, bajo un esquema temporal, que permite obtener finalmente el esquema del modelo de ciclo de vida Iterativo Incremental, con sus actividades genéricas asociadas. Aquí se observa claramente cada ciclo de vida en cascada que es desarrollado para la obtención de un incremento. Éstos últimos se van integrando para obtener el producto final completo.

Con este modelo de desarrollo primeramente se determinan las actividades a realizar y posteriormente el orden de estas tareas. Después de cada entrega al cliente conoceremos de antemano las tareas de la siguiente fase de desarrollo en base a las pruebas realizadas. Esta rutina debe ser repetida hasta que el proyecto quede completamente terminado de acuerdo a las condiciones pactadas inicialmente con los clientes.

Para facilitar el desarrollo del ciclo de vida del proyecto, se ha decidido diseñar e implementar el proyecto utilizando el estilo de programación de la filosofía de Arquitectura orientada a servicios basada en el paradigma pregunta/respuesta de aplicaciones sincronizadas. En ella la lógica de negocios o las funciones individuales son modulares y presentadas como servicios para aplicaciones consumidoras/clientes. Lo

que es clave de estos servicios es su naturaleza desacoplada; la interfaz de servicios es independiente de la implementación.

Una estrategia de aplicaciones debe facilitar su integración. Además que debe motivar la construcción de servicios, más que aplicaciones. Estos servicios se encargarían de exponer una funcionalidad bien definida a la aplicación que la requiera. Un servicio es una funcionalidad concreta que puede ser descubierta en la red y que describe tanto lo que puede hacer como el modo de interactuar con ella [60].

3.2.3 Metodología de diseño

Con el objetivo de modelar formalmente la aplicación web basada en servicios a desarrollar, se utilizarán las herramientas que proporciona UML (Unified Modeling Language). Éste es un lenguaje de modelado no ligado a ningún lenguaje de desarrollo particular, y que permite abordar el diseño de las distintas fases del ciclo de vida del proyecto en diferentes niveles de abstracción.

Una de las ventajas que proporciona UML en términos de diseño, es que permite definir el sistema desde dos puntos de vista: uno estático, que se refiere a la manera en la que la información se relaciona entre sí y cómo es almacenada, y otro dinámico, que permite representar cómo la información según su modo de flujo dentro del sistema global.

Se ha decidido hacer uso de UML debido a la necesidad de definir correctamente los planos del sistema software, así como debido a la importancia que posee el modelado en el proceso de desarrollo. Por lo tanto, el modelado UML proporcionará las distintas vistas o enfoques a partir de los cuales se puede abarcar el desarrollo del proyecto para cada una de sus partes, es decir, permitirá visualizar, especificar, construir y documentar el sistema con mayor facilidad [61] [62][63].

3.2.4 Gestión del tiempo (Gantt)

El diagrama de Gantt, introducido por Henry Laurence Gantt en 1917 [55], es una herramienta gráfica cuyo objetivo es mostrar el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado.

A pesar de que, en principio, el diagrama de Gantt no indica las relaciones existentes entre actividades, la posición de cada tarea a lo largo del tiempo permite identificar dichas relaciones e interdependencias. Por tanto el proyecto se puede descomponer en diferentes tareas y subtareas que poseen una precedencia entre ellas de forma que algunas no podrán empezar hasta que hayan concluido las actividades de sus tareas previas.

En el campo de gestión de proyectos, el diagrama de Gantt muestra el origen y el final de las diferentes unidades mínimas de trabajo y los grupos de tareas (llamados Summary Elements) o las dependencias entre unidades mínimas de trabajo.

Desde su introducción, los diagramas de Gantt, se han convertido en una herramienta básica en la gestión de proyectos de todo tipo, con la finalidad de representar las diferentes fases, tareas y actividades programadas como parte de un proyecto o para mostrar una línea de tiempo que relaciona las diferentes actividades haciendo el método más eficiente [59].

En la Fig. 26 podemos observar el diagrama de Gantt correspondiente al calendario planificado para este proyecto. Debemos tener presente que la jornada laboral empleada para la construcción del cronograma es de 8 horas diarias de lunes a viernes, lo que hace un total de 40 horas semanales. Esta jornada laboral se aplicó a partir del 01/11/2016, fecha en la que dio comienzo la fase del proyecto relacionada con el cliente SOS.

Por otro lado se incluye la Fig. 27 en la que se puede observar el diagrama planificado de asignación de recursos a tareas, que detalla el porcentaje de tiempo asignado a cada recurso y para cada una de las tareas del proyecto.

Las reuniones que se definen en estos diagramas han sido recogidas y detalladas en el apéndice del presente documento, en la sección de actas de reunión.

3.2.4.1 Calendario previsto

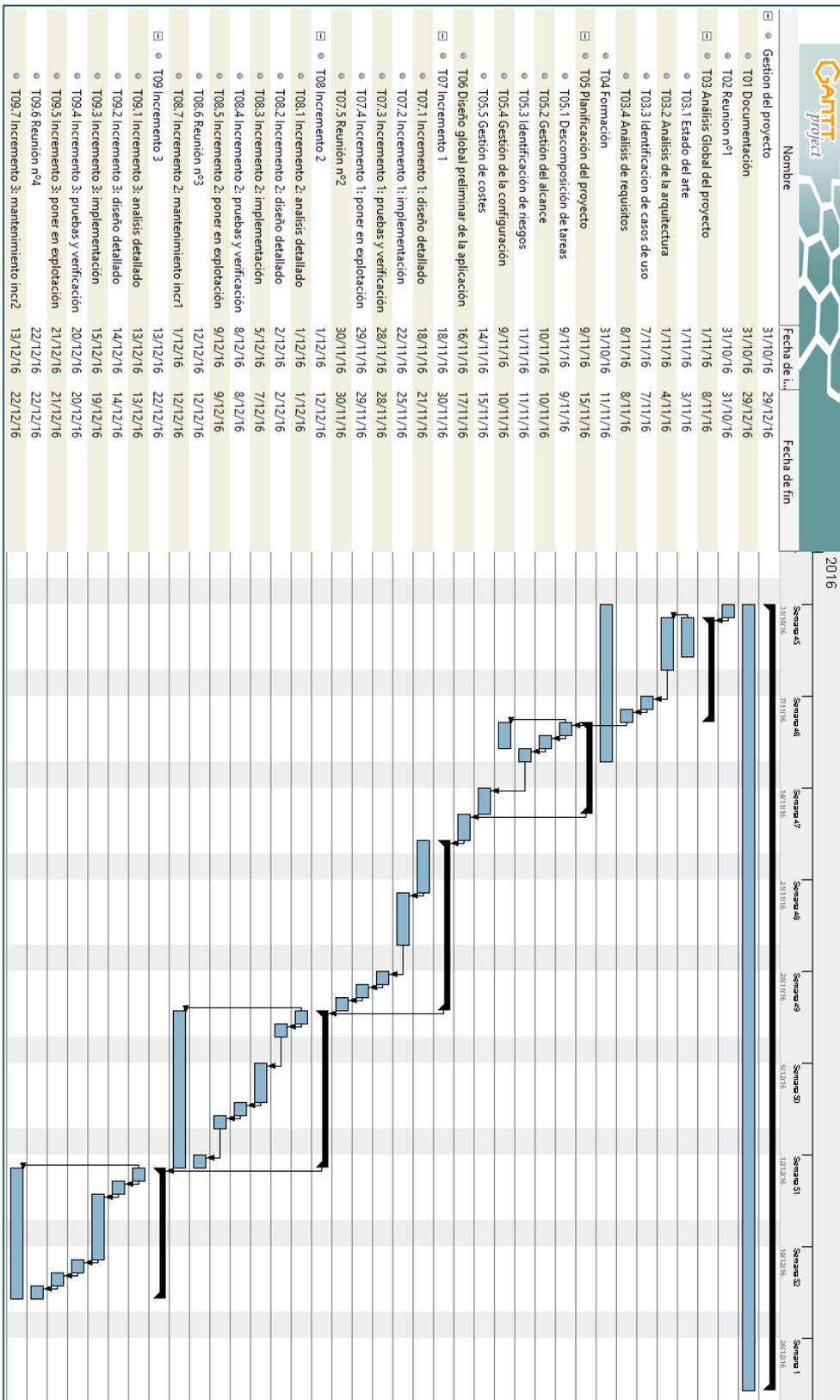


Fig. 26 Diagrama de Gantt planificado

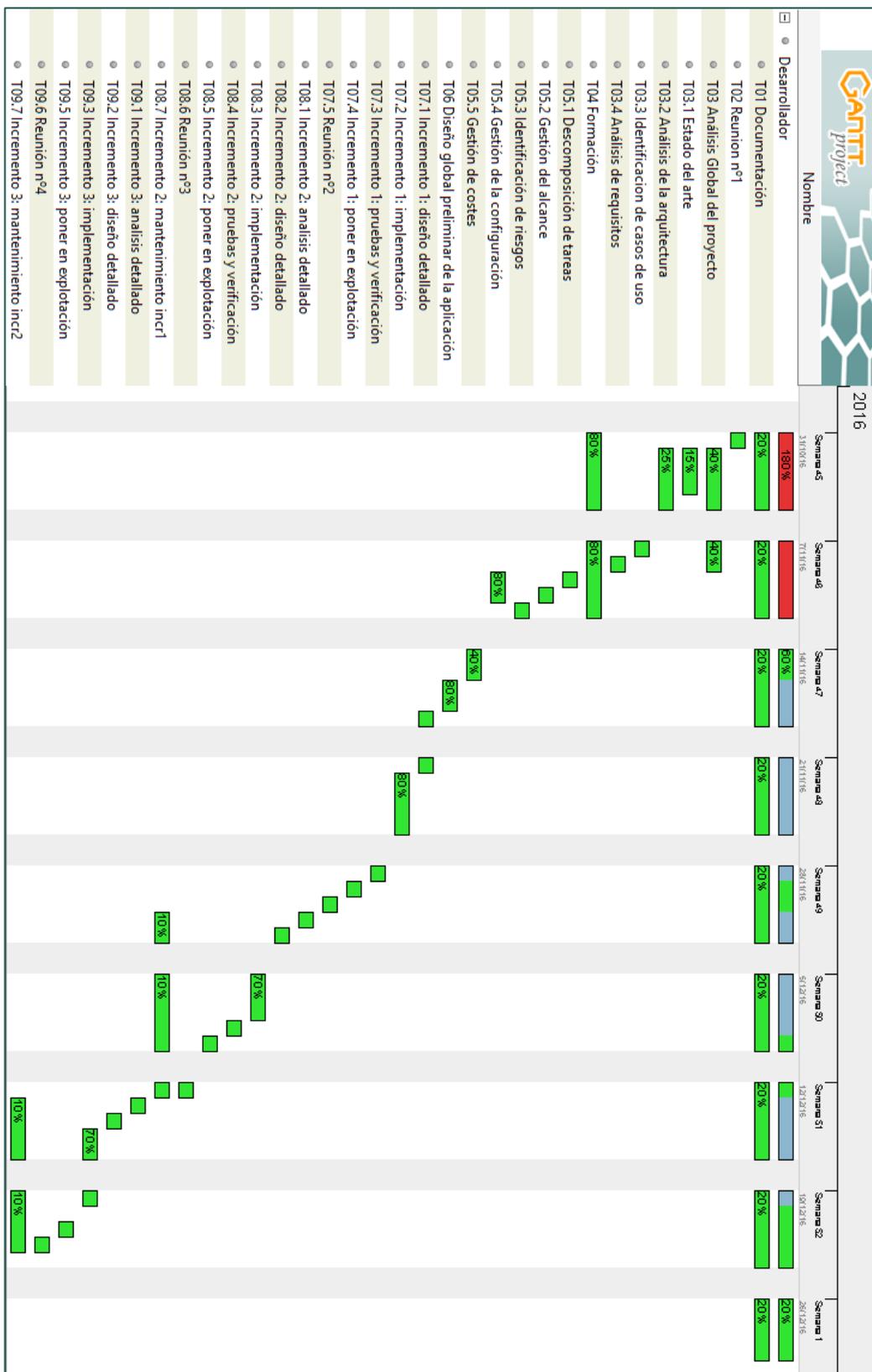


Fig. 27 Diagrama de recursos humanos planificado

3.2.4.2 Calendario real

Sobre la planificación inicial generada a partir del análisis de tareas explicado en apartados anteriores de este documento, se ha producido un conjunto de retrasos que han provocado que el proyecto no haya finalizado en la fecha prevista. Aun habiendo tenido en cuenta los riesgos expuestos en apartados anteriores, fue imposible mitigar el impacto de estos retrasos. Por tanto se modificó el calendario previsto, extendiendo su duración para desarrollar completamente el proyecto. En las Fig. 28 y Fig. 29 se pueden observar los diagramas de Gantt con los ajustes mencionados.

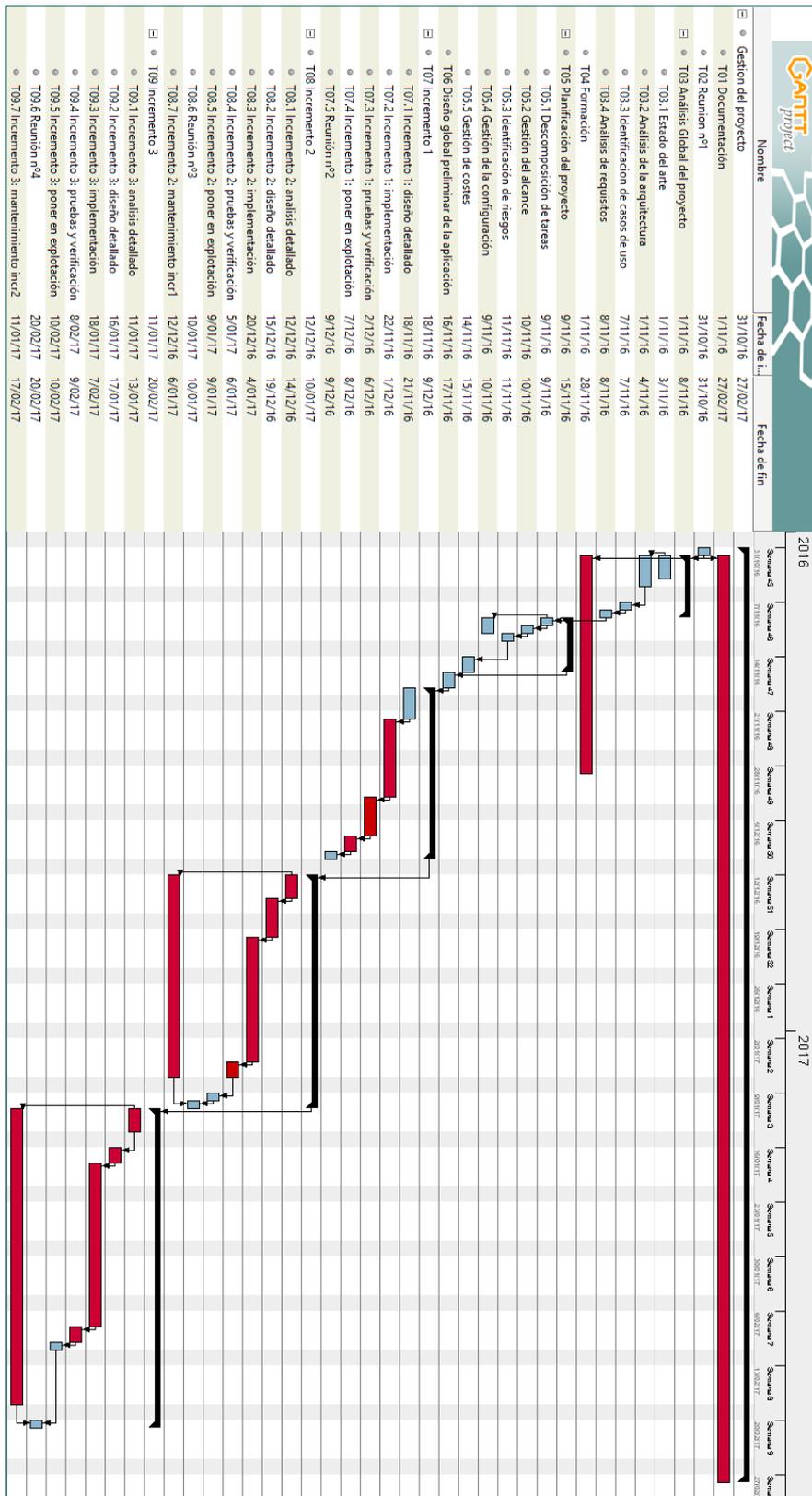


Fig. 28 Diagrama de Gantt seguido

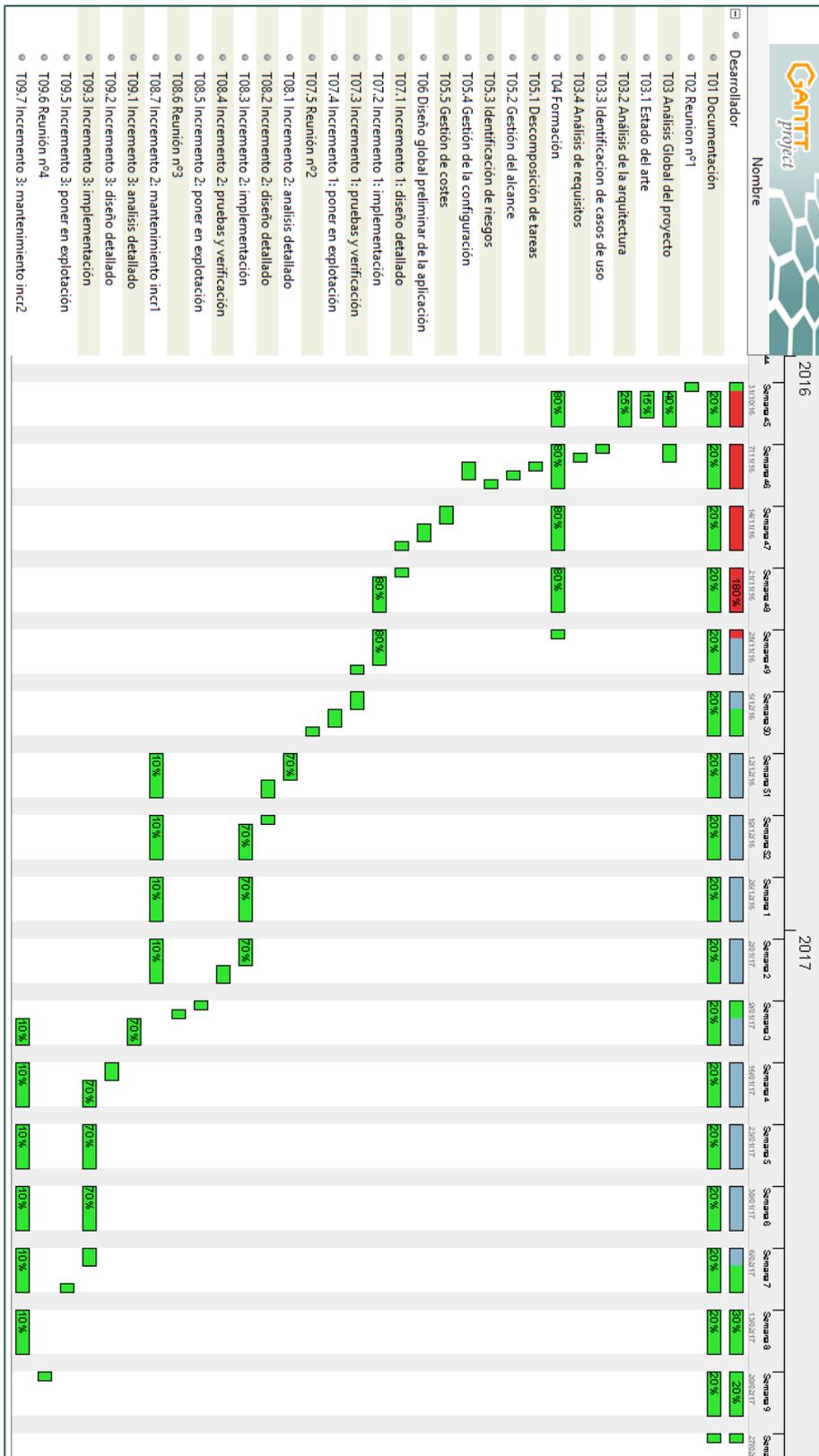


Fig. 29 Diagrama de recursos humanos seguido

3.3 Gestión de costes

La tarea de gestión de costes tiene como objetivo determinar cuál es el presupuesto necesario para llevar a cabo el proyecto. Para ello, se realiza un análisis por separado de costes derivados de material, costes de recursos humanos y otros costes del proyecto.

3.3.1 Costes de material

En esta sección se reflejan los costes derivados de los materiales empleados para el desarrollo del proyecto (ver Tabla 18). Se ha tenido en cuenta el gasto de material con respecto a la duración del TFM, puesto que el material de investigación suele emplearse en varios proyectos de investigación y/o desarrollo.

Recurso	Coste	Vida útil	Duración TFM	Coste
Pc Sobremesa	900€	5 años	0,5 años	90€
Microsoft Windows 10	130€	5 años	0,5 años	13€
Material de oficina	15€	0.5años	0,5 años	15€
			Total	118€

Tabla 18 Desglose de los costes de material

3.3.2 Costes de recursos humanos

En esta sección se reflejan los costes asociados a recursos humanos empleados para la realización del proyecto (ver Tabla 19)Tabla 19. Se contabiliza al investigador único del proyecto de investigación encargado de realizar las tareas de gestión del proyecto, análisis, diseño, implementación, evaluación y generación de la documentación.

Para la determinación de los costes se ha tomado como referencia que un analista-programador sin experiencia cobra un salario de 20.000 € / año brutos. A este sueldo bruto anual se le añade un 30% destinado a cubrir los costes de cotizaciones. Tanto el sueldo bruto como el porcentaje de cotizaciones son valores aproximados intermedios. Por tanto el gasto asciende a 26.000 € / año. Así tenemos que si un analista acumula un total de 1760 horas anuales trabajadas, el coste por hora es de 14,80 € / h.

En la siguiente tabla se resume el coste del TFM en cuanto a recursos humanos:

Recurso	Horas Dedicadas	Coste/hora	Coste Total
Investigador	480	14.80€	7.104€
Total			7.104€

Tabla 19 Desglose de los costes de recursos humanos

3.3.3 Coste total

En esta sección se especifica el coste total del proyecto (ver Tabla 20) teniendo en cuenta el coste de los recursos humanos y materiales y la gestión de costes sobre el total asociado a otros gastos como el consumo eléctrico, calefacción, aire acondicionado, internet, alquiler de oficina, etc. Éstos últimos se asumen como un 25% del total del precio de los recursos humanos y no humanos.

Fuente	Coste
Recursos humanos	7.104€
Material	118€
Subtotal	7222€
Otros 25%	1.805,5€
Total	9.027,5€

Tabla 20 Desglose de los costes totales

3.4 Gestión de la configuración

La gestión de la configuración es el proceso encargado de establecer los mecanismos y herramientas necesarios para mantener la integridad de todos los elementos de trabajo que intervienen en el desarrollo del proyecto de investigación. Con ella se consigue una correcta configuración del software. Un elemento de configuración es todo aquel producto de trabajo que mediante un cambio puede ocasionar un fallo en el desarrollo del proyecto [55][59].

A continuación se exponen las diferentes acciones que se han llevado a cabo para garantizar la integridad y calidad de los resultados finales obtenidos durante el desarrollo del proyecto. Entre estas acciones se encuentran la gestión de integridad del código fuente, la gestión de la documentación generada durante el transcurso del proyecto, el control de cambios durante el desarrollo del proyecto, etc.

3.4.1 Control de código fuente

Para garantizar la integridad del código fuente de la aplicación durante la fase de implementación del mismo se ha empleado el sistema de control de versiones GitLab, ver Fig. 30.

Se llama control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Una versión, revisión o edición de un producto, es el estado en el que se encuentra el mismo en un momento dado de su desarrollo o modificación. Aunque un sistema de control de versiones puede realizarse de forma manual, es muy aconsejable disponer de herramientas que faciliten esta gestión dando lugar a los llamados sistemas de control de versiones. Estos sistemas facilitan la administración de las distintas versiones de cada producto desarrollado, así como las posibles especializaciones realizadas (por ejemplo, determinada configuración y adaptación específica de un producto para un cliente determinado).

El control de versiones es empleado en la industria informática para controlar las distintas versiones del código fuente dando lugar a los sistemas de control de código fuente o SCM (siglas del inglés Source Code Management). Sin embargo, los mismos conceptos son aplicables a otros ámbitos como documentos, imágenes, sitios web, etc.

Hemos utilizado GitLab en su modo web y desde explorador web, permitiendo gestionar el versionado de los diferentes ficheros del proyecto en un repositorio privado que mantiene actualizados los ficheros en su última versión. Este sistema permite también consultar o comparar entre si las versiones previas de los ficheros. Si se llegara a dar el caso, se pueden revertir los cambios realizados en el código y regresar a versiones anteriores del código. Un repositorio es un almacén centralizado donde se atesora y mantiene información digital.

Esta actividad de control de versiones asegura la trazabilidad de los cambios del proyecto a la vez que proporciona la seguridad de la capacidad de recuperación de versiones anteriores completas en caso de producirse cualquier error durante el proceso de desarrollo. Con este sistema se evita el riesgo técnico RT-4 explicado en secciones anteriores de este documento.

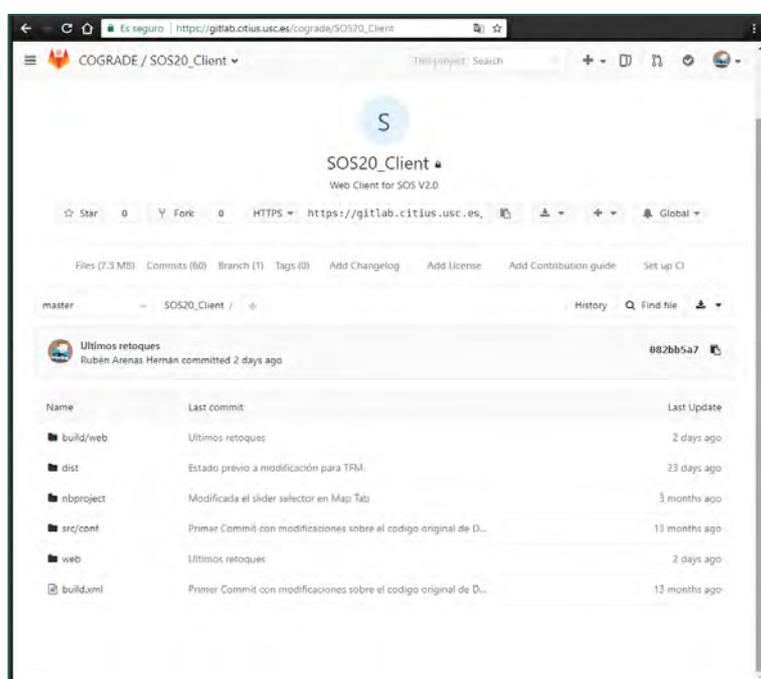


Fig. 30 Herramienta GitLab empleada para el control de versiones de código

3.4.2 Control de la documentación

Toda la documentación generada durante el ciclo de vida de este proyecto se ha gestionado de igual manera que el código fuente: se ha utilizado el sistema de control de versiones Git bajo la implementación gratuita GitLab.

GitLab permite integrar un conjunto de directorios y ficheros almacenados de forma local y de forma independiente al sistema de ficheros del sistema operativo referenciando éstos directorios hacia copias almacenadas en máquinas remotas. GitLab permite la sincronización de los ficheros según el estándar Git. Así todos los documentos generados, como diagramas, esquemas, imágenes y textos elaborados permanecen actualizados y versionados, manteniendo una copia de respaldo en un servidor ajeno a la máquina de desarrollo. De esta forma se garantiza la disponibilidad de todos los documentos creados independientemente de la máquina, evitando así el riesgo de asociado a pérdidas involuntarias de información.

El empleo de esta herramienta hace posible el acceso desde cualquier máquina o navegador a la documentación generada garantizando en todo momento la existencia de copias de respaldo, evitando el riesgo técnico RT-4 explicado en secciones anteriores de este documento.

La Fig. 31 como GitLab gestiona los diferentes directorios y archivos:

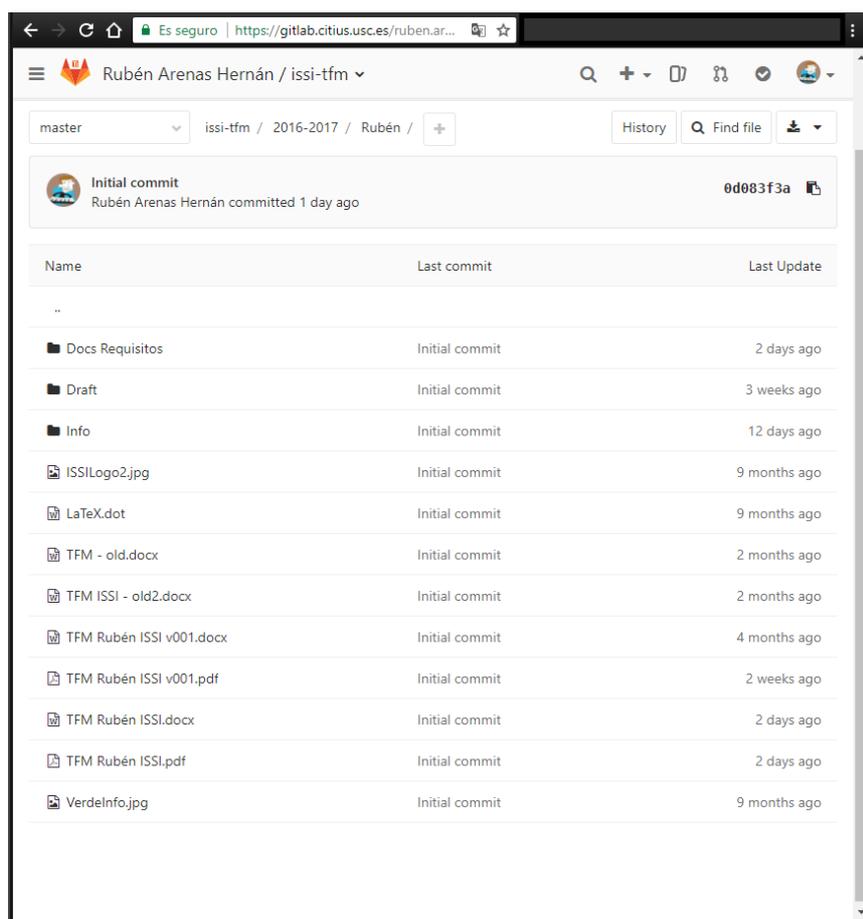


Fig. 31 Herramienta GitLab empleada para el control de versiones de la documentación

3.4.3 Gestión de los cambios

En esta sección se especifica la forma de actuación ante los cambios que han ido sufriendo tanto el proyecto como sus requisitos a lo largo del ciclo de vida de desarrollo del mismo.

La gestión de los cambios es una actividad que se desarrolla durante todas y cada una de las tareas de las etapas que forman parte del ciclo de vida del proceso de desarrollo, ya que es imposible determinar en qué momento concretó se puede producir un cambio en el mismo.

Las actividades en el proceso de gestión de los cambios se desarrollan para:

- Identificar el cambio
- Controlar el cambio
- Garantizar que el cambio se realizará de manera adecuada
- Reportar los cambios a todos los interesados

La gestión de cambios es una tarea crítica, determinante y fundamental de los proyectos software puesto que su ausencia genera caos e incertidumbre y provoca que se desarrollen productos software de mala calidad. La gestión de cambios es una práctica sólida y consolidada contemplada en ingeniería de software.

Existen cuatro fuentes fundamentales que pueden originar cambios en la planificación de un proyecto software:

- Nuevas situaciones en el negocio, que conllevan variaciones en los requisitos del producto o en las directrices comerciales.
- Nuevas necesidades del cliente que originan la modificación de la información producida por los sistemas software.
- Reestructuración o cambios del volumen del negocio, que generan cambios en las precedencias del proyecto o en la organización del equipo de desarrollo.
- Limitaciones de planificación o presupuestarias, que obligan a la redefinición de las condiciones del proyecto.

