

UNIVERSIDADE DE SANTIAGO DE
COMPOSTELA



ESCOLA TÉCNICA SUPERIOR DE ENXEÑARÍA

Visualización de observacións en SIX de escritorio

Autor:

Rubén Mosquera Varela

Directores:

José Ramón Ríos Viqueira

Manuel Antonio Regueiro Seoane

Grao en Enxeñaría Informática

Xullo 2015

Traballo de Fin de Grao presentado na Escola Técnica Superior de Enxeñaría
da Universidade de Santiago de Compostela para a obtención do Grao en
Enxeñaría Informática



D. José Ramón Ríos Viqueira, Profesor do Departamento de Electrónica e Computación da Universidade de Santiago de Compostela e investigador do CiTIUS, e **D. Manuel Antonio Regueiro Seoane**, investigador do CiTIUS,

INFORMAN:

Que a presente memoria, titulada *Visualización de observacións en SIX de escritorio*, presentada por **D. Rubén Mosquera Varela** para superar os créditos correspondentes ao Traballo de Fin de Grao da titulación de Grao en Enxeñaría Informática, realizouse baixo nosa dirección no Departamento de Electrónica e Computación da Universidade de Santiago de Compostela.

E para que así conste aos efectos oportunos, expiden o presente informe en Santiago de Compostela, a 10 de Xullo de 2015:

O director,

O codirector,

O alumno,

José Ramón Ríos Viqueira Manuel Antonio Regueiro Seoane Rubén Mosquera Varela

Agradecementos

A Olatz, pola motivación, o ánimo e o apoio.

Resumo

No eido da observación medioambiental xéranse a cada instante inxentes cantidades de datos, e a estimación de crecemento a curto prazo é considerable. Este gran volume de datos é tremendamente heteroxéneo, en gran medida pola ampla variedade de dispositivos e sistemas de sensorización empregados. O acceso a toda esta información dende ferramentas de axuda á toma de decisións está, en xeral, dificultado pola falta de interoperabilidade derivada dos diferentes formatos de representación.

Co fin de facilitar o intercambio de datos de observación, o Open Geospatial Consortium (OGC), mediante a súa iniciativa Sensor Web Enablement (SWE), propón a adopción de diferentes estándares de servizos xeoespaciais a través da web. De gran relevancia para este traballo son o estándar Observations and Measurements (O&M), que define o modelo de datos e a súa codificación XML para a representación de observacións medioambientais, e o estándar Sensor Observation Service (SOS), que define a interface de servizo web para a publicación de todo tipo de información relativa a calquera tipo de sensor e as observacións realizadas polo mesmo.

O nivel de implantación destes estándares é aínda moi baixo, entre outros motivos pola a falta de soporte dos mesmos nos Sistemas de Información Xeográfica (SIX) de propósito xeral.

Así pois, o obxectivo principal deste traballo de fin de grao é dotar ó QGIS da capacidade de consultar diferentes servidores de datos medioambientais a través da interface SOS, e representar os datos obtidos no contorno de mapas propio da ferramenta. Deste xeito permítese a integración de datos de observación de diferente natureza de forma sinxela para a súa posterior exploración e análise por parte do usuario, aproveitando toda a funcionalidade de QGIS, o SIX de escritorio máis empregado e coa comunidade, tanto de usuarios como de desenvolvedores, máis numerosa, dentro do ámbito do software libre.

O *plugin* resultado deste proxecto incorporase ó repositorio oficial de *plugins* da ferramenta QGIS co nome SOS Client.

Índice xeral

1. Introducción	1
1.1. Contextualización	1
1.2. Motivación e Obxectivos	3
1.3. Entorno tecnolóxico	4
1.3.1. QGIS	4
1.3.2. Sensor Observation Service	4
1.3.3. Observations and Measurements	6
1.4. Estrutura da memoria	8
2. Xestión do proxecto	9
2.1. Alcance do proxecto	9
2.1.1. Definición do alcance	9
2.1.2. Estrutura de Descomposición do Traballo	10
2.2. Metodoloxía	12
2.2.1. Aplicación da metodoloxía Scrum	12
2.3. Xestión do tempo	13
2.3.1. Planificación temporal	13
2.3.2. Sprint 0	15
2.3.3. Sprints 1, 2 e 3	16
2.3.4. Sprint release	16
2.4. Xestión do custo	16
2.4.1. Custo de recursos humanos	17
2.4.2. Custo de recursos materiais	17
2.4.3. Custo de recursos software	17
2.4.4. Presuposto	17
2.5. Xestión de riscos	18
2.5.1. Especificación de riscos	19
2.6. Xestión da configuración	21
2.6.1. Control do código fonte	22
2.6.2. Control dos entregables	22

3. Análise de requisitos	23
3.1. Casos de uso	23
3.2. Catálogo de requisitos	25
3.2.1. Requisitos funcionais	26
3.2.2. Requisitos non funcionais	29
3.3. Matriz de trazabilidade	31
4. Deseño do software	33
4.1. Arquitectura do sistema	33
4.1.1. Patrón de arquitectura	34
4.2. Interacción entre compoñentes	35
5. Desenvolvemento dos sprints	39
5.1. Sprint 1	39
5.2. Sprint 2	43
5.3. Sprint 3	45
6. Conclusións e traballo futuro	49
6.1. Conclusións	49
6.2. Traballo futuro	49
A. Manual técnico	51
A.1. Entorno de desenvolvemento	51
A.2. Ampliación do módulo de procesamento XML	55
B. Manual de usuario	59
B.1. Instalación	59
B.2. Crear capa de observacións	60
B.3. Gráficos en dúas dimensións	64
Bibliografía	67

Índice de figuras

1.1. Diagrama de secuencia dun consumidor SOS	6
1.2. Diagrama UML dunha Observation	7
2.1. Estrutura de Descomposición do Traballo	11
2.2. Ciclo da metodoloxía Scrum	13
2.3. Diagrama de Gantt	14
3.1. Diagrama de casos de uso	25
4.1. Diagrama de compoñentes	34
4.2. Diferencias entre MVC e Model/View	35
4.3. Diagrama de secuencia para os casos de uso CU.01 e CU.02	36
4.4. Diagrama de secuencia para o caso de uso CU.03	36
4.5. Diagrama de secuencia para o caso de uso CU.04	37
5.1. Diagrama de clases da compoñente SOS Client	41
5.2. Diagrama de clases da compoñente SOS Plot	46
A.1. Importación do repositorio en Eclipse, paso 1	52
A.2. Importación do repositorio en Eclipse, paso 2	52
A.3. Importación do repositorio en Eclipse, paso 3	53
A.4. Importación do repositorio en Eclipse, paso 4	53
A.5. Configuración de Python en Eclipse	54
A.6. Configuración do PYTHONPATH	54
A.7. Definición da clase XMLParser	56
B.1. Pantalla de instalación do plugin	59
B.2. Barra de ferramentas do plugin	60
B.3. Diálogo de conexión co servidor, sen datos	61
B.4. Diálogo de conexión co servidor, con datos	61
B.5. Ferramenta de selección espacial	63
B.6. Engadir capa SOS	63
B.7. Executar ferramenta SOS Plot	64
B.8. SOS Plot: Gráfica de varias series	65

Índice de cadros

2.1. Pila inicial do produto	15
2.2. Previsión de historias a incluír en cada sprint	16
2.3. Custos totais	18
2.4. Nivel de exposición dun risco en base a Probabilidade e Impacto	18
R.TEC.01 Complexidade do estándar SOS	19
R.TEC.02 Escaseza de servidores SOS	19
R.EXT.01 Soporte de SOS en QGIS	20
R.PER.01 Descoñecemento da ferramenta QGIS	20
R.PER.02 Descoñecemento da linguaxe Python	21
R.PER.03 Situación persoal dos membros do equipo	21
R.XES.01 Error na estimación temporal	21
CU.01 Consultar as capacidades dun servidor SOS	24
CU.02 Obter observacións do servidor SOS	24
CU.03 Visualizar gráficas en dúas dimensións	25
CU.04 Visualizar mapas animados	25
3.1. Niveis de relevancia dun requisito	26
RF.01 Conectar con servidor SOS	26
RF.02 Gardar lista de servidores SOS	27
RF.03 Visualizar XML das capacidades do servidor	27
RF.04 Xerar capa vectorial cas observacións do servidor	27
RF.05 Permitir filtrado básico das observacións a descargar	27
RF.06 Permitir filtrado avanzado das observacións a descargar	28
RF.07 Xerar petición de observacións manualmente	28
RF.08 Xerar gráfica propiedade <i>vs</i> tempo	28
RF.09 Xerar gráfica para enfrontar dúas propiedades	29
RF.10 Xerar gráfica con varias series	29
RF.11 Xerar animación no visor de mapas	29
RNF.01 Versión de SOS 1.0	30
RNF.02 Linguaxe de programación Python	30
RNF.03 Consistencia da interface gráfica	30
RNF.04 Manual de usuario	30
RNF.05 Internacionalización da aplicación	31
RNF.06 Data límite de execución do proxecto	31

3.2. Matriz de trazabilidade de requisitos	31
PR.01 Conexión con servidor SOS	42
PR.02 Xestión de servidores	42
PR.03 Descargar observacións	42
PR.04 Procesamento de XML con observacións en formato <i>Observations</i> e <i>Measurements</i>	44
PR.05 Xeración de filtros para <i>GetObservations</i>	44
PR.06 Xestionar capa co <i>TimeManager</i>	44
PR.07 Gráfica tempo <i>vs</i> propiedade	45
PR.08 Filtros espaciais	46
PR.09 Xeración de gráficas con varias series	47
PR.10 Personalización das propiedades das gráficas	47
B.1. Lapelas do formulario de consulta ó SOS	62

Capítulo 1

Introdución

1.1. Contextualización

Dende sempre o ser humano tivo inquietude por coñecer e entender como funciona o medio que o rodea, tarefa para a que resulta imprescindible a observación do mesmo. A observación defínese como a captación activa e rexistro de información dende unha fonte primaria, a través dos sentidos ou da utilización de instrumentos.

