

## Visualización de series temporales oceanográficas mediante SOS.js

S. Soto Alonso<sup>(1)</sup>, O. Chic Giménez<sup>(1)</sup>, O. Mulet Morales<sup>(1)</sup> y J. Guillén Aranda<sup>(1)</sup>

<sup>(1)</sup> Observatorio Marino Costero (COO). Instituto de Ciencias del Mar (ICM-CSIC). Passeig Marítim de la Barceloneta, 37-49, 08003 Barcelona, coo@listas.csic.es.

### RESUMEN

*La duración de los proyectos científicos (media de 3 años) supone una limitación para obtener largas series temporales que son imprescindibles para alcanzar un buen conocimiento de los procesos oceanográficos. La existencia de estaciones marinas permanentes, como el Observatorio Marino Costero (COO) del ICM-CSIC, hace posible la obtención de estas mediciones a largo plazo de diversos parámetros oceanográficos. Desde abril de 2002, se realiza una campaña oceanográfica mensual, para obtener una serie temporal de datos en la costa frente a la ciudad de Barcelona. Con el fin de visualizar estas observaciones se utiliza la tecnología Sensor Web Enablement (SWE) del Open Geospatial Consortium (OGC) que ha definido especificaciones para estandarizar datos generados por sensores. Dentro del conjunto de estas especificaciones, Sensor Observation Service (SOS) ha sido útil para describir las observaciones y los sensores de una manera estándar. Para la visualización de los datos hemos utilizado el cliente SOS.js desarrollado por Paul Breen del British Antarctic Survey (BAS) y el SOS 1.0 TestClient v2 de 52º North como cliente-servidor SOS. El desarrollo del sistema resuelve la complejidad inherente al uso de estas especificaciones debida a la falta de herramientas ágiles a la hora de la publicación y actualización de los datos. La aplicación de estas herramientas de visualización ha demostrado su utilidad en el análisis e interpretación de procesos oceanográficos. La estandarización de los datos supone un gran esfuerzo que queda compensado por la interoperabilidad que se consigue.*

**Palabras clave:** SOS, OGC, SWE, SensorML, O&M, series temporales, oceanografía.

## INTRODUCCIÓN

Hasta hace pocos años, los datos adquiridos en campañas oceanográficas sólo se podían visualizar, previo tratamiento, con el software de la empresa fabricante de los instrumentos y los sensores. Posteriormente, estos datos se analizaban más a fondo haciendo uso de herramientas software que han ido cambiando en el tiempo como Grapher, Surfer, Matlab, IDL... Durante años, las tecnologías software usadas en oceanografía han ido paralelas y sin converger con las del mundo geoespacial. Ambos han utilizado diferentes modelos de datos, software, etc. El primero modeliza los sistemas como variables 3D evolucionando en el tiempo y el segundo como estructuras 2D fijos cartografiables. Poco a poco las tecnologías han convergido y formatos o frameworks como el NetCDF, muy utilizado en meteorología y oceanografía, a día de hoy es un estándar de la OGC.

El proyecto que se presenta aquí también responde a esta convergencia de tecnologías y estándares entre el ámbito de la oceanografía y el de los Sistema de Información Geográfica (SIG). La oceanografía se beneficia de las herramientas y estándares Open Source desarrollados bajo el amparo del OGC en la gestión y visualización de sus datos.

En este sentido, el Observatorio Marino Costero (COO) viene trabajando desde hace unos años. El COO fue creado en 2001 como un servicio del Instituto de Ciencias del Mar (ICM-CSIC) dedicado a la adquisición, gestión y visualización de datos oceanográficos. Estos datos se pueden visualizar en tiempo casi-real por Internet con aplicaciones web con el objetivo de contribuir a las actividades científicas del instituto. El objetivo del COO es realizar un seguimiento físico, morfológico y biológico con la mayor resolución posible y con una perspectiva a largo plazo con el fin de determinar el comportamiento y la evolución del sistema costero donde las influencias naturales y humanas coexisten.