En nuestro proyecto se han contemplado los cambios únicamente a través de correo electrónico. El protocolo de gestión de cambios adoptado fue el siguiente:

- El cliente encuentra un cambio y lo notifica formalmente al equipo de desarrollo mediante correo electrónico.
- El equipo de desarrollo archiva el cambio y lo incluye en la agenda de tareas a tratar en la siguiente reunión.
- En cada reunión se discuten, analizan, diseñan y presupuestan cada uno de los cambios que el cliente ha solicitado durante el incremento del ciclo de vida actual.
- El cliente decide la inclusión de cada uno de los cambios, sujetos a las condiciones negociadas.

- El equipo de desarrollo incluye todos los cambios en el acta de reunión, con el fin de documentar los cambios y de facilitar futuras variaciones sobre la planificación del proyecto.

Con el objetivo de garantizar el correcto tratamiento de los cambios se han incluido las tareas de mantenimiento en la planificación temporal del proyecto. Así podemos observar en el diagrama de Gantt de la Fig. 26 las tareas de mantenimiento T09.7 y T10.7. La duración de estas tareas ha sido estimada teniendo en cuenta lo siguiente:

- El número de cambios puede ser variable e indeterminado, y por tanto la cantidad de recursos temporales implicados en la implementación de los mismos podrá variar sustancialmente la planificación original. De ahí que se haya considerado la reserva de una cantidad de recursos temporales considerables. En el caso de emplear menor cantidad de recursos temporales de los reservados inicialmente, ésta podrá ser destinada a la finalización de otras tareas del incremento actual.
- La posibilidad de ser necesarios recursos temporales adicionales para la finalización del conjunto restante de tareas del incremento. De esta manera se contempla un margen de tiempo para gestionar retrasos sobre la planificación temporal inicial.

3.5 Sistema de seguimiento

Los sistemas de seguimiento hacen referencia a los diferentes medios empleados para la coordinación de los miembros de todas las partes intervinientes en el proyecto y para la comunicación a los mismos de los diferentes informes de situación del proyecto. Las líneas de seguimiento del proyecto se establecieron mediante tres vías:

- Correo electrónico.
- Conversación telefónica.
- Reuniones periódicas al final de cada uno de los incrementos del ciclo de vida

Este seguimiento continuo ha proporcionado conocimientos sobre la salud del proyecto y ha permitido identificar las áreas que requieren especial atención. Así mismo han permitido un refinamiento en cuanto a definición de los requisitos dado el constante contacto con el cliente.

El contacto directo a través de correo electrónico ha sido el método principal de seguimiento entre el equipo de investigación y los grupos involucrados en las agencias INTECMAR y MeteoGalicia. De esta forma se ha utilizado el correo electrónico como medio de comunicación formal sobre los diferentes aspectos del proyecto. Mediante dicha herramienta se ha informado de la evolución del proyecto, de la resolución de problemas del proyecto y se ha empleado como método de contacto para el establecimiento de las reuniones. Los documentos asociados a los correos electrónicos han sido almacenados y tratados como documentación del proyecto.

Por otro lado la vía telefónica se ha empleado en casos puntuales, para la comunicación en tiempo real de contratiempos y de sus posibles soluciones.

En las reuniones se adoptó un modo de comunicación interactiva, donde cada una de las partes intervinientes intercambiaron ordenada y bidireccionalmente la información deseada. Este método es la manera más eficiente de asegurar que todos los participantes alcancen una comprensión común acerca de los diferentes elementos específicos del proyecto.

Para concertar reuniones de seguimiento se han establecido convocatorias de reunión por medio de correo electrónico en los que se incluyen los datos de la misma:

- Hora y lugar de la reunión
- Personal convocado
- Asunto principal a tratar en la reunión
- Agenda de los temas a tratar listados según orden de discusión
- Estimación temporal para la argumentación de los mismos.
- Tiempo dedicado a dudas y novedades

Tras cada una de las reuniones se ha procedido a la redacción de un acta de reunión con el objetivo de documentar y registrar la información ofrecida por todas las partes. Estas actas se comparten con los diferentes integrantes del proyecto a través del correo electrónico, puesto que ha sido el medio que mejor se ajustaba a los conocimientos técnicos todos los miembros intervinientes. Simultáneamente se guardaron en el repositorio de control de versiones de documentación, para garantizar su almacenamiento seguro.

En dichas actas de reunión se detallan:

- Los cambios en el desarrollo del proyecto estudiados en el incremento del ciclo de vida actual.
- El plazo estimado para el cumplimiento de cada una de las nuevas tareas y objetivos acordados.
- El personal asignado a las nuevas tareas, así como el tiempo de dedicación a cada una de ellas.

Estas actas se pueden consultar en la sección de actas del apéndice del presente documento. Se corresponden con el Apéndice E.

Capítulo 4. Análisis del proyecto

El análisis de requisitos es una de las tareas más importantes en el ciclo de vida del desarrollo de software, ya que constituye la base sobre la que se construirá el sistema.

En cualquier proyecto software los requisitos son las necesidades del producto que se deben implementar. Por ello, en la fase de análisis de requisitos es crítico identificar y documentar correcta y claramente estas necesidades. Como resultado de esta fase se debe producir un documento de especificación de requisitos en el que se describan todas y cada una de las acciones que el futuro sistema debe realizar. Esta especificación determinará en gran medida el éxito o fracaso del proyecto. Por tanto, no se trata simplemente de una actividad de análisis, sino también de síntesis [64].

El análisis de requisitos se puede definir como el proceso del estudio de las necesidades de los usuarios para llegar a una definición de los requisitos del sistema, hardware o software, así como el proceso de estudio y refinamiento de dichos requisitos, definición proporcionada por el estándar IEEE 830 [65]. Asimismo, se define requisito como una condición o capacidad que necesita el usuario para resolver un problema o conseguir un objetivo determinado. Esta definición se extiende y se aplica a las condiciones que debe cumplir o poseer un sistema o uno de sus componentes para satisfacer un contrato, una norma o una especificación [66].

En la determinación de los requisitos no sólo deben actuar el equipo de desarrollo, es muy importante la participación de los propios usuarios, porque son éstos los que mejor conocen el sistema que se va a automatizar. Equipo de desarrollo y cliente deben ponerse de acuerdo en las necesidades del sistema: el cliente no suele entender el proceso de diseño y desarrollo del software como para redactar formalmente una adecuada especificación de requisitos software (en adelante ERS) y el equipo de desarrollo no suele entender profundamente el problema que el cliente necesita resolver, debido en gran medida a la inexperiencia en el área de trabajo del cliente.

Es vital que el documento de especificación de requisitos pueda ser entendido en su totalidad por el cliente, permitiendo evitar malentendidos en determinadas situaciones, ya que el cliente participa activamente en la extracción de dichos requisitos.

Basándose en estos requisitos, el ingeniero de software deberá proceder al modelado de la futura aplicación. Para ello, se pueden utilizar diferentes tipos de metodologías como pueden ser la metodología estructurada y la metodología orientada a objetos (por ejemplo DFDs y UML respectivamente).

La metodología estructurada está basada en la representación de las funciones que debe realizar el sistema y los datos que fluyen entre ellas.

Los principales objetivos que se identifican en la especificación de requisitos software son [55][67][68]:

- Ayudar a los clientes a describir claramente lo que se desea obtener mediante un determinado software: El cliente debe participar activamente en la especificación de requisitos, ya que éste tiene una visión mucho más detallada de los procesos que se llevan a cabo. Asimismo, el cliente se siente partícipe del propio desarrollo.
- Ayudar a los desarrolladores a entender qué quiere exactamente el cliente: En muchas ocasiones el cliente no sabe exactamente qué es lo que quiere. La ERS permite al cliente definir todos los requisitos que desea y al mismo tiempo los desarrolladores tienen una base fija en la que trabajar. Si no se realiza una buena especificación de requisitos, los costes de desarrollo pueden incrementarse considerablemente, ya que los cambios emergentes serán continuos durante todo el ciclo de desarrollo de la aplicación.
- Servir de base para desarrollos de estándares de ERS particulares para cada organización: Cada entidad puede desarrollar sus propios estándares para definir sus necesidades.

En esta actividad de ERS se deben concertar varias entrevistas entre el equipo de desarrollo y el cliente con el fin de conocer detalladamente los requisitos funcionales del sistema, así como también para extraer los requisitos de calidad, diseño, interfaz y de proyecto que se pretenden alcanzar durante el desarrollo del proyecto.

Una vez detallados los objetivos a cumplir y las funcionalidades de la aplicación debemos plasmar estas ideas gráficamente en forma de casos de uso, con diferentes escenarios, en los que se describirá la forma en la que interactúan los usuarios con la aplicación. A partir de los casos de uso construidos se deben definir los requisitos funcionales, que darán forma a la aplicación mediante la descripción de las actividades que podrán ser ejecutadas y mediante la descripción de restricciones en el funcionamiento y en el diseño.

Una ERS correcta debe originar:

- El acuerdo a cumplir entre cliente y equipo de desarrollado.
- La reducción del esfuerzo durante el resto de ciclo de vida del proyecto.
- Una base sólida para planificación del proyecto y la estimación de costes.
- Un punto de referencia para procesos de verificación y validación de los diferentes elementos del proyecto.
- Una base para la identificación de posibles mejoras en los procesos analizados.

4.1 Casos de uso

La presente sección incluye una especificación formal de los casos de uso que se encargarán de explicar todas las interacciones que tendrán los usuarios con la aplicación o el propio sistema consigo mismo de forma automática. Esta tarea es una de las fases más importantes en el desarrollo del proyecto que condiciona las etapas posteriores. Es por esto que debe realizarse un análisis exhaustivo en el que se recopilen todos los casos de uso del proyecto.

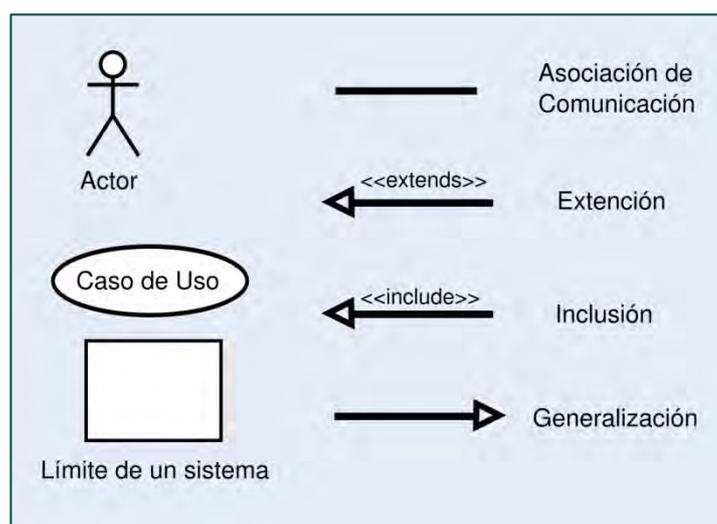


Fig. 32 Notación de los diagramas de casos de uso UML

Los casos de uso son una técnica excelente para identificar las funcionalidades que el sistema debe implementar. Estos casos de uso se corresponden con diagramas en los que se identifica la interacción deseable que debe existir entre el usuario y el sistema (ver Fig. 32), sin atender a cuestiones específicas de implementación, y que se utilizan para identificar los requisitos del sistema [55][59].

En la Fig. 33 se presenta el diagrama de casos de uso global que incluye todos los casos de uso del sistema que se han identificado, así como las relaciones existentes entre los mismos. Hay que indicar que el actor principal se corresponde con el usuario del sistema en todos y cada uno de estos casos de uso. A continuación se incluye una descripción pormenorizada de cada uno de ellos.

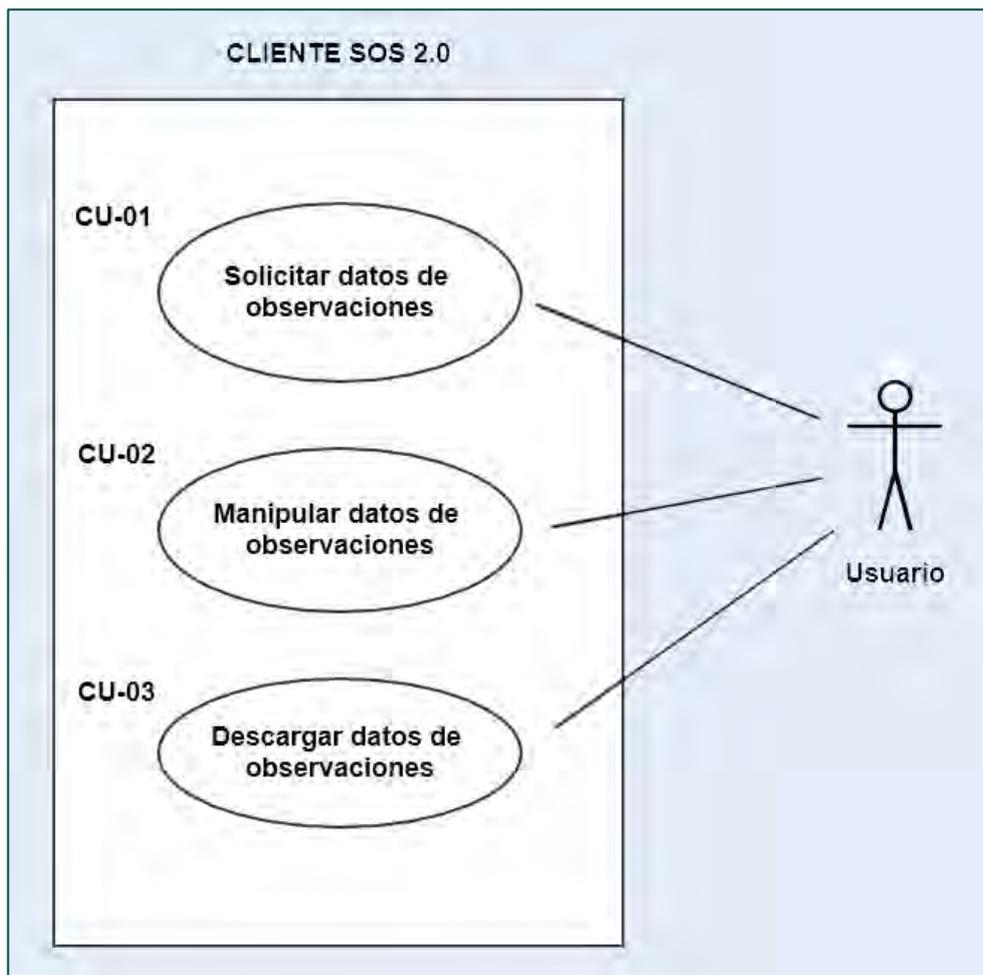


Fig. 33 Diagrama de casos de uso del presente proyecto

A continuación se incluye una descripción detallada de cada uno de los casos de uso incluidos en la figura anterior:

Caso de uso:	CU-01
Descripción:	El actor del sistema solicita la obtención de datos de observación del servidor o servidores SOS.
Actor:	Usuario

<p>Precondiciones:</p>	<p>Los servidores SOS 2.0 deben estar arrancados y funcionalmente operativos.</p> <p>El cliente SOS 2.0 debe estar desplegada, con conexión a red.</p> <p>La cliente SOS 2.0 debe presentar una configuración correcta de los servidores SOS 2.0</p>
<p>Flujo Normal:</p>	<p>Una vez que el actor ingresa la URI del cliente SOS 2.0 en el navegador web, se carga el cliente SOS 2.0 encargándose de dialogar de forma automática con los diferentes servidores SOS 2.0 identificados en la sección de configuración.</p> <p>Como respuesta de dicho diálogo el cliente SOS muestra en un formulario diferentes menús desplegados con las propiedades de los servidores SOS que el usuario puede consultar.</p> <p>El usuario selecciona diferentes valores relacionados con las entidades de observación.</p> <p>Se activa el botón de consulta.</p> <p>El actor pulsa el botón para realizar la solicitud.</p>
<p>Flujo Alternativo:</p>	<p>-</p>
<p>Postcondiciones:</p>	<p>El cliente SOS 2.0 se encarga de realizar la petición a los servidores SOS 2.0</p> <p>Una vez recibida la respuesta se muestran los datos de observaciones en diferentes modos.</p>

Tabla 21 Detalles del caso de uso CU-01

Caso de uso: CU-02	
Descripción:	El actor podrá manipular los datos de observaciones mostrados en la interfaz web
Actor:	Usuario
Precondiciones:	<p>El usuario ha solicitado datos de observaciones a Servidores SOS 2.0</p> <p>Los servidores SOS 2.0 han enviado una respuesta</p> <p>El cliente SOS 2.0 ha procesado los datos contenidos en la respuesta</p> <p>El cliente SOS 2.0 ha formateado y mostrado los datos de observaciones</p>
Flujo Normal:	<p>El actor puede escoger entre diferentes representaciones de los datos de observaciones: formato de tabla, formato de gráfica, formato de mapa.</p> <p>En el formato tabla podrá ordenar y filtrar los datos según columnas.</p> <p>En el formato de gráfica el actor selecciona los elementos relacionados con la petición previa. El cliente SOS 2.0 filtra las observaciones, genera interpolaciones de los datos de observaciones y las muestra en una gráfica de dos dimensiones.</p> <p>En el formato de mapa el actor realiza una selección en los menús mostrados, el cliente SOS filtra los datos de observaciones y los muestra en el mapa</p>
Flujo Alternativo:	-

Postcondiciones:	-
------------------	---

Tabla 22 Detalles del caso de uso CU-02

Caso de uso: CU-03	
Descripción:	<p>El actor selecciona en los diferentes menús la opción de descarga de datos de observación que desee.</p> <p>El cliente SOS 2.0 procesa dicha acción y descarga los datos de observaciones.</p>
Actor:	Usuario
Precondiciones:	<p>Se ha realizado una petición a un servidor SOS 2.0</p> <p>El cliente ha recibido, procesado y mostrado los datos de la petición.</p>
Flujo Normal:	<p>El actor escoge entre los menús de las representaciones de los datos de observación la opción deseada.</p> <p>El cliente SOS 2.0 procesa la solicitud, filtra los datos almacenados localmente y los descarga a la maquina.</p> <p>El cliente SOS 2.0 permitirá la descarga de datos en:</p> <ul style="list-style-type: none"> - formato tabla de cálculo CSV, TSV - formato de gráfica en dos dimensiones
Flujo Alternativo:	-
Postcondiciones:	-

Tabla 23 Detalles del caso de uso CU-03

4.2 Catálogo de requisitos

En el catálogo de requisitos que se presenta a continuación se recogen todas las especificaciones que deben ser implementadas en el proyecto. Se describirán los requisitos de una forma detallada, sin entrar en cuestiones técnicas y estableciendo un nivel de importancia entre ellos.

Esta especificación de requisitos ha sido llevada a cabo según el estándar IEEE 830-1998 [69][66][64] de buenas prácticas, que divide los requisitos, según su ámbito de influencia, en las siguientes categorías:

- Requisitos funcionales. Requisitos que definen las funcionalidades del sistema y de sus componentes.
- Requisitos de diseño: También conocidos como restricciones de diseño, indican las limitaciones a las que se enfrenta el desarrollador a la hora de llevar a cabo la elaboración del producto.
- Requisitos de interfaz gráfica. Requerimientos relacionados con la interfaz gráfica del usuario.
- Requisitos de calidad: Requerimientos que establecen la calidad del producto; como por ejemplo, la usabilidad o el rendimiento.
- Requisitos de evolución: Estos requisitos se establecen con el fin de desarrollar un sistema fácilmente extensible.
- Requisitos de soporte: Son los requisitos que ha de cumplir el cliente para llevar a cabo el desarrollo del software.
- Requisitos de proyecto: Se refiere a los requisitos que afectan al proyecto.
- Requisitos emergentes. Requisitos que surgen al incrementarse la comprensión del cliente en el desarrollo del sistema. El proceso de diseño puede revelar requerimientos emergentes nuevos.

Los diferentes requisitos expuestos en apartados posteriores de este documento se clasificarán según su categoría bajo la nomenclatura descrita en la siguiente tabla:

Categoría	Identificador
Requisitos de proyecto	RQ-PR
Requisitos funcionales	RQ-FN
Requisitos de calidad	RQ-CA
Requisitos de evolución	RQ-EV
Requisitos de diseño	RQ-DI
Requisitos de soporte	RQ-SO
Requisitos de interfaz gráfica	RQ-IG
Requisitos emergentes	RQ-EM

Tabla 24 Identificadores de categorías de requisitos

En la Tabla 24 podemos observar el esquema de descripción de requisitos que será utilizado más adelante.

Identificador	Código único que identifica al requisito. Se emplea la codificación de la Tabla 24 junto al número de requisito.
Título	Frase corta que describe al requisito
Descripción	Breve descripción del requisito de forma clara, concisa e inequívoca.
Casos de uso relacionados	Indica los casos de uso con los cuales el requisito tiene relación.
Prioridad	De menor a mayor grado puede ser: opcional, condicional o esencial.
Estabilidad	Probabilidad de que el requisito varíe a lo largo del proyecto; puede ser baja, media o alta.
Comentarios	Comentarios que puedan ayudar a complementar la información dada en la descripción.

Tabla 25 Plantilla de descripción de requisitos

4.2.1 Requisitos del proyecto

Requisito RQ-PR-01	
Título	Paquete Software
Descripción	Deberá entregarse el software resultado de la implementación de los requisitos funcionales del proyecto en formato empaquetado.
Casos de uso relacionados	-
Prioridad	Esencial
Estabilidad	Alta
Comentarios	-

Tabla 26 Requisito del proyecto RQ-PR-01

Requisito RQ-PR-02	
Título	Fecha de entrega
Descripción	La fecha de entrega de la herramienta será Enero de 2017
Casos de uso relacionados	-
Prioridad	Opcional

Estabilidad	Baja
Comentarios	-

Tabla 27 Requisito del proyecto RQ-PR-02

Requisito RQ-PR-03	
Título	Memoria del proyecto
Descripción	En el proceso de entrega del proyecto se debe adjuntar una memoria que documente fielmente el ciclo de vida y transcurso del proyecto
Casos de uso relacionados	-
Prioridad	Esencial
Estabilidad	Alta
Comentarios	

Tabla 28 Requisito del proyecto RQ-PR-03

Requisito RQ-PR-04	
Título	Manual de usuario
Descripción	En la entrega del proyecto se debe incluir un manual de usuario que describa los métodos de utilización de la aplicación y sus funcionalidades
Casos de uso	-

relacionados	
Prioridad	Esencial
Estabilidad	Alta
Comentarios	-

Tabla 29 Requisito del proyecto RQ-PR-04

Requisito RQ-PR-05	
Título	Manual de instalación
Descripción	En la entrega del proyecto se debe incluir un manual de instalación de la aplicación, que detalle el método de instalación de la herramienta
Casos de uso relacionados	-
Prioridad	Esencial
Estabilidad	Alta
Comentarios	

Tabla 30 Requisito del proyecto RQ-PR-05

Requisito RQ-PR-06	
Título	Tecnología a emplear
Descripción	Se deben emplear tecnologías de desarrollo web siguiendo el

	interfaz estándar proporcionado por el SOS API, definido por el OGC. Este será la guía para la implementación de los diferentes módulos que formarán parte del cliente SOS.
Casos de uso relacionados	-
Prioridad	Esencial
Estabilidad	Alta
Comentarios	-

Tabla 31 Requisito del proyecto RQ-PR-06

Requisito RQ-PR-07	
Título	Duración
Descripción	El tiempo dedicado a la realización del proyecto será aproximadamente de 420 horas de trabajo autónomo y 20 horas de trabajo presencial con los directores del proyecto
Casos de uso relacionados	-
Prioridad	Esencial
Estabilidad	Alta
Comentarios	-

Tabla 32 Requisito del proyecto RQ-PR-07

4.2.2 Requisitos funcionales

Requisito RQ-FN-01	
Título	Configuración del cliente SOS
Descripción	La herramienta permitirá su configuración mediante algún sistema, ya sea con fichero de configuración o desde formulario de interfaz web. Se permitirá la configuración del servidor SOS 2.0 al que se pretende realizar el acceso a datos de observaciones de sensores
Casos de uso relacionados	-
Prioridad	Condicional
Estabilidad	Baja
Comentarios	-

Tabla 33 Requisito del proyecto RQ-FN-01

Requisito RQ-FN-02	
Título	Conectar con servicio SOS 2.0
Descripción	El cliente SOS deberá poder conectarse al servidor SOS 2.0 configurado. Se empleará únicamente las operaciones que define la interfaz SOS 2.0. Esta conexión se realizará de forma automática, sin intervención del usuario

Casos de uso relacionados	-
Prioridad	Esencial
Estabilidad	Alta
Comentarios	-

Tabla 34 Requisito del proyecto RQ-FN-02

Requisito	RQ-FN-03
Título	Comprobar estado del servicio SOS 2.0
Descripción	El cliente SOS 2.0 puro deberá comprobar automáticamente el estado del servidor SOS 2.0, sin ninguna interacción del usuario.
Casos de uso relacionados	-
Prioridad	Esencial
Estabilidad	Alta
Comentarios	-

Tabla 35 Requisito del proyecto RQ-FN-03

Requisito	RQ-FN-04
Título	Consultar metadatos del servicio SOS 2.0
Descripción	El cliente SOS 2.0 puro deberá consultar los metadatos asociados

	al servicio SOS 2.0 del que se pretende consultar los datos de observaciones.
Casos de uso relacionados	-
Prioridad	Esencial
Estabilidad	Alta
Comentarios	-

Tabla 36 Requisito del proyecto RQ-FN-04

Requisito RQ-FN-05	
Título	Consultar localización de estaciones de observación
Descripción	El cliente SOS 2.0 puro deberá poder consultar las estaciones de observación que están registradas en el servicio SOS 2.0
Casos de uso relacionados	-
Prioridad	Esencial
Estabilidad	Alta
Comentarios	-

Tabla 37 Requisito del proyecto RQ-FN-05

Requisito RQ-FN-06	
--------------------	--

Título	Formulario de peticiones SOS 2.0
Descripción	El cliente SOS 2.0 deberá implementar un formulario desde el cual el usuario introduzca los parámetros que formarán parte de la consulta al servicio SOS 2.0 y desde el cual se lanzará la ejecución de dicha consulta.
Casos de uso relacionados	CU-01
Prioridad	Esencial
Estabilidad	Alta
Comentarios	-

Tabla 38 Requisito del proyecto RQ-FN-06

Requisito RQ-FN-07	
Título	Recogida de la respuesta del servicio SOS 2.0
Descripción	El cliente SOS 2.0 puro deberá ser capaz de recoger la respuesta del servicio SOS 2.0. Se determinará en qué momento se espera una respuesta y se deberán implementar mecanismos que permitan mostrar mensajes de error en caso de que el servicio presente algún fallo en el proceso de consulta de datos de observación.
Casos de uso relacionados	-
Prioridad	Esencial

Estabilidad	Alta
Comentarios	-

Tabla 39 Requisito del proyecto RQ-FN-07

Requisito RQ-FN-08	
Título	Analizar la sintaxis de la respuesta de servicio SOS 2.0 y almacenar su información
Descripción	El cliente SOS 2.0 puro deberá parsear la respuesta recibida del servicio SOS 2.0, para extraer la información de datos de observaciones solicitados previamente. Además se deberán almacenar en caché del navegador los datos bajo una estructura acorde que permita su manipulación posterior.
Casos de uso relacionados	CU-02
Prioridad	Esencial
Estabilidad	Alta
Comentarios	-

Tabla 40 Requisito del proyecto RQ-FN-08

Requisito RQ-FN-09	
Título	Manipular datos de observaciones de respuestas del servicio SOS 2.0

Descripción	El cliente SOS 2.0 puro deberá implementar algoritmos de manipulación de datos de observaciones que permitan mostrar dichos datos con diferentes resoluciones, según la información que haya introducido al respecto el usuario
Casos de uso relacionados	CU-02
Prioridad	Esencial
Estabilidad	Alta
Comentarios	-

Tabla 41 Requisito del proyecto RQ-FN-09

Requisito	RQ-FN-10
Título	Mostrar datos crudos de la respuesta de observaciones del servicio SOS 2.0
Descripción	El cliente SOS 2.0 puro deberá ofrecer la opción de mostrar los datos crudos de la respuesta recibida tras la consulta al servicio SOS 2.0 Esta información debe ser mostrada de forma gráfica
Casos de uso relacionados	-
Prioridad	Esencial
Estabilidad	Alta

Comentarios	-
-------------	---

Tabla 42 Requisito del proyecto RQ-FN-10

Requisito RQ-FN-11	
Título	Mostrar datos interpolados de la respuesta de observaciones del servicio SOS 2.0
Descripción	El cliente SOS 2.0 puro deberá ofrecer la opción de mostrar los datos interpolados de la respuesta recibida tras la consulta al servicio SOS 2.0 Esta información debe ser mostrada de forma gráfica.
Casos de uso relacionados	-
Prioridad	Esencial
Estabilidad	Alta
Comentarios	-

Tabla 43 Requisito del proyecto RQ-FN-11

Requisito RQ-FN-12	
Título	Descarga de datos crudos de la respuesta de observaciones del servicio SOS 2.0
Descripción	El cliente SOS 2.0 puro deberá ofrecer la opción de descargar los datos crudos de la respuesta recibida tras la consulta al servicio SOS 2.0 Estos datos serán descargados en formato imagen, CSV, TSV y/o

	XLS
Casos de uso relacionados	CU-03
Prioridad	Esencial
Estabilidad	Alta
Comentarios	-

Tabla 44 Requisito del proyecto RQ-FN-12

Requisito	RQ-FN-13
Título	Descarga de datos interpolados de la respuesta de observaciones del servicio SOS 2.0
Descripción	El cliente SOS 2.0 puro deberá ofrecer la opción de descargar los datos interpolados de la respuesta recibida tras la consulta al servicio SOS 2.0 Estos datos serán descargados en formato imagen, CSV, TSV y/o XLS
Casos de uso relacionados	CU-03
Prioridad	Esencial
Estabilidad	Alta
Comentarios	-

Tabla 45 Requisito del proyecto RQ-FN-13

4.2.3 Requisitos de calidad

Requisito RQ-CA-01	
Título	Eficiencia
Descripción	Se requiere que la ejecución de los procesos implementados sea eficiente
Casos de uso relacionados	-
Prioridad	Esencial
Estabilidad	Alta
Comentarios	-

Tabla 46 Requisito de calidad RQ-CA-01

Requisito RQ-CA-02	
Título	Interfaz de usuario usable
Descripción	Se busca que la comunicación entre usuario y aplicación sea intuitiva, ágil, cómoda y natural
Casos de uso relacionados	-
Prioridad	Esencial

Estabilidad	Alta
Comentarios	-

Tabla 47 Requisito de calidad RQ-CA-02

Requisito RQ-CA-03	
Título	Lag mínimo
Descripción	La comunicación entre cliente SOS 2.0 y servidor SOS 2.0 debe implicar el menor tiempo entre consulta y respuesta. Cuanto mayor sea el tiempo de retraso entre petición y representación de los datos solicitados menor será la satisfacción del usuario con la herramienta
Casos de uso relacionados	-
Prioridad	Esencial
Estabilidad	Alta
Comentarios	-

Tabla 48 Requisito de calidad RQ-CA-03

4.2.4 Requisitos de evolución

Requisito RQ-EV-01	
Título	Introducción sencilla de nuevas funcionalidades
Descripción	El diseño del sistema debe facilitar la introducción de nuevas funcionalidades
Casos de uso relacionados	-
Prioridad	Esencial
Estabilidad	Alta
Comentarios	-

Tabla 49 Requisito de calidad RQ-EV-01

Requisito RQ-EV-02	
Título	Facilidad de modificación
Descripción	El diseño del sistema debe contemplar la posibilidad de reemplazo de determinadas funcionalidades con la mayor facilidad posible. Para ello se exige desacoplamiento entre funcionalidades no relacionadas
Casos de uso relacionados	-

Prioridad	Esencial
Estabilidad	Alta
Comentarios	-

Tabla 50 Requisito de calidad RQ-EV-02

4.2.5 Requisitos de diseño

Requisito	RQ-DI-01
Título	Interfaz SOS 2.0
Descripción	La herramienta debe implementar las funcionalidades obligatorias del interfaz SOS 2.0 expuestas en la documentación del estándar OGC SOS 2.0
Casos de uso relacionados	-
Prioridad	Esencial
Estabilidad	Alta
Comentarios	-

Tabla 51 Requisito de diseño RQ-DI-01

Requisito RQ-DI-02	
Título	Independencia de navegador
Descripción	La herramienta debe poder ejecutarse en la mayoría de los navegadores web empleados en sistemas de escritorio. Por lo tanto la visualización y comportamiento de la herramienta debe ser correcta en todos ellos.
Casos de uso relacionados	-
Prioridad	Opcional
Estabilidad	Media
Comentarios	-

Tabla 52 Requisito de diseño RQ-DI-02

Requisito RQ-DI-03	
Título	Datos de observaciones bien estructurados
Descripción	Los datos de observaciones extraídos de las respuestas obtenidas del servicio SOS 2.0 deben ser almacenados en caché del navegador web mediante una estructura datos eficiente. Los datos serán accedidos por los diferentes módulos mediante DOM y una estructuración ineficiente provocará tiempos de procesado y representación elevados.