Dado o crecente interese polo estado do medio ambiente e os avances tecnolóxicos que se produciron nas últimas décadas, na actualidade xérase a cada instante unha inxente cantidade de datos que rexistra medicións dunha ampla variedade de fenómenos, empregando para elo os máis diversos métodos, dende modernos sensores montados en satélites ata o rexistro manual das medicións realizadas por un sinxelo termómetro de mercurio.

As maiores dificultades para converter este gran volume de datos en información útil son debidas a heteroxeneidade dos mesmos, tanto pola súa natureza física como polos medios empregados para capturalos e rexistralos.

Para atallar este problema xurdiu dende o Open Geospatial Consortium¹ (OGC) a iniciativa Sensor Web Enablement (SWE), que ten como obxectivo principal a estandarización de modelos, formatos e interfaces que teñan que ver coa xestión de información de observación medioambiental. De importancia para o presente traballo de fin de grado son os estándares Observations and Measurements (O&M) e Sensor Observation Service (SOS).

O estándar O&M define un modelo de datos e a súa codificación XML para a representación de observacións medioambientais. Cada observación modelada en O&M proporciona un valor observado para unha propiedade concreta (Observed Property) dunha entidade observada (Feature Of Interest - FOI). Por exemplo,

¹O OGC é un consorcio formado por máis de cincocentas empresas privadas, universidades e axencias gobernamentais co obxectivo de desenvolver e publicar estándares no eido da información xeoespacial (<http://www.opengeospatial.org/> e <https://www.youtube.com/watch?v=bfkCdir-y08>).

un valor real para a temperatura (Observed Property) dunha estación meteorolóxica (FOI). Outros metadatos asociados á observación son, por exemplo, o instante de tempo no que a observación se aplica á FOI (Phenomenon Time) e o proceso utilizado para xerar a observación. O proceso utilizado normalmente é a combinación de algún dispositivo físico (Sensor) con algún algoritmo de procesamento; por exemplo, para calcular a temperatura diaria na estación combínase un sensor de temperatura que ofrece medidas cada dez minutos con un algoritmo de agregación de datos.

O estándar SOS define a interface do servizo web que permite consultar observacións, metadatos dos sensores e a representación das entidades observadas. Amais define os medios para rexistrar novos sensores ou eliminar os existentes, así como para inserir novas observacións para os sensores rexistrados. Este estándar apoíase en outros, tamén desenvolvidos polo OGC, que definen os modelos para a representación dos datos a comunicar, así pois as xeometrías das entidades observadas represéntase a través do Geography Markup Language (GML), os metadatos dos sensores a través do SensorML e as observacións a través do xa citado Observations and Measurements (O&M).

As observacións de fenómenos medioambientais son, polo tanto, un tipo de información xeográfica; de feito para poder interpretar os datos de observacións de forma correcta, durante os procesos de apoio á toma de decisións, é fundamental dispoñer de metadatos que proporcionen o seu contexto temporal e espacial. Como se describiu anteriormente, o contexto temporal ven proporcionado polo Phenomenon Time, mentres que o contexto espacial ven dado polas propiedades de tipo xeográfico dispoñibles na FOI. Pódese abordar, polo tanto, a análise e explotación destes datos empregando como ferramenta un Sistema de Información Xeográfica (SIX) de propósito xeral.

Un SIX é un sistema deseñado para capturar, almacenar, manipular, analizar e presentar calquera tipo de información xeograficamente referenciada. Dentro do software SIX distínguense varios grandes grupos segundo o seu propósito, como poden ser sistemas de xestión de bases de datos espaciais, servidores cartográficos, visores web, móbiles ou de escritorio. Dentro deste último grupo, os SIX de escritorio, inclúense un amplo número de programas que permiten visualizar, editar e analizar datos xeográficos e que se empregan en moitos eidos e con moi diversos fins, por exemplo para investigacións científicas, meteoroloxía, cartografía, hidroloxía, xestión de recursos, marketing, loxística, avaliación do impacto ambiental, etc.

Os SIX traballan principalmente con dous formatos de información, segundo os obxectos do mundo real que se vaian representar sexan de natureza continua ou discreta. Así pois, para representar unha variable de natureza continua, como por exemplo un mapa da elevación do terreo, empregase o formato raster, que divide o espazo en celas regulares asociando un valor a cada unha delas. Pola contra, para representar obxectos de natureza discreta, como pode ser un río, emprégase xeralmente o formato vectorial, no que se representa a xeometría en base a pun-

tos, liñas ou polígonos, e se lle asocian a esta xeometría os atributos e valores que sexan necesarios. Cabe destacar que estes dous formatos poden converterse entre si en función da análise a realizar, por exemplo, en base ás observacións de temperaturas en determinados puntos pódese xerar por interpolación un raster que represente o mapa térmico da zona (rasterización), ou pola contra, a partir dun mapa de elevación pódense trazar as curvas de nivel que o representan (vectorización).

En canto ás ferramentas de análise proporcionadas polos paquetes SIX, existe unha moi ampla gama de técnicas que se desenvolveron principalmente no último medio século, e é un campo que cambia rapidamente na actualidade, incluíndose cada vez máis e máis ferramentas, ben proporcionadas polo propio provedor orixinal do software ou en moitos casos desenvolvidas por terceiros a través das posibilidades de ampliación que os propios programas SIX poñen a disposición dos desenvolvedores.

Na maioría dos SIX o modelado dos fenómenos xeográficos realizase de forma estática, sen embargo na actualidade requírense modelos que engadan o tempo como variable, de xeito que permitan observar o comportamento dinámico dos eventos xeográficos, como por exemplo no caso das observacións de fenómenos medioambientais. Os SIX que incorporan a representación do tempo, amais das dimensións espaciais, denomínanse SIX temporais e permiten facer unha representación da información en 4D, o que abre un amplo abano de posibilidades. A incorporación da compoñente temporal é un dos aspectos ó que máis esforzos se están a adicar actualmente, no que á evolución dos SIX se refire.

1.2. Motivación e Obxectivos

Na actualidade, a área de análise e explotación da información obtida a través das observacións medioambientais está moi fragmentada. Cada organismo ou axencia desenvolve as súas propias solucións para a captación, o almacenamento e a análise das cada vez máis amplas redes de sensores despregadas por todo o planeta. Este feito supón unha gran limitación en canto ó aproveitamento de toda esta información, o restrinxir o acceso á mesma as organizacións que dispoñen dos recursos necesarios para desenvolver e implantar os sistemas necesarios.

O estándar SOS pretende permitir a interoperabilidade entre os sistemas encargados de captar e almacenar esta información e os sistemas empregados para a análise da mesma, non obstante o grao de implantación deste estándar é aínda reducido. Como motivo, ou consecuencia, deste baixo grao de implantación existen na actualidade moi poucos sistemas de información xeográfica que inclúan o servizo SOS como unha fonte de datos soportada, a diferenza de outros estándares desenvolvidos polo OGC que son soportados pola ampla maioría dos sistemas, como o Web Map Service (WMS) ou o Web Feature Service (WFS).

As escasas solucións dispoñibles para operar con datos de observación servi-

dos pola interface SOS en sistemas de información xeográfica pasan polo uso de extensións de terceiros para software comercial, como por exemplo ArcGIS², que supón un elevado custe de licencia.

Este proxecto ven motivado, por tanto, pola inexistencia dun sistema de información xeográfica libre e de propósito xeral que permita a incorporación e análise de datos de observacións dispoñibles a través da interface SOS.

O obxectivo deste traballo é o desenvolvemento dunha extensión para a ferramenta SIX libre QGIS³ que permita a conexión a fontes de datos SOS e a exploración dos seus contidos no contorno de mapas proporcionado pola ferramenta.

1.3. Entorno tecnolóxico

1.3.1. QGIS

QGIS é un Sistema de Información Xeográfica de escritorio multiplataforma, libre e de código aberto. É un proxecto oficial da organización non gobernamental Open Source Geospatial Foundation⁴ (OSGeo) nacido no ano 2002 co obxectivo principal de proporcionar unha ferramenta que permitise visualizar, editar e analizar datos xeográficos en calquera ordenador persoal.