Desde abril de 2002, una de las actividades del COO es la realización de una campaña oceanográfica mensual en la costa frente a la ciudad de Barcelona. En esta, se utiliza un equipo CTD (Conductivity Temperature Depth), con sensores adicionales añadidos al equipo, y se realizan ocho perfiles de la columna de agua a lo largo de dos cortes transversales separados 2.5 kilómetros desde los 10 a los 40 m. de profundidad (Fig. 1).



Figura 1: Mapa de localización de los perfiles

Este conjunto de perfiles se pueden codificar como series temporales de diferentes variables oceanográficas de la columna de agua.

Los datos brutos son adquiridos durante las campañas con el software del instrumento que proporciona el fabricante. Estos datos se filtran para eliminar distintos tipos de errores (ruidos, datos fuera de los rangos climatológicos...) en el proceso del control de calidad. Incluso procediendo de una manera muy rigurosa en la estrategia de la planificación de adquisición, pueden quedar errores asociados a la misma técnica de medición, pudiendo modificar los parámetros que se quieren medir.

Las variables que se miden directa o indirectamente son: conductividad, temperatura, profundidad, salinidad, fluorescencia, turbidez, irradiancia y oxígeno. El resultado es una tabla de datos como la que se muestra en la Figura 2.

Estos datos son complementarios de los obtenidos mediante otros equipamientos científicos del observatorio en la zona costera de Barcelona. Estos incluyen una estación meteorológica, correntímetros (con sensores de temperatura, salinidad, turbidez, presión, oleaje, velocidad y dirección de la corriente), un sistema EUMETCast de recepción de imágenes de satélite, un sistema de video monitorización de las playas de Barcelona y una embarcación ligera.

Scan	TimeS	PrM	Tnc68C	Sal100	SeaTurbMtr	F1SP	Par	Sbeox0Mg/L
977	122.001	2.000	13.6724	38.2207	0.608	6.3418e-01	2.7245e+02	5.26357
994	124.121	2.500	13.6610	38.2219	1.113	6.2737e-01	2.2645e+02	5.26061
1009	125.959	3.000	13.6541	38.2217	1.407	4.9990e-01	2.1250e+02	5.26036
1023	127.707	3.500	13.6507	38.2214	1.410	2.7802e-01	2.0722e+02	5.27558
1040	129.911	4.000	13.6497	38.2209	1.418	1.9018e-01	1.9020e+02	5.28397
1061	132.488	4.500	13.6419	38.2212	1.436	1.4448e-01	1.6487e+02	5.29428
1074	134.147	5.000	13.6349	38.2211	1.390	1.2233e-01	1.4087e+02	5.29942
1082	135.145	5.500	13.6345	38.2192	1.453	2.9526e-01	1.2304e+02	5.30327
1099	137.205	6.000	13.6323	38.2195	1.585	2.0923e-01	1.1285e+02	5.30313
1113	138.945	6.500	13.6301	38.2191	1.629	2.2400e-01	9.7090e+01	5.33056
1122	140.166	7.000	13.6296	38.2190	1.589	2.3647e-01	8.5122e+01	5.33958
1138	142.086	7.500	13.6315	38.2189	1.751	2.2671e-01	7.1655e+01	5.34165
1155	144.216	8.000	13.6332	38.2191	1.819	2.7682e-01	6.5092e+01	5.33895
1168	145.892	8.500	13.6343	38.2191	1.834	3.2932e-01	5.2372e+01	5.34466
1175	146.803	9.000	13.6327	38.2195	1.680	3.1258e-01	4.5617e+01	5.35141
1199	149.691	9.500	13.6288	38.2182	1.794	3.3332e-01	4.3262e+01	5.34258
1212	151.365	10.000	13.6287	38.2184	1.876	2.7567e-01	4.0627e+01	5.34750

Figura 2: Datos brutos obtenidos con el CTD

## METODOLOGÍA

### Tecnología

El objetivo de este proyecto es la visualización web de los datos recogidos en las campañas mediante el uso de herramientas geoespaciales Open Source. En este trabajo se ha hecho uso de la iniciativa de OGC SWE que define especificaciones para estandarizar el uso de los datos generados por los sensores que permitan su interoperabilidad.