Casos de uso relacionados	-
Prioridad	Esencial
Estabilidad	Alta
Comentarios	-

Tabla 53 Requisito de diseño RQ-DI-03

4.2.6 Requisitos de soporte

Requisito	RQ-SO-01
Título	Asistencia a reuniones
Descripción	Los clientes deben acudir a todas las reuniones donde expondrá todas sus ideas sobre la agenda de la reunión
Casos de uso relacionados	-
Prioridad	Esencial
Estabilidad	Alta
Comentarios	-

Tabla 54 Requisito de soporte RQ-SO-01

Requisito RQ-SO-02	
Título	Respuesta a notificaciones del cliente
Descripción	Ante una comunicación escrita o telefónica por parte del cliente, el investigador responderá tan pronto termine la tarea que en el momento de la notificación estuviera realizando
Casos de uso relacionados	-
Prioridad	Opcional
Estabilidad	Media
Comentarios	-

Tabla 55 Requisito de soporte RQ-SO-02

Requisito RQ-SO-03	
Título	Manual de usuario
Descripción	Debe desarrollarse un manual de usuario con el modo de utilización y las FAQs correspondientes con el objetivo de minimizar las consultas sobre soporte técnico del cliente
Casos de uso relacionados	-

Prioridad	Opcional
Estabilidad	Media
Comentarios	-

Tabla 56 Requisito de soporte RQ-SO-03

4.2.7 Requisitos de interfaz gráfica

Requisito	RQ-IG-01
Título	Formulario de consulta de observaciones
Descripción	<p>La herramienta deberá contemplar un formulario que el usuario deberá cumplimentar para poder realizar peticiones al servicio SOS 2.0</p> <p>El objetivo es abstraer al usuario de la implementación de la consulta, sea en el lenguaje que sea: KVP, POX, XML, JSON, SOAP o REST.</p>
Casos de uso relacionados	CU-01
Prioridad	Esencial
Estabilidad	Alta
Comentarios	-

Tabla 57 Requisito de interfaz gráfica RQ-IG-01

Requisito RQ-IG-02	
Título	Visualización de los datos de respuesta de consulta en formato tabla
Descripción	El cliente SOS 2.0 desarrollado debe mostrar los datos obtenidos del servicio SOS 2.0 en formato tabla. Esta tabla podrá ser ordenada según el usuario y atendiendo a los datos que se contemplen en ella.
Casos de uso relacionados	CU-02
Prioridad	Esencial
Estabilidad	Alta
Comentarios	-

Tabla 58 Requisito de interfaz gráfica RQ-IG-02

Requisito RQ-IG-03	
Título	Visualización de los datos de respuesta de consulta en formato gráfica
Descripción	El cliente SOS 2.0 desarrollado debe mostrar los datos obtenidos del servicio SOS 2.0 en formato gráfica. Esta gráfica podrá ser manipulada en cuanto a zoom y resolución de los datos según los parámetros que el usuario indique sobre la

	misma.
Casos de uso relacionados	CU-02
Prioridad	Esencial
Estabilidad	Alta
Comentarios	-

Tabla 59 Requisito de interfaz gráfica RQ-IG-03

Requisito	RQ-IG-04
Título	Visualización de los datos de respuesta de consulta en formato mapa
Descripción	El cliente SOS 2.0 desarrollado debe mostrar los datos obtenidos del servicio SOS 2.0 en formato mapa. Los datos representados en el mapa variarán según los parámetros que introduzca el usuario en los controles de visualización.
Casos de uso relacionados	CU-02
Prioridad	Condicional
Estabilidad	Media
Comentarios	-

Tabla 60 Requisito de interfaz gráfica RQ-IG-04

Requisito RQ-IG-05	
Título	Elementos navegables
Descripción	Dado que el cliente SOS 2.0 podrá mostrar diferentes conjuntos de formularios y visualizaciones, cada uno de estos conjuntos o modos de visualización se mostrará independientemente del resto. Se precisa algún mecanismo de ventanas, pestañas o paneles que ofrezcan la distribución ordenada de los datos en la interfaz web.
Casos de uso relacionados	CU-02
Prioridad	Esencial
Estabilidad	Alta
Comentarios	-

Tabla 61 Requisito de interfaz gráfica RQ-IG-05

Requisito RQ-IG-06	
Título	Interfaz gráfica usable
Descripción	La interfaz web del cliente SOS 2.0 debe ser lo más usable posible. Para ello el diseño se debe realizar teniendo en cuenta las normas

	básicas de usabilidad.
Casos de uso relacionados	-
Prioridad	Esencial
Estabilidad	Alta
Comentarios	-

Tabla 62 Requisito de interfaz gráfica RQ-IG-06

4.2.8 Requisitos emergentes

Requisito	RQ-EM-01
Título	Tamaño de petición
Descripción	<p>Se conoce de antemano que el servicio SOS 2.0 presenta un control sobre el volumen de datos que una petición puede realizar. Aún así, el cliente SOS 2.0 debe gestionar el volumen de las peticiones, indicando al usuario que no se podrá ejecutar una consulta si su tamaño excede un máximo.</p> <p>Esto se busca con el objetivo de no consumir recursos excesivos en la máquina de ejecución del cliente SOS 2.0</p>

Casos de uso relacionados	-
Prioridad	Condicional
Estabilidad	Media
Comentarios	Hay que consensuar el modo de funcionamiento entre equipo de desarrollo y clientes

Tabla 63 Requisito emergente RQ-EM-01

Requisito RQ-EM-02	
Título	Optimización de la estructura de datos de observaciones
Descripción	Hay que optimizar la estructura de datos de los datos de observaciones resultado de una consulta para contener los datos de los parámetros y que éstos sean fácilmente accesibles
Casos de uso relacionados	-
Prioridad	Condicional
Estabilidad	Media
Comentarios	Puesto que las fuentes de datos únicamente contemplan datos de profundidad, se sobreentiende que todos los parámetros de los datos de observaciones contemplan un único tipo de parámetro. El estándar OGC SOS 2.0 no dice nada al respecto de la heterogeneidad de los parámetros entre observaciones de la misma

	estación o incluso de la misma fuente de datos
--	--

Tabla 64 Requisito emergente RQ-EM-02

Requisito RQ-EM-03	
Título	Interfaz gráfica basada en GeoExt v1.1
Descripción	El equipo de Intecmar requiere que la interfaz web de usuario esté basada en la versión 1.1 de la herramienta GeoExt, dado que en caso de extensibilidad, les resultará sencillo, dado que tienen otras herramientas con esta tecnología.
Casos de uso relacionados	-
Prioridad	Esencial
Estabilidad	Alta
Comentarios	El equipo de MeteoGalicia no tiene preferencia por este framework de desarrollo web, por lo que no les influye la tecnología de desarrollo de la interfaz web de usuario

Tabla 65 Requisito emergente RQ-EM-03

4.3 Matriz de trazabilidad

Para verificar que todos los casos de uso del proyecto están cubiertos con los requisitos funcionales y que no se han diseñado casos de uso innecesarios, se ha generado una matriz de trazabilidad que enfrenta los casos de uso y los requisitos funcionales del catálogo de requisitos (ver Tabla 66) [55][59].

	RQ-FN-01	RQ-FN-02	RQ-FN-03	RQ-FN-04	RQ-FN-05	RQ-FN-06	RQ-FN-07	RQ-FN-08	RQ-FN-09	RQ-FN-10	RQ-FN-11	RQ-FN-12	RQ-FN-13
CU-01						X							
CU-02							X	X					
CU-03												X	X

Tabla 66 Matriz de Trazabilidad

Capítulo 5. Diseño del proyecto

Se puede definir el proceso de diseño como el procedimiento mediante el cual se aplican determinadas técnicas y principios con el propósito de definir un dispositivo, una técnica, una herramienta o un sistema; con suficientes detalles como para permitir su interpretación y realización física. La fase de diseño debe formalizar la aplicación a partir de los requerimientos definidos en la etapa anterior del ciclo de vida de software, con el fin de cumplirlos y satisfacer los deseos del cliente.

Es vital que esta fase del ciclo de vida del software esté desarrollada con rigurosidad y esmero, puesto que el éxito o fracaso del software en desarrollo dependerá completamente de ella. Una vez terminada esta fase, el diseño debe ser capaz de proporcionar los suficientes detalles del proyecto que permitan la construcción de la aplicación. Si el diseño es correcto y presenta un nivel alto de calidad el esfuerzo necesario a realizar en la fase de implementación será mínimo.

Los diferentes apartados de este capítulo se enfocan en especificar el diseño de los diferentes elementos del proyecto y desde perspectivas complementarias.

A la hora de completar el diseño debemos tener en cuenta el modelo de ciclo de vida elegido en la planificación. En apartados anteriores de este documento se ha detallado la elección del modelo del ciclo de vida. Se determinó que el ciclo de vida incremental iterativo (ver Fig. 25) sería el modelo de ciclo de vida que mejor se ajusta a la tipología del proyecto y que se utilizará para desarrollarlo.

La elección del ciclo de vida incremental iterativo tiene consecuencias directas sobre la metodología de diseño. Así pues se realizará un diseño también incremental iterativo, basado en tres fases de diseño, una por cada uno de los incrementos que hemos definido en las tareas de planificación, descritas en apartados anteriores del presente documento.

Como ya hemos explicado, los tres incrementos en los que se divide el proyecto, y que dan lugar a tres versiones ejecutables, se basarán en la implementación de

funcionalidades propias del cliente web SOS V2.0, para facilitar el acceso, visualización, manipulación y descarga de los datos que se ofrecen en el servicio SOS 2.0 semántico desarrollado por el grupo de investigación COGRADE de la USC.

La primera fase del diseño será la más importante y completa ya que su resultado debe facilitar el acoplamiento de los futuros incrementos a desarrollar. Esta fase se encarga de definir el conjunto del sistema desde un punto de vista general y con un nivel de abstracción alto. Por lo tanto en esta fase se debe definir la arquitectura del sistema (Fig. 34) y la arquitectura de la aplicación (Fig. 35), además de definir los procesos de interacción del usuario sobre la interfaz gráfica (ver Fig. 39, Fig. 40, Fig. 41 y Fig. 42).

Esta parte del diseño se dividirá en dos apartados: Arquitectura del Sistema, sección en la que se definen los distintos elementos que componen la arquitectura orientada a servicio SOS V2.0, y Arquitectura de la Aplicación, que muestra la estructura del cliente pesado SOS V2.0 desarrollado, detallando el conjunto de elementos modulares que lo componen.

5.1 Diseño de la Arquitectura del Sistema

Como arquitectura del sistema se ha utilizado una arquitectura en tres capas dado que la implementación del servicio SOS 2.0 realizada por la organización 52^oNorth así lo contempla (ver Fig. 11, Fig. 8 y Fig. 21). Además se contempla la integración directa del cliente SOS V2.0 con el servidor SOS V2.0

El cliente SOS V2.0 se encargará de la capa de presentación con funcionalidad de procesamiento de los datos de observaciones. Las otras dos capas, a priori no nos influyen en nada. Es por ello que en primer lugar se probará el cliente SOS V2.0 contra la implementación de servidor SOS V2.0 definida bajo la arquitectura que se ha presentado en la Fig. 7. Puesto que las tareas de investigación y desarrollo de esta herramienta se llevan en paralelo con el desarrollo del presente proyecto, las primeras pruebas se realizarán contra dicha implementación sin las capacidades de integración

semántica. De esta forma la arquitectura completa del sistema sería la detallada en Fig. 34:

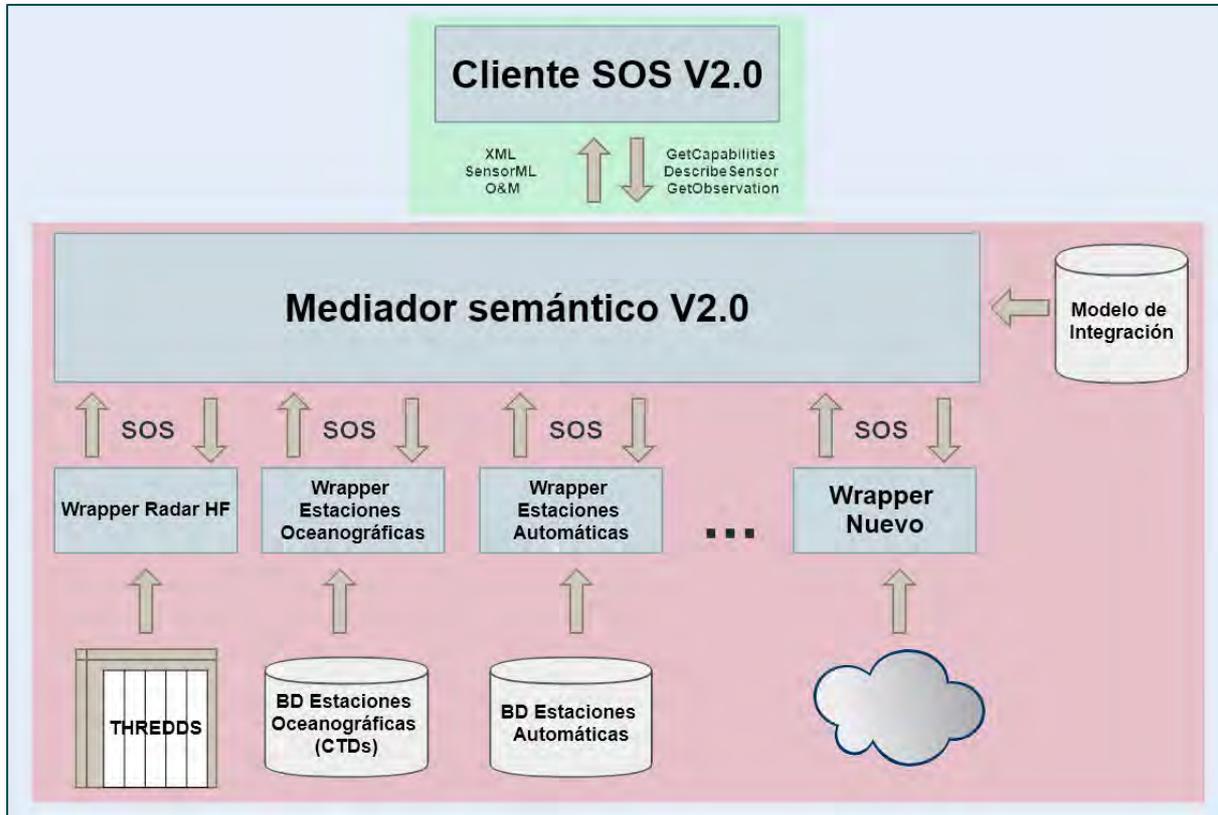


Fig. 34 Arquitectura completa del sistema SOS V2.0 semántico

Tal y como hemos indicado nuestro cliente SOS V2.0 se situará en la capa de presentación, que en la anterior imagen se corresponde con el la parte de color verde. A efectos de simplificación trataremos toda la parte del sistema coloreada con color de fondo rojo como un servidor SOS V2.0. Siempre que hagamos referencia a servidor SOS V2.0 en las fases posteriores del ciclo de vida de desarrollo estaremos haciendo referencia a toda esta arquitectura.

5.2 Diseño de la Arquitectura de la Aplicación

En esta sección mostraremos el diseño de la estructura interna que compone la aplicación. El diseño se ha desarrollado teniendo en cuenta que la aplicación resultante será una aplicación estructurada. Se dice que una aplicación software es estructurada si cumple los principios de la programación estructurada, que son los siguientes:

- La aplicación presenta una estructura jerárquica top-down.
- El diseño de la aplicación es modular
- Las salidas y entradas están claramente definidas
- La aplicación posee un único punto de entrada y un único punto de salida
- La aplicación utiliza exclusivamente los constructores de programas estructurados: secuencia, selección, iteración y llamadas a procedimientos.

Para diseñar una aplicación estructurada se emplean las siguientes técnicas:

- **Recursos abstractos.** Descomponer una aplicación en términos de recursos abstractos consiste en descomponer acciones complejas en términos de acciones más simples capaces de ser ejecutadas en una computadora.
- **Diseño descendente.** El diseño descendente se encarga de resolver un problema realizando una descomposición del mismo en otros problemas más sencillos utilizando para ello una estructura jerárquica de módulos. Así cada módulo de la estructura es refinado por los módulos de niveles inferiores, que resuelven problemas más pequeños y contienen más detalles sobre los mismos.
- **Estructuras básicas de control.** Las estructuras básicas de control sirven para especificar el orden en que se ejecutarán las distintas instrucciones de un algoritmo. Este orden de ejecución determina el flujo de control del programa.

Por tanto el diseño de nuestra aplicación quedará definido como una serie de funcionalidades que se relacionan entre sí para conformar la estructura jerárquica de los diferentes módulos software de la aplicación (ver Fig. 35).

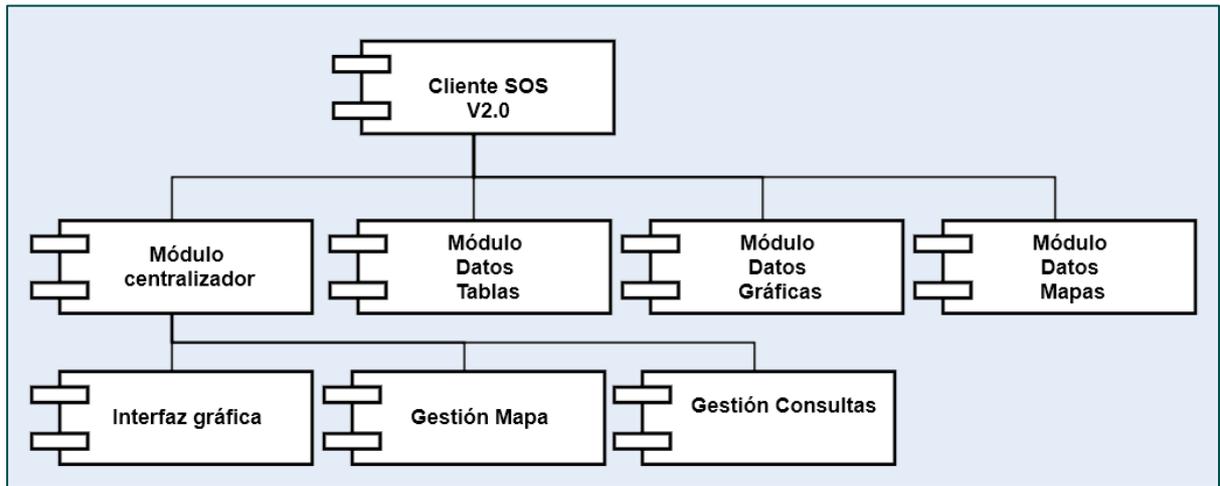


Fig. 35 Estructura jerárquica de los módulos de la aplicación

Con el fin de modelar los diferentes aspectos de la aplicación haremos uso de los diagramas de UML. En concreto he empleado los diagramas de secuencia y los diagramas de actividades.

5.2.1 Diagrama de secuencia

Con el diagrama de secuencia detallado en esta sección se define la interacción entre los módulos software de la aplicación, el servidor SOS y el usuario del sistema para la las actividades de generación y ejecución de consulta de observaciones. En este diagrama se reflejan las diferentes llamadas a procedimientos y acciones para lograr el propósito de obtención de datos de observación.

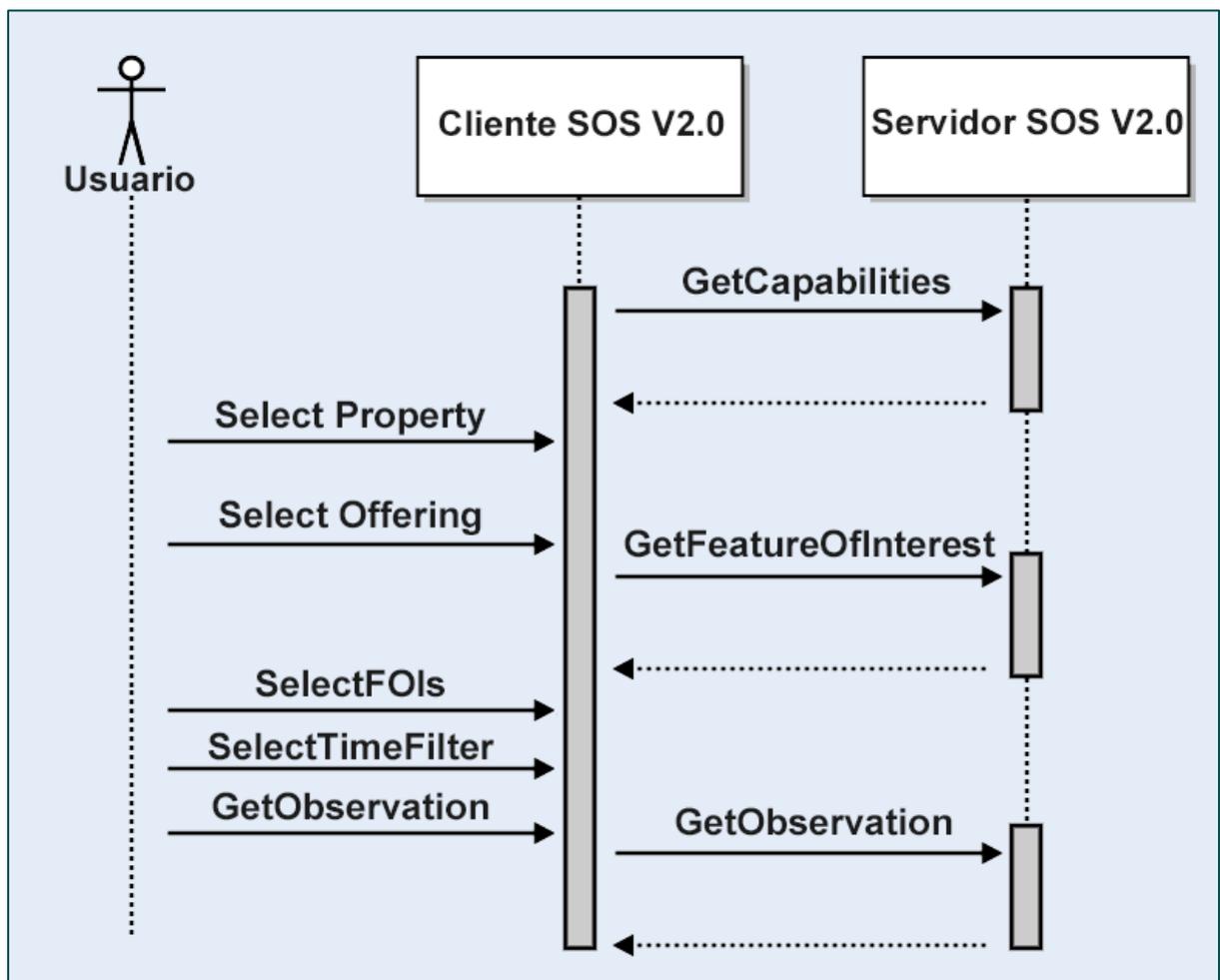


Fig. 36 Diagrama de Secuencia de petición de observaciones

5.2.2 Diagrama de actividades

En los diagramas de actividades de UML se muestra un proceso de software como un flujo de trabajo a través de una serie de acciones. Por tanto estos diagramas muestran este flujo desde el punto inicial hasta el punto final detallando las rutas de decisiones que existen en el progreso de eventos contenidos en la actividad. Además pueden describir situaciones donde el proceso paralelo puede ocurrir en la ejecución de algunas actividades [70] [71].

Las acciones del diagrama pueden ser llevadas a cabo por personas, componentes de software o equipos. Puede usarse un diagrama de actividades para describir procesos de diversos tipos, como los siguientes ejemplos:

- Procesos de negocio o un flujo de trabajo entre los usuarios y el sistema.
- Los pasos realizados en un caso de uso.
- Un protocolo de software, es decir, las secuencias de interacciones permitidas entre los componentes.
- Un algoritmo de software.

A continuación se incluyen diagramas de actividades para las tareas de adquisición de datos de observación del servidor SOS V2.0 (ver Fig. 38) y para las tareas de descarga de datos de observaciones crudos y generados (ver Fig. 37),

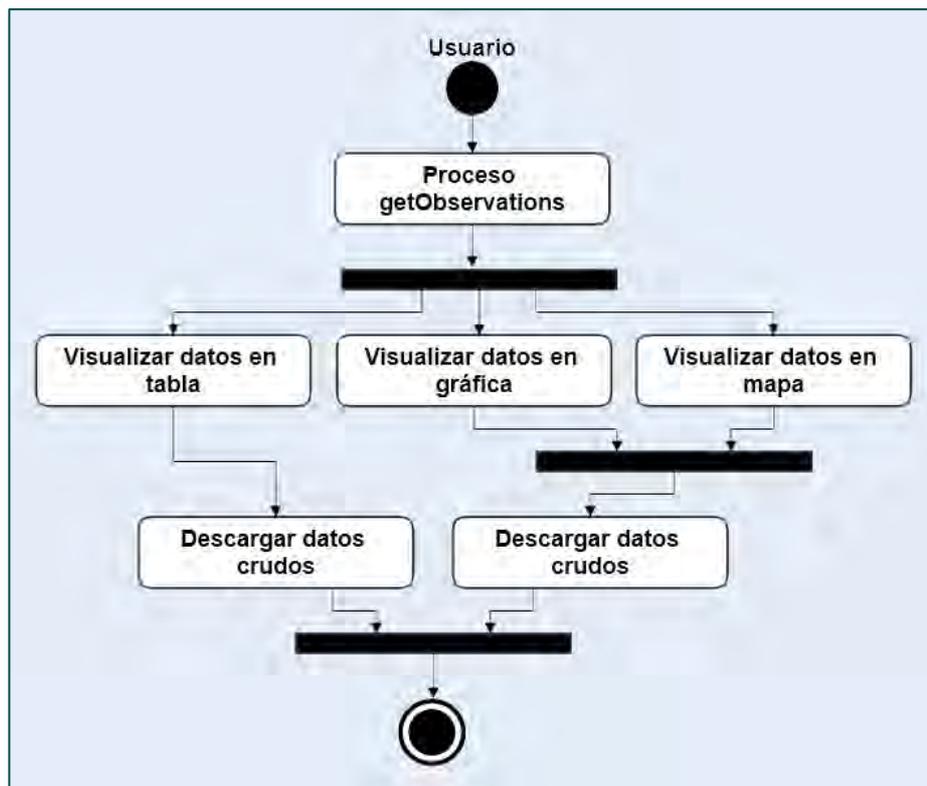


Fig. 37 Diagrama de actividad para el proceso de descarga de datos de observaciones

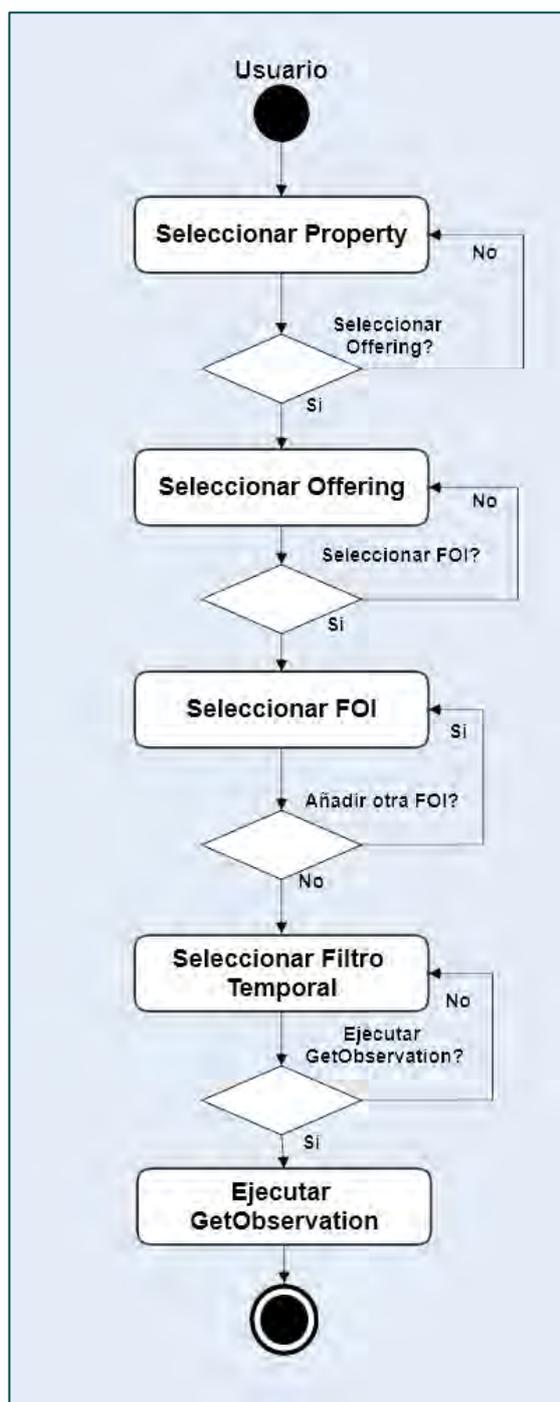


Fig. 38 Diagrama de actividad para el proceso de obtención de datos de observaciones del servidor SOS V2.0

5.3 Diseño de la interfaz gráfica de usuario

La interfaz gráfica es el elemento software que controla el usuario y que se encarga de manipular al resto de elementos que forman parte del sistema. Es por tanto, el punto de enlace entre el usuario y la aplicación. Por este motivo su diseño es fundamental, puesto que los errores en la interfaz perjudicarán la experiencia del usuario con la aplicación. Es importante realizar un diseño de calidad que se pueda traducir en una interfaz simple y amigable, fácil de comprender y que ofrezca de forma clara al usuario el conjunto de funcionalidades implementadas en la aplicación.

El objetivo final de la interfaz gráfica es permitir a cualquier usuario la utilización de las funcionalidades de la aplicación con la única ayuda del manual de usuario incluido en el apéndice de este documento.

El diseño de la aplicación se basará en los requisitos y funcionalidades de la aplicación descritos en apartados anteriores de esta memoria. Por tanto para completar este diseño hemos realizado una serie de bocetos informales o mockups en los que se muestran los distintos elementos que conformarán la aplicación gráfica. Se ha diseñado un mockup por cada iteración del ciclo de vida del proyecto, así tendremos un boceto por cada incremento desarrollado.

A continuación se incluyen los bocetos para la interfaz gráfica web del incremento 1 (Fig. 39 y Fig. 40), del incremento 2 (Fig. 41) y del incremento 3 (Fig. 42).