Amais de proporcionar de por si soporte para un amplo número de formatos de datos e funcionalidades para operar cos mesmos, QGIS proporciona unha API completa para implementación de *plugins* que amplíen a funcionalidade da ferramenta e os formatos de información xeográficos soportados. Os *plugins* poden programarse en C++ ou en Python, e no caso de cumprir a normativa e ser aprobados polos responsables poden incluírse no repositorio oficial da aplicación, de xeito que se podan instalar directamente dende o propio QGIS.

QGIS é na actualidade o SIX libre de referencia, está apoiado por un gran número de organizacións que patrocinan o proxecto, e que son, xunto coas doazóns puntuais de particulares e organizacións o método de financiamento do proxecto. Arredor do QGIS desenvolveuse unha ampla e activa comunidade tanto de usuarios como de desenvolvedores, do propio QGIS ou de *plugins*.

1.3.2. Sensor Observation Service

O estándar Sensor Observation Service⁵ define a interface do servizo web para descubrir e recuperar datos en tempo real ou arquivados producidos por calquera clase de sensor, tanto móbiles como estacionarios, in-situ ou remotos.

²<http://www.esri.es/es/productos/arcgis/>

³<http://www.qgis.org/>

⁴<http://www.osgeo.org/>

⁵<http://www.opengeospatial.org/standards/sos>

Os datos dos sensores poden ser tanto observacións realizadas como descricións dos sensores ou metadatos dos mesmos relativos a calibracións, mantementos, etc. As observacións devólvense codificadas segundo o estándar Observations and Measurements, e a información sobre sensores a través de SensorML. O SOS define, na súa versión 1.0, tres grupos de operacións:

Core: Estas operacións deben ser implementadas obrigatoriamente por todos os servizos SOS.

- *GetCapabilities*: Esta operación é a que proporciona a descrición das capacidades do SOS, ademais de información xeral sobre o propio servizo. Entre outros datos indica as versións de SOS soportadas, os datos identificativos do servizo e do provedor do mesmo, as operacións que implementa (cos parámetros soportados por cada unha delas), os filtros soportados e a lista de ofertas provistas polo servizo, con información detallada de cada unha delas.
- *DescribeSensor*: Proporciona información relativa ó sensor requirido, segundo o estándar Sensor Model Language (SensorML). Entre outra información indícase a posición, nome e descrición do sensor ou sistema, o instante temporal da instalación e se proporciona medidas válidas, os fenómenos que pode medir e as unidades nas que os mide, e outra información relacionada, como un breve historial de reparacións e calibracións, etc
- *GetObservations*: É a operación encargada de proporcionar os datos medidos polos sensores dunha determinada oferta e que cumpran os filtros definidos na consulta. Os valores destas observacións realizadas polos sensores devólvense segundo o estándar Observations and Measurements (O&M). Este estándar define un formato de XML no que cada observación se describe polo fenómeno que mide, o sensor que o mide, o instante temporal da medida, a posición e nome do lugar no que se toma a medida, e a unidade na que se mide o valor correspondente.

Transactional: Son as operacións que permiten rexistrar información no servizo. A súa implementación é opcional.

- *RegisterSensor*: Permite rexistrar un novo sensor no servizo, mandando a súa descrición no formato SensorML.
- *InsertObservation*: Permite ó cliente inserir novas observacións no sistema de sensores previamente rexistrados.

Extended: Son operacións adicionais opcionais, para requirir información ó servizo.

A figura 1.1 representa a interacción entre un consumidor e os servizos SOS que interroga.

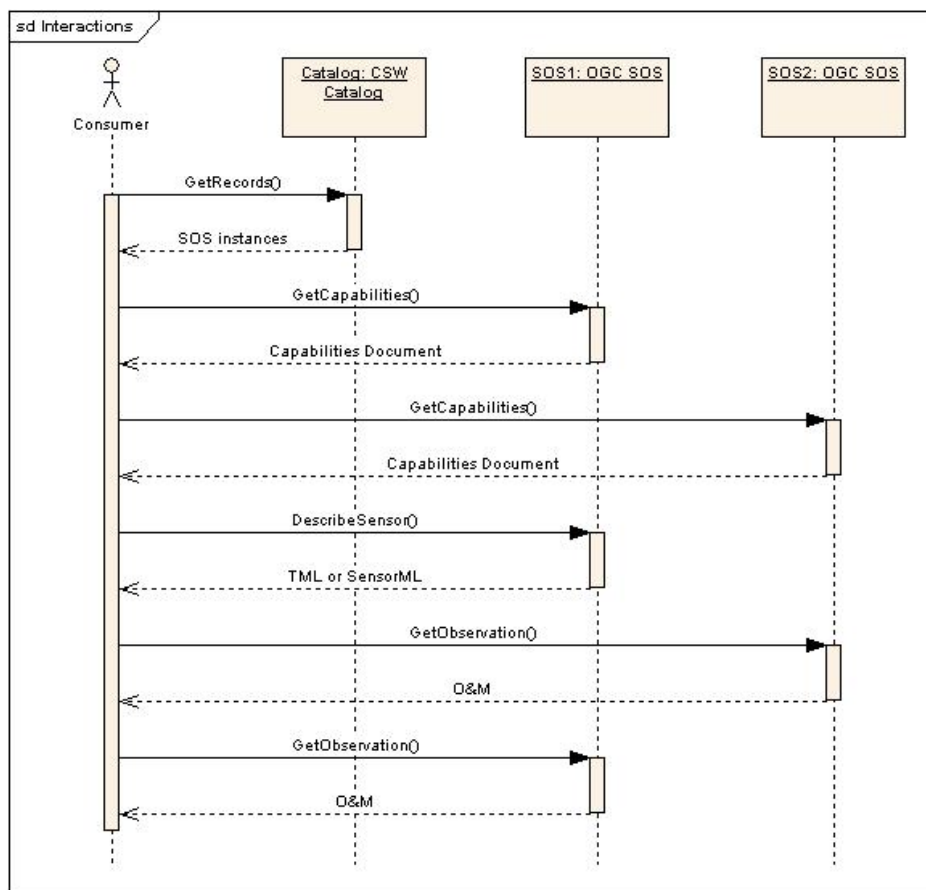


Figura 1.1: Diagrama de secuencia dun consumidor SOS

Fonte: http://www.ogcnetwork.net/SOS_Intro

1.3.3. Observations and Measurements

O estándar Observations and Measurements⁶ define o modelo de datos e a súa codificación XML para a representación de observacións medioambientais. Define polo tanto modelos de documentos para o intercambio de información, tanto de descrición do proceso de observación coma os seus resultados, entre diferentes comunidades técnicas e científicas.

Constitúe unha dependencia esencial para o estándar SOS.

O modelo de datos proposto é o que se mostra na figura 1.2.

⁶<http://www.opengeospatial.org/standards/om>

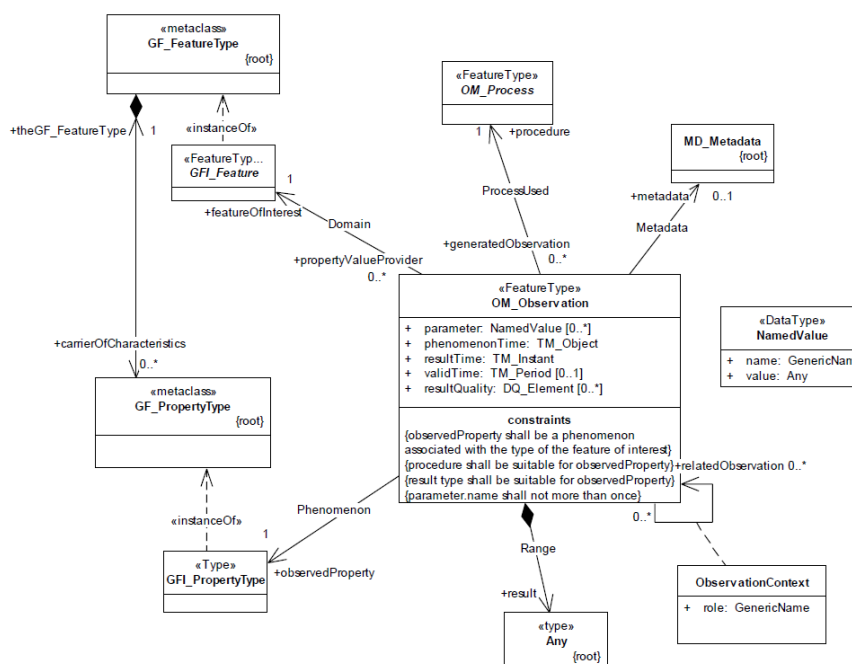


Figura 1.2: Diagrama UML dunha Observation

Fonte: <http://www.openqgis.net/doc/IS/om-eo-metadata>

Segundo o modelo, unha observación é un evento que estima unha propiedade observada (Property) dalgunha entidade de interese (Feature Of Interest), utilizando un procedemento especificado (Process) e xera un resultado (Result).

Feature Of Interest: Representa a entidade de interese ou área observada. Por exemplo, unha estación meteorolóxica.