En nuestro caso de estudio no tenemos sensores fijos sino 8 estaciones fijas a las que se acude en una barca y en donde se captura la información de la columna de agua a través de un CTD. La tecnología SWE es muy útil tanto para trabajar con datos históricos de series temporales que se recogen durante estas campañas como para sensores que adquieren datos en tiempo real.

Dentro del SWE existen varias especificaciones entre las que se han utilizado las siguientes:

**SOS:** es un servicio web que permite consultar los datos de sensores en tiempo real o casi real y también series temporales de datos históricos. La especificación permite, por lado, hacer una descripción del sensor, es decir construir los metadatos y codificarlos con la especificación Sensor Model Language (SensorML) y, por otro, codificar los valores y formatos de las observaciones y mediciones con la especificación Observations and Measurements (O&M).

La ventaja del SOS es que los datos y la descripción del sensor - de cualquier tipo - están disponibles en un formato estandarizado utilizando operaciones estandarizadas. De este modo, el acceso a los datos del sensor a través de web services se simplifica y además se consigue una fácil integración en las Infraestructuras de Datos Espaciales (IDE) o SIG.

**SensorML:** es el lenguaje que establece los modelos estándar y el esquema XML necesario para describir los sistemas de sensores, los procesos asociados a las observaciones de dichos sensores, la ubicación de las observaciones, el procesamiento de las observaciones de sensores de bajo nivel y el listado de propiedades de las funciones, así como soportar el procesamiento bajo demanda de las observaciones del sensor.

**O&M:** son los modelos estándar y el esquema XML para codificar observaciones y mediciones de un sensor, tanto en tiempo real como histórico (Fig. 3).

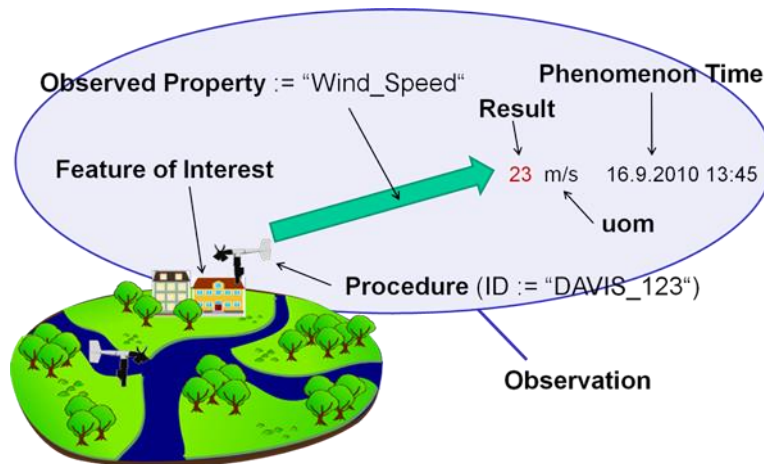


Figura 3: Modelo de observaciones de SOS (Fuente: OGC Network)

Entre las opciones de sistemas SOS que había en el momento de realizar este trabajo se escogió el SOS 1.0 TestClient v2 de 52°North que tiene arquitectura cliente-servidor. Es cliente ya que se pueden introducir los datos del muestreo y es servidor en tanto que permite hacer consultas SOS desde la aplicación directamente o desde otras aplicaciones como las de visualización probadas en este proyecto. Esta aplicación tiene asociada una base de datos PostgreSQL-PostGIS en la que se insertan los datos con la descripción del sensor dentro de una petición de registro SOS, RegisterSensor (RS) y los datos que adquiere dicho sensor con una petición InsertObservation (IO) todo bajo los estándares SensorML y O&M.