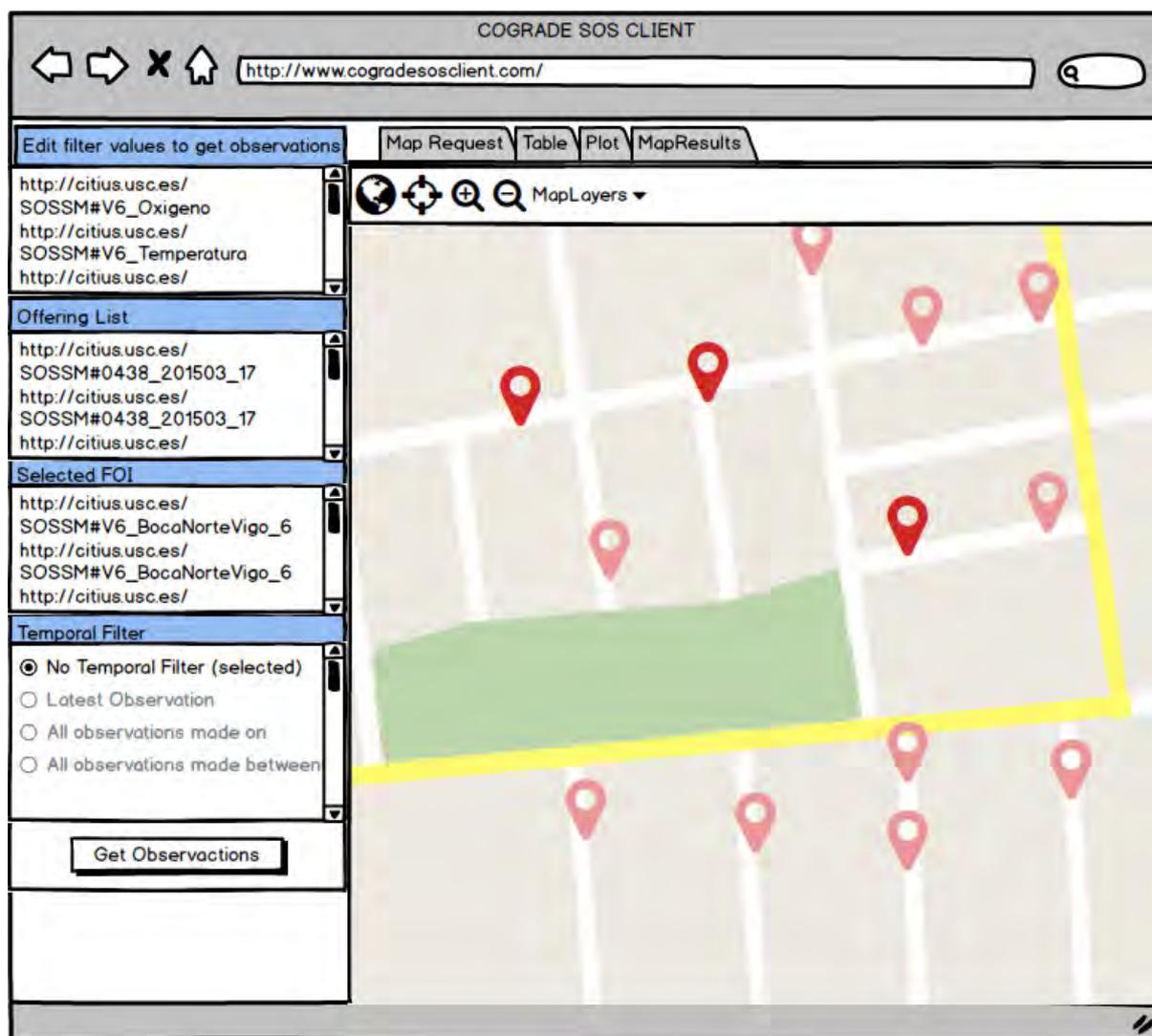


Fig. 39 Mockup de la interfaz inicial desde la que se construye la consulta de observaciones

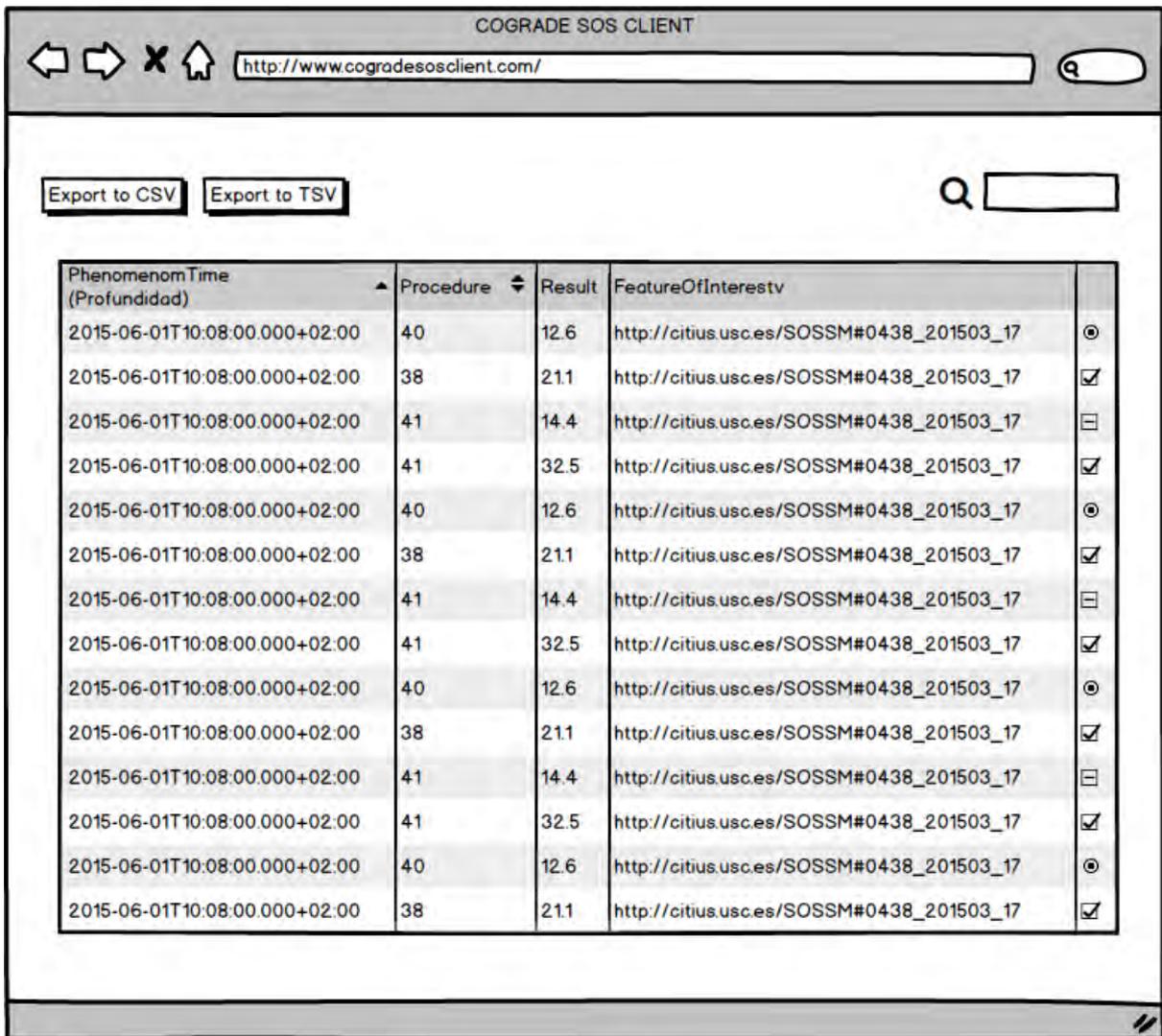


Fig. 40 Mockup de la herramienta de visualización de datos de observación en tablas

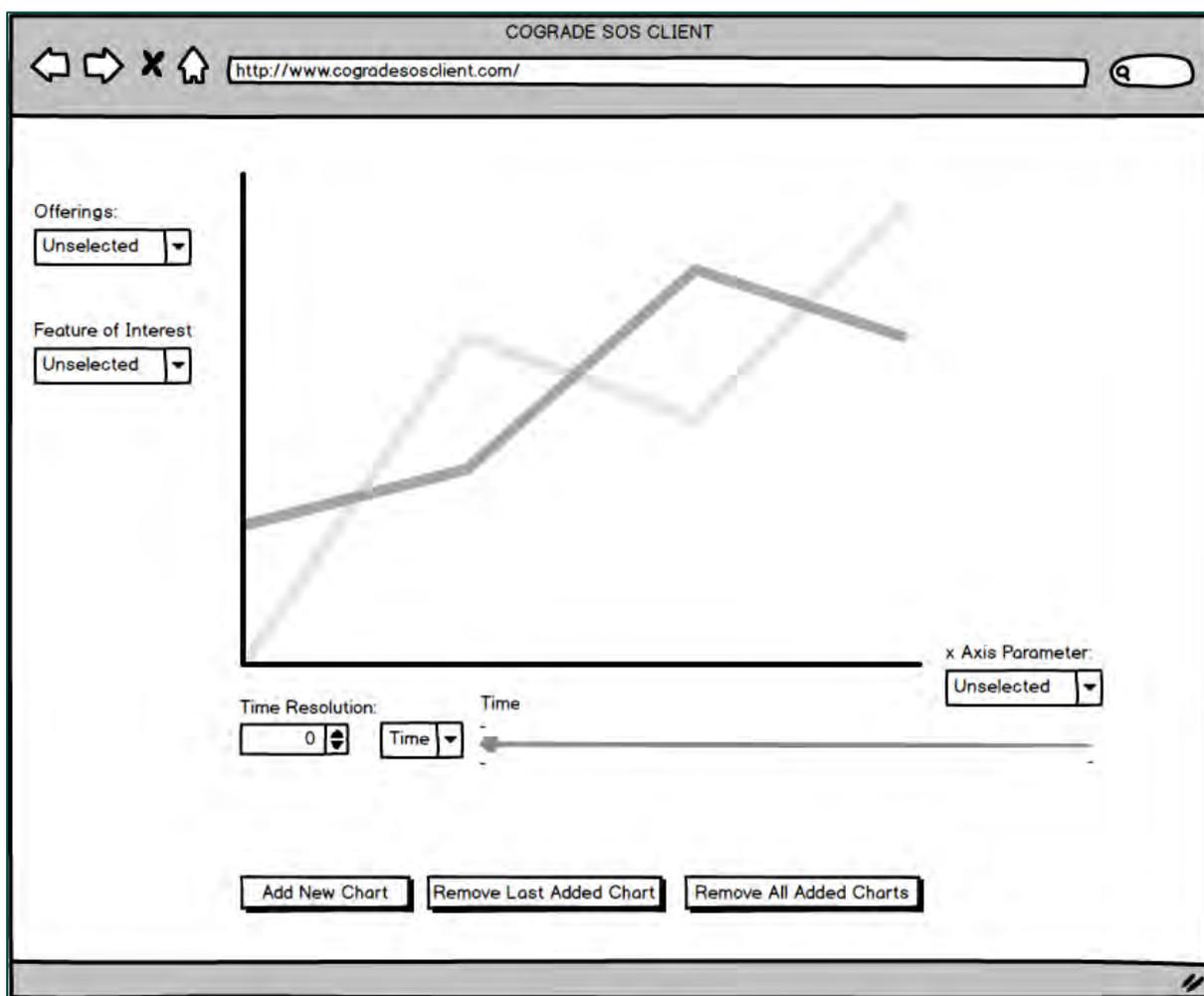


Fig. 41 Mockup de la herramienta de visualización de datos de observación en gráficas



Fig. 42 Mockup de la herramienta de visualización de datos de observación en mapas

Capítulo 6. Implementación del proyecto

La fase de implementación es la parte del ciclo de vida de los proyectos informáticos que se encarga de la transformación de los diferentes diseños en componentes de hardware, software o de ambos. Por tanto esta fase incluye la codificación de los diseños previos en lenguajes de programación y/o la implementación del hardware diseñado. Es importante haber llevado a cabo de forma correcta todas las fases anteriores del ciclo de vida del proyecto, ya que esta etapa se encarga únicamente de la traducción a lenguaje de programación de los resultados de fases anteriores y del desarrollo el prototipo hardware diseñado. Esta etapa se enfoca exclusivamente en transformar los requerimientos analizados y los diseños desarrollados en piezas de software, sin pararse a pensar sobre las alternativas de diseño. Esto permite que el desarrollador encargado de codificar estas piezas pueda centrarse en resolver los problemas técnicos propios de la codificación.

En este capítulo se especifica el proceso de implementación atendiendo al ciclo de vida incremental iterativo, utilizado en el desarrollo de este proyecto. Tal y como se describe en el diagrama de Gantt (Fig. 26) se han desarrollado tres etapas de implementación correspondientes a las tareas T07.3, T08.3 y T9.3 asociadas al incremento 1, incremento 2 e incremento 3 respectivamente.

6.1 Incremento 1

El objetivo de esta primera versión de la aplicación es implementar una aplicación que sirva como base para el resto de incrementos del ciclo de vida y que pueda realizar operaciones de consulta al servidor SOS V2.0. A continuación detallaremos las funcionalidades implementadas en cada uno de los módulos de la aplicación definidos en la fase de diseño.

Módulo Central

En este incremento se ha creado el módulo central encargado de las funcionalidades core del cliente SOS V2.0. De esta forma se han creado las siguientes funcionalidades asociadas al cliente SOS V2.0.

Se ha implementado la interfaz gráfica base mediante GeoExt V1.1. A partir de esta se podrán crear y añadir los módulos de visualización de datos de observación obtenidos del servicio SOS V2.0

Se ha implementado el formulario de consulta de datos de observaciones

Se ha implementado la función de inicialización del cliente SOS V2.0. Éste se encarga de configurar sus parámetros en relación al servidor SOS V2.0 leyendo el fichero de configuración almacenado en la carpeta de configuración. A continuación el código de dicho fichero a modo ilustrativo:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <!-- Configuración del servidor SOS -->
  <serverurl id="Cognate SOS">162.16.143.74:8080/SOS20/service</serverurl>
  <!-- Configuración de los archivos de consulta-->
  <getCapabilities id="SOS20Request">requests/GetCapabilities_allSections_V20.xml</getCapabilities>
  <getFeatureOfInterest
id="getFeatureOfInterest">requests/GetFeatureOfInterest.xml</getFeatureOfInterest>
  <getObservations id="getObservations">requests/GetObservation2.xml</getObservations>
  <!-- DEBUG MODE: verbose, console, none -->
  <debugmode>verbose</debugmode>
</root>
```

La función de inicialización se encarga de solicitar, recibir y analizar las capacidades del servidor SOS V2.0 a través de la función GetCapabilities definida en el estándar OGC SOS V2.0

Una vez se parsea la respuesta asociada al GetCapabilities y se mantienen los metadatos del servicio SOS V2.0, se actualizan los elementos correspondientes de la

interfaz gráfica. Concretamente se actualizan los campos de Server Identification, Server Provider y Property List, tal y como se ilustra en la Fig. 43

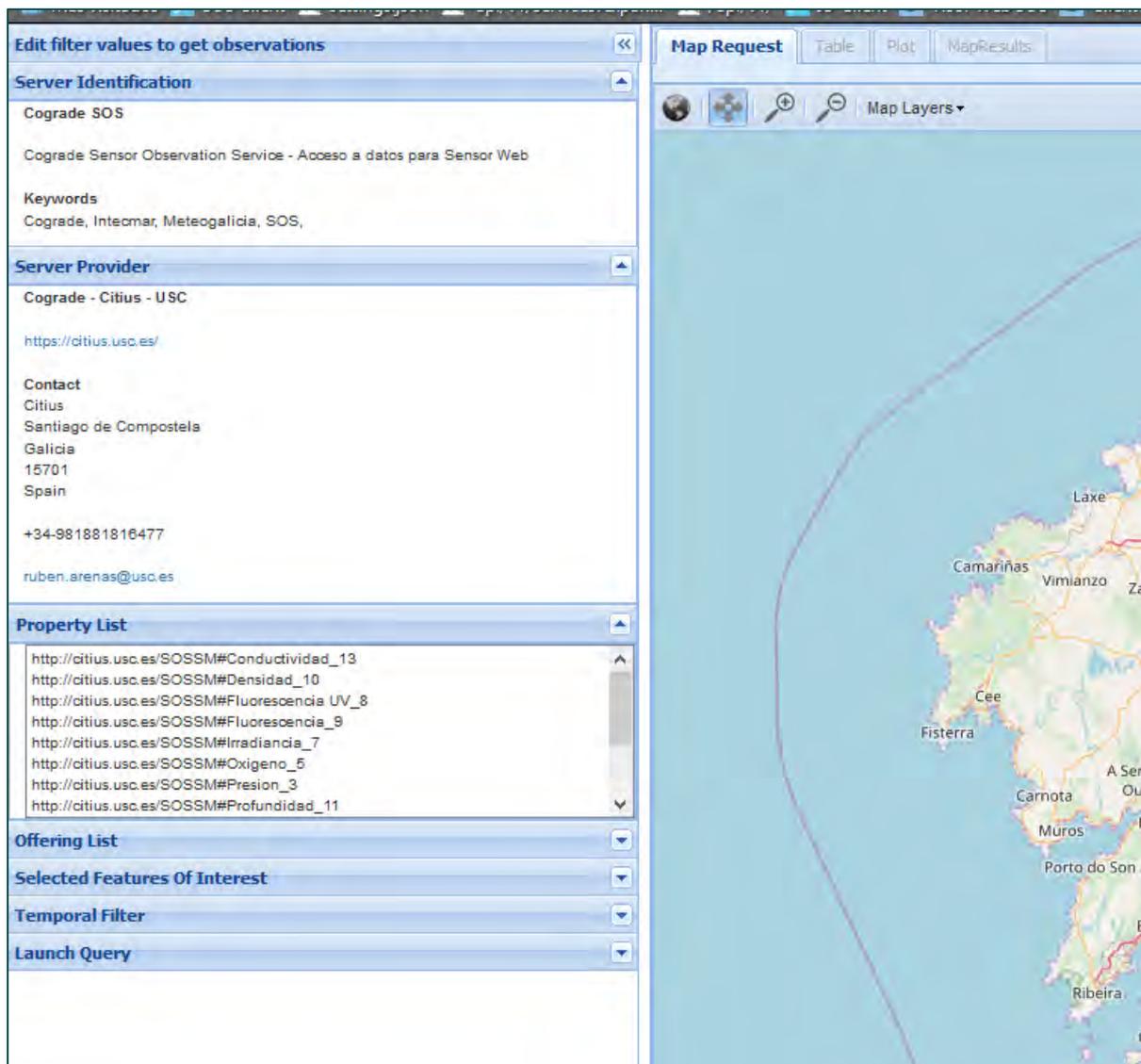


Fig. 43 Interfaz de formulario de consulta de datos de observación

Además se han implementado las funciones:

- GetFeatureOfInterest, para la obtención de las estaciones de observación
- Parseo de la respuesta asociada y relleno del listado de FOIs de la interfaz gráfica

- Parseo de Property y Offering y relleno de interfaz gráfica con los datos asociados
- Interacción en mapa para la selección de las FOIs desplegando menú con sus propiedades.
- Se han implementado 4 tipos diferentes de filtros temporales:
 - Completo
 - Última observación
 - Todas las observaciones realizadas en un momento determinado
 - Todas las observaciones de un periodo de tiempo
- Se ha implementado la función de generación de la consulta de GetObservations a partir de plantilla XML y los datos que el usuario introduce en el formulario. A continuación se incluye el código de dicha plantilla:

```
<?xml version="1.0" encoding="UTF-8"?>
<sos:GetObservation
  xmlns:sos="http://www.opengis.net/sos/2.0"
  xmlns:fes="http://www.opengis.net/fes/2.0"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:swe="http://www.opengis.net/swe/2.0"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:swes="http://www.opengis.net/swes/2.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" service="SOS" version="2.0.0"
  xsi:schemaLocation="http://www.opengis.net/sos/2.0 http://schemas.opengis.net/sos/2.0/sos.xsd">

<!-- 1 PROCEDURE -->
<sos:procedure>http://www.52north.org/test/procedure/1</sos:procedure>
<sos:procedure>http://www.52north.org/test/procedure/2</sos:procedure>

<!-- 2 OFFERINGS -->
<sos:offering>http://www.52north.org/test/offering/1</sos:offering>
<sos:offering>http://www.52north.org/test/offering/2</sos:offering>

<!-- 3 OBSERVEDPROPERTYs -->
<sos:observedProperty>http://www.52north.org/test/observableProperty/1</sos:observedProperty>
```

```

<sos:observedProperty>http://www.52north.org/test/observableProperty/2</sos:observedProperty>
<!-- 4 TEMPORALFILTERs -->
<sos:temporalFilter>
  <fes:During>
    <fes:ValueReference>phenomenonTime</fes:ValueReference>
    <gml:TimePeriod gml:id="tp_1">
      <gml:beginPosition>2012-11-19T14:00:00.000+01:00</gml:beginPosition>
      <gml:endPosition>2012-11-19T15:00:00.000+01:00</gml:endPosition>
    </gml:TimePeriod>
  </fes:During>
</sos:temporalFilter>

<!-- 5 FEATUREOFINTEREST -->
<sos:featureOfInterest>http://www.52north.org/test/featureOfInterest/1</sos:featureOfInterest>
<sos:featureOfInterest>http://www.52north.org/test/featureOfInterest/2</sos:featureOfInterest>

<!-- 6 SPATIALFILTERs -->
<sos:spatialFilter>
  <fes:BBOX>
    <fes:ValueReference>
      om:featureOfInterest/sams:SF_SpatialSamplingFeature/sams:shape
    </fes:ValueReference>
    <gml:Envelope srsName="http://www.opengis.net/def/crs/EPSSG/0/4326">
      <gml:lowerCorner>0 0</gml:lowerCorner>
      <gml:upperCorner>60 60</gml:upperCorner>
    </gml:Envelope>
  </fes:BBOX>
</sos:spatialFilter>
<sos:responseFormat>http://www.opengis.net/om/2.0</sos:responseFormat>
</sos:GetObservation>

```

de realizar una conexión con el servidor SOS V2.0 definido en los ficheros de configuración del cliente SOS V2.0. Para ello se realiza una consulta Ajax que recoge la información almacenada en la plantilla de GetCapabilities y

para ello se realiza una consulta Ajax de las capacidades que ofrecen el servicio sos 2.0

dicha consulta Ajax se encarga de ejecutar una plantilla de la función getcapabilities definida en el interfaz sos

además se ha desarrollado una función de paseo de la respuesta asociada al getcapabilities

una vez que el servidor sos procesa la consulta de capabilities devuelve al cliente un fichero XML con sus propiedades

por tanto esta función se encarga de recorrer el fichero XML en busca de las propiedades relevantes del servidor sos 2.0

una vez que se tienen almacenadas en memoria todas las propiedades relevantes del servicio sos 2.0 se rellena automáticamente los elementos de la interfaz gráfica relacionados con las propiedades que ese servicio sos contempla

Plantilla de consulta de observaciones

```
<?xml version="1.0" encoding="UTF-8"?>
<sos:GetCapabilities service="SOS"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:sos="http://www.opengis.net/sos/2.0"
  xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:swe="http://www.opengis.net/swe/2.0"      xsi:schemaLocation="http://www.opengis.net/sos/2.0
http://schemas.opengis.net/sos/2.0/sosGetCapabilities.xsd">
  <ows:AcceptVersions>
    <ows:Version>2.0.0</ows:Version>
  </ows:AcceptVersions>
  <ows:Sections>
```