Property: Representa o fenómeno físico observado. Existen varios tipos, sendo o máis habitual o *Measurement*, que é a combinación de un valor numérico real con unha unidade de medida. Por exemplo, a temperatura.

Proces: Describe o proceso empregado para xerar o resultado. Pode ser un sensor, un conxunto de sensores, un algoritmo, etc. Por exemplo, un termómetro; ou ben, un algoritmo de agregación que calcule temperaturas medias, mínimas e máximas para un día.

Result: É o valor para a propiedade observada na entidade de interese, obtido aplicando o procedemento especificado. Por exemplo, o rexistro de temperatura media para unha estación meteorolóxica nun día concreto.

1.4. Estrutura da memoria

Este documento estrutúrase en 6 capítulos, dous apéndices e a bibliografía empregada.

Neste capítulo de introdución descríbese a motivación e obxectivos do proxecto e contextualízase o mesmo describindo brevemente os conceptos empregados e a situación actual da técnica nas áreas de coñecemento relacionadas.

No capítulo 2 abórdanse os aspectos relativos á xestión do proxecto: a definición do alcance, a metodoloxía, planificación temporal e xestión de riscos e da configuración.

No capítulo 3 detállase a análise do software, describindo os casos de uso identificados e os requisitos extraídos dos mesmos.

No capítulo 4 descríbese o deseño do software, tanto a arquitectura do mesmo como o comportamento de cada unha das compoñentes para satisfacer os casos de uso definidos.

No capítulo 5 documéntase, en orde cronolóxico, o proceso de deseño e implementación, así como as probas realizadas para cumprir e validar os requisitos establecidos.

No capítulo 6 expóñense as conclusións extraídas da realización do proxecto e detállanse varias propostas de traballo futuro.

O apéndice A é unha guía para a configuración do entorno de desenvolvemento e para engadir novas capacidades ó módulo de procesamento de XML.

O apéndice B é unha guía para a instalación e manexo básico do *plugin* desenvolvido.

Capítulo 2

Xestión do proxecto

A xestión do proxecto ten como finalidade definir e alcanzar os obxectivos do mesmo, ó tempo que se optimiza o uso de recursos, tanto humanos como materiais, co obxectivo último de lograr un proxecto e un produto de calidade no tempo e custe establecido.

Na sección III do PMBOK^[3] descríbense amplamente os distintos tipos de procesos que se inclúen dentro da xestión do proxecto, non obstante, para cada proxecto concreto será necesario seleccionar os máis apropiados para cumprir co obxectivo do mesmo. Nas seguintes seccións detállanse os procesos que se levaron a cabo para a xestión do presente proxecto.

2.1. Alcance do proxecto

Os procesos de xestión do alcance do proxecto son os encargados de asegurar que o proxecto inclúa o traballo requirido, e so o requirido, para completar o proxecto de forma satisfactoria.

2.1.1. Definición do alcance

O obxectivo deste proxecto é dotar ó Sistema de Información Xeográfica QGIS da capacidade de consultar fontes de datos de observacións a través da interface SOS e representar os datos obtidos no contorno de mapas proporcionado pola ferramenta, de xeito que podan ser explorados e analizados polo usuario. Para acadar este obxectivo desenvolverase un *plugin* para o programa QGIS, coas seguintes funcionalidades básicas:

- Conectarse a un servidor SOS e obter as capacidades do servizo a través da operación GetCapabilities.
- En base ás capacidades do servidor, permitir xerar unha petición de observacións, de forma sinxela e sen necesidade de coñecementos técnicos do SOS.

- Permitir modificar a petición xerada manualmente se o usuario o desexa.
- Obter as observacións a través da operación `GetObservations`.
- Coas observacións descargadas xerar unha capa vectorial que conteña a información xeográfica, temporal e os valores das propiedades observadas.
- Xerar gráficos en dúas dimensións cos datos da capa, tanto para ver a evolución dunha propiedade con respecto ó tempo como dunha propiedade con respecto a outra.
- Permitir visualizar a capa co *plugin* *TimeManager* de xeito que se podan facer animacións.

Criterios de aceptación

A aceptación do produto está condicionada a que o *plugin* satisfaga as funcionalidades descritas no apartado do alcance do proxecto, sempre e cando ó longo do desenvolvemento do proxecto non se amose que son irrealizables nas 420 horas de traballo das que consta o Traballo de Fin de Grao. En caso de que os obxectivos iniciais resultasen demasiado ambiciosos redefiniranse os mesmos de acordo cos titores.

Entregables do proxecto

Entregaranse ó cliente para a súa validación os distintos incrementos resultado de cada ciclo de desenvolvemento, que consistirá no *plugin* empaquetado en formato instalable para o QGIS.

Ó remate do proxecto entregarase, ademais do produto software, o código fonte correspondente, a memoria do traballo realizado e demais documentación necesaria segundo o regulamento do Traballo de Fin de Grao en Enxeñería Informática da Universidade de Santiago de Compostela^[16].

2.1.2. Estrutura de Descomposición do Traballo

A Estrutura de Descomposición do Traballo (EDT) axuda a dividir o proxecto en paquetes de traballo de forma xerárquica. Dado que se empregará unha metodoloxía áxil para a xestión do proxecto (Sección 2.2) non é necesario detallar a EDT máis do que xa se fixo no anteproxecto, que se representa na figura 2.1.

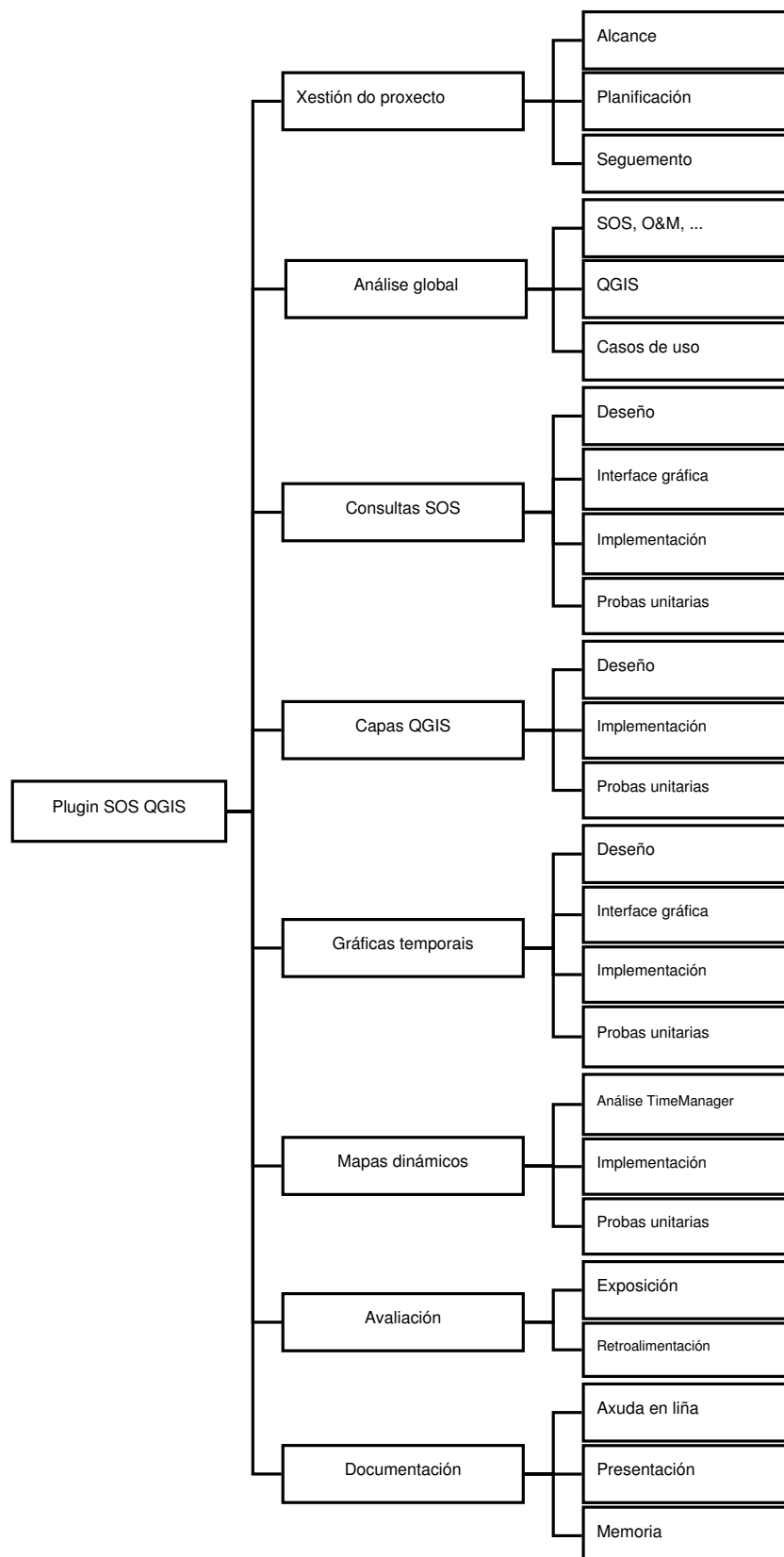


Figura 2.1: Estructura de Descomposición do Trabalho

2.2. Metodoloxía

Unha das primeiras decisións a tomar á hora de iniciar un proxecto é a metodoloxía a seguir para levalo a cabo. A metodoloxía define o ciclo de vida do proxecto, e por tanto, as fases que conectan o inicio co fin do mesmo. É moi importante polo tanto que a metodoloxía elixida se adapte á natureza do proxecto, dos produtos e dos demais aspectos relacionados.