### Preparación de los esquemas

Tanto el RS como el IO han sido editados, en parte, a partir de un modelo plantilla ofrecido por el TestClient. En el caso del RS, se ha seguido el Register Sensor\_withParents. Este se compone de los siguientes atributos (Fig. 4):

```
<?xml version="1.0" encoding="UTF-8"?>
<RegisterSensor service="SOS" version="1.0.0" xmlns="http://www.opengis.net/sos/1.0"
  xmlns:swe="http://www.opengis.net/swe/1.0.1"
  xmlns:ows="http://www.opengeospatial.net/ows"
  xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:gml="http://www.opengis.net/gml"
  xmlns:ogc="http://www.opengis.net/ogc" xmlns:om="http://www.opengis.net/om/1.0"
  xmlns:sml="http://www.opengis.net/sensorML/1.0.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/sos/1.0
http://schemas.opengis.net/sos/1.0.0/sosRegisterSensor.xsd
http://www.opengis.net/om/1.0
http://schemas.opengis.net/om/1.0.0/extensions/observationSpecialization\_override.xsd">
<!-- Sensor Description parameter; Currently, this has to be a sml:System -->
<SensorDescription>
  <sml:SensorML version="1.0.1">
    <sml:member>
      <sml:System xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <sml:identification></sml:identification>
        <sml:capabilities>
          <swe:SimpleDataRecord></swe:SimpleDataRecord>
        </sml:capabilities>
        <sml:inputs></sml:inputs>
        <sml:outputs></sml:outputs>
      </sml:System>
    </sml:member>
  </sml:SensorML>
</SensorDescription>
</RegisterSensor>
```

Figura 4: Esquema del Registro del Sensor utilizado (RS)

Para el IO, optamos por la plantilla InsertObservation\_GenericObs\_Derwert.xml, que contiene los siguientes atributos (Fig. 5):

```
<InsertObservation xmlns="http://www.opengis.net/sos/1.0"
  xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:om="http://www.opengis.net/om/1.0" xmlns:sos="http://www.opengis.net/sos/1.0"
  xmlns:sa="http://www.opengis.net/sampling/1.0" xmlns:gml="http://www.opengis.net/gml"
  xmlns:swe="http://www.opengis.net/swe/1.0.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/sos/1.0
http://schemas.opengis.net/sos/1.0.0/sosInsert.xsd http://www.opengis.net/sampling/1.0
http://schemas.opengis.net/sampling/1.0.0/sampling.xsd http://www.opengis.net/om/1.0
http://schemas.opengis.net/om/1.0.0/extensions/observationSpecialization\_override.xsd"
  service="SOS" version="1.0.0">
<AssignedSensorId></AssignedSensorId>
  <om:Observation>
    <om:samplingTime></om:samplingTime>
    <om:procedure xlink:href="urn:ogc:object:feature:Sensor:ICM:icm-ctd-sbe25-pu11"/>
    <om:observedProperty></om:observedProperty>
    <om:featureOfInterest></om:featureOfInterest>
    <om:result>
      <swe:DataArray>
        <swe:elementCount></swe:elementCount>
        <swe:elementType name="Components"></swe:elementType>
        <swe:encoding></swe:encoding>
        <swe:values></swe:values>
      </swe:DataArray>
    </om:result>
  </om:Observation>
</InsertObservation>
```

Figura 5: Esquema de la Observación Insertada (IO)

La edición de estas plantillas es tediosa y se echa en falta herramientas que faciliten la edición de los ficheros XML, que son muy adecuados para el intercambio de datos entre servidores pero no para ser editados por humanos. Sin embargo, esta dificultad queda contrarrestada con la estandarización de los datos que pueden ser suministrados por cualquier servidor OGC y por su utilidad en el análisis e interpretación de procesos oceanográficos.

### **Preparación de los datos**

Previamente a la introducción de los datos a través de una petición IO, éstos se preparan para que el formato esté acorde con la estandarización marcada por la OGC.

Se ha desarrollado un programa en Python, concretamente con la tecnología iPython Notebook, para crear las tablas de datos que van en el atributo o campo `swe:values` del `swe:DataArray`.