```

<ows:Section>OperationsMetadata</ows:Section>
<ows:Section>ServiceIdentification</ows:Section>
<ows:Section>ServiceProvider</ows:Section>
<ows:Section>FilterCapabilities</ows:Section>
<ows:Section>Contents</ows:Section>
</ows:Sections>
<!--
  <sos:extension
xsi:type="swe:CountPropertyType"><swe:Count><swe:identifier>crs</swe:identifier><swe:value>4326</swe:
value></swe:Count></sos:extension>
  -->
<!--
  <sos:extension
xsi:type="swe:TextPropertyType"><swe:Text><swe:identifier>language</swe:identifier><swe:value>ger</swe
:value></swe:Text></sos:extension>
  -->
<!--
  <sos:extension
xsi:type="swe:BooleanPropertyType"><swe:Boolean><swe:identifier>returnHumanReadableIdentifier</swe:ide
ntifier><swe:value>true</swe:value></swe:Boolean></sos:extension>
  -->
</sos:GetCapabilities>

```

Una vez que se obtiene la respuesta del servidor SOS V2.0 asociada a la petición de observaciones. Se parsea el fichero XML devuelto, según la siguiente estructura de datos:

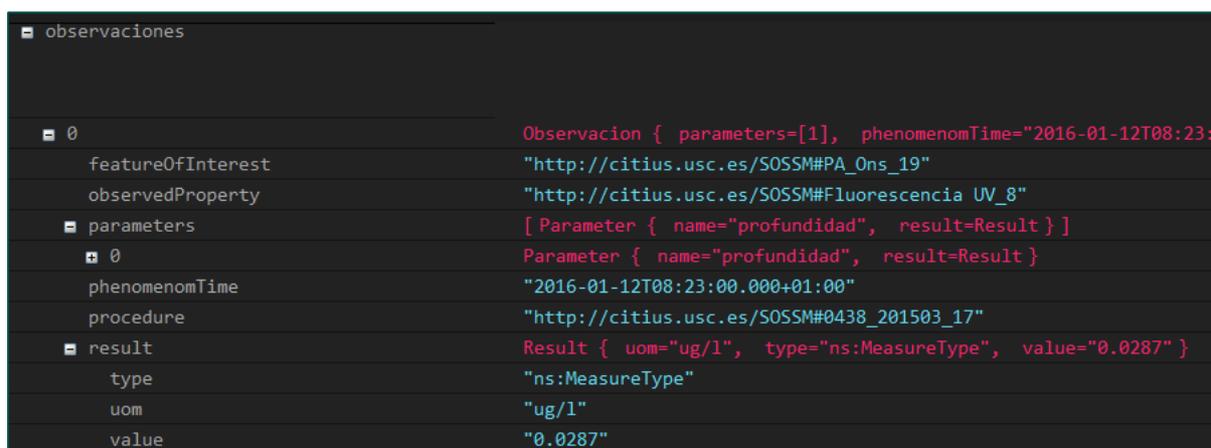


Fig. 44 Ejemplo de dato de observación parseado y almacenado en caché del navegador

Módulo Datos Tablas

En este primer incremento no se ha añadido funcionalidad asociada a este módulo.

Módulo Datos Gráficas

En este primer incremento no se ha añadido funcionalidad asociada a este módulo.

Módulo Datos Mapas

En este primer incremento no se ha añadido funcionalidad asociada a este módulo.

6.2 Incremento 2

El objetivo de esta segunda versión de la aplicación es añadir diversas funcionalidades a la estructura implementada en el incremento 1 ampliando las funcionalidades que ofrece. A continuación detallaremos las funcionalidades implementadas en cada uno de los módulos de la aplicación definidos en la fase de diseño.

Módulo Central

Se ha añadido funcionalidad de logging de mensajes atendiendo a la etiqueta debugmode almacenada en el fichero de configuración config.xml. Por tanto se contemplan 3 niveles diferentes de modo de debug: verbose, console y none.

Con el modo verbose se mostrarán mensajes de debug en diálogos emergentes según se sucedan las diferentes actividades del cliente SOS v2.0. Por el contrario, el modo console únicamente mostrará los datos de debug por consola, siendo accesible a través de las herramientas de desarrollador que los navegadores actuales presentan. El tercer modo de debug se corresponde con none. Tal y como su nombre indica, no se mostrarán mensajes de debug.

Módulo Datos Tablas

En este segundo incremento se ha implementado la funcionalidad de visualización de datos de observaciones mediante herramienta de tablas. Para ello, una vez que se almacenan los datos de observaciones parseados en memoria caché del servidor se lanza automáticamente la herramienta visualizadora de datos en formato tabla. La siguiente imagen muestra la herramienta implementada:

PhenomenonTime	Procedure	profundidad Value	profundidad Unit	FeatureOfInterest	Result	Result Unit
2016-01-12T08:23:00.000+01:00	http://citi.usc.es/SOSSM#0438_201503_17	2.257	m	http://citi.usc.es/SOSSM#PA_Ons_19	0.0287	ug/l
2016-01-12T08:23:00.000+01:00	http://citi.usc.es/SOSSM#0438_201503_17	2.284	m	http://citi.usc.es/SOSSM#PA_Ons_19	0.0296	ug/l
2016-01-12T08:23:00.000+01:00	http://citi.usc.es/SOSSM#0438_201503_17	2.294	m	http://citi.usc.es/SOSSM#PA_Ons_19	0.0279	ug/l
2016-01-12T08:23:00.000+01:00	http://citi.usc.es/SOSSM#0438_201503_17	2.319	m	http://citi.usc.es/SOSSM#PA_Ons_19	0.0281	ug/l
2016-01-12T08:23:00.000+01:00	http://citi.usc.es/SOSSM#0438_201503_17	2.338	m	http://citi.usc.es/SOSSM#PA_Ons_19	0.0299	ug/l
2016-01-12T08:23:00.000+01:00	http://citi.usc.es/SOSSM#0438_201503_17	2.356	m	http://citi.usc.es/SOSSM#PA_Ons_19	0.0304	ug/l
2016-01-12T08:23:00.000+01:00	http://citi.usc.es/SOSSM#0438_201503_17	2.382	m	http://citi.usc.es/SOSSM#PA_Ons_19	0.0277	ug/l
2016-01-12T08:23:00.000+01:00	http://citi.usc.es/SOSSM#0438_201503_17	2.407	m	http://citi.usc.es/SOSSM#PA_Ons_19	0.0281	ug/l
2016-01-12T08:23:00.000+01:00	http://citi.usc.es/SOSSM#0438_201503_17	2.474	m	http://citi.usc.es/SOSSM#PA_Ons_19	0.0308	ug/l
2016-01-12T08:23:00.000+01:00	http://citi.usc.es/SOSSM#0438_201503_17	2.55	m	http://citi.usc.es/SOSSM#PA_Ons_19	0.0304	ug/l
2016-01-12T08:23:00.000+01:00	http://citi.usc.es/SOSSM#0438_201503_17	2.591	m	http://citi.usc.es/SOSSM#PA_Ons_19	0.0302	ug/l
2016-01-12T08:23:00.000+01:00	http://citi.usc.es/SOSSM#0438_201503_17	2.63	m	http://citi.usc.es/SOSSM#PA_Ons_19	0.032	ug/l
2016-01-12T08:23:00.000+01:00	http://citi.usc.es/SOSSM#0438_201503_17	2.701	m	http://citi.usc.es/SOSSM#PA_Ons_19	0.0341	ug/l
2016-01-12T08:23:00.000+01:00	http://citi.usc.es/SOSSM#0438_201503_17	2.799	m	http://citi.usc.es/SOSSM#PA_Ons_19	0.0356	ug/l

Fig. 45 Ejemplo de observaciones visualizadas con la herramienta de tablas

Mediante esta herramienta podemos organizar los datos de observaciones ascendente o descendientemente según columnas. Así mismo podremos realizar filtrados de los datos a través de la caja de búsqueda situada en la parte superior derecha de la herramienta. Hay que destacar que esta herramienta se encarga de mostrar los datos de observaciones crudos, tal y como el servidor SOS 2.0 los devuelve.

Módulo Datos Gráficas

En este segundo incremento se ha implementado la funcionalidad de visualización de datos de observaciones mediante herramienta de tablas. Esta herramienta se encarga de procesar los datos según las resoluciones que el usuario introduzca en la interfaz de la herramienta. En la imagen incluida a continuación se ilustra la herramienta implementada, así como los filtros de resolución desarrollados.

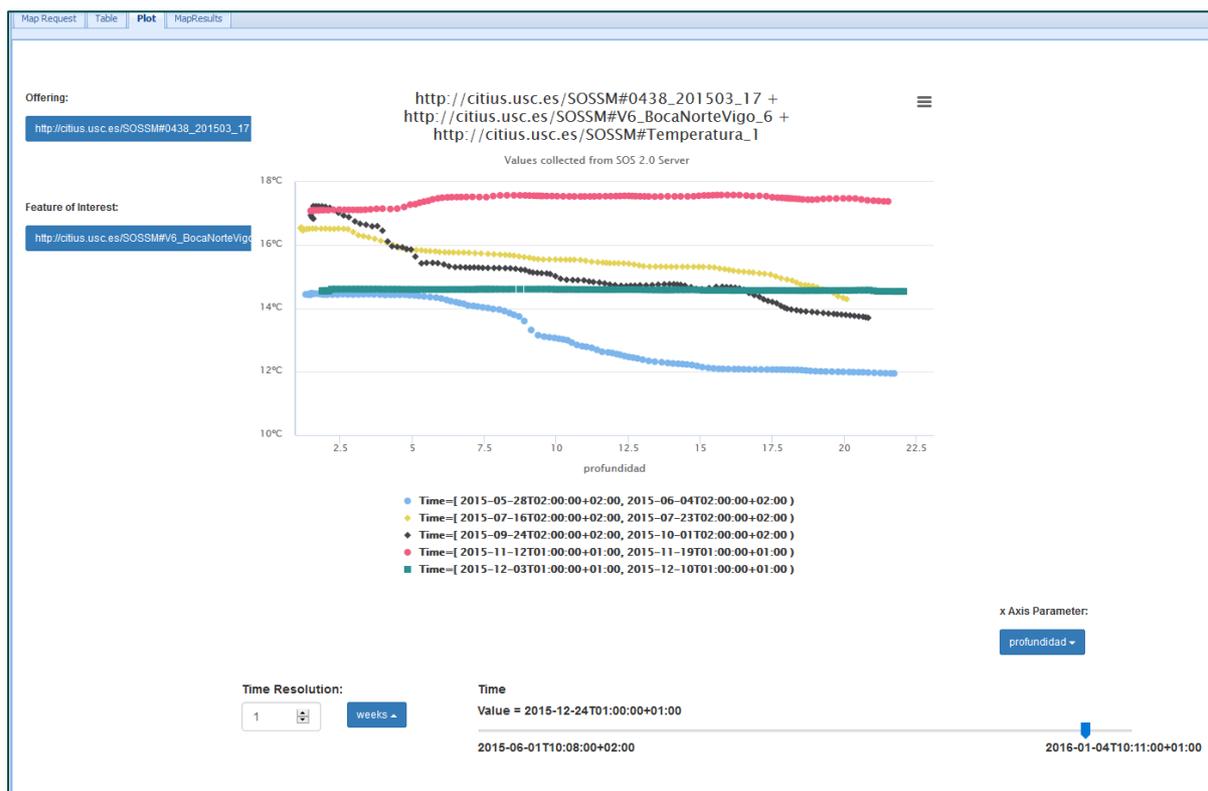


Fig. 46 Herramienta de visualización de datos mediante gráfica

Hay que indicar que esta herramienta de visualización se encarga de procesar mediante técnicas estadísticas básicas los datos de observaciones. Se utiliza el valor que el usuario introduzca en la agrupación de elementos web relativos a la resolución. Para ello se realizan agrupaciones de los datos mediante la operación de media, y atendiendo a la resolución. De esta manera el dato representado en la gráfica se corresponde con el dato medio que se ha registrado en un intervalo teniendo en cuenta la resolución del parámetro.

Hay que indicar que se pueden realizar graficas de series temporales. Para ello, y a diferencia de los clientes ligeros que existen en la actualidad, no se ha empleado la interfaz TimeSeries.

Además hay que destacar que esta herramienta, además de series temporales, es capaz de representar series paramétricas. En el caso concreto de la fuente de datos que estamos empleando, podremos realizar gráficas atendiendo al parámetro de profundidad, tal y como se detalla en la siguiente imagen:

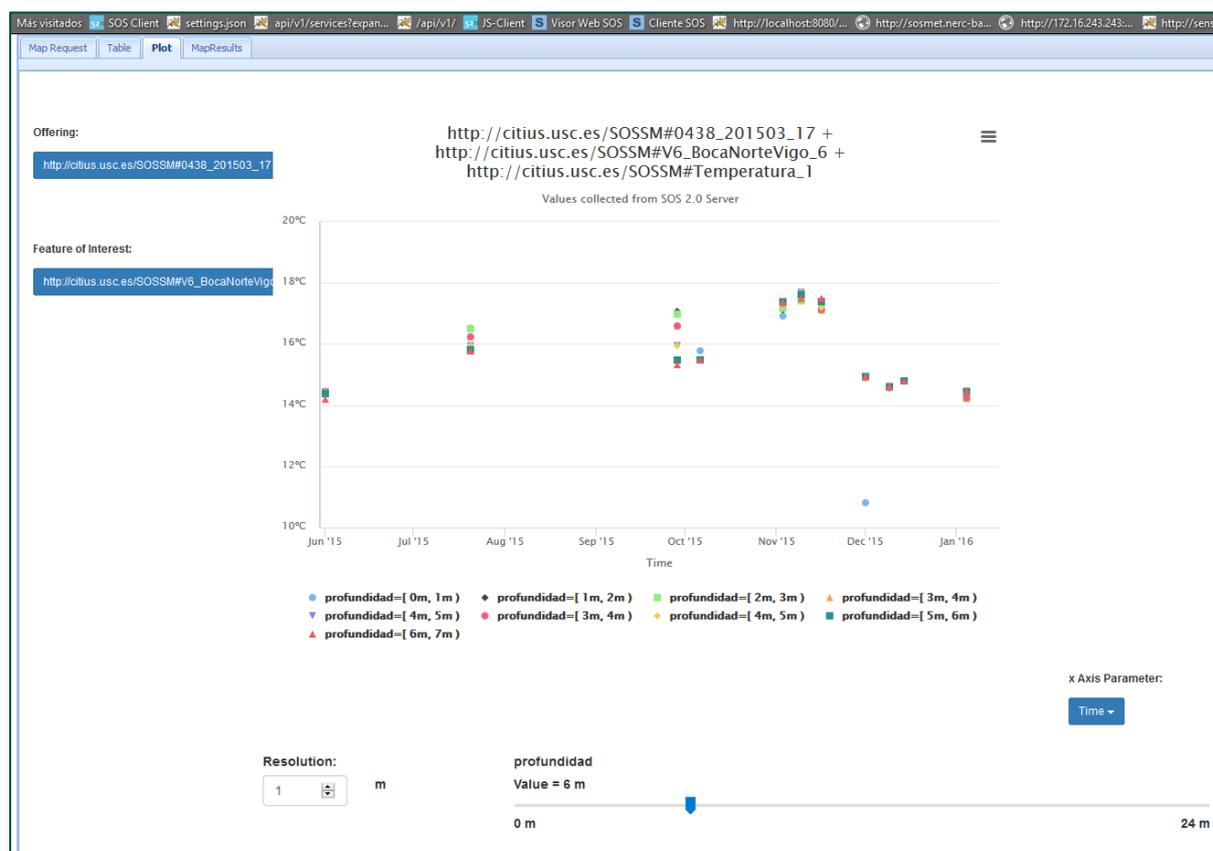


Fig. 47 Ejemplo de gráfica con datos de series de profundidad para intervalo de tiempo de 1 año.

Esta herramienta ha sido implementada partiendo de la base que ofrecer la librería highcharts. A partir de ejemplos básicos se ha desarrollado toda la algoritmia asociada a los parámetros.

Hay que indicar que la herramienta está implementada para que se puedan tratar datos de observaciones con un número ilimitado de parámetros. Para ello se ha definido la construcción de los elementos de la interfaz de forma dinámica. Así si los datos que queremos representar contienen 2 o más parámetro, tendremos que introducir valores de resolución en 2 o más grupos de sliders y textboxes de resolución.

Módulo Datos Mapas

En este segundo incremento no se ha añadido funcionalidad asociada a este módulo.

6.3 Incremento 3

El objetivo de esta tercera versión de la aplicación es añadir diversas funcionalidades a la versión previa ampliando el número de operaciones de carga, visualización, tratamiento y almacenamiento de imágenes.

A continuación detallaremos las funcionalidades implementadas en cada uno de los módulos de la aplicación definidos en la fase de diseño.

Módulo Central

En este incremento se han refinado tanto las funciones de consulta como las funciones de parseo de respuestas. El objetivo ha sido optimizar el funcionamiento de este conjunto de funciones. Se han eliminado ciertos bucles for que provocaban retrasos en el parseo y almacenamiento en caché de los datos de observaciones.

Módulo Datos Tablas

En este incremento se han añadido las funcionalidades de descarga de datos observaciones en los formatos csv y tsv. De esta manera se ofrece al usuario la opción de poder gestionar los datos de observaciones bajo su propio criterio y con herramientas destinadas a realizar diferentes procesamientos de los datos de los contemplados en el cliente SOS V2.0 que hemos desarrollado.

Módulo Datos Gráficas

Al igual que en el modulo de datos de tablas, en este incremento se ha añadido la funcionalidad de descarga de los datos en diferentes formatos, tal y como ilustra la imagen introducida a continuación:

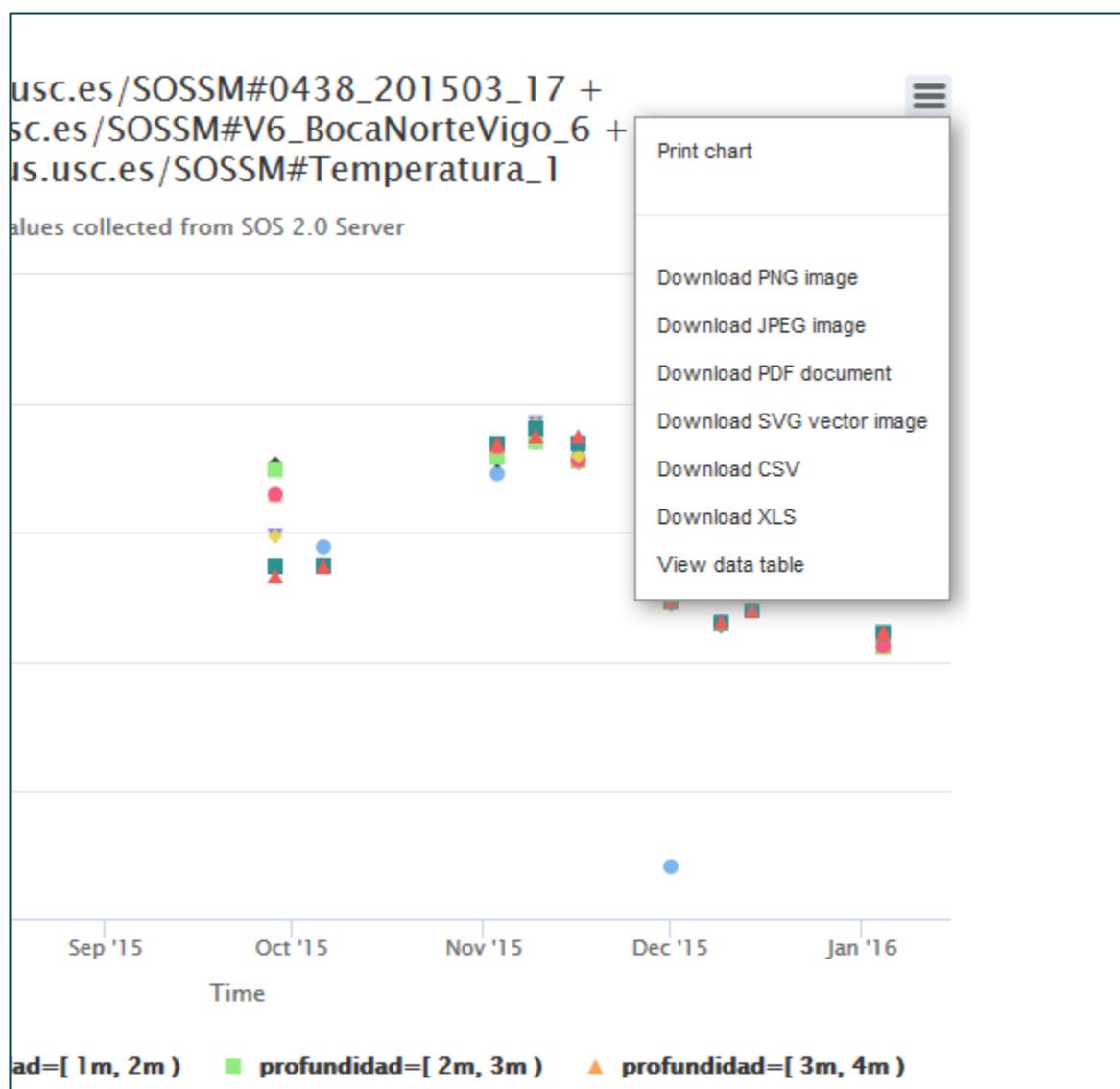


Fig. 48 Menú desplegable de la herramienta de gráficas que permite descargar los datos generados

Tal y como se detalla en la anterior imagen, la herramienta de gráficas permite descargar los datos en los formatos:

- Imagen PNG
- Imagen JPG
- Impresión directa
- Documento PDF

- Imagen vectorial SVG
- Documento CSV
- Documento XLS
- Visualización de los datos generados en formato tabla.

Módulo Datos Mapas

En este tercer incremento se ha implementado la visualización de los datos de observación en mapas de forma análoga a la implementada en la herramienta de graficas. Así se generan grupos de cajas de texto y slider asociados a parámetros. Estos grupos de elementos web se generan dinámicamente atendiendo al tipo de datos de observaciones que el servidor SOS V2.0 devuelve, tal y como detalla la imagen a continuación:

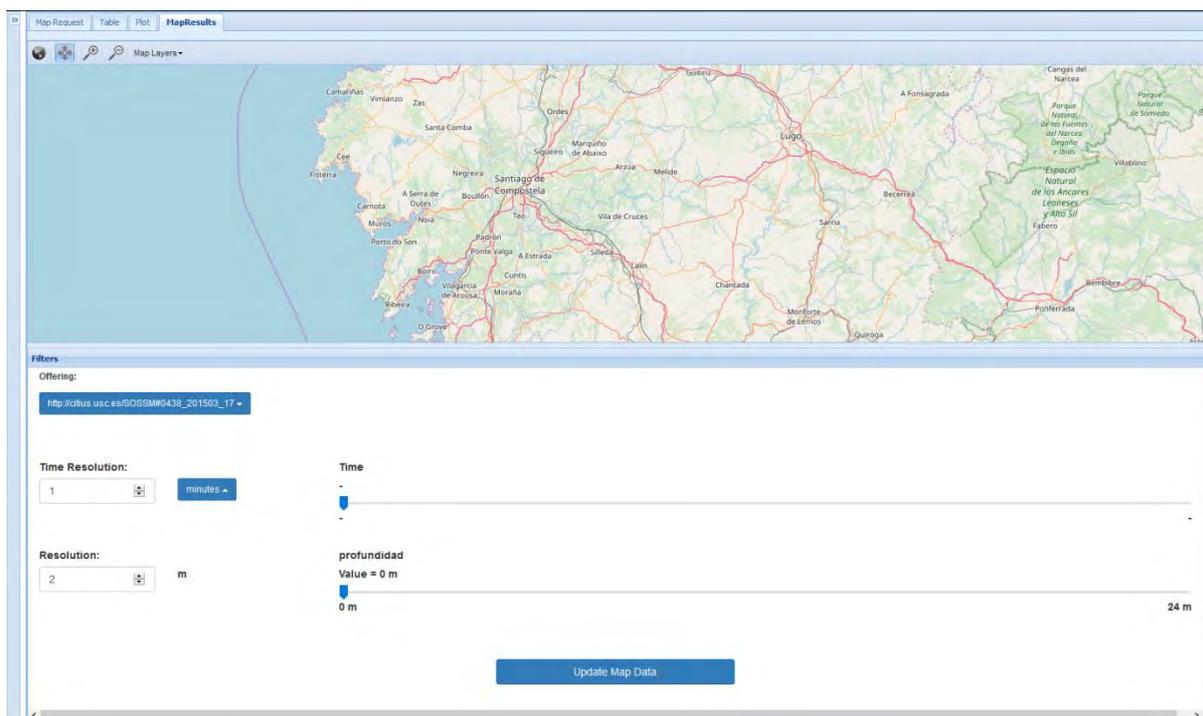


Fig. 49 Herramienta de visualización de datos de observaciones en mapa

Capítulo 7. Pruebas

Las pruebas son ejecuciones o ensayos de funcionamiento que se realizan en una fase posterior a la implementación del código. Éstas efectúan una evaluación de la calidad del producto generado para poder detectar posibles defectos antes de la puesta en explotación. Son por tanto, un método para poder verificar y validar el software creado.

En este capítulo se expondrán las pruebas realizadas detallando el grado de cumplimiento de las funcionalidades implementadas con respecto a las funcionalidades del proyecto establecidas en la fase de análisis del mismo.

La validación del software se ha realizado conjuntamente con el cliente durante las diferentes reuniones del proyecto con el fin de comprobar que éste cumple con las necesidades para las que ha sido desarrollado.

Se ha realizado una fase de pruebas y validación por cada uno de los incrementos realizados durante el ciclo de vida del proyecto. Así podemos observar en el diagrama de Gantt del proyecto (ver Fig. 26) las tareas de pruebas y validación T07.3, T08.4 y T9.4 asociadas al incremento 1, incremento 2 e incremento 3 respectivamente. El resultado de estas tres tareas será detallado en secciones posteriores de este capítulo.

Como último apartado de esta sección se describirá la evaluación heurística de la interfaz gráfica realizada basándose en los diez principios de Nielsen. El objetivo es garantizar la usabilidad de la implementación de la interfaz gráfica. Entendemos por usabilidad como la facilidad con que las personas pueden utilizar una herramienta particular o cualquier otro objeto fabricado por humanos con el fin de alcanzar un objetivo concreto.

7.1 Verificación

Las pruebas realizadas se describirán bajo la estructura de la Tabla 67:

Identificador	ID
Descripción	Breve resumen de la prueba
Requisitos	Requerimientos analizados en la prueba
Entradas	Datos con los que se ejecutará la prueba
Procedimiento	Descripción de los pasos para realizar la prueba. Cada paso se identificará con ID-X, donde ID es el identificador de la prueba y X una letra única.
Resultado	Describe el estado final del sistema tras la ejecución de cada paso de la prueba. Cada resultado está identificado con el mismo código ID-X del paso del procedimiento de la prueba al que da respuesta.

Tabla 67 Plantilla de descripción de pruebas

ID	P-01
Descripción	Prueba de petición de datos de observación de servidor SOS V2.0
Requisitos	RQ-FN-01, RQ-FN-02, RQ-FN-03, RQ-FN-04, RQ-FN-05, RQ-FN-06, RQ-FN-07, RQ-FN-08, RQ-FN-09, RQ-CA-01, RQ-CA-02, RQ-CA-03, RQ-DI-01, RQ-DI-03, RQ-IG-06, RQ-IG-05, RQ-IG-01, RQ-EM-01, RQ-EM-02, RQ-EM-03
Entradas	-

Procedimiento	<p>P-01-A: El usuario invoca al cliente SOS V2.0 en navegador web</p> <p>P-01-B: El cliente SOS V2.0 configura automáticamente la conexión con el servidor SOS V2.0 según el fichero de configuración</p> <p>P-01-C: El cliente SOS V2.0 ejecuta un GetCapabilities a partir de plantilla almacenada en carpeta de configuración</p> <p>P-01-D: El cliente SOS V2.0 recibe respuesta del Servidor SOS V2.0 y la parsea en memoria caché del navegador</p>
Resultado	<p>P-01-A: El cliente SOS V2.0 está cargado en el navegador, con los paneles de visualización desactivados.</p> <p>P-01-B: El cliente SOS V2.0 continua con la interfaz desactivada</p> <p>P-01-C: El cliente SOS V2.0 continua con la interfaz desactivada</p> <p>P-01-D: El cliente SOS V2.0 activa los paneles del formulario de solicitud de observaciones y rellena los paneles con los datos parseados de la repuesta del GetCapabilities del servidor SOS V2.0</p>

Tabla 68 Prueba P-01

ID	P-02
Descripción	Prueba de visualización de datos en los tres formatos admitidos
Requisitos	RQ-FN-01, RQ-FN-02, RQ-FN-03, RQ-FN-04, RQ-FN-05, RQ-FN-06, RQ-FN-07, RQ-FN-08, RQ-FN-09, RQ-CA-01, RQ-CA-02, RQ-CA-03, RQ-DI-01, RQ-DI-03, RQ-IG-06, RQ-IG-05, RQ-IG-01, RQ-EM-01, RQ-EM-02, RQ-EM-03, RQ-FN-10, RQ-FN-11, RQ-FN-12, RQ-FN-13, RQ-PR-01, RQ-PR-06, RQ-EV-01, RQ-EV-02, RQ-IG-02, RQ-IG-03, RQ-IG-04
Entradas	-

<p>Procedimiento</p>	<p>P-02-A El usuario accede al cliente web SOS V2.0 en su navegador</p> <p>P-02-B: El usuario rellena el formulario de solicitud de dato de observación</p> <p>P-02-C: El usuario lanza la petición de datos de observación</p> <p>P-02-D: El cliente SOS V2.0 espera la respuesta del servidor SOS V2.0 desactivando temporalmente la interfaz gráfica.</p> <p>P-02-E: El cliente SOS V2.0 recibe la respuesta y parsea los datos</p> <p>P-02-F: El cliente SOS V2.0 activa la herramienta de visualización de datos de observación en formato tabla y representa en ella los datos crudos obtenidos del servidor.</p> <p>P-02-G: El usuario juega con la interfaz manipulando los diferentes elementos web.</p> <p>P-02-H: El usuario descarga datos en formato csv y tsv de la herramienta de visualización de datos de observación en formato tabla</p> <p>P-02-I: El usuario accede a la herramienta de visualización de datos de observación en formato gráfica</p> <p>P-02-J: El usuario manipula los controles de la herramienta y visualiza datos de observación.</p> <p>P-02-K: El usuario descarga datos en formato csv, xls, pn y pdf de la herramienta de visualización de datos de observación en formato grafica</p> <p>P-02-L: El usuario accede a la herramienta de visualización de datos de observación en formato mapa</p> <p>P-02-M: El usuario manipula los controles de la herramienta y visualiza datos de observación.</p>
----------------------	---

<p>Resultado</p>	<p>P-01-A: El cliente SOS V2.0 está cargado en el navegador, con los paneles de visualización desactivados.</p> <p>P-02-B: Se resaltan los diferentes elementos que el usuario ha seleccionado o establecido.</p> <p>P-02-C: El cliente SOS V2.0 queda a la espera de respuesta del servidor SOS V2.0</p> <p>P-02-D: La interfaz del cliente SOS V2.0 está desactivada.</p> <p>P-02-E: La interfaz del cliente SOS V2.0 está desactivada</p> <p>P-02-F: El usuario observa automáticamente los datos de la respuesta en el visualizador en formato tabla.</p> <p>P-02-G: Se refrescan y ordenan los datos a antojo del usuario.</p> <p>P-02-H: Se almacenan en disco local los ficheros de datos de observación</p> <p>P-02-I: El cliente web SOS V2.0 oculta la herramienta de visualización de datos en formato tabla y muestra la herramienta de visualización de datos en formato gráfica sin datos en ella.</p> <p>P-02-J: La interfaz de la herramienta de visualización de datos en formato gráfica se actualiza conforme el usuario introduce datos.</p> <p>P-02-K: Se almacenan en disco local los ficheros de datos de observación</p> <p>P-02-L: El cliente web SOS V2.0 oculta la herramienta de visualización de datos en formato gráfica y muestra la herramienta de visualización de datos en formato mapa sin datos en ella.</p> <p>P-02-M: La interfaz de la herramienta de visualización de datos en formato gráfica se actualiza conforme el usuario introduce datos.</p>
------------------	--

Tabla 69 Prueba P-02

ID P-03	
Descripción	Prueba de lectura de memoria
Requisitos	RQ-PR-01, RQ-PR-02, RQ-PR-03, RQ-PR-04, RQ-PR-05, RQ-PR-06, RQ-PR-07, RQ-SO-01, RQ-SO-02, RQ-SO-03
Entradas	-
Procedimiento	P-01-A: El investigador lee el presente documento
Resultado	P-01-A: El documento está listo para ser entregado

Tabla 70 Prueba P-03

7.2 Validación

En este apartado se realizará una comprobación del estado de cumplimiento de cada requisito en función de las pruebas realizadas. La validación de requisitos se ha desarrollado conjuntamente con el cliente durante las diferentes reuniones del proyecto con el fin de comprobar que éste cumple con las necesidades para las que ha sido desarrollado.

Requisito	Estado	Prueba justificadora
RQ-PR-01	Completo	P-02, P-03
RQ-PR-02	Completo	P-03
RQ-PR-03	Completo	P-03
RQ-PR-04	Completo	P-03
RQ-PR-05	Completo	P-03
RQ-PR-06	Completo	P-02, P-03
RQ-PR-07	Completo	P-03
RQ-FN-01	Completo	P-01, P-02
RQ-FN-02	Completo	P-01, P-02