Existen dous tipos xerais de metodoloxías, as predictivas, que consisten nunha planificación inicial ríxida a seguir durante todo o ciclo de vida do proxecto, e as áxiles, que asumen que existirán cambios o longo do ciclo de vida do proxecto polo que son máis flexibles para axilizar o desenvolvemento e a capacidade de adaptación ós cambios. Dada a nula experiencia previa nas tecnoloxías a empregar e na área de coñecemento do proxecto o uso dunha metodoloxía predictiva non é aconsellable. Entre as metodoloxías áxiles seleccionase *Scrum*^[10], por ser unha das máis utilizadas e porque o equipo de desenvolvemento xa está familiarizado con ela.

2.2.1. Aplicación da metodoloxía Scrum

Só se aplicarán os conceptos da metodoloxía de xestión de proxectos *Scrum* que resulten beneficiosos para este traballo concreto, pois debido as particularidades que presenta por realizarse no entorno dun Traballo de Fin de Grao, algúns dos conceptos son dificilmente aplicables ou inútiles. Así pois, a continuación descríbense brevemente os conceptos dos que se fará uso:

Historia de usuario ou *User history*: Representación dun requisito funcional do software mediante unha breve descrición textual. Debe ser o suficientemente sinxela como para poder estimar o tempo necesario para completala.

Pila do produto ou *Product Backlog*: Lista ordenada de historias de usuario que compoñen o proxecto.

Sprint: É o período de tempo durante o cal se leva a cabo o traballo, xeralmente de 2 a 4 semanas. Cada sprint produce un incremento software, é dicir, unha nova versión potencialmente entregable.

Pila do sprint ou *Sprint Backlog*: É o subconxunto de historias de usuario que serán acometidas nun sprint. Xeralmente as historias divídense en tarefas cando pasan á pila do sprint. Esta pila, así como os requisitos incluídos na mesma non poden modificarse mentres dure o sprint.

Na figura 2.2 móstrase graficamente o ciclo de vida da metodoloxía *Scrum*.

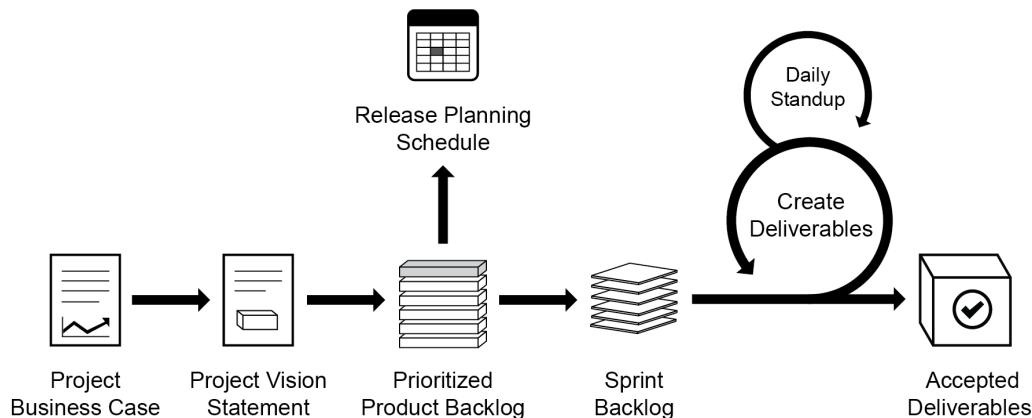


Figura 2.2: Ciclo da metodoloxía Scrum

Fonte: https://commons.wikimedia.org/wiki/File:Scrum_Flow_for_one_Sprint.png

2.3. Xestión do tempo

A xestión do tempo do proxecto inclúe os procesos necesarios para lograr a conclusión do proxecto a tempo.

2.3.1. Planificación temporal

Tendo en conta que na metodoloxía *Scrum* non se planifica inicialmente a duración nin o contido de cada sprint debemos considerar esta planificación como unha guía a seguir para organizar o traballo cronoloxicamente, non como unha planificación estrita e vinculante para o desenvolvemento do proxecto.

Para levar a cabo a planificación do proxecto faise unha revisión xeral das distintas fases, tomando como base a Estrutura de Descomposición do Traballo da figura 2.1. Existen diferencias significativas entre a *EDT* e o diagrama de *Gantt* (Figura 2.3) debido a que no primeiro móstranse os paquetes de traballo necesarios para acadar os obxectivos do proxecto, mentres que no segundo se organizan cronoloxicamente as tarefas necesarias para completar ditos paquetes de traballo.

En canto ás estimacións temporais, considérase unha xornada semanal de 35 horas, que, por simplicidade, se visualiza no *Gantt* como 5 horas diarias os 7 días a semana. En realidade esta dedicación será variable diariamente dependendo das obrigas laborais dos membros do equipo pero co compromiso de cumprir a planificación semanalmente.

Planifícase o desenvolvemento do proxecto en cinco sprints, o *Sprint 0* inicial, tres sprints normais de 70 horas e o *Sprint release* final.

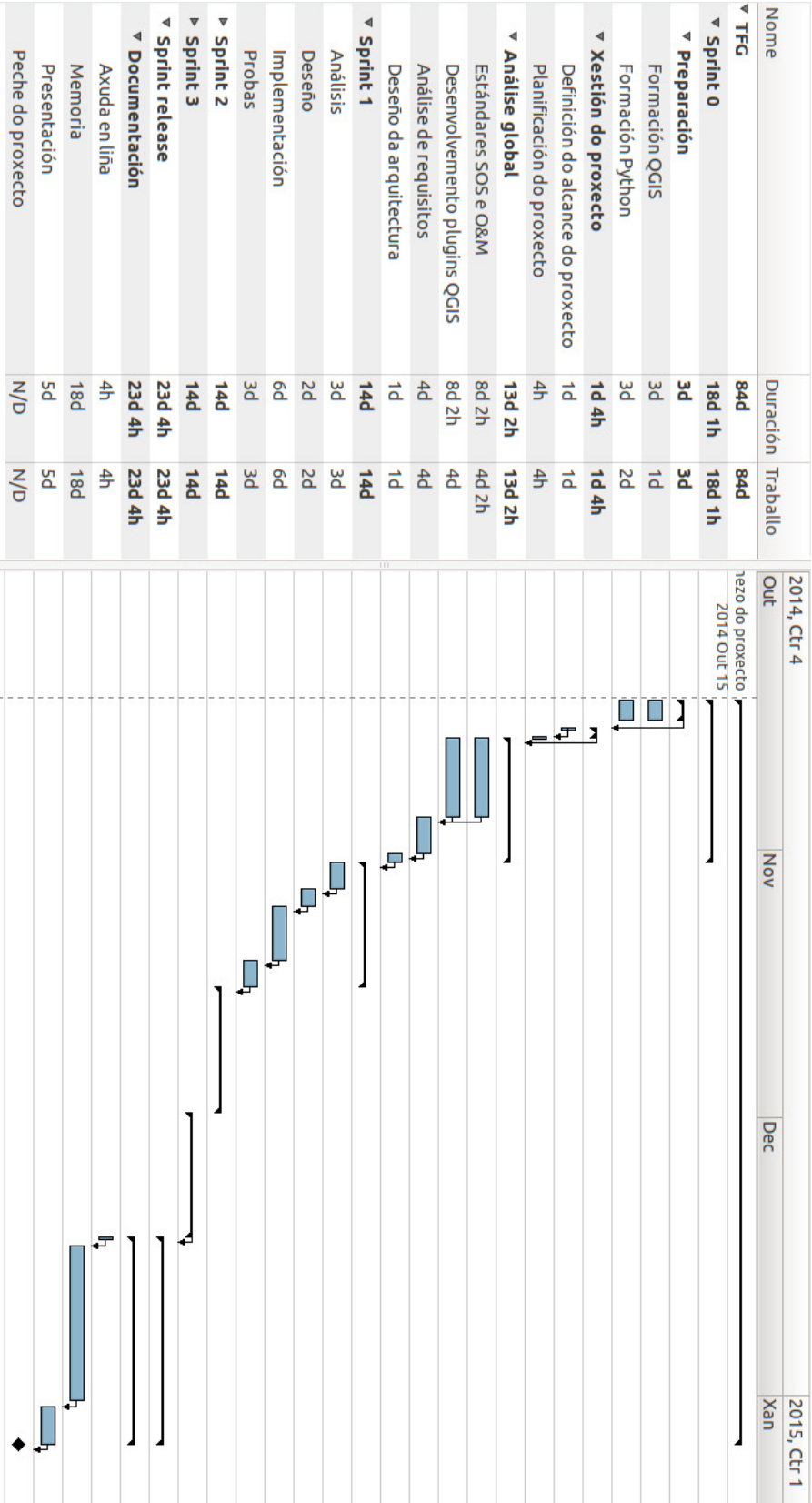


Figura 2.3: Diagrama de Gantt

#	Historias de usuario	Horas
1	Conexión con servidor SOS	25
2	Procesamento do xml de capacidades do servizo	30
3	Xeración de peticións básicas	15
4	Xeración de peticións con filtros complexos	20
5	Creación de peticións personalizadas	5
6	Xeración dunha capa vectorial a partir dos datos de observacións	35
7	Xeración dunha capa válida para o <i>plugin TimeManager</i>	10
8	Visualización gráfica de propiedades respecto ó tempo	20
9	Visualización gráfica de unha propiedade con respecto a outra	25
10	Visualización de varias series de observacións simultaneamente	25

Cadro 2.1: Pila inicial do produto

2.3.2. Sprint 0

Este sprint é o inicial do proxecto. É un sprint especial, pois non xerará ningún incremento, senón que se levarán a cabo os distintos procesos de iniciación do proxecto.