Una de las conversiones a realizar es la del tiempo. En los datos brutos el tiempo es un campo `TimeS` codificado en segundos desde que se empieza a bajar el CTD muestreando la columna de agua. El estándar obliga a convertir el tiempo según la ISO 8601 en el que tenemos un único codificador para la fecha y el tiempo del tipo: `YYYY-MM-DDThh:mm:ss:ms` con los milisegundos incluidos y sin espacios en blanco. Esta sección del IO también lleva la información del perfil, el llamado Feature of Interest (FOI).

Tras ejecutar este script de Python aún hay que editar el fichero XML introduciendo el número total de registros que vamos a introducir (`swe:elementCount`). En algunos casos se debe introducir un código de final de línea y añadir la cadena "noData" donde no hubiese datos o no fuesen de buena calidad.

### **Visualización de los datos**

Tras hacer una evaluación de los softwares del mercado se optó por la tecnología de 52° North por ser una de las más maduras y, además, tener un cliente que permite visualizar los datos. Sin embargo, después de hacer pruebas con el visor `ThinSweClient` de 52° North se comprobó que no cumplía los mínimos requerimientos necesarios ni tenía la flexibilidad solicitada para ver series temporales (Fig. 6).

La frecuencia mensual de las campañas no permite una visualización cómoda de las observaciones al tener que desplazar una barra que como máximo permite desplazar el tiempo hacia atrás un mes. La falta de un mecanismo que permita definir fácilmente el rango temporal hace difícil utilizar este visor en producción. Es posible que este visor esté más diseñado para ver datos de tiempo real o casi-real. Sin embargo, la aplicación tiene características útiles ya que permite visualizar de forma gráfica diversas variables a la vez y de diferentes sensores o perfiles.

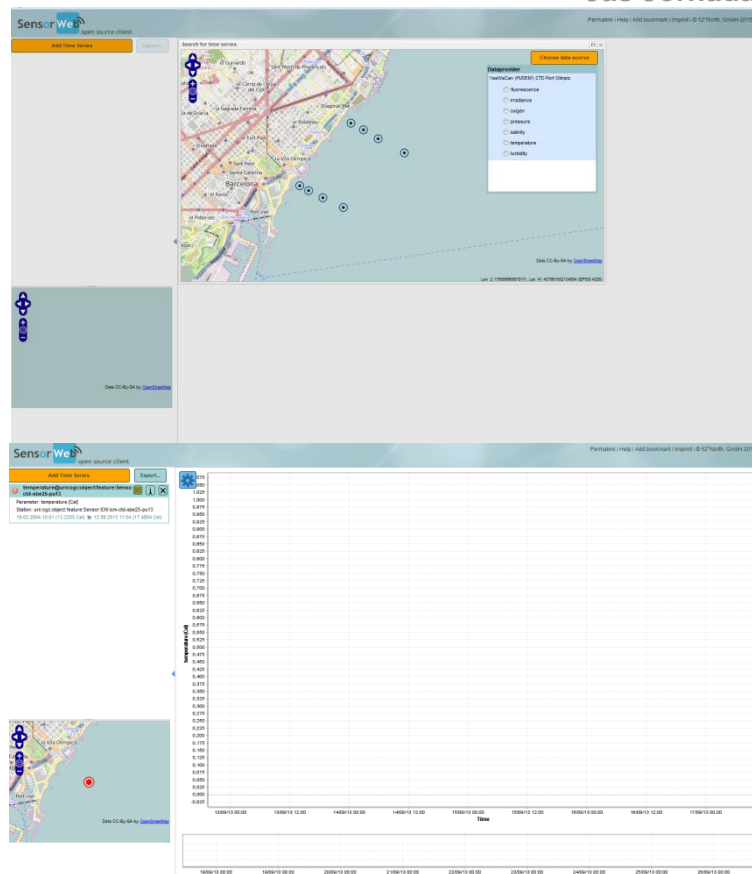


Figura 6: Aplicación de visualización de 52° North

Tras buscar en el fórum de 52° North, se decide optar por la aplicación SOS.js basada en las librerías JavaScript OpenLayers, jQuery, y flot para navegar, visualizar y acceder a los datos a través de un servicio SOS, desarrollada por Paul Breen del British Antarctic Survey (BAS).