RQ-FN-03	Completo	P-01, P-02
RQ-FN-04	Completo	P-01, P-02
RQ-FN-05	Completo	P-01, P-02
RQ-FN-06	Completo	P-01, P-02
RQ-FN-07	Completo	P-01, P-02
RQ-FN-08	Completo	P-01, P-02
RQ-FN-09	Completo	P-01, P-02
RQ-FN-10	Completo	P-02
RQ-FN-11	Completo	P-02

RQ-FN-12	Completo	P-02
RQ-FN-13	Completo	P-02
RQ-CA-01	Completo	P-01, P-02
RQ-CA-02	Completo	P-01, P-02
RQ-CA-03	Completo	P-01, P-02
RQ-EV-01	Completo	P-02
RQ-EV-02	Completo	P-02
RQ-DI-01	Completo	P-01, P-02
RQ-DI-02	Completo	P-02
RQ-DI-03	Completo	P-01, P-02
RQ-SO-01	Completo	P-03
RQ-SO-02	Completo	P-03
RQ-SO-03	Completo	P-03
RQ-IG-01	Completo	P-01, P-02
RQ-IG-02	Completo	P-02
RQ-IG-03	Completo	P-02
RQ-IG-04	Completo	P-02
RQ-IG-05	Completo	P-01, P-02
RQ-IG-06	Completo	P-01, P-02

RQ-EM-01	Completo	P-01, P-02
RQ-EM-02	Completo	P-01, P-02
RQ-EM-03	Completo	P-01, P-02

Tabla 71 Tabla de descripción de comprobaciones

Capítulo 8. Conclusiones

Esta sección se encarga de cerrar el presente documento de trabajo final de máster exponiendo los logros alcanzados mediante las actividades de investigación y desarrollo detalladas en las secciones anteriores.

Desde un punto de vista general, el problema resuelto en este trabajo ha sido la investigación, desarrollo e implantación en las plataformas de laboratorio de las agencias MeteoGalicia e Intecmar de una herramienta web innovadora basada en el estándar OGC SOS 2.0. Esta herramienta se encarga de poner en valor las capacidades y bondades de la tecnología de integración semántica de datos de fuentes heterogéneas de datos de observación desarrollada en el grupo de investigación COGRADE de la Universidad de Santiago de Compostela. Para ello se ha desarrollado un prototipo funcional de cliente web pesado basado en arquitectura en tres capas de conexión directa con servidor SOS 2.0. Este modelo de arquitectura delega buena parte de las tareas de procesamiento de los datos en el lado del cliente web.

Las actividades de desarrollo han sido estrictamente diseñadas y guiadas por las bases de las especificaciones establecidas en el estándar OGC SOS V2.0. Por tanto se ha implementado la interfaz SOS como único nexo de unión entre las capas de la arquitectura. La herramienta desarrollada se corresponde con una aplicación plenamente innovadora, ya que es la primera solución real con esta característica, dado que el resto de clientes SOS V2.0 se corresponden con clientes ligeros que implementan la interfaz SOS además de la interfaz Timeseries. Ésta última no está contemplada en el estándar OGC SOS V2.0.

El cliente SOS 2.0 desarrollado en este trabajo se encarga de configurar la conexión a un servicio SOS 2.0 y de forma automática realizar las operaciones necesarias para la consulta de capacidades del mismo. Así el cliente SOS 2.0 muestra los metadatos relevantes del servicio SOS 2.0 en un formulario que el usuario debe completar con el

objetivo de facilitar y automatizar la formalización de peticiones de datos de observaciones del servidor SOS 2.0.

Una vez se ejecutan las peticiones que el usuario ha configurado y posteriormente se reciben las respuestas procedentes del servidor SOS 2.0, el cliente SOS se encarga de parsear automáticamente y de forma eficiente la información contenida del fichero de respuesta. Para ello, el contenido de los ficheros de respuesta de consultas de datos de observación es analizado y almacenado en la memoria caché del navegador según la estructura de datos detallada en secciones anteriores. Estos datos pueden ser mostrados y manipulados mediante tres herramientas diferentes diseñadas y desarrolladas para este propósito concreto. Estas herramientas son:

- Visualizador de datos crudos en formato tabla
- Visualizador de datos generados en formato gráfica
- Visualizador de datos generados en formato mapa

Para el procesamiento y generación de los datos se ha investigado en cuanto a la forma del tratamiento de datos de observaciones que actualmente se ofrece en herramientas del ámbito. Por tanto, se ha optado por implementar algoritmos de tratamiento y generación de datos basados en agrupamiento e interpolación según filtros de resolución temporales y/o de parámetros. Esta característica es la que dota de pesadez y valor funcional al cliente SOS V2.0 desarrollado. Recordemos que todos los clientes existentes en la actualidad se corresponden con clientes ligeros, que únicamente permiten la descarga de los datos crudos obtenidos del servidor SOS V2.0

8.1 Conocimientos y habilidades adquiridas

Durante los procesos de investigación y desarrollo del proyecto detallado en la presente memoria se han adquirido habilidades y conocimientos en diferentes ámbitos. A continuación se realiza un resumen de cada una de ellas.

- Se ha profundizado en el conocimiento sobre infraestructuras de datos espaciales orientadas a datos de observación
- Se ha continuado con la formación en aspectos relacionados con las arquitecturas orientadas a servicios. Durante el máster ISSI el autor de este TFM cursó la asignatura de AOS. En este TFM se ha profundizado en el conocimiento sobre este ámbito en cuanto a las arquitecturas en tres capas aplicadas a servicios de datos de observaciones y los protocolos de petición y respuesta más empleados en la actualidad.
- También se ha ahondado en las normativas actuales sobre la publicación de datos financiados con subvención pública. Concretamente, se ha revisado la directiva Directiva 2007/2/CE comúnmente denominada directiva INSPIRE.
- Se han estudiado los conceptos de estándares dedicado a definir un marco interoperable en el ámbito del acceso y difusión de datos públicos de sensores. El estándar aplicado en este TFM se corresponde con el OGC S.O.S. V2.0
- Se ha profundizado sobre las actividades de investigación y desarrollo de iniciativas y organizaciones que promulgan el avance tecnológico de la sociedad, encarando la transformación digital hacia un futuro donde los datos abiertos y la transparencia de las administraciones jueguen un papel fundamental.
- Se han aplicado los conocimientos de ingeniería de software adquiridos en las diferentes materias del máster ISSI aplicados a la resolución de procesos de un proyecto de investigación real. Se ha podido comprobar empíricamente la utilidad y validez de los diferentes métodos estudiados en el mundo real. Con esto se ha conseguido obtener una visión integradora y global de los procesos de ingeniería de software.

- En cuanto a desarrollo se ha profundizado en la construcción de herramientas modulares de integración directa con servicios existentes de antemano. Se ha interiorizado la forma de trabajo basada en contratos que promueven los estándares de desarrollo.
- Además se ha comprobado como las tecnologías de integración de datos pueden ser una útil solución a problemas que hasta el momento se resuelven mediante métodos menos elegantes basados en procesos reiterativos, solapados y que en multitud de ocasiones continúan desarrollándose de forma semiautomática implicando la supervisión humana.
- En cuanto a conocimiento técnico hay que detallar que se han experimentado los métodos de desarrollo de aplicaciones web basados en integración de tecnologías, frameworks y librerías de diferente índole.
- Se ha adquirido experiencia en cuanto a habilidades y conocimientos sobre la actividad investigadora desarrollada en centros universitarios y centros tecnológicos, dado el carácter especial del presente proyecto.
- Se han aplicado métodos de análisis matemático y estadístico para la generación de datos mediante resoluciones o zoom. Para ello se manipulan los datos obtenidos de la consulta de recogida de datos del servidor SOS V2.0 con el objetivo de ajustarse a dichas resoluciones o zooms.

8.2 Trabajo futuro

La herramienta desarrollada a partir de los conocimientos resultantes de las tareas de investigación realizadas en este TFM se considera como una primera versión prototípica de lo que en un futuro cercano (debido a la entrada en vigor de la directiva INSPIRE) será una herramienta disponible por la ciudadanía. Como todo prototipo presenta aspectos que deben ser mejorados y extendidos. Por tanto el trabajo futuro de este proyecto se puede resumir en las siguientes ideas:

- Será necesaria una modificación de la herramienta de visualización de datos generados a partir de los datos de observación que devuelve el servidor SOS V2.0 dado que la usabilidad de la misma no permite explotarlos de la forma más eficiente y útil posible.
- Se contemplan tareas de migración del contenedor web Tomcat 8 hacia el servidor corporativo de aplicaciones JBoss. El proyecto se encuentra en este momento en entorno de laboratorio contemplado en el protocolo de desarrollo de aplicaciones de la Xunta de Galicia. Para poner el prototipo en entorno de preproducción y posteriormente producción es necesario incluir garantías de servicio y seguridad que Tomcat no ofrece. Por tanto la solución pasa por migrar la herramienta cliente SOS V2.0 a la plataforma corporativa bajo la versión EAP 6.2 de JBoss.
- Se prevén pruebas completamente funcionales de la tecnología de integración semántica de datos de servidor SOS V2.0 en un lapso de tiempo medio. Actualmente el grupo COGRADE se encuentra en las últimas fases de implementación de la migración de la plataforma de integración de datos de la versión 1.0 del estándar OGC SOS a la versión 2.0
- Como requisito de mejora del cliente SOS V2.0 se contempla la creación de un sistema que permita el almacenamiento de diferentes consultas prototipo por parte de los usuarios. Se ha detectado que este tipo de prácticas es habitual en los diferentes miembros de las agencias meteorológicas MeteoGalicia e Intecmar que están empleando en la actualidad la herramienta.
- Se contempla la modificación de la arquitectura de la aplicación cliente SOS V2.0 para extenderla y permitir emplear servicios de alertas y consultas periódicas automatizadas.

- Por otro lado, y dada la relación con el personal de desarrollo de la organización alemana 52^oNorth, se contempla la inclusión de la herramienta cliente SOS V2.0 en sus servidores para ser distribuida como una herramienta más de la suite.
- Se prevé que sea necesario realizar un estudio de optimización del código desarrollado para la herramienta cliente SOS V2.0 con el objetivo de aumentar la velocidad de procesamiento y disminuir los problemas de sobrecarga de la memoria física del navegador durante la realización de peticiones con un volumen de datos de observación excesivamente elevado.
- Se contempla además la inclusión de herramientas innovadoras que puedan ser especialmente útiles en ámbitos determinados empleando otros métodos de visualización y representación de los datos de observaciones; como pueden ser diagramas de rosas, diagramas de tartas, representaciones 3D, etc.

Capítulo 9. Referencias

- [1] Raia - Observatorio Oceanográfico da Marxe Ibérica, «Observatorio Oceanográfico da Marxe Ibérica,» [En línea]. Available: <http://www.marnaraia.org/>. [Último acceso: 08 01 2017].
- [2] CiTIUS, «SOSSI: Integración semántica de datos de observación a través de SOS,» [En línea]. Available: <https://citius.usc.es/investigacion/proxectos/listado/integracion-semantica-datos-observacion-traves-sos>. [Último acceso: 21 04 2017].
- [3] Meteogalicia, «meteogalicia.gal,» Meteogalicia, [En línea]. Available: <http://www.meteogalicia.gal/web/informacion/glosario/pred4.action>. [Último acceso: 05 07 2017].
- [4] datos.gob.es, «Open data y datos ambientales, avances y oportunidades,» 05 06 2017. [En línea]. Available: <http://datos.gob.es/es/noticia/open-data-y-datos-ambientales-avances-y-oportunidades>. [Último acceso: 19 08 2017].
- [5] Desconocido, «<https://es.wikipedia.org/wiki/MeteoGalicia>,» [En línea]. Available: <https://es.wikipedia.org/wiki/MeteoGalicia>. [Último acceso: 12 5 2017].
- [6] Intecmar, «Intecmar - Funciones,» [En línea]. Available: <http://www.intecmar.gal/Intecmar/Funciones.aspx?sm=b>. [Último acceso: 10 12 2016].
- [7] Anónimo, «wikipedia - Vigilancia Tecnológica,» [En línea]. Available: https://es.wikipedia.org/wiki/Vigilancia_tecnol%C3%B3gica. [Último acceso: 12 08 2017].
- [8] M. A. Regueiro, J. R. R. Viqueira, C. Stasch y J. A. Taboada, «Semantic mediation of observation datasets through Sensor Observation Services,» *Future Generation Computer Systems - The International Journal of Grid Computing and eScience* ,

- vol. 67, pp. 47-56, 2017.
- [9] M. A. Regueiro, S. Villarroya, G. Sanmartín y J. R. Ríos Viqueira, «Integración de observaciones medioambientales: Solución inicial y retos futuros,» de *XVII Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2012)*, Almería, 2012.
- [10] Official Journal of the European Union, «Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007,» Official Journal of the European Union, L 108: 50, 25 April 2007. [En línea]. Available: <http://eurlex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2007:108:0001:0014:EN:PDF> . [Último acceso: 14 09 2016].
- [11] Boletín Oficial del Estado, «Ley 14/2010, de 5 de julio, sobre las infraestructuras y los servicios de información geográfica,» 06 07 2010. [En línea]. Available: <http://www.boe.es/boe/dias/2010/07/06/pdfs/BOE-A-2010-10707.pdf>. [Último acceso: 27 11 2016].
- [12] R. Vallejo Bombín y S. Mas Mayoral, «Resumen de El proyecto de Directiva europea INSPIRE,» Fundación Dialnet - Universidad de La Rioja, [En línea]. Available: <https://dialnet.unirioja.es/servlet/articulo?codigo=499660&info=resumen>. [Último acceso: 18 04 2017].
- [13] MASTER ISSI - UNED, «Reglamento para el Trabajo Fin de Máster del Máster Universitario de Investigación en Ingeniería de Software y Sistemas Informáticos,» [En línea]. Available: http://www.issi.uned.es/Master_ISSI/WebMISSI/ReglamentoTFdMISSI17.htm. [Último acceso: 07 09 2017].
- [14] A. Anguita Sánchez, *Modelo de mediación semántica para la integración de fuentes de datos heterogéneas*, Madrid: Universidad Politécnica de MADRID - Facultad de Informática, 2012.
- [15] Wikipedia, «Programación por capas,» 05 05 2017. [En línea]. Available: https://es.wikipedia.org/wiki/Programaci%C3%B3n_por_capas. [Último acceso: 06 07 2017].

-
- [16] M. Iniesto, A. Núñez, J. C. González, F. J. Ariza, M. A. Ureña, A. F. Rodríguez, P. Abad, J. R. Rodríguez, M. F. Álvarez, C. Pérez, A. Bastarrika, Á. Rodríguez, L. Torre, M. Á. Manso, D. Rivas, G. Píriz, E. Coll y J. C. Martínez, *Introducción a las Infraestructuras de Datos Espaciales*, Centro Nacional de Información Geográfica (CNIG), Dirección General del Instituto Geográfico Nacional (IGN), 2014.
- [17] D. D. Nebert, *Global Spatial Data Infrastructure Association (GSDI), The SDI Cookbook*, Global Spatial Data Infrastructure , 2001.
- [18] Departamento de la Guajira - Colombia, «MEJORA DE LOS SISTEMAS DE CARTOGRAFÍA DEL TERRITORIO COLOMBIANO,» Rioacha, 2007.
- [19] Anónimo, «Wikipedia,» [En línea]. Available: https://es.wikipedia.org/wiki/Infraestructura_de_Datos_Espaciales. [Último acceso: 09 07 2017].
- [20] Ministerio de agricultura y pesca, alimentacion y medio ambiente, «mapama.es,» [En línea]. Available: <http://www.mapama.gob.es/es/cartografia-y-sig/ide/presentacion/que-es-ide.aspx>. [Último acceso: 04 08 2017].
- [21] Universidad de Barcelona, «Scripta Nova - REVISTA ELECTRÓNICA DE GEOGRAFÍA Y CIENCIAS SOCIALES - INFRAESTRUCTURA DE DATOS ESPACIALES (IDE).,» [En línea]. Available: <http://www.ub.edu/geocrit/sn/sn-170-61.htm>. [Último acceso: 17 01 2017].
- [22] X. Méndez, J. Touriño, J. Parapar, M. Hermida, V. García, M. Regueiro, J. R. Ríos Viqueira y F. Landeira, «Building a meteorological SDI for the region of Galicia (Spain),» de *INSPIRE conference 2011*, 2011.
- [23] Inspire, «Inspire Knowledge Base,» [En línea]. Available: <https://inspire.ec.europa.eu/>. [Último acceso: 14 03 2016].
- [24] Libertic, «Introducing “Inspire for dummies”,» 22 01 2010. [En línea]. Available: <https://edemocratie.wordpress.com/2010/01/22/introducing-inspire-for-dummies/>. [Último acceso: 02 08 2017].
- [25] Gobierno de España, «Presentación resumen de la Directiva Inspire,» [En línea].
-

- Available: <http://datos.gob.es/es/noticia/presentacion-resumen-de-la-directiva-inspire>. [Último acceso: 21 06 2017].
- [26] J. Muñoz Negrete y J. Capdevila Subirana, «Trayectoria de la implementación de la Directiva INSPIRE en España,» de *Jornadas Ibéricas de Infraestructuras de Datos Espaciales - JIIDE*, Madrid, 2012.
- [27] P. Baumann, «How to INSPIRE Citizens: Towards Open Access to High-Volume Public Sector Geodata,» de *Conference: Electronic Government and the Information Systems Perspective - Second International Conference, EGOVIS 2011*, Toulouse, France, 2011.
- [28] Malta Government, «Malta Spatial data Infrastructure,» [En línea]. Available: <https://msdi.data.gov.mt/>. [Último acceso: 06 04 2017].
- [29] OGC, «OGC Members,» [En línea]. Available: <http://www.opengeospatial.org/ogc/members>. [Último acceso: 04 10 2016].
- [30] Wikipedia Anónimo, [En línea]. Available: https://es.wikipedia.org/wiki/Open_Geospatial_Consortium. [Último acceso: 22 08 2017].
- [31] The Open Data Foundation (ODaF), «The Open Data Foundation (ODaF),» [En línea]. Available: <http://www.odaf.org/>. [Último acceso: 27 08 2017].
- [32] Anónimo Wikipedia, «Datos abiertos,» [En línea]. Available: https://es.wikipedia.org/wiki/Datos_abiertos. [Último acceso: 24 05 2017].
- [33] Gobierno Abierto de Navarra, «Open Data y Reutilización de Información del Sector Público,» [En línea]. Available: <http://www.gobiernoabierto.navarra.es/es/open-data/que-es-open-data/open-data-y-risp>. [Último acceso: 15 06 2018].
- [34] datos.gob.es, «Conceptos básicos, beneficios del Open Data y barreras para su aplicación,» 03 23 2017. [En línea]. Available: <http://datos.gob.es/es/documentacion/conceptos-basicos-beneficios-del-open-data-y-barreras-para-su-aplicacion>. [Último acceso: 2017 06 15].