Preparación

15 horas

Capacitación do equipos nas tecnoloxías empregadas no proxecto e iniciación na area de coñecemento dos Sistemas de Información Xeográfica. Tamén se configura o entorno de desenvolvemento integrado *Eclipse* co *plugin PyDev*.

Xestión do proxecto

9 horas

Definición do alcance do proxecto, metodoloxía, xestión de custos, riscos e da configuración, e planificación do mesmo. Para realizar esta planificación é necesario crear a versión inicial da pila do produto (cadro 2.1).

Análise global

69 horas

Realízase o estudo dos estándares SOS e O&M, e realízase o estudo da API de QGIS^[2] para a programación de *plugins* en *Python* e demais documentación dispoñible^[1].

Tamén se leva a cabo nesta fase a análise de casos de uso e de requisitos, así como o deseño preliminar da arquitectura do software a desenvolver.

#	Historias de usuario	Horas
Sprint 1		
1	Conexión con servidor SOS	25
2	Procesamento do xml de capacidades do servizo	30
3	Xeración de peticións básicas	15
Sprint 2		
5	Creación de peticións personalizadas	5
6	Xeración dunha capa vectorial a partir dos datos de observacións	35
7	Xeración dunha capa válida para o <i>plugin TimeManager</i>	10
8	Visualización gráfica de propiedades respecto ó tempo	20
Sprint 3		
4	Xeración de peticións con filtros complexos	20
9	Visualización gráfica de unha propiedade con respecto a outra	25
10	Visualización de varias series de observacións simultaneamente	25

Cadro 2.2: Previsión de historias a incluír en cada sprint

2.3.3. Sprints 1, 2 e 3

Estes 3 sprints forman a fase de desenvolvemento do proxecto, durante a que se codifican os distintos compoñentes que conforman o sistema. Cada un dos sprints ten unha duración de 70 horas de horas de traballo (2 semanas).

En cada iteración realizaranse as etapas de análise, deseño, implementación e probas para as historias seleccionadas. A previsión de historias a incluír en cada sprint é a que se mostra no cadro 2.2. É importante destacar que esta división das tarefas nos distintos sprints é preliminar e susceptible de ser revisada e modificada o longo das distintas planificacións do sprint que se realizan durante a execución do proxecto, cando o coñecemento do problema a resolver sexa máis profundo e polo tanto a estimación máis precisa.

2.3.4. Sprint release

119 horas

Este sprint tamén é especial, pois está adicado a levar a cabo os procesos de documentación e peche do proxecto. As tarefas a desenvolver son a redacción da axuda, da memoria e da presentación para a exposición do proxecto, e a preparación de todos os entregables necesarios (ver páxina 10).

2.4. Xestión do custo

Debido a que este proxecto é un Traballo de Fin de Grao os custos manexados son teóricos e non se considerarán os custos indirectos (electricidade, internet e

similares) ou gastos de desprazamento. A xestión de custos faise co único obxectivo de dar unha valoración económica realista do traballo realizado polo que se contemplarán os recursos humanos, os recursos materiais e os recursos software necesarios para a execución do proxecto.

2.4.1. Custo de recursos humanos

Se consideramos, a nivel de custos, ós titores do proxecto como clientes, o equipo de desenvolvemento para a realización do proxecto consta dunha soa persoa, que realizará as distintas tarefas de análise, programación e documentación do mesmo.

Consideraremos como salario bruto anual para o recurso 24.000 €, que é segundo [tecnoempleo.com](#)^[14] o salario bruto medio para un analista/programador. Ó salario bruto débense engadir os custos do mesmo para a empresa, que tomando como referencia os datos da Seguridade Social^[13] suporía un 29,9 % do mesmo. Para o cálculo do custo por hora considerase a xornada máxima indicada no Convenio Colectivo^[6], que son 1.800 horas de traballo ó ano.

O custo de recursos humanos é por tanto de 17,32 €/hora.

2.4.2. Custo de recursos materiais

Para o desenvolvemento do proxecto é necesario un ordenador capaz de executar QGIS e o entorno de desenvolvemento integrado Eclipse. QGIS non especifica formalmente uns requirimentos mínimos e é capaz de funcionar de xeito fluído nun ordenador de gama media que se pode adquirir por uns 600 €. Considerando unha porcentaxe de amortización anual do 25 %, como indica a Lei 27/2014^[7], pódense imputar como custes 12,50 €/mes. A estes efectos deben computarse os meses naturais de duración do proxecto.

Os materiais funxibles necesarios para a realización do proxecto e os gastos de impresión e CDs para a presentación do mesmo supoñen un custe de 140 €.

2.4.3. Custo de recursos software

Todas as ferramentas software empregadas para a realización deste proxecto son de uso gratuíto.

2.4.4. Presuposto

No cadro 2.3 amosase o resumo dos custos do proxecto, que suman un total de 7489,40 €.

Concepto	Cantidade	Custo unitario	Total
Custos de persoal	420 horas	17,32 €	7274,40 €
Amortización do ordenador	6 meses	12,50 €	75,00 €
Custos doutros materiais	1	140,00 €	140,00 €
Total			7489,40 €

Cadro 2.3: Custos totais

2.5. Xestión de riscos

A xestión de riscos ten como finalidade aumentar a probabilidade e o impacto de eventos positivos e diminuír a probabilidade e impacto dos eventos adversos para o proxecto. Implica, polo tanto, prever e xestionar os eventos que poden influír na planificación temporal, no esforzo ou no custo do proxecto, ou na calidade do produto, e tomar as accións necesarias para evitalos ou minimizar o seu impacto. Os catro pasos básicos a seguir para levar a cabo a xestión de riscos son:

- Identificación
- Análise e catalogación
- Planificación da resposta
- Seguimento e control

Para catalogar os riscos en base a súa relevancia empréganse tres medidores: a probabilidade de que ocorra, o impacto que supón sobre o tempo ou esforzo, e o nivel de exposición, que é unha combinación dos valores de probabilidade e impacto. Os distintos valores para estes medidores recóllense no cadro 2.4.

		Probabilidade		
		Case seguro ≥80 %	Moi probable <80 % e >30 %	Pouco probable ≤30 %
Impacto	Alto ≥20 %	Alto	Alto	Medio
	Medio <20 % e >10 %	Alto	Medio	Baixo
	Baixo ≤10 %	Medio	Baixo	Baixo

Cadro 2.4: Nivel de exposición dun risco en base a Probabilidade e Impacto

2.5.1. Especificación de riscos

Co obxectivo de non engadir complexidade ó seguimento do proxecto so se planifican os riscos adversos con un nivel de exposición alto, é dicir, os riscos que é bastante probable que ocorran e que teñen un impacto negativo considerable sobre o desenvolvemento do proxecto. Para desenvolver respostas efectivas ós riscos é de moita utilidade agrupalos por causas comúns, neste caso clasificándoos en base á fonte do risco.

Riscos técnicos

R.TEC.01.- COMPLEXIDADE DO ESTÁNDAR SOS

Descrición	O nivel de madurez e uso do estándar SOS fai que sexa un estándar extremadamente amplo e con un alto grao de liberdade. Non existe ningunha implementación que cubra o estándar na súa totalidade.
Probabilidade	Moi probable
Impacto	Alto
Nivel de exposición	Alto
Resposta ao risco	Só se comprometerá como imprescindible soportar as operacións básicas para obter os datos de observacións provistos pola implementación SOS realizada polo CiTIUS ¹ .

R.TEC.02.- ESCASEZA DE SERVIDORES SOS

Descrición	Existen moi poucos servidores que implementen SOS abertos ó público cos que poder validar a implementación realizada.
Probabilidade	Case seguro
Impacto	Medio
Nivel de exposición	Alto
Resposta ao risco	Instalar en local un servidor SOS para poder probar a aplicación, e solicitar acceso ós xestionados polo CiTIUS.