Se trata de un SOS ligero que posteriormente 52°North ha adoptado como parte de su toolkit licenciado bajo Apache License 2.0 de acuerdo con el BAS. La aplicación permite visualizar observaciones de diferentes sensores, seleccionar dichos sensores de forma interactiva y representarlos gráficamente, explorar las series temporales de las diferentes variables en un mismo gráfico, visualizar direcciones como rosa de vientos, hacer estadísticas sencillas...

Su salida gráfica para series temporales es continua y muy adecuada para datos de frecuencia variable como es el caso de estudio y permite fijar el rango temporal y visualizar las variables aunque no tengamos un registro continuo. Además existe la opción de mostrar los datos en formato de tabla (Fig. 7), así como descargarlos.



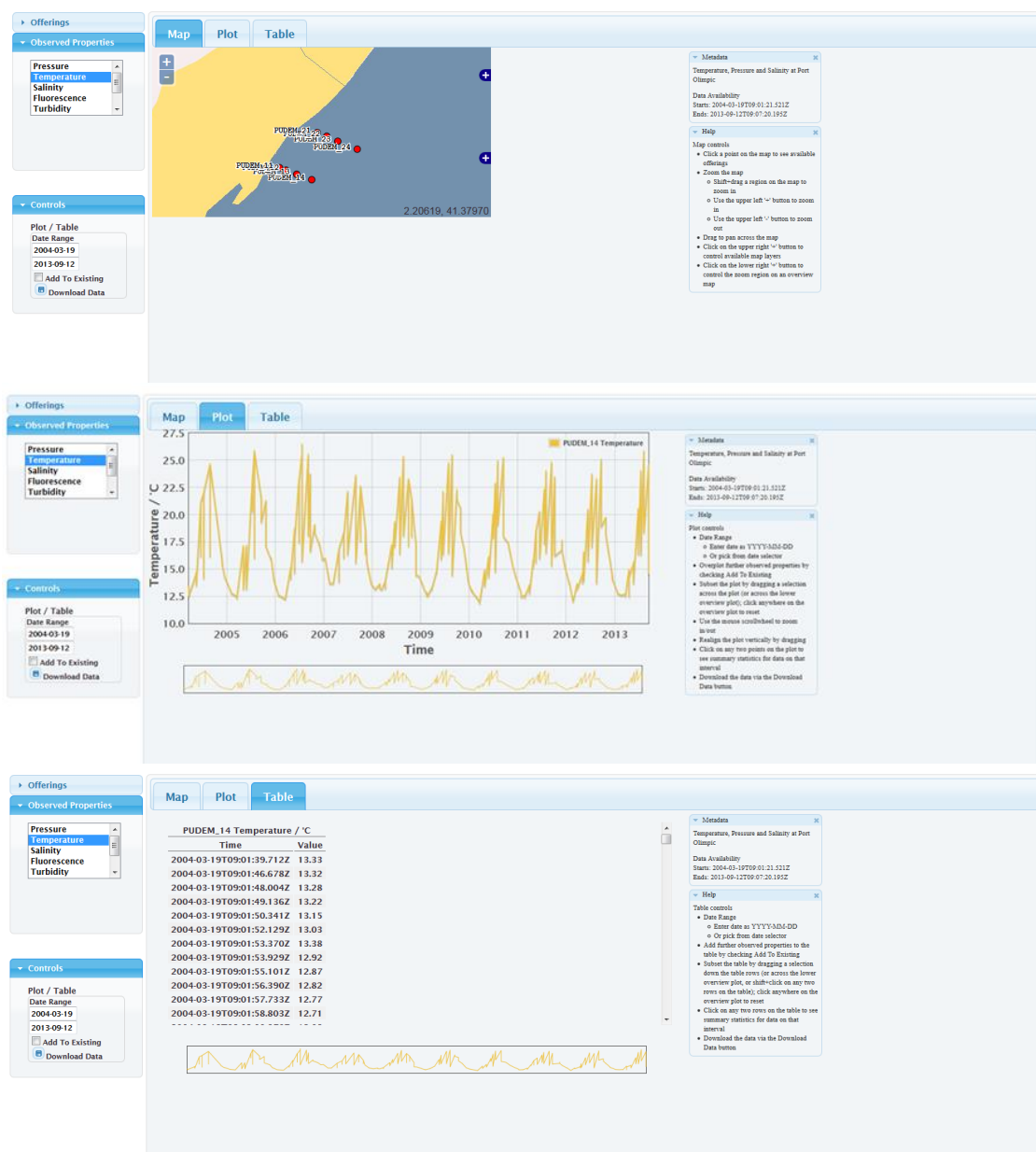


Figura 7: Aplicación de visualización de SOS.js

Al igual que en el visor de 52°North, es posible visualizar varias variables simultáneamente en un mismo gráfico con la mejora, en referencia al cliente de 52° North, de poder representar periodos determinados. Igualmente, permite la comparación de datos de diferentes sensores o, en nuestro caso, perfiles en un mismo gráfico. También permite obtener una estadística básica de un rango temporal de las variables. (Fig. 8 y 9)

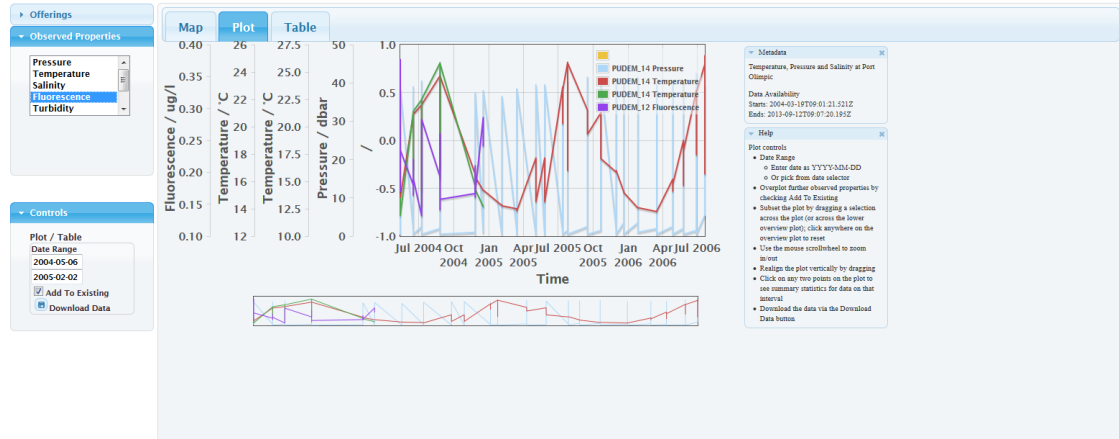


Figura 8: Representación gráfica de varias variables de un sensor en un rango temporal