-

-
- [35] C. Gil Navarro, A. Pequeño Pulleiro y A. J. del Valle Silva, *Visualización de datos de comunidades colaborativas*, Madrid: Universidad Complutense de Madrid - Facultad de Informática, 2016.
- [36] World Wide Web Consortium, «World Wide Web Consortium. Linked Data,» [En línea]. Available: <https://www.w3.org/standards/semanticweb/data>. [Último acceso: 03 09 2017].
- [37] Open Geospatial Consortium (OGC), «Sensor Web Enablement (SWE),» [En línea]. Available: <http://www.opengeospatial.org/ogc/markets-technologies/swe>. [Último acceso: 04 06 2017].
- [38] M. Botts, G. Percivall, C. Reed y J. Davidson, «OGC® Sensor Web Enablement: Overview and High Level Architecture,» *GeoSensor Networks*, vol. 4540, pp. 175-190.
- [39] P. Fraga-Lamas, T. Fernández-Caramés, A. Carro-Lagoa, C. Escudero y M. González-López, «Estándares para interoperabilidad de redes de sensores: IEEE 1451 y Sensor Web Enablement (SWE),» de *CIUDAD2020: HACIA UN NUEVO MODELO DE CIUDAD INTELIGENTE SOSTENIBLE*, 2014.
- [40] A. Bröring, C. Stasch y J. Echterhof, «Open Geospatial Consortium (OGC),» [En línea]. Available: <http://www.opengeospatial.org/standards/sos> . [Último acceso: 22 01 2016].
- [41] OGC, «Sensor Observation Service,» [En línea]. Available: <http://www.opengeospatial.org/standards/sos>. [Último acceso: 11 06 2017].
- [42] OGC, «OpenGIS Sensor Observation Service,» [En línea]. Available: http://portal.opengeospatial.org/files/?artifact_id=26667. [Último acceso: 21 06 2017].
- [43] OGC, «OGC® Sensor Observation Service Interface Standard,» [En línea]. Available: https://portal.opengeospatial.org/files/?artifact_id=47599. [Último acceso: 12 04 2017].
- [44] 52°North, «52°North SOS 4.X Documentation,» [En línea]. Available:
-

- <https://wiki.52north.org/SensorWeb/SensorObservationServiceIVDocumentation>.
[Último acceso: 02 04 2016].
- [45] Open Geospatial Consortium (OGC), «OGC Sensor Observation Service (SOS) Standard Version 2.0 Adopted,» [En línea]. Available: <http://www.opengeospatial.org/node/1601>. [Último acceso: 24 03 2017].
- [46] OGC Network, «How to model your observation data in SOS 2.0?,» [En línea]. Available: http://www.ogcnetwork.net/sos_2_0/tutorial/om. [Último acceso: 14 05 2017].
- [47] OGC Network, [En línea]. Available: http://www.ogcnetwork.net/sos_2_0/tutorial/sensorML. [Último acceso: 06 04 2017].
- [48] O. Fonts, 2015. [En línea]. Available: <http://sensor-widgets.readthedocs.io/es/latest/sos.html>. [Último acceso: 06 03 2017].
- [49] 52°North, «52°North History,» [En línea]. Available: <http://52north.org/about/52north/history>. [Último acceso: 19 04 2017].
- [50] 50°North, [En línea]. Available: <http://52north.org/about/52north>. [Último acceso: 24 05 2017].
- [51] OSGeo-es, «Panorama SIG libre - Servidores,» 2014. [En línea]. Available: <http://panorama-sig-libre.readthedocs.io/es/latest/servidores/>. [Último acceso: 27 07 2017].
- [52] A. Bröring, C. Stasch y E. Johannes, «SOS 2.0 Tutorial,» 18 01 2012. [En línea]. Available: http://www.ogcnetwork.net/SOS_2_0/tutorial. [Último acceso: 19 10 2016].
- [53] 52° North, «52°North downloads,» 14 12 2015. [En línea]. Available: <http://52north.org/downloads/category/3-sos>. [Último acceso: 17 09 2016].
- [54] L. E. Castillo León, «Gestión del Riesgo,» 27 05 2008. [En línea]. Available: <https://es.slideshare.net/lecastillox/gestion-del-riesgo>. [Último acceso: 19 02 2017].
-

-
- [55] R. Pressman, *Ingeniería del Software: Un enfoque práctico*, McGraw-Hill, 2005.
- [56] Wikiversity, «Gestión de riesgos de proyectos software,» 30 10 2014. [En línea]. Available: https://es.wikiversity.org/wiki/Gesti%C3%B3n_de_riesgos_de_proyectos_softwar_e#Riesgos_t.C3.A9cnicos. [Último acceso: 24 01 2017].
- [57] hrubeneiva21, «Planificación de proyectos de software,» 7 06 2012. [En línea]. Available: <https://es.slideshare.net/hrubeneiva21/planificacin-de-proyectos-de-software>. [Último acceso: 17 01 2017].
- [58] I. Sanchez Vera, «Planificacion De Proyectos De Software,» 01 08 2009. [En línea]. Available: <https://es.slideshare.net/ivansanchez1988/planificacion-de-proyectos-de-software>. [Último acceso: 03 02 2017].
- [59] I. Sommerville, *Ingeniería del software*, Pearson Educación, 2005.
- [60] Ingeniería Del Software UAH, «Arquitectura Orientada a Servicios (SOA),» 22 03 2015. [En línea]. Available: <https://ingenieriadelsoftwareuah2015.wordpress.com/2015/03/22/arquitectura-orientada-a-servicios-soa/>. [Último acceso: 06 21 2017].
- [61] UML, «Unified Modeling Language,» [En línea]. Available: <http://www.uml.org/>. [Último acceso: 09 02 2017].
- [62] J. Siegel, «INTRODUCTION TO OMG'S UNIFIED MODELING LANGUAGE™ (UML®),» 07 2005. [En línea]. Available: <http://www.uml.org/what-is-uml.htm>. [Último acceso: 24 01 2017].
- [63] «Lenguaje unificado de modelado,» 12 09 2017. [En línea]. Available: https://es.wikipedia.org/wiki/Lenguaje_unificado_de_modelado. [Último acceso: 15 09 2017].
- [64] G. Méndez, «Especificacion de Requisitos según el estándar de IEEE 830-1998,» Dpto. de Ingeniería del Software e Inteligencia Artificial - Facultad de Informática - Universidad Complutense de Madrid, Madrid, 2008.
-

- [65] M. G. Piattini Velthuis, J. A. Calvo-Manzano, J. Cervera Bravo y L. Fernández Sanz, *Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión*, RA-MA, S.A. Editorial y Publicaciones, 2007.
- [66] R. Monferrer Agut, «Especificación de Requisitos Software según el estándar de IEEE 830,» Departament d'Informàtica - Universitat Jaume I, Castellón, 2000.
- [67] G. Vidal, «Ingeniería de Requerimientos,» 2007. [En línea]. Available: http://users.dsic.upv.es/~jsilva/fin/idr/IDR_practica1.pdf. [Último acceso: 18 02 2017].
- [68] UNIDAD EDUCATIVA "JUAN FRANCISCO MONTALVO", «Análisis y diseño detallado de aplicaciones informáticas de gestión,» 2009. [En línea]. Available: <https://sites.google.com/site/adai6jfm/home/introduccion-al-analisis-de-requisitos-ar>. [Último acceso: 17 01 2017].
- [69] IEEE, *IEEE Recommender Practice for Software Requirements Specifications (IEEE Std 930-1998)*, New York: IEEE Standards Board, 1998.
- [70] M. Fowler y K. Scott, *UML gota a gota*, Pearson Educación, 1999.
- [71] L. Debrauwer y F. van der Heyde, *UML 2: iniciación, ejemplos y ejercicios corregidos*, Ediciones ENI, 2005.
- [72] T. Soddemann, *Web Services and Service Oriented Architectures*, Delaman Workshop, 2004.
- [73] T. Erl, *SOA: Principles of Service Design*, Prentice Hall, 2007.
- [74] T. Erl, B. Carlyle, C. Pautasso, R. Balasubramanian, H. Wilhelmsen y D. Booth, *SOA with REST: Principles, Patterns & Constraints for Building Enterprise Solutions with REST*, Prentice Hall, 2012.
- [75] T. Rischbeck y T. Erl, *SOA Design Patterns*, Prentice Hall PTR, 2009.
- [76] D. Adams y R. McNamara, *Service-Oriented Architecture and Best Practices*,

-
- TIBCO Software, Inc., 2006.
- [77] T. Erl, *Service-Oriented Architecture (SOA): Concepts, Technology, and Design*, Prentice Hall, 2005.
- [78] Microsoft, *Service Oriented Architecture: Right on Track*, Microsoft, 2008.
- [79] J. Hurwitz, R. Bloor, M. Kaufman y F. Halper, *Service Oriented Architecture for dummies*, Wiley Publishing, Inc., 2009.
- [80] M. Rosen, *Service Oriented Architecture Based Integration*, AZORA Technologies, Inc., 2005.
- [81] G. C. Lin, K. T. Desmond, N. T. Htoon y N. V. Thuat, «Service Oriented Architecture,» de *A Fresh Graduate's Guide to Software Development Tools and Technologies*, National University of Singapore, 2012, p. Chapter 10.
- [82] BearingPoint, *Positions on Service Oriented Architecture*, BearingPoint Management & Technology Consultants, 2012.
- [83] Microsoft Corporation, «La Arquitectura Orientada a Servicios (SOA) de Microsoft,» *Whitepaper: La arquitectura SOA de Microsoft aplicada al mundo real*, pp. 1-21, 2006.
- [84] Intecmar, «Instituto tecnológico par el control del medio marino de Galicia,» [En línea]. Available: <http://www.intecmar.gal/intecmar/>. [Último acceso: 15 03 2017].
- [85] S. E. Institute, *Getting Started with ServiceOriented Architecture (SOA) Terminology*, Carnegie Mellon, 2010.
- [86] D. Slama, D. Krafzig y K. Banke, *Enterprise SOA: Service-oriented Architecture Best Practices*, Prentice Hall PTR, 2004.
- [87] M. Corral y R. Poncela, «El camino hacia la arquitectura orientada a servicios,» *Tecnimap Sevilla*, nº 398, pp. 1-8, 2006.
- [88] A. Delgado, L. Gonzalez y F. Piedrabuena, «Desarrollo de aplicaciones con enfoque SOA (Service Oriented Architecture),» de *JIIISIC'06 - V Jornadas Iberoamericanas*
-

de Ingeniería de Software e Ingeniería del Conocimiento, México, 2006.

- [89] Meteogalicia, «MeteoSIX,» [En línea]. Available: <http://www.meteogalicia.gal/web/proyectos/meteosix.action>. [Último acceso: 05 02 2017].
- [90] I. M. Burguan Valverde, «Gestión del Cambio Software,» [En línea]. Available: <https://es.slideshare.net/imburguan/gestin-del-cambio-del-software>. [Último acceso: 02 06 2017].
- [91] S. Jirka, «SOS Clients,» de *INSPIRE Conference 2014 - The Sensor Observation Service as INSPIRE Download Service for Sharing Observation Data*, Aalborg (Dinamarca), 2014.
- [92] J. R. Parker, *Algorithms for image processing and computer vision*, Wiley, 1997.
- [93] W. K. Pratt, *Digital image processing*, Wiley-Interscience, 2001.
- [94] R. Szeliski, *Computer Vision: Algorithms and Applications*, Springer, 2010.
- [95] M. S. Nixon y A. S. Aguado, *Feature Extraction & Image Processing for Computer Vision*, Academic Press, 2012.
- [96] C. Solomon y T. Breckon, *Fundamentals of Digital Image Processing*, John Wiley & Sons, 2011.
- [97] F. Y. Shih, *Image Processing and Pattern Recognition*, John Wiley & Sons, 2010.
- [98] P. Malay K., *Digital Image Processing and Pattern Recognition*, PHI Learning Pvt. Ltd., 2011.
- [99] R. C. González, *Digital Image Processing*, Pearson Education, 2009.
- [100] R. C. Gonzalez, R. Eugene Woods y S. L. Eddins, *Digital Image Processing Using MATLAB*, Pearson Education, 2004.
- [101] T. Acharya y A. K. Ray, *Image Processing: Principles and Applications*, John Wiley

-
- & Sons, 2005.
- [102] B. Jähne, *Digital Image Processing*, Springer Science & Business Media, 2005.
- [103] J. N. Wilson y G. X. Ritter, *Handbook of Computer Vision Algorithms in Image Algebra*, CRC Press, 2000.
- [104] W. K. Pratt, *Digital Image Processing: PIKS Scientific Inside*, Wiley, 2007.
- [105] R. Pressman, *Ingeniería Del Software: Un Enfoque Práctico*, McGraw-Hill, 2005.
- [106] I. Sommerville, *Ingeniería del software*, Pearson Educación, 2005.
- [107] C. Rother, V. Kolmogorov y A. Blake, «“GrabCut” — Interactive Foreground Extraction using Iterated Graph Cuts,» *ACM Transactions on Graphics*, vol. 23, nº 3, pp. 309-314, 2004.
- [108] E. Ramírez J., D. Martínez R. y R. Carmona S., «Segmentación de imágenes a color basada en el algoritmo de GrabCut,» *Tekhne Revista de la Facultad de Ingeniería UCAB*, nº 15, pp. 21-37, 2012.
- [109] M. G. Piattini Velthuis, J. A. Calvo-Manzano, J. Cervera Bravo y L. Fernández Sanz, *Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión*, RA-MA S.A. Editorial y Publicaciones, 2007.
- [110] J. Nielsen y H. Loranger, *Usabilidad*, Volumen 268 de *Diseño y creatividad*, Anaya Multimedia, 2007.
- [111] I. 830, *IEEE Recommended Practice for Software Requeriments Specifications (IEEE Std 930-1998)*, New York: IEEE Standards Board, 1998.
-

Apéndice A. Código de la herramienta

Todo el código fuente del cliente pesado SOS 2.0 desarrollado en este trabajo final de máster puede ser descargado desde el repositorio GitLab del CiTIUS. Puesto que es el código de la aplicación ha estado gestionado mediante este sistema de versiones, se podrá descargar cualquiera de las versiones del código almacenadas.

La url de enlace es:

https://gitlab.CiTIUS.usc.es/cograde/SOS20_Client

Apéndice B. Siglas, abreviaturas y acrónimos

- **MeteoGalicia:** Agencia meteorológica de la comunidad autónoma de Galicia
- **INTECMAR:** Instituto Tecnológico para el Control del Medio Marino de Galicia
- **AEMET:** Agencia Estatal de Meteorología
- **GHCN:** Global Historical Climatology Network
- **NOAA:** National Oceanic and Atmospheric Administration
- **COGRADE:** Grupo de investigación de Computación Gráfica e Ingeniería de Datos, del inglés Computer Graphics and Data Engineering
- **USC:** Universidad de Santiago de Compostela
- **OGC:** Open Geospatial Consortium
- **SWE:** Sensor Web Enablement
- **SOS:** Sensor Observation Service
- **AOS:** Arquitectura Orientada a Servicios
- **INSPIRE:** Infrastructure for Spatial Information in Europe
- **GIS:** Geographic Information System
- **OGF:** Open GIS Foundation
- **GRASS:** Geographic Resources Analysis Support System
- **API:** Application Program Interface
- **XML:** Extensible Markup Language
- **UML:** Unified Modeling Language
- **TML:** Transducer Modeling Language

- **WNS: Web Notification Service**
- **SAS: Sensor Alert Service**
- **SPS: Sensor Planning Service**
- **O&M: Observations and Measurements**
- **SDI: Spatial Data Infrastructure**
- **IDE: Infraestructura de Datos Espaciales**
- **FOI: Feature Of Interest**
- **UOM: Unit Of Measure**
- **IFGI: Institute for Geoinformatics**
- **52°N: 52° North**
- **JRE: Java SE Runtime Environment**
- **TSQL: Transact-SQL, Transact Structured Query Language**
- **POM: Project Object Model**
- **HTML: HyperText Markup Language**
- **DOM: Document Object Model**
- **AJAX: Asynchronous JavaScript And XML**
- **CSS: Cascading Style Sheets**
- **EDT: Estructura de descomposición del trabajo**
- **WBS: Work Breakdown Structure**
- **SCM: Software Configuration Management**
- **ERS: Especificación de Requerimientos de Software**
- **DFD: Diagrama de Flujo de Datos**
- **CSV: Comma-Separated Values**
- **TSV: Tab-Separated Values**
- **XLS: Microsoft Excel Spreadsheet**
- **FAQs: Frequently Asked Questions**

Apéndice C. Manual de instalación

En este manual se explicaran los pasos a seguir para poner en funcionamiento la herramienta desarrollada en el presente TFM. Las instrucciones presentadas han sido diseñadas para ser utilizadas en máquinas con sistemas operativos tanto Windows como Linux.

Como paso previo a la instalación hay que preparar la máquina con las herramientas necesarias para el despliegue del cliente SOS V2.0 Por ello será necesario descargar Apache Tomcat 8.0.x desde la página oficial de Apache:

<https://tomcat.apache.org/download-80.cgi>

Según el sistema operativo presente en la máquina de instalación habrá que realizar diferentes procedimientos para la instalación del contenedor web Tomcat. En la página oficial de Apache se detalla el proceso de instalación del mismo atendiendo al sistema operativo destino.

El cliente pesado SOS V2.0 se distribuye a través de la URL especificada en el Apéndice A. Por lo tanto habrá que descargar de dicha URL el archivo clienteSOS.war, localizado en la carpeta raíz de esa web.

Procederemos a la instalación de Apache Tomcat y una vez instalado accederemos a su página principal introduciendo las credenciales de administrador en el cuadro de diálogo que aparece en su entrada (ver Fig. 50). En ese punto nos encontraremos en la página principal de Tomcat.

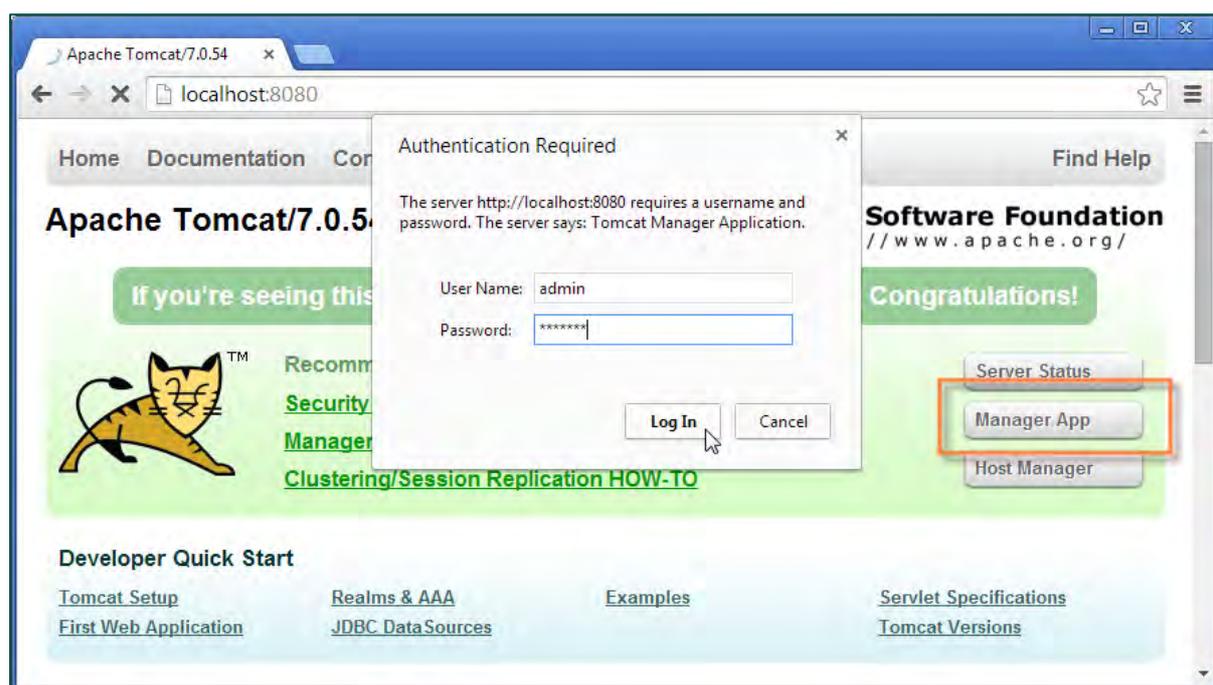


Fig. 50 Página de inicio de Apache Tomcat

Para acceder al gestor de aplicaciones desde su página principal, desde donde procederemos a realizar el despliegue del archivo war del cliente SOS. Para ello debemos pulsar sobre el botón “Manage App”, localizado en la parte lateral derecha de la página (ver Fig. 50).

Una vez hayamos accedido al gestor de aplicaciones tendremos que dirigirnos a la sección “WAR file to deploy”, situada en la sección inferior de la web (ver Fig. 51). Tendremos que pulsar el botón “Choose File”. Seguidamente se mostrará un cuadro de diálogo del navegador que tendremos que utilizar para localizar el archivo .war correspondiente al cliente SOS V2.0 descargado previamente.

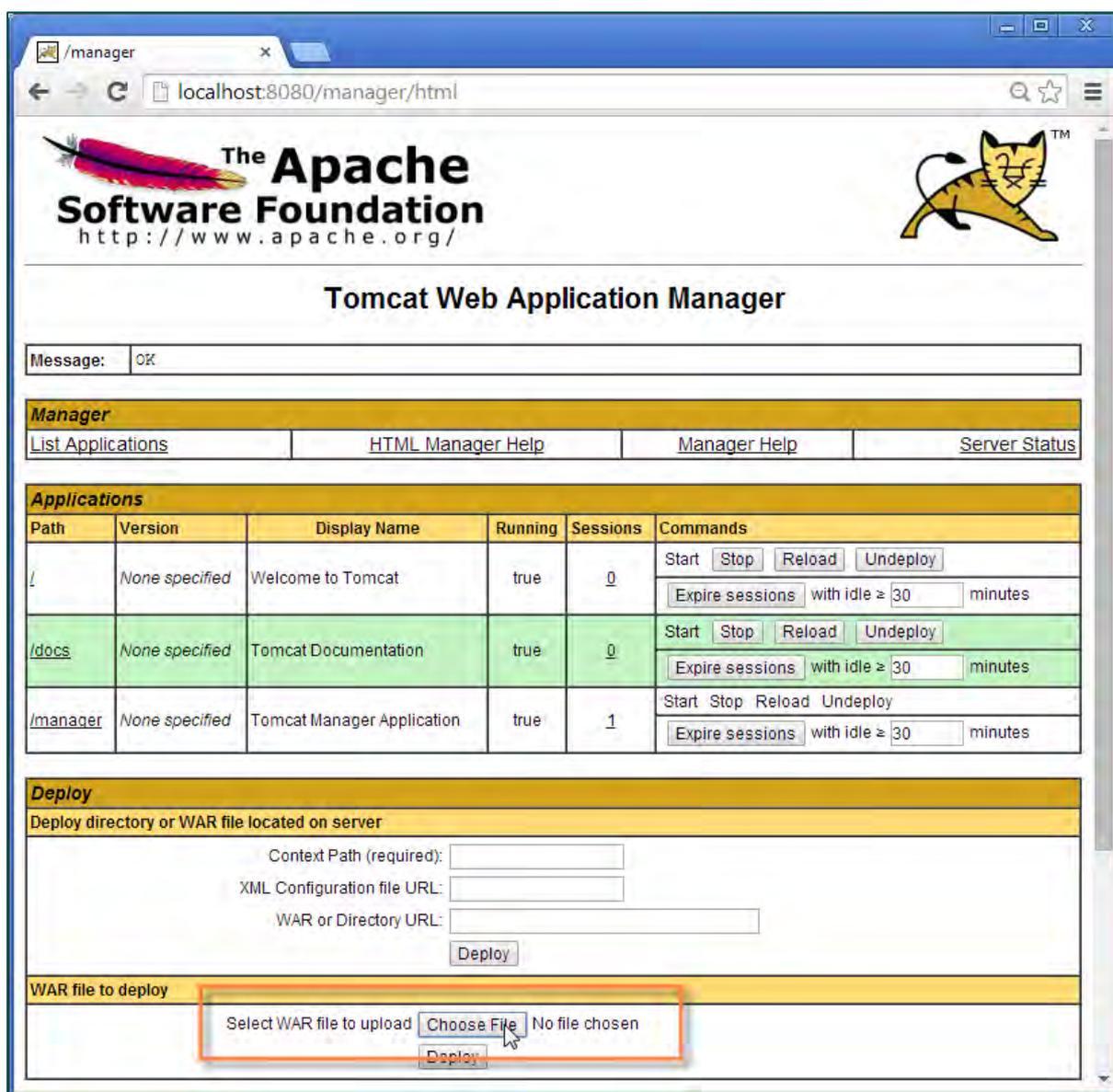


Fig. 51 Gestor de aplicaciones de Tomcat

Una vez se haya seleccionado el archivo WAR habrá que hacer clic en el botón “Deploy” para comenzar con el despliegue de la aplicación en Tomcat. Este proceso de despliegue se encarga de copiar el archivo WAR seleccionado en el directorio webapps de Tomcat, para descomprimir dicho archivo en su correspondiente subdirectorio.

Una vez finalice el proceso de despliegue se accederá al cliente SOS V2.0 a través de la url que indica el gestor de aplicaciones de Tomcat.

En este punto se recomienda consultar el manual de usuario del Apéndice D.

Apéndice D. Manual de Usuario

Para comenzar a utilizar el cliente SOS es importante cumplir los siguientes requisitos:

- Disponer de una URL de servicio SOS V2.0
- Tener la máquina de ejecución del cliente SOS completamente configurada. Para ello tendremos que:
 - Realizar el proceso de instalación detallado en el Apéndice C
 - Comprobar que el servicio SOS V2.0 esté accesible desde la máquina de ejecución
 - Tener instalado un navegador web, preferiblemente Firefox, que es donde se han realizado la gran parte de las pruebas de la herramienta.

Una vez se cumplan los requisitos habrá que acceder a la URL del cliente web desplegado en el Tomcat. Una vez ingresamos la primera vez al cliente SOS V2.0 aparecerá la siguiente imagen:

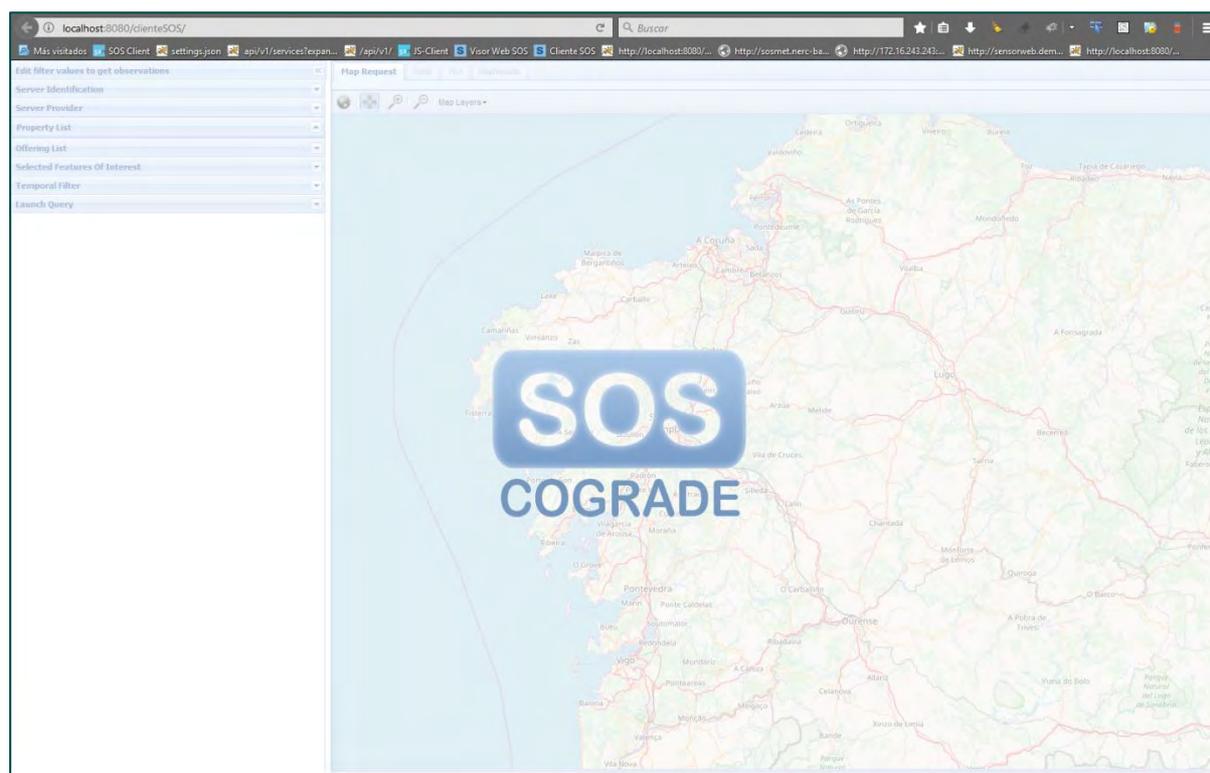


Fig. 52 Ventana inicial del cliente SOS V2.0

En ella aparecerá un logotipo parpadeante y una interfaz web que estará durante unos segundos inaccesible. Habrá que esperar puesto que el cliente SOS V2.0 ha realizado la petición de GetCapabilities al servidor SOS V2.0 de forma automática.