¹<https://citius.usc.es/>

Riscos externos**R.EXT.01.- SOPORTE DE SOS EN QGIS**

Descrición	Implementación nativa dentro do QGIS do soporte para SOS ou a publicación de algún plugin que implemente dito soporte.
Probabilidade	Moi probable
Impacto	Alto
Nivel de exposición	Alto
Resposta ao risco	Reorientar o proxecto para centralo en dotar o QGIS de ferramentas específicas para interacción cos datos de sensores, sempre e cando a solución SOS implantada de soporte completo á implementación realizada polo CiTIUS.

Riscos de persoal**R.PER.01.- DESCOÑECEMENTO DA FERRAMENTA QGIS**

Descrición	O equipo de desenvolvemento non ten experiencia no manexo da ferramenta QGIS nin outras ferramentas de información xeográfica.
Probabilidade	Case seguro
Impacto	Medio
Nivel de exposición	Alto
Resposta ao risco	No plan de traballo do proxecto inclúese a capacitación na ferramenta QGIS, así como nos conceptos básicos sobre sistemas de información xeográfica.

R.PER.02.- DESCOÑECEMENTO DA LINGUAXE PYTHON

Descrición	O equipo de desenvolvemento non ten experiencia na linguaxe de programación Python.
Probabilidade	Case seguro
Impacto	Alto
Nivel de exposición	Alto

Resposta ao risco	No plan de traballo do proxecto inclúese a capacitación na linguaxe de programación Python.
--------------------------	---

R.PER.03.- SITUACIÓN PERSOAL DOS MEMBROS DO EQUIPO

Descrición	Cambios na situación persoal ou laboral dos membros do equipo que impidan levar a cabo a dedicación estimada.
Probabilidade	Case seguro
Impacto	Alto
Nivel de exposición	Alto
Resposta ao risco	Volver a planificar os prazos de execución do proxecto.

Riscos na xestión do proxecto

R.XES.01.- ERROR NA ESTIMACIÓN TEMPORAL

Descrición	A estimación temporal inicial para a execución das tarefas é pouco precisa a causa da inexperiencia en proxectos similares.
Probabilidade	Case seguro
Impacto	Medio
Nivel de exposición	Alto
Resposta ao risco	A estimación inicial realizase de forma pesimista. A metodoloxía <i>Scrum</i> minimiza o impacto ó permitir o refinamento de requisitos e estimacións ó longo do proxecto.

2.6. Xestión da configuración

A xestión da configuración^[4] ten como propósito establecer e manter a integridade dos elementos de traballo. No proxecto que nos ocupa os elementos de traballo a xestionar son o código fonte do *plugin*, o código fonte da documentación, e os distintos entregables xerados.

2.6.1. Control do código fonte

Dentro do código fonte a xestionar inclúese tanto o código da aplicación, imaxes, documentos de axuda e demais que se inclúen no *plugin* empaquetado, como o código fonte para xerar a documentación.

En ambos casos o procedemento de xestión da configuración se fai a través do sistema de xestión de versións *Git*. Mantéñense en local dous repositorios diferentes, un para aplicación e outro para documentación, cada un coa súa réplica remota correspondente aloxada en *GitHub*². As carpetas dos repositorios locais replícanse en *Dropbox*³ co obxectivo de ter unha copia redundante na nube e acceso desde distintos ordenadores en todo momento.

O método de traballo consiste en realizar un *Commit* local cada vez que se remata unha tarefa e realizar un *Push* ó servidor remoto cada vez que se remata unha versión susceptible de ser publicada. Deste xeito no repositorio público sempre hai unha versión estable do produto.

2.6.2. Control dos entregables

Os distintos obxectos entregables xerados o longo do proxecto, dende o anteproxecto ata a presentación do mesmo son arquivados tanto en local como na carpeta de *Dropbox*. Non se realiza máis control de versións sobre os produtos entregables que o propio control realizado por *Dropbox*. Esta ferramenta permite o acceso ás distintas versións gardadas de calquera tipo de ficheiro.

O servizo *Dropbox* tamén permite a creación de carpetas públicas. Esta funcionalidade permítenos crear un repositorio público para QGIS no que aloxar o *plugin* empaquetado de xeito que sexa moi sinxelo de instalar.

²<https://github.com/>

³<https://www.dropbox.com/>

Capítulo 3

Análise de requisitos

A análise dos requisitos dun produto software é unha parte fundamental do proceso de desenvolvemento, pois consiste en definir de forma detallada o que debe facer o produto para satisfacer as necesidades e cumprir as expectativas do cliente.

O proceso de xestión de requisitos^[5] consistirá en definir conxuntamente co cliente os distintos casos de uso do software. A partir dos casos de uso identificados extraírense e documéntanse os requisitos, que se usarán como referencia para o desenvolvemento do sistema.

3.1. Casos de uso

Os casos de uso^[8] describen as interaccións entre os actores e o sistema nunha linguaxe non técnica, centrándose en que fai o sistema para o actor, entendendo por actor calquera entidade externa ó sistema que lle demande algunha funcionalidade.

Neste proxecto identifícase un único actor, que é o usuario xenérico da aplicación, pois nin existen distintos roles de usuarios nin outras entidades externas.

A continuación descríbense os casos de uso identificados en colaboración cos directores do proxecto, que exercen o rol de cliente, e que se resumen graficamente na figura 3.1.

CU.01.- CONSULTAR AS CAPACIDADES DUN SERVIDOR SOS

Propósito	Consultar as capacidades dun servidor SOS para ver as ofertas que realiza e as propiedades dispoñibles nelas, e demais información do servizo.
Actores	Usuario
Relacións	É estendido polo CU.02
Precondicións	-

Poscondicións	Visualízanse as capacidades do SOS nun formato amigable para o usuario.
Escenario	O usuario introduce o enderezo do servidor SOS, ou selecciónao de entre os gardados. A aplicación descarga as capacidades do mesmo para amosalas en pantalla.

CU.02.- OBTENIR OBSERVACIÓNS DO SERVIDOR SOS

Propósito	Xerar unha capa vectorial cos datos das observacións rexistradas no SOS.
Actores	Usuario
Relacións	Estende o CU.01
Precondicións	Ter consultado as capacidades do servizo.
Poscondicións	Capa vectorial coas observacións obtidas.
Escenario	O usuario selecciona a través da interface gráfica a oferta e propiedades a consultar, así como outros filtros máis avanzados, por entidade de interese, sensor, período de tempo, filtros espaciais, etc. A aplicación descarga as observacións e xera unha capa vectorial en QGIS coa información procesada.

CU.03.- VISUALIZAR GRÁFICAS EN DÚAS DIMENSIÓNS

Propósito	Visualizar nunha gráfica en dúas dimensións a información das observacións, a partir da capa xerada. Débese poder visualizar a evolución dunha propiedade con respecto ó tempo, e tamén sería de utilidade en algúns casos poder enfrontar unha propiedade contra outra.
Actores	Usuario
Relacións	-
Precondicións	Ter xerada unha capa con datos de observacións.
Poscondicións	Gráfico en dúas dimensións dos datos seleccionados.
Escenario	Sobre a capa xerada o usuario seleccionará entidades de interese para poder visualizar nunha nova ventá un gráfico que represente o tempo no eixo X e a propiedade indicada no eixo Y. O usuario pode seleccionar outra propiedade para representar en cada un dos eixos.

CU.04.- VISUALIZAR MAPAS ANIMADOS

Propósito	Facer visualizacións animadas dos datos da capa dentro do propio visor de mapas de QGIS.
Actores	Usuario
Relacións	-
Precondicións	Ter xerada unha capa con datos de observacións.
Poscondicións	Visualización animada das observacións.
Escenario	O usuario disporá duns controis de reprodución para as capas con datos de observacións de xeito que poda reproducir a evolución cronolóxica das observacións sobre o propio visor de mapas de QGIS.

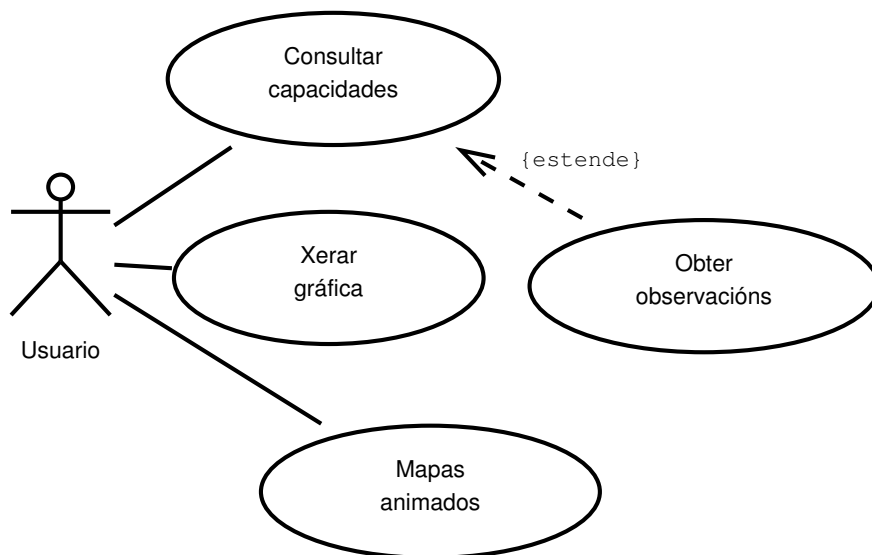


Figura 3.1: Diagrama de casos de uso

3.2. Catálogo de requisitos

A partir dos casos de uso documentados realizase a extracción dos requisitos da aplicación. Estes requisitos son as características que debe presentar a aplicación, e definen o software desde o punto de vista do cliente. Dividimos o catálogo de requisitos en dúas partes, segundo a natureza dos mesmos:

Funcionais: Son as características da aplicación para proporcionar a funcionalidade requirida.