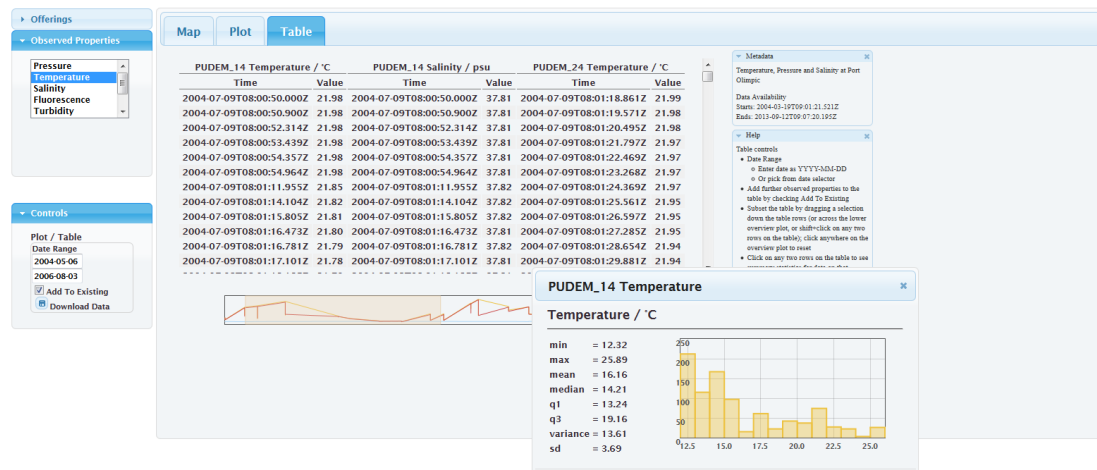


Figura 9: Representación de los datos en tablas y resumen estadístico de una variable

Toda la información del SOS 1.0 TestClient utilizado se almacena en una base de datos PostgreSQL-PostGIS. Analizando las tablas de la base de datos y los cambios observados al añadir nuevos datos, se abre la posibilidad a buscar un método alternativo a través de la programación clásica o Shell scripts para hacer la importación de nuevos datos directamente haciendo INSERTs en la base de datos y no teniendo que pasar por la laboriosa edición de ficheros XML.

## CONCLUSIONES

La confluencia de la oceanografía con las geociencias ha permitido a la primera avanzar tecnológicamente y disfrutar de nuevos formatos, modelos y, sobre todo, nuevas herramientas de visualización de los datos. Los estándares de la OGC basados en herramientas Open Source permiten interoperabilidad y mejorar la gestión de los datos.

SOS.js permite la visualización de más de una variable conjuntamente de las series temporales obtenidas de las campañas. Este visor es útil para la interpretación y el análisis de los datos obtenidos por los sensores y para la detección de errores instrumentales en la recogida de los mismos.

Queda patente la falta de modelos y visores de las geociencias que permitan representar la complejidad de los datos oceanográficos 4D (x,y,z,t). En este sentido, los modelos de datos de NetCDF y ArcMarine pueden ser de utilidad. En esta línea, un objetivo de futuro es la mejora de las capacidades gráficas de la librería Flot.

El desarrollo del sistema resuelve la complejidad inherente al uso de estas especificaciones debida a la falta de herramientas ágiles a la hora de la publicación y actualización de los datos.

Aun así, es necesario mejorar las herramientas de creación de los ficheros IO y RS para que sea más útil todo este esquema. Otro objetivo de futuro es migrar el proyecto a SensorML versión 2.0 y encontrar una manera más ágil de editar los esquemas y definir plantillas acorde al perfil de la oceanografía. Otra idea a explorar es crear software para poder hacer importaciones a la base de datos sin tener que editar ficheros XML.

En la línea de trabajo futura se propone realizar una comparativa con el istSOS desarrollado en Python por el Istituto Scienze della Terra de Cannobio, Suiza.

## AGRADECIMIENTOS

Agradecemos a Maribel Lloret del Servicio de Instrumentación y a Neus Maestro del Departamento de Geociencias Marinas su trabajo y dedicación durante las campañas en representación de todas las personas que han colaborado.

## REFERENCIAS

- ◆ Especificaciones OGC: <http://www.opengeospatial.org/standards/is>
- ◆ Open Geospatial: <http://www.opengeospatial.org/ogc/markets-technologies/swe>
- ◆ OGC Network: <http://www.ogcnetwork.net/>
- ◆ Sensor ML: <http://en.wikipedia.org/wiki/SensorML>
- ◆ Presentación de GeoCENS: <http://www.slideshare.net/cybera/geo-cens-ogc-standards-and-sensor-web-enablementgeorge-percivall>
- ◆ SOS.js: <http://sosjs.readthedocs.org/en/latest/overview.html>
- ◆ 52north: <http://blog.52north.org/2014/02/21/sos-js/>
- ◆ Librería Flot: <http://www.flotcharts.org/>
- ◆ Fórum 52º North: <http://sensorweb.forum.52north.org/>
- ◆ NetCDF: <http://www.unidata.ucar.edu/software/netcdf/>
- ◆ ArcMarine: <http://dusk.geo.orst.edu/djl/arcgis/>