Una vez que se ha recibido la respuesta del servidor SOS V2.0, el cliente web tendrá el siguiente aspecto:

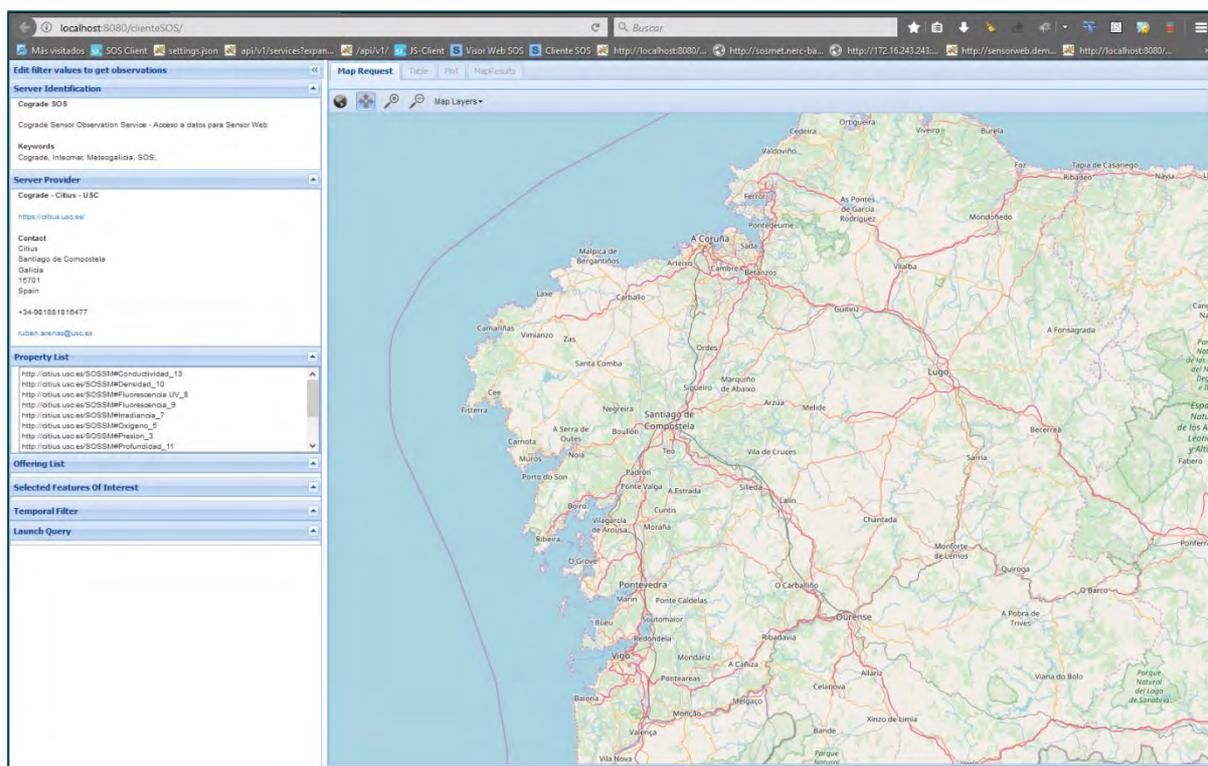


Fig. 53 Estado del cliente SOS V2.0 una vez ha recibido la respuesta de GetCapabilities

En este punto podemos consultar en el panel lateral izquierdo los datos de Identificación del Servidor SOS v2.0, así como los datos de proveedor. Ambos conjuntos de datos se corresponden fielmente a la descripción del estándar OGC SOS V2.0

Situado en la posición inmediatamente a estos dos paneles encontramos el desplegable con leyenda “Property List”. Para construir la consulta de observaciones debemos seleccionar un elemento de dicha lista, que se corresponderá con la magnitud que se desea consultar en el servidor SOS V2.0

Tras seleccionar una de dichas magnitudes, el cliente SOS v2.0 desplegará automáticamente el listado de Offerings. Para continuar el usuario deberá seleccionar al menos una de las ofertas del listado (ver Fig. 54). Hay que indicar que para seleccionar más de una oferta habrá que mantener pulsado el botón Control del teclado y pulsar sobre todas las ofertas que se deseen.

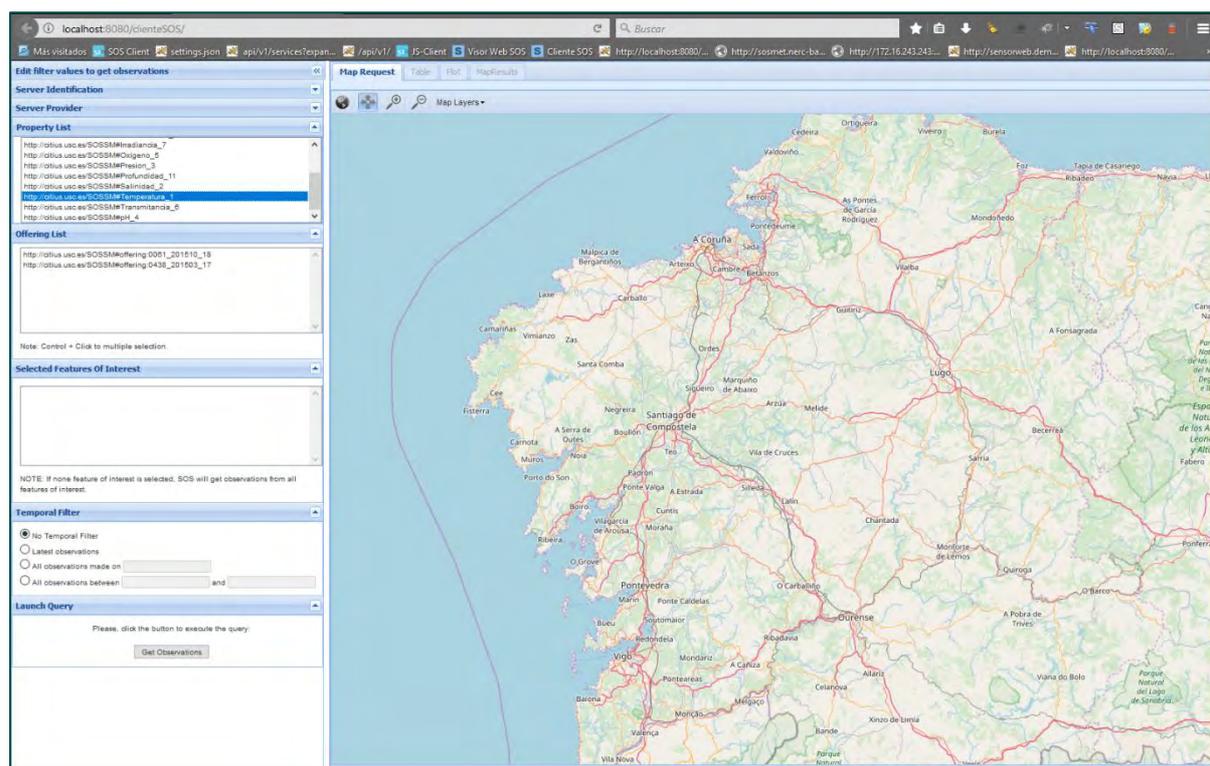


Fig. 54 Estado del cliente SOS V2.0 esperando la selección de ofertas.

Tras seleccionar las ofertas que se deseen el cliente SOS V2.0 realiza una petición de GetFeatureOfInterest al servidor SOS V2.0. Esto es así ya que debemos completar el filtro de la consulta con las FOIs que se deseen. En este estado la interfaz web quedará bloqueada hasta que se reciba la respuesta del servidor SOS V2.0 (ver Fig. 55)

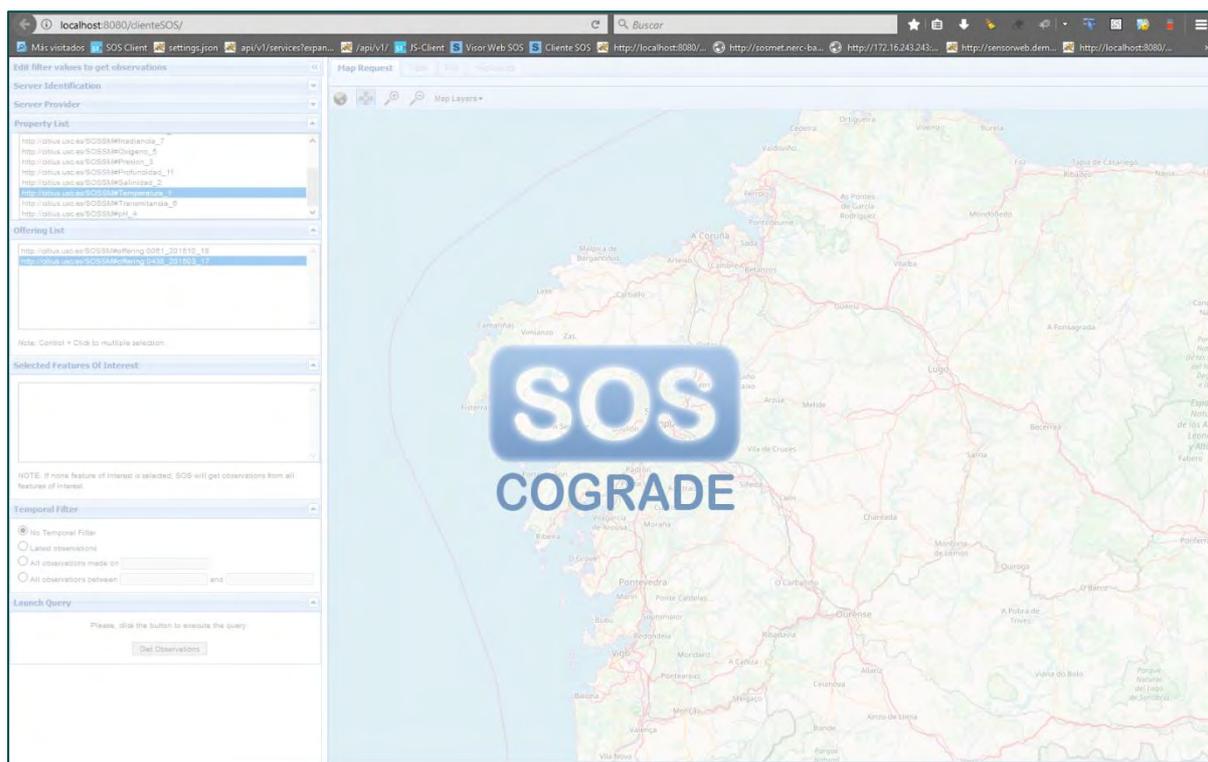


Fig. 55 Cliente SOS esperando la respuesta de la consulta GetFeatureOfInterest

Una vez que se recibe la respuesta de la operación GetFeatureOfInterest, se muestran geoposicionadas en el mapa todas y cada una de las FOIs pertenecientes a las Offerings seleccionadas (ver Fig. 56).

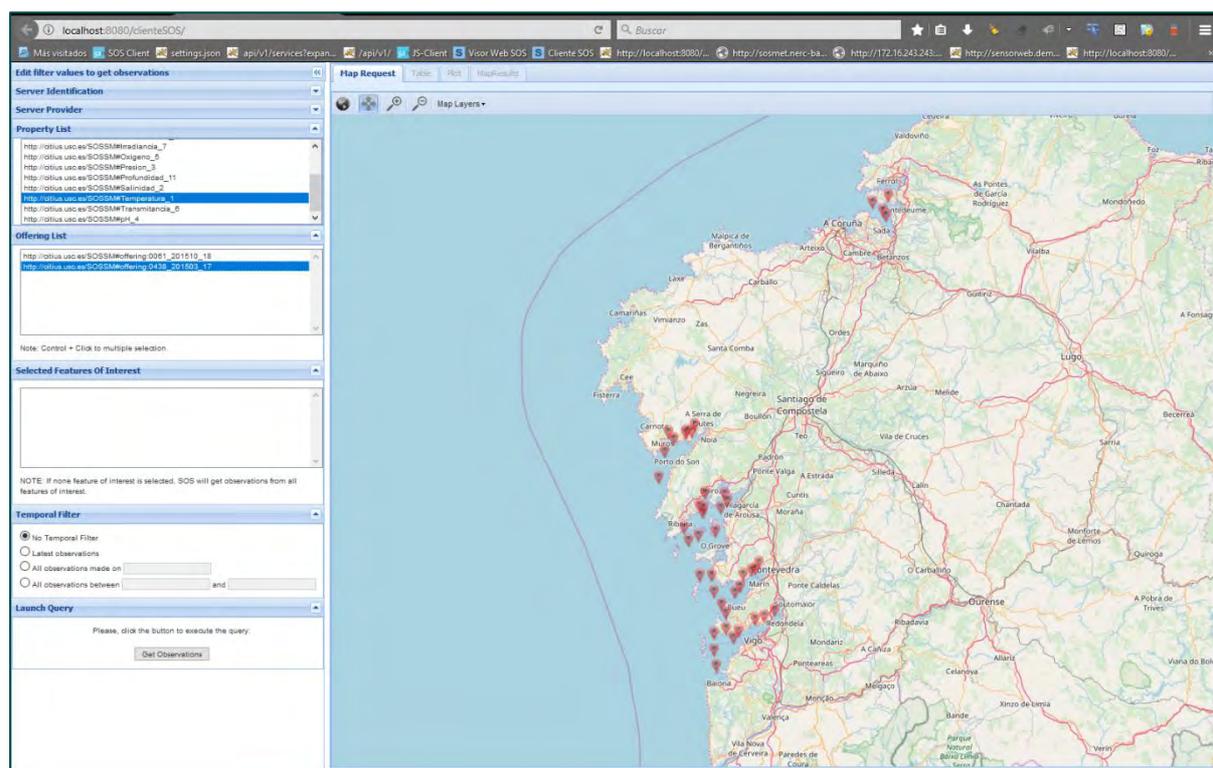


Fig. 56 Cliente SOS con las FOIs asociadas a las Offerings seleccionadas

Para continuar con la construcción de la consulta de observaciones debemos seleccionar las FOIs de las que se desee consultar datos de observación. Para ello se seleccionará con el ratón todas y cada una de las FOIs mostradas en el mapa (ver Fig. 57). Cada vez que se selecciona una FOI se despliega un menú popUp con la información obligatoria que define el estándar OGC SOS V2.0 sobre la FOI. Se muestra por tanto, el identificador de la FOI en ser servicio SOS y sus coordenadas GPS (ver Fig. 58).

Hay que indicar que si no se selecciona ninguna FOI la consulta de observaciones se realizará sin ningún filtro y por lo tanto se devolverán datos de observaciones de todas las FOIs contempladas en las Offerings seleccionadas.

Además hay que indicar, que para borrar la selección de FOIs simplemente habrá que hacer clic en cualquier parte del mapa que no contenga ninguna FOI. De esta manera se limpiará el panel de FOIs seleccionadas.

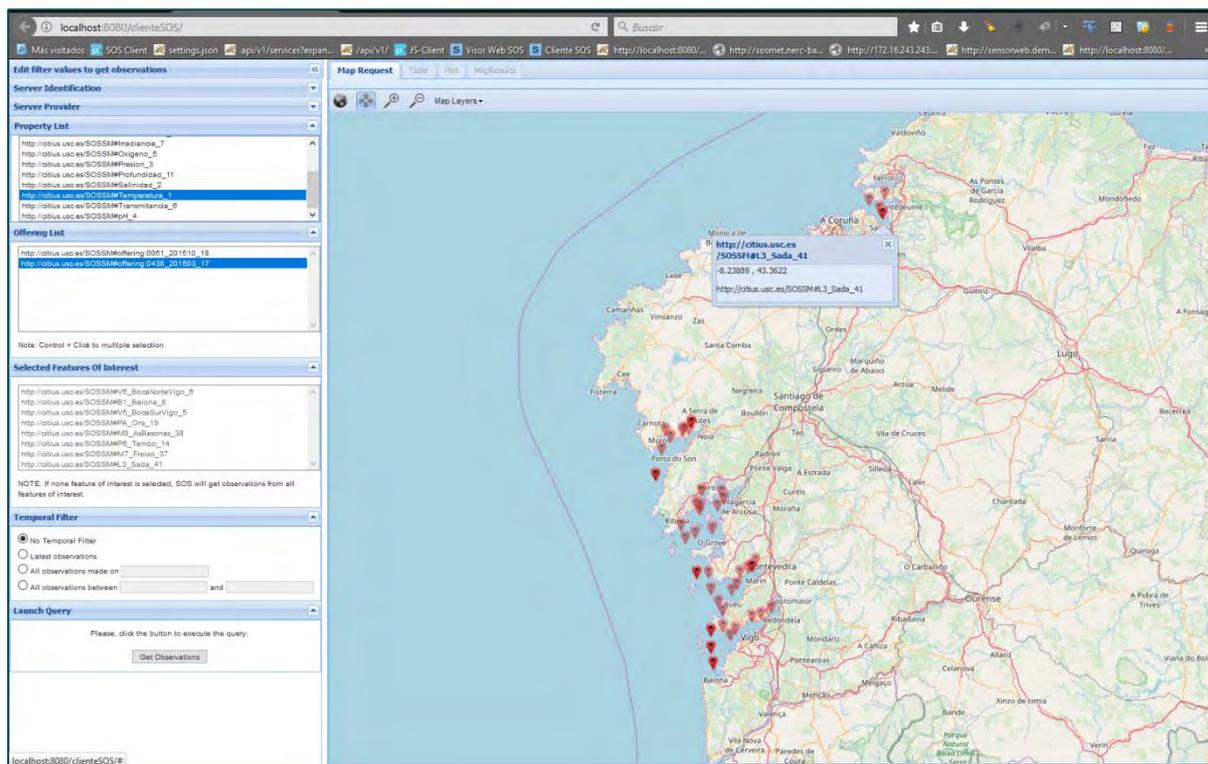


Fig. 57 Cliente SOS con FOIs seleccionadas



Fig. 58 Panel de detalles de FOI

Una vez se tengan las FOIs seleccionadas se activará en el panel lateral izquierdo el desplegable que permitirá incluir los filtros temporales a la consulta de datos de observaciones (ver Fig. 59). Podremos realizar filtros temporales atendiendo:

- a la última observación registrada
- a todas las observaciones realizadas en un momento determinado
- a todas las observaciones realizadas en un intervalo concreto de fechas

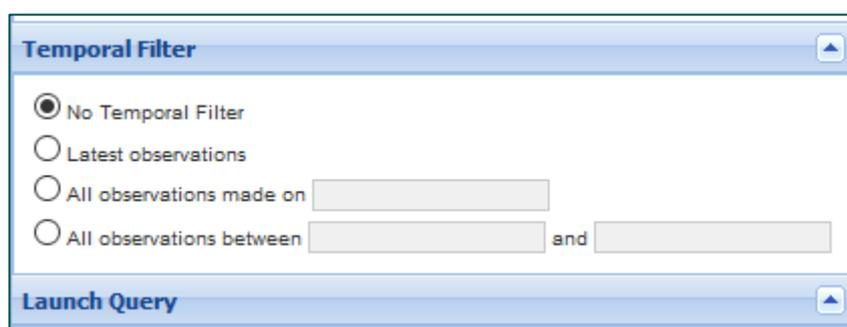


Fig. 59 Panel de filtro temporal activo

Una vez se haya configurado el filtro temporal a gusto del usuario se procederá a relizar la petición de observaciones pulsando en el botón GetObservation, situado en la parte inferior del panel lateral izquierdo.

En este punto el cliente SOS se encarga de generar la consulta de observaciones según los datos que el usuario ha introducido en el formulario, para posteriormente realizar la consulta de GetObservation automáticamente al servidor SOS V2.0. Como ocurre con el resto de operaciones de consulta al servidor SOS V2.0, hasta que no se reciba la respuesta por parte del servidor SOS V2.0 la interfaz web quedará bloqueada, tal y como se ilustra en la Fig. 60.

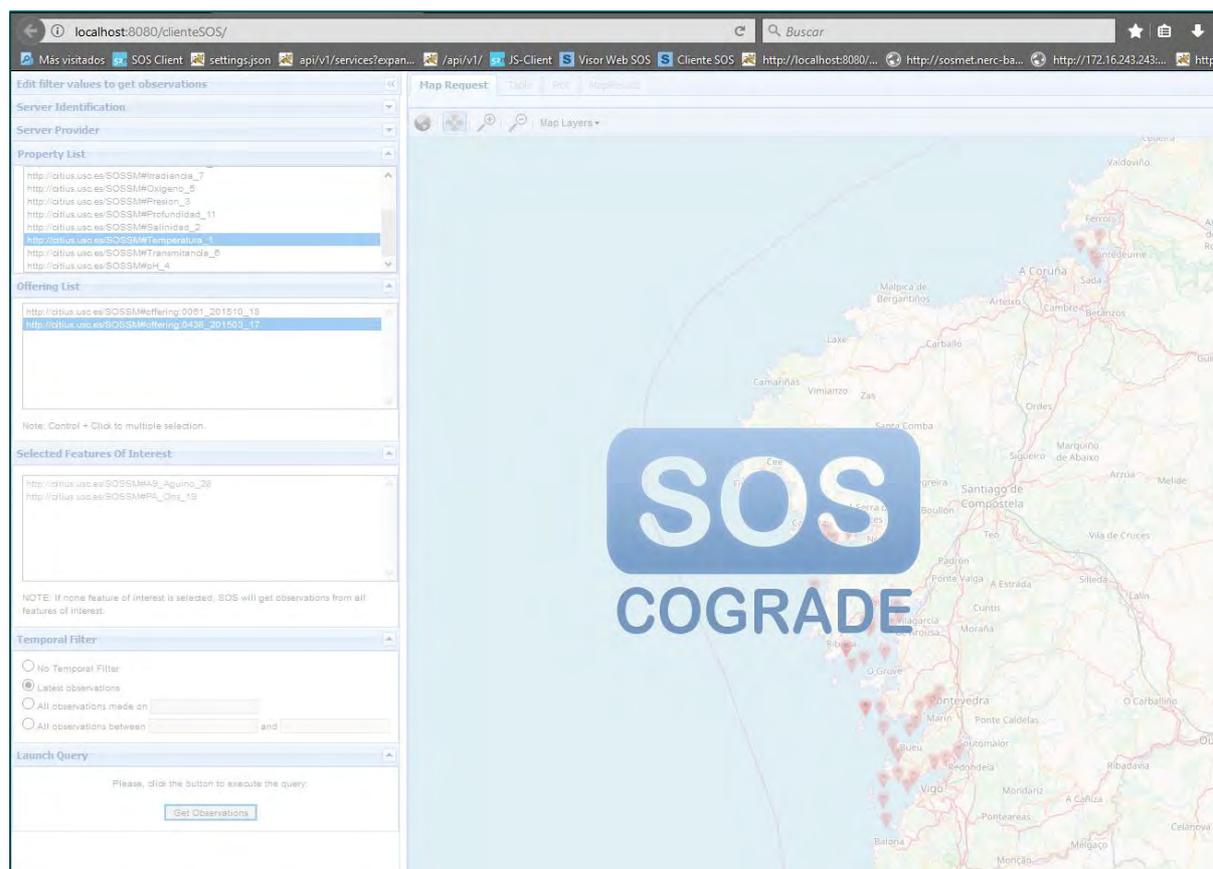


Fig. 60 Cliente SOS V2.0 ejecutando la consulta GetObservation

Una vez se recibe la respuesta del servidor SOS V2.0, ésta se procesa mediante las tareas de parseo y se almacenan sus datos en la caché del navegador web. En este punto la interfaz web se desbloquea y se puede acceder a las herramientas de visualización y manipulación de los datos de observaciones.

Por defecto la herramienta que se muestra es el visualizador de los datos de observaciones en crudo, en formato tabla (ver Fig. 61).

PhenomenomTime	Procedure	profundidad Value	profundidad Unit	FeatureOfInterest	Result	Result Unit
2016-01-12T08:23:00.000+01:00	http://citi.usc.es/SOSSM#0438_201503_17	2.257	m	http://citi.usc.es/SOSSM#PA_Ons_19	13.0075	°C
2016-01-12T08:23:00.000+01:00	http://citi.usc.es/SOSSM#0438_201503_17	2.284	m	http://citi.usc.es/SOSSM#PA_Ons_19	13.1248	°C
2016-01-12T08:23:00.000+01:00	http://citi.usc.es/SOSSM#0438_201503_17	2.294	m	http://citi.usc.es/SOSSM#PA_Ons_19	13.3402	°C
2016-01-12T08:23:00.000+01:00	http://citi.usc.es/SOSSM#0438_201503_17	2.319	m	http://citi.usc.es/SOSSM#PA_Ons_19	13.3841	°C
2016-01-12T08:23:00.000+01:00	http://citi.usc.es/SOSSM#0438_201503_17	2.338	m	http://citi.usc.es/SOSSM#PA_Ons_19	13.3908	°C
2016-01-12T08:23:00.000+01:00	http://citi.usc.es/SOSSM#0438_201503_17	2.356	m	http://citi.usc.es/SOSSM#PA_Ons_19	13.3854	°C
2016-01-12T08:23:00.000+01:00	http://citi.usc.es/SOSSM#0438_201503_17	2.382	m	http://citi.usc.es/SOSSM#PA_Ons_19	13.3416	°C
2016-01-12T08:23:00.000+01:00	http://citi.usc.es/SOSSM#0438_201503_17	2.407	m	http://citi.usc.es/SOSSM#PA_Ons_19	13.5696	°C
2016-01-12T08:23:00.000+01:00	http://citi.usc.es/SOSSM#0438_201503_17	2.474	m	http://citi.usc.es/SOSSM#PA_Ons_19	13.1905	°C
2016-01-12T08:23:00.000+01:00	http://citi.usc.es/SOSSM#0438_201503_17	2.55	m	http://citi.usc.es/SOSSM#PA_Ons_19	13.354	°C
2016-01-12T08:23:00.000+01:00	http://citi.usc.es/SOSSM#0438_201503_17	2.581	m	http://citi.usc.es/SOSSM#PA_Ons_19	13.3762	°C
2016-01-12T08:23:00.000+01:00	http://citi.usc.es/SOSSM#0438_201503_17	2.63	m	http://citi.usc.es/SOSSM#PA_Ons_19	13.1175	°C
2016-01-12T08:23:00.000+01:00	http://citi.usc.es/SOSSM#0438_201503_17	2.701	m	http://citi.usc.es/SOSSM#PA_Ons_19	13.0571	°C
2016-01-12T08:23:00.000+01:00	http://citi.usc.es/SOSSM#0438_201503_17	2.799	m	http://citi.usc.es/SOSSM#PA_Ons_19	13.0912	°C

Fig. 61 Datos crudos de observaciones en formato Tabla

En este visualizador podremos realizar las siguientes tareas:

- Ordenar los datos ascendente o descendente según cada una de las columnas.
- Realizar búsquedas sobre los datos de la tabla, a modo de filtrado de datos
- Descargar los datos en formato CSV y TSV. Los datos que se descargan son el resultado de aplicar filtros de búsqueda y ordenación de la tabla. De esta forma se permite que el usuario pueda refinar la búsqueda inicial sin necesidad de ejecutar una nueva consulta GetObservation.

Los datos mostrados en esta tabla se corresponden con los datos de:

- PhenomenomTime
- Procedure
- Valor de profundidad
- Unidad de profundidad

- FeatureOfInterest
- Resultado
- UOM del resultado

Por otro lado, otra de las herramientas de visualización se corresponde con la herramienta de graficación de los datos de observación (ver Fig. 62).

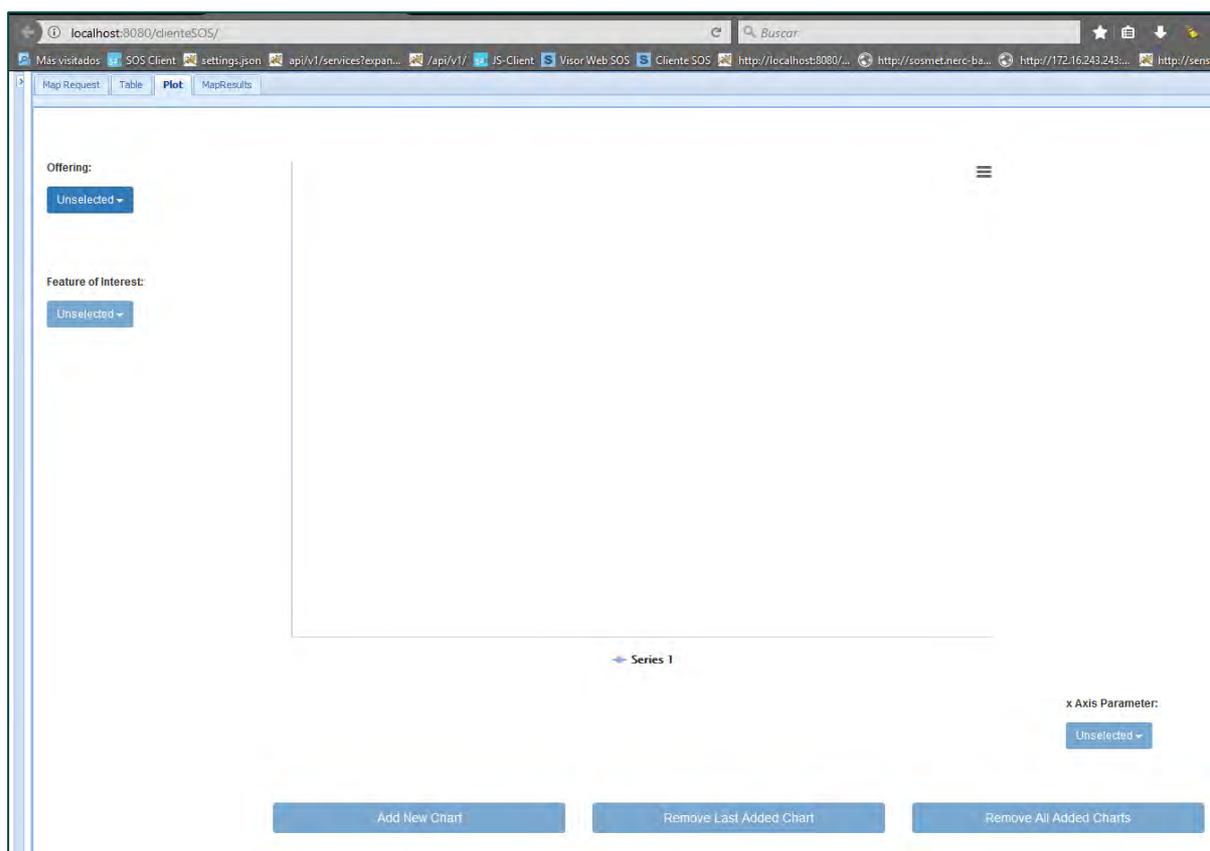


Fig. 62 Herramienta de graficación de datos de observación sin aplicar ningún filtro

En este punto se observa como la gráfica está vacía y todos los botones de la herramienta salvo uno están desactivados. Por lo tanto el usuario tendrá que escoger una Offering de este único botón activo, que está situado en la parte superior izquierda de la herramienta. El procedimiento de filtrado es similar al realizado en la consulta de observaciones. Habrá que seleccionar por tanto una Offering y una FOI de los menús

dropdown de la parte izquierda. Una vez seleccionados ambos se muestra lo siguiente (ver Fig. 63):

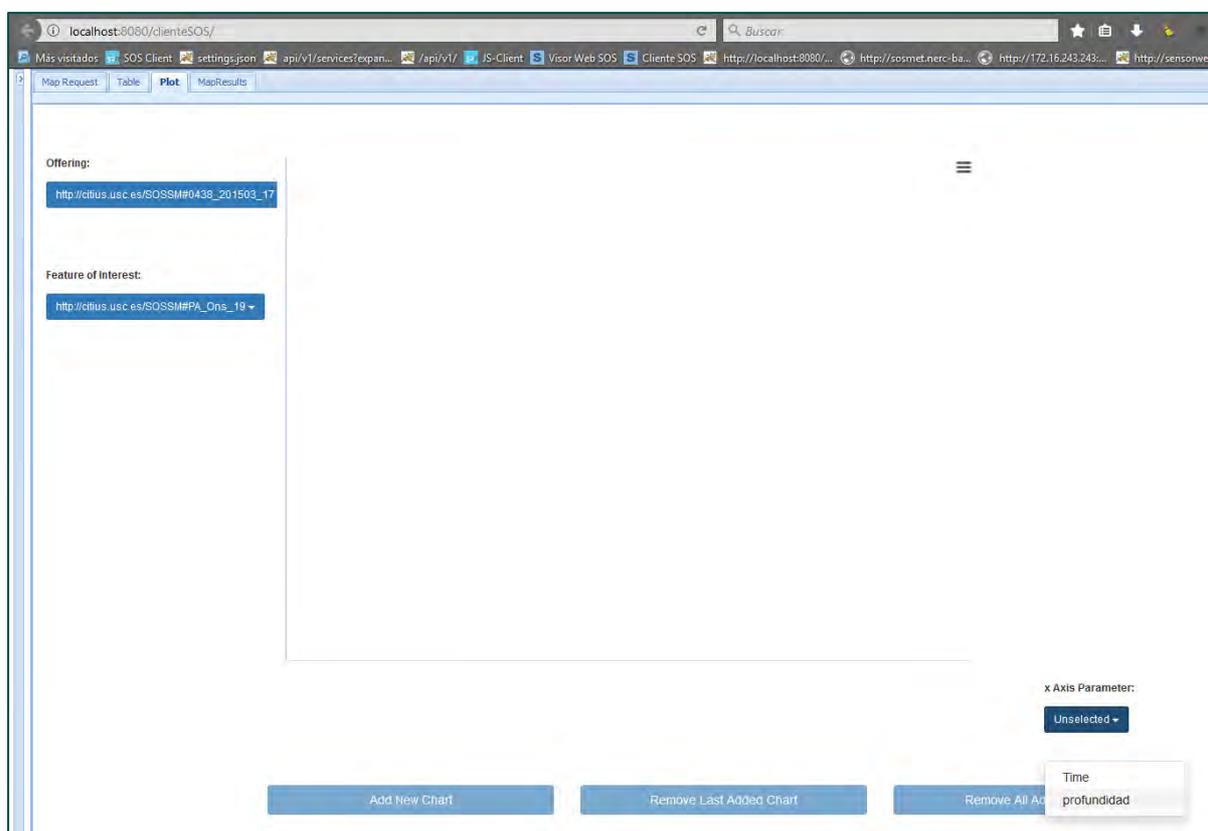


Fig. 63 Herramienta de gráficas esperando por la selección del parámetro a representar en el eje X

En este punto el usuario debe seleccionar que parámetro desea representar en el eje X: tiempo o el parámetro de profundidad, específico de nuestra fuente de datos.

Según la selección se mostrarán unos u otros menús de resolución en la parte inferior de la herramienta. A continuación se detalla el procedimiento para el graficado del parámetro de profundidad. En el otro caso el procedimiento de utilización es muy similar.

De esta manera, tras seleccionar el parámetro de profundidad par el eje X se muestran los siguientes controles en la parte inferior de la herramienta:

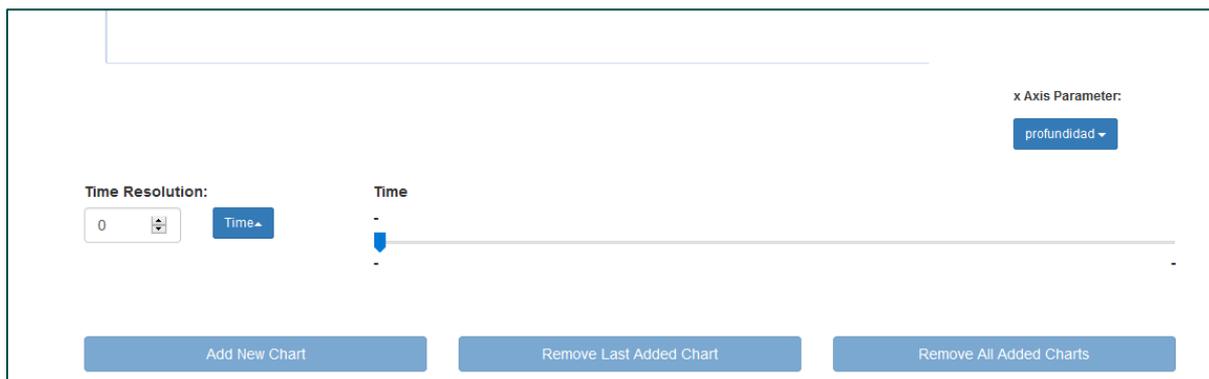


Fig. 64 Opciones de resolución para el muestreo de datos de profundidad en la herramienta de gráficos

En este estado el usuario deberá introducir los valores del filtro de resolución temporal y seleccionar posteriormente el momento de tiempo del que quiere graficar los valores asociados a la profundidad y la propiedad previamente seleccionada en la consulta de GetObservation (ver Fig. 65).

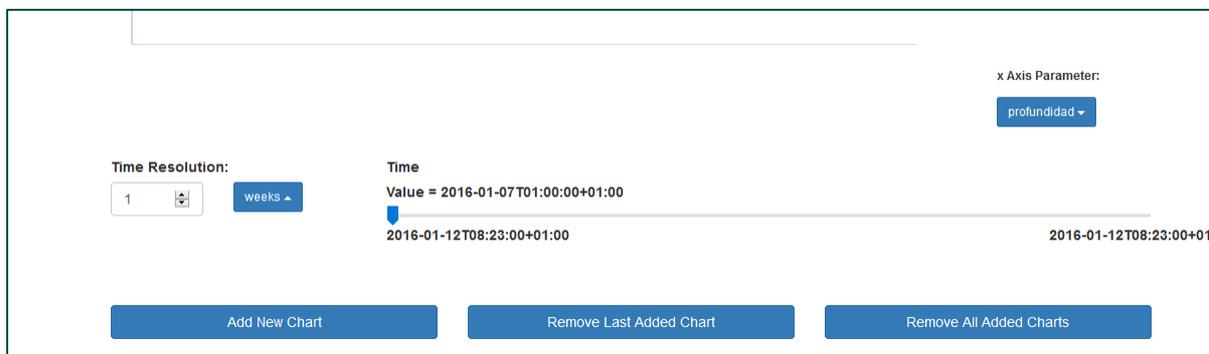


Fig. 65 Opciones de resolución introducidas para el muestreo de datos de profundidad en la herramienta de gráficos

Una vez que se tiene lo ilustrado en la Fig. 65 se activarán los botones situados en el panel inferior. En este punto se permite:

- Añadir nueva gráfica al diagrama
- Eliminar la ultima gráfica añadida
- Eliminar todas las gráficas del diagrama

Veamos que ocurre al añadir nueva gráfica (ver Fig. 66)

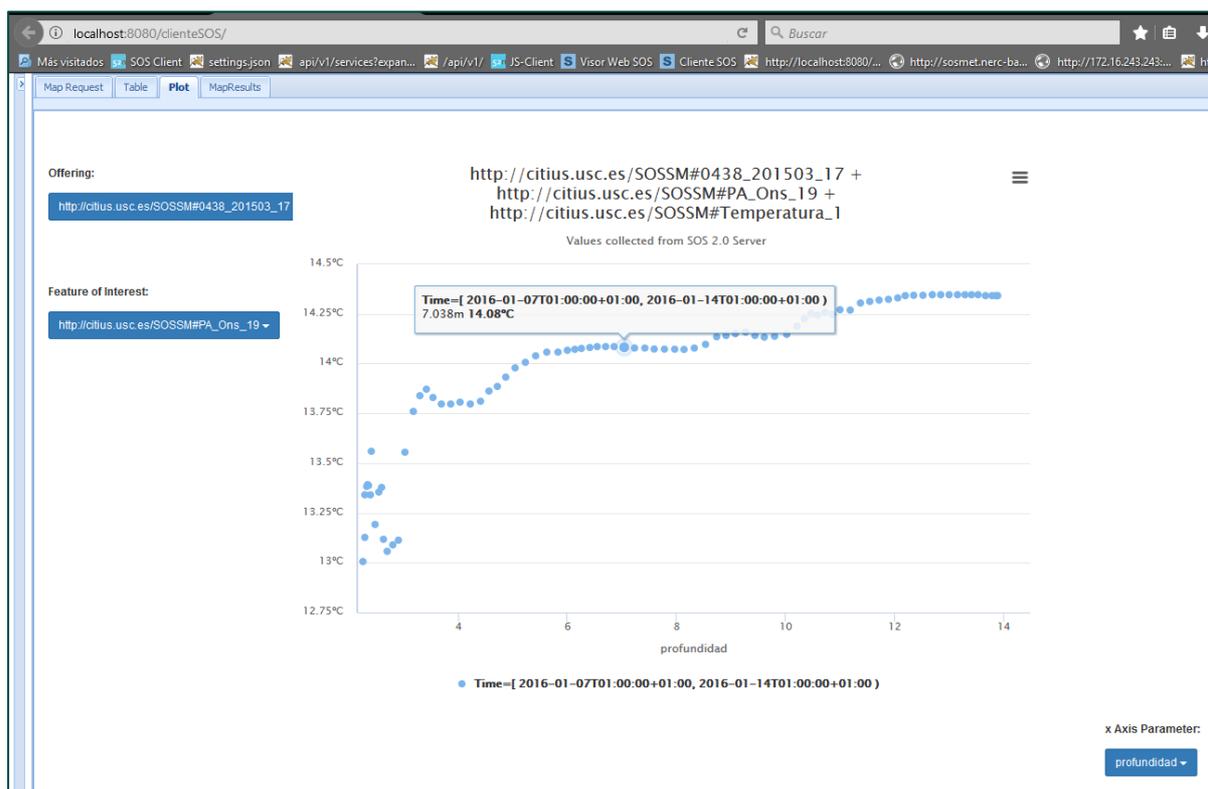


Fig. 66 gráfica generada según los filtros y resoluciones que el usuario ha introducido

Una vez que hemos generado un gráfico podemos observar los datos de cada uno de los intervalos generados situando el ratón por encima de cada uno de las figuras de la gráfica. Hay que indicar que los datos que se representan en esta gráfica han sido tratados empleando técnicas de interpolación y agrupación.

En este punto el usuario podrá introducir una nueva gráfica en el diagrama para confrontar los datos, tal y como se ilustra en la Fig. 67

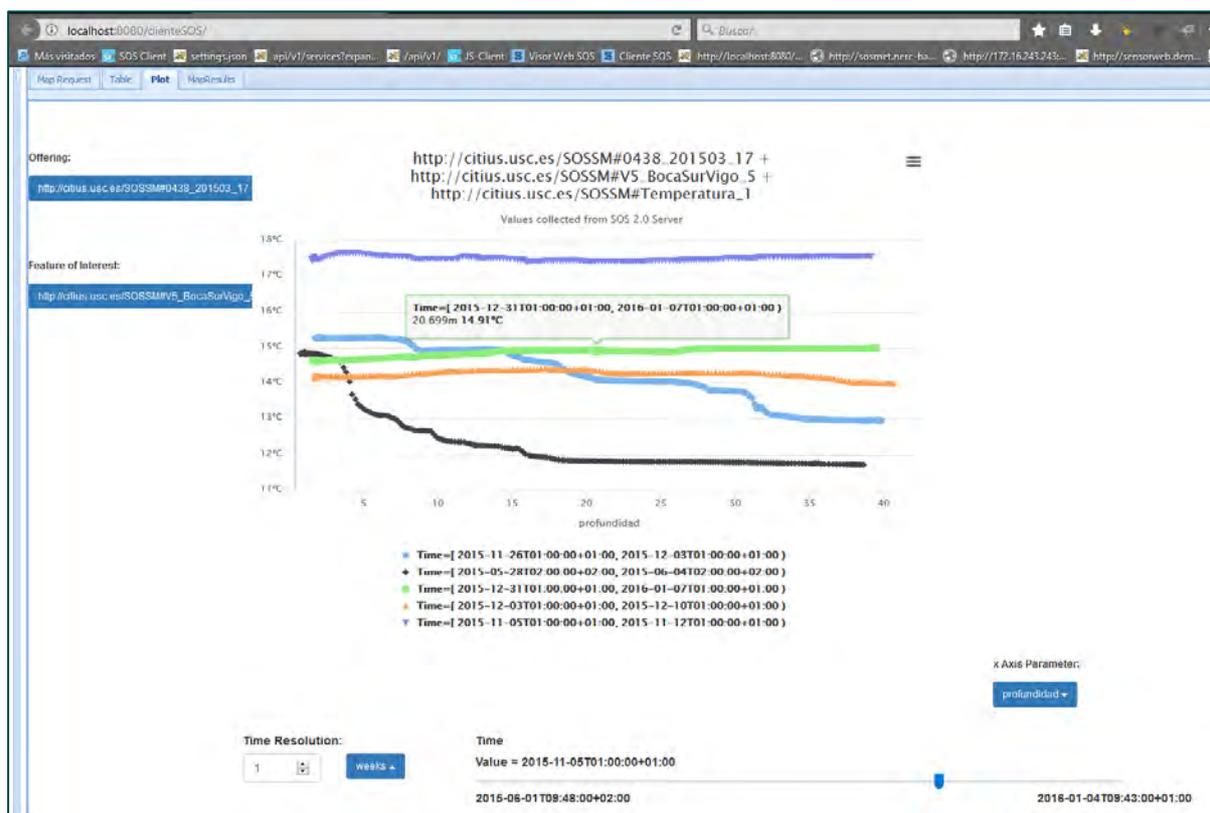


Fig. 67 Herramienta de graficación con diferentes gráficas añadidas al diagrama

Hay que destacar que el usuario podrá descargar estos datos procesados en diferentes formatos. Para ello deberá pulsar el icono situado en la parte superior derecha de la herramienta. Se desplegará por tanto el menú ilustrado en la Fig. 68

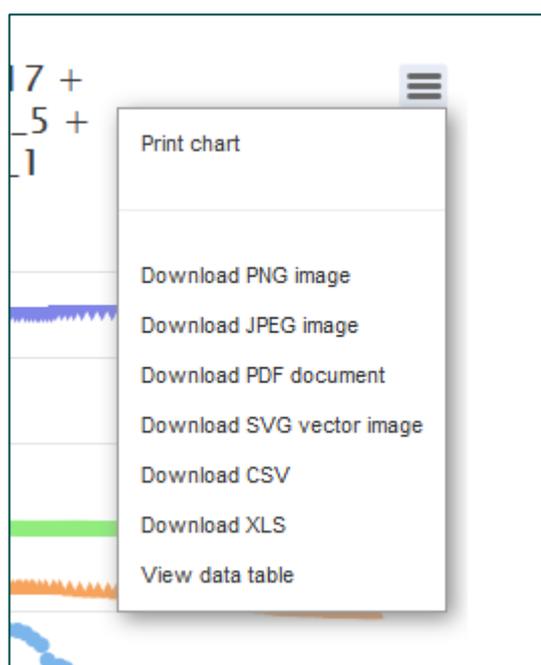


Fig. 68 Menú de exportación de datos procesados de la herramienta de gráficas

Desde este menú el usuario tiene acceso a las siguientes tareas:

- Imprimir gráfica
- Descargar gráfica en formato imagen PNG
- Descargar gráfica en formato imagen JPEG
- Descargar gráfica en formato documento PDF
- Descargar gráfica en formato imagen SVG
- Descargar gráfica en formato documento CSV
- Descargar gráfica en formato documento XLS
- Visualizar los datos generados en formato tabla

En cuanto a la herramienta de representación de datos de observación en formato mapa, indicar que el proceso de filtrado y representación es similar al implementado en la herramienta de graficación (ver Fig. 69). La explicación de la herramienta gráficas es igualmente válida para esta otra herramienta.

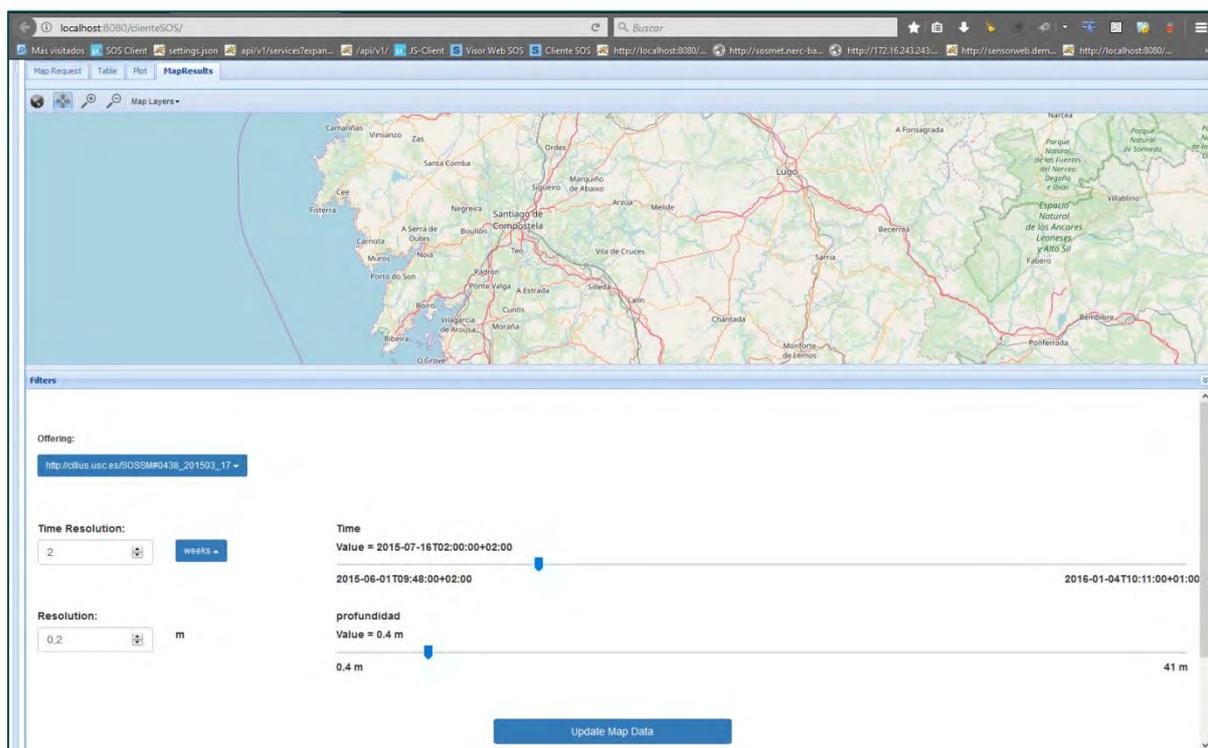


Fig. 69 Herramienta de representación de datos de observaciones en formato mapa

Apéndice E. Actas de reuniones

Reunión N°1 – Tarea T02	
Fecha / Hora	31/10/2016 11:00
Lugar	Sala de reuniones CiTIUS
Asistentes	<ul style="list-style-type: none">▪ Rubén Arenas Hernán▪ José Ramón Ríos Viqueira▪ Manuel Regueiro Seoane▪ Representante 1 de MeteoGalicia▪ Representante 2 de Intecmar
Agenda	<ul style="list-style-type: none">▪ Presentación del proyecto
Discusión de la agenda	<ul style="list-style-type: none">▪ Presentación de las partes implicadas en el proyecto▪ Realización de una presentación introductoria al proyecto Los representantes 1 y 2 introdujeron al equipo investigador nuevo en el ámbito del proyecto explicando sus motivaciones, alcance y objetivos▪ Se completó la fase de captura de requisitos▪ Creación de los primeros bocetos sobre la investigación a realizar y sobre la herramienta deseada

Resumen de los acuerdos	<ul style="list-style-type: none"> ▪ Especificación de los requisitos del proyecto ▪ Aceptación de la fecha de la siguiente reunión 	
Resumen de acciones		
Descripción de actuación	Fecha	Responsable
Redacción de Anteproyecto	-	Rubén Arenas
Redacción de los requisitos del proyecto	-	Rubén Arenas

Tabla 72 Acta de la reunión nº1 - Tarea T02

Reunión N°2 - Tarea T07.5	
Fecha / Hora	09/12/2016 10:30
Lugar	Sala de reuniones CiTIUS
Asistentes	<ul style="list-style-type: none"> ▪ Rubén Arenas Hernán ▪ José Ramón Ríos Viqueira ▪ Representante 1 de MeteoGalicia ▪ Representante 2 de Intecmar
Agenda	<ul style="list-style-type: none"> ▪ Presentación de resultados de investigación sobre clientes SOS 2.0 previamente compartidos via electrónica

	<ul style="list-style-type: none"> ▪ Presentación del diseño global del sistema atendiendo a las especificaciones previamente capturadas ▪ Exposición de la planificación futura del proyecto ▪ Exposición de los detalles de la entrega del primer incremento 									
Discusión de la agenda	<ul style="list-style-type: none"> ▪ Discusión sobre el diseño del sistema ▪ Refinamiento de requisitos de los módulos que componen el sistema ▪ Detección de requisitos emergentes del proyecto: RQ-EM-01 RQ-EM-03 									
Resumen de los acuerdos	<ul style="list-style-type: none"> ▪ Especificación pormenorizada de los requisitos de los módulos del sistema ▪ Establecimiento de requisitos emergentes ▪ Especificación detallada del incremento 2 ▪ Aceptación de la fecha de la siguiente reunión 									
Resumen de acciones										
<table border="1"> <thead> <tr> <th>Descripción de actuación</th> <th>Fecha</th> <th>Responsable</th> </tr> </thead> <tbody> <tr> <td>Desarrollo del primer incremento</td> <td>-</td> <td>Rubén Arenas</td> </tr> <tr> <td>Revisión de los requisitos del proyecto</td> <td>-</td> <td>Rubén Arenas</td> </tr> </tbody> </table>		Descripción de actuación	Fecha	Responsable	Desarrollo del primer incremento	-	Rubén Arenas	Revisión de los requisitos del proyecto	-	Rubén Arenas
Descripción de actuación	Fecha	Responsable								
Desarrollo del primer incremento	-	Rubén Arenas								
Revisión de los requisitos del proyecto	-	Rubén Arenas								

Tabla 73 Acta de la reunión nº2 - T07.5

Reunión N°3 - Tarea T08.6	
Fecha / Hora	10/01/2017 10:30
Lugar	Sala de reuniones CiTIUS
Asistentes	<ul style="list-style-type: none"> ▪ Rubén Arenas Hernán ▪ José Ramón Ríos Viqueira ▪ Representante 1 de MeteoGalicia ▪ Representante 2 de Intecmar
Agenda	<ul style="list-style-type: none"> ▪ Presentación de los resultados de las pruebas del primer incremento por parte de los representantes 1 y 2 ▪ Validación con los representantes 1 y 2 de la herramienta software entregada ▪ Exposición de los detalles de la entrega del segundo incremento
Discusión de la agenda	<ul style="list-style-type: none"> ▪ Puesta en conocimiento entre las partes sobre la experiencia del primer incremento desarrollado y entregado. ▪ Detección de requisitos emergentes del proyecto: RQ-EM-02
Resumen de los acuerdos	<ul style="list-style-type: none"> ▪ Validación de los requisitos implementados ▪ Establecimiento de requisitos emergentes ▪ Especificación pormenorizada del incremento 3 ▪ Aceptación de la fecha de la siguiente reunión
Resumen de acciones	

Descripción de actuación	Fecha	Responsable
Desarrollo del segundo incremento	-	Rubén Arenas
Revisión de los requisitos del proyecto	-	Rubén Arenas

Tabla 74 Acta de la reunión nº3 - T08.6

Reunión Nº3 - Tarea T08.6	
Fecha / Hora	20/02/2017 11:00
Lugar	Sala de reuniones CiTIUS
Asistentes	<ul style="list-style-type: none"> ▪ Rubén Arenas Hernán ▪ José Ramón Ríos Viqueira ▪ Representante 1 de MeteoGalicia ▪ Representante 2 de Intecmar
Agenda	<ul style="list-style-type: none"> ▪ Presentación de los resultados de las pruebas del segundo incremento por parte de los representantes 1 y 2 ▪ Validación con los representantes 1 y 2 de la herramienta software entregada ▪ Exposición de los detalles de la entrega del tercer

	incremento <ul style="list-style-type: none"> ▪ Toma de decisiones de trabajo futuro 									
Discusión de la agenda	<ul style="list-style-type: none"> ▪ Puesta en conocimiento entre las partes sobre la experiencia del segundo incremento desarrollado y entregado. 									
Resumen de los acuerdos	<ul style="list-style-type: none"> ▪ Validación de los requisitos implementados ▪ Establecimiento de requisitos emergentes ▪ Feedback telemático sobre las pruebas del tercer incremento 									
Resumen de acciones										
<table border="1"> <thead> <tr> <th>Descripción de actuación</th> <th>Fecha</th> <th>Responsable</th> </tr> </thead> <tbody> <tr> <td>Desarrollo del segundo incremento</td> <td>-</td> <td>Rubén Arenas</td> </tr> <tr> <td>Revisión de los requisitos del proyecto</td> <td>-</td> <td>Rubén Arenas</td> </tr> </tbody> </table>		Descripción de actuación	Fecha	Responsable	Desarrollo del segundo incremento	-	Rubén Arenas	Revisión de los requisitos del proyecto	-	Rubén Arenas
Descripción de actuación	Fecha	Responsable								
Desarrollo del segundo incremento	-	Rubén Arenas								
Revisión de los requisitos del proyecto	-	Rubén Arenas								

Tabla 75 Acta de la reunión nº4 - T09.