Relevancia	Descrición
Esixido	Requisito que está directamente relacionado co cumprimento dos obxectivos do proxecto.
Desexado	Requisito que mellora a calidade do proxecto, pero non está relacionado directamente cos obxectivos do mesmo.
Opcional	Requisito que non supón unha mellora substancial na calidade do proxecto, aínda que si resulta de utilidade.

Cadro 3.1: Niveis de relevancia dun requisito

Non funcionais: Non describen funcionalidades, se non que representan outras necesidades como seguridade, rendemento, usabilidade, ou restricións de plataforma, tecnolóxicas, e similares.

Clasificaremos os requisitos pola súa relevancia segundo o cadro 3.1, que se terá en conta na planificación dos sprints.

3.2.1. Requisitos funcionais

Os requisitos funcionais definidos, verificados e revisados para o presente proxecto son os que seguen:

RF.01.- CONECTAR CON SERVIDOR SOS

Descrición	O usuario pode introducir o enderezo dun servidor que proporcione o servizo SOS e obter do mesmo as súas capacidades. Estas capacidades deben visualizarse nun formato amigable para o usuario.
Relevancia	Esixido
Criterio de validación	O requisito cumprirase se a aplicación é capaz de conectarse cos servidores indicados e procesar o resultado da operación <i>GetCapabilities</i> para visualizala nun formato lexible.

RF.02.- GARDAR LISTA DE SERVIDORES SOS

Descrición	O usuario poderá xestionar unha lista de servidores SOS para os que se indicará o enderezo e un nome para identificalo.
Relevancia	Opcional

Criterio de validación	O requisito cumprirase se a aplicación permite gardar unha lista de servidores e facer o mantemento da mesma.
-------------------------------	---

RF.03.- VISUALIZAR XML DAS CAPACIDADES DO SERVIDOR

Descrición	O usuario poderá visualizar en formato texto, con resaltado de sintaxe, e nun visor de árbore o XML que describe as capacidades do servidor.
Relevancia	Desexado
Criterio de validación	O requisito cumprirase se a aplicación dispón dun visor de XML con resaltado de sintaxe e visor de árbore no que se amose o ficheiro resposta do servidor á petición <i>GetCapabilities</i> .

RF.04.- XERAR CAPA VECTORIAL CAS OBSERVACIÓNS DO SERVIDOR

Descrición	Unha vez consultadas as capacidades do servidor, o usuario poderá seleccionar unha oferta e unha ou varias das propiedades da mesma para descargar os datos e xerar unha capa vectorial coas observacións descargadas.
Relevancia	Esixido
Criterio de validación	O requisito cumprirase se a aplicación descarga os datos solicitados do servidor e procesa o XML para crear un rexistro na capa vectorial para cada observación, cos datos de xeometría, tempo e o valor de cada propiedade.

RF.05.- PERMITIR FILTRADO BÁSICO DAS OBSERVACIÓNS A DESCARGAR

Descrición	O usuario poderá indicar o rango de tempo do que lle interesan as observacións, así como un ou máis sensores e unha ou máis entidades.
Relevancia	Esixido
Criterio de validación	O requisito cumprirase se a través de interface se pode seleccionar un rango de tempo, unha lista de sensores e unha lista de entidades e estes datos se inclúen no XML para a operación <i>GetObservations</i> .

RF.06.- PERMITIR FILTRADO AVANZADO DAS OBSERVACIÓNS A DESCARGAR

Descrición	O usuario poderá indicar un filtro espacial que delimite a zona da que quere obter as observacións. A interface mostrará os operadores e tipos de operandos soportados polo servizo. Tamén se poderá indicar un filtro escalar por algunha das propiedades da oferta seleccionada. A interface so mostrará os operandos soportados polo servizo.
Relevancia	Desexado
Criterio de validación	O requisito cumprirase se a través de interface se pode indicar un filtro espacial, seleccionando a xeometría sobre o mapa, e un filtro escalar para algunha das propiedades observadas, e estes datos se inclúen no XML para a operación <i>GetObservations</i> .

RF.07.- XERAR PETICIÓN DE OBSERVACIÓNS MANUALMENTE

Descrición	O usuario poderá modificar en modo texto o XML xerado a través da interface gráfica, para obter as observacións desexadas.
Relevancia	Opcional
Criterio de validación	O requisito cumprirase se a aplicación permite visualizar e modificar o XML a usar para a operación <i>GetObservations</i> .

RF.08.- XERAR GRÁFICA PROPIEDAD *vs* TEMPO

Descrición	O usuario poderá, a partir dunha entidade seleccionada no mapa, visualizar unha gráfica que mostre unha liña coa evolución temporal da propiedade observada.
Relevancia	Esixido
Criterio de validación	O requisito cumprirase se a aplicación mostra unha gráfica de liña representando o tempo no eixo X e os valores da propiedade observada no eixo Y.

RF.09.- XERAR GRÁFICA PARA ENFRONTAR DÚAS PROPIEDADES

Descrición	O usuario poderá, a partir dunha entidade seleccionada no mapa, visualizar unha gráfica de puntos indicando unha propiedade para o eixo X e outra para o Y.
-------------------	---

Relevancia	Desexado
Criterio de validación	O requisito cumprirase se a aplicación mostra unha gráfica de puntos cos datos relativos a entidade seleccionada para as propiedades seleccionadas en cada eixo.

RF.10.- XERAR GRÁFICA CON VARIAS SERIES

Descrición	O usuario poderá seleccionar varias entidades no mapa, e cada unha delas será unha serie de datos na gráfica visualizada. O usuario poderá configurar o estilo de cada unha das series.
Relevancia	Desexado
Criterio de validación	O requisito cumprirase se a aplicación permite xerar gráficas con varias entidades seleccionadas, de xeito que as observacións de cada entidade teñan un formato diferente. O tipo de liña, de marcador, e a cor de cada serie de datos ten que poder modificarse.

RF.11.- XERAR ANIMACIÓN NO VISOR DE MAPAS

Descrición	O usuario disporá dunha barra de tempo integrada no visor de mapas, que permitirá visualizar so as observacións correspondentes ó tempo indicado nesta barra. Ademais disporá dun botón de reprodución que desprace a barra de tempo automaticamente.
Relevancia	Opcional
Criterio de validación	O requisito cumprirase se a aplicación xera unha capa válida para ser engadida ó <i>plugin TimeManager</i> de QGIS.

3.2.2. Requisitos non funcionais

Os requisitos non funcionais definidos, verificados e revisados para o presente proxecto son os que seguen:

RNF.01.- VERSIÓN DE SOS 1.0

Descrición	A pesar de existir a versión 2.0 do estándar SOS debe soportarse a versión 1.0, pois esta é a que implementan os servidores desenvolvidos polo CiTIUS.
Relevancia	Esixido
Criterio de validación	O requisito cumprirase se a aplicación é capaz de comunicarse con servidores que implementen a versión 1.0 do estándar SOS.

RNF.02.- LINGUAXE DE PROGRAMACIÓN PYTHON

Descrición	QGIS soporta <i>plugins</i> programados en C++ ou en Python, pero so os programados en Python son instalables desde o xestor de <i>plugins</i> incorporado na aplicación.
Relevancia	Desexado
Criterio de validación	A aplicación programarase en linguaxe Python.

RNF.03.- CONSISTENCIA DA INTERFACE GRÁFICA

Descrición	Dado que a aplicación a desenvolver é un <i>plugin</i> que se integrará dentro da propia aplicación QGIS o aspecto e deseño da interface gráfica debe ser consistente coa do propio QGIS.
Relevancia	Desexado
Criterio de validación	O requisito cumprirase se usuarios habituais ó QGIS son capaces de usar o <i>plugin</i> sin necesidade de indicacións.

RNF.04.- MANUAL DE USUARIO

Descrición	O usuario terá a súa disposición un manual de uso da aplicación.
Relevancia	Esixido
Criterio de validación	O requisito cumprirase se a memoria do proxecto inclúe un apéndice no que se explique o funcionamento básico do <i>plugin</i> .

RNF.05.- INTERNACIONALIZACIÓN DA APLICACIÓN

Descrición	A aplicación deberá ser deseñada de xeito que poida ser adaptada para outras linguas sen a necesidade de facer cambios a nivel de código.
Relevancia	Opcional
Criterio de validación	O requisito cumprase se o <i>plugin</i> pode ser traducido a distintos idiomas sen modificar o código fonte.

RNF.06.- DATA LÍMITE DE EXECUCIÓN DO PROXECTO

Descrición	A data límite de entrega da documentación do proxecto é o venres 10 de Xullo de 2015, segundo o publicado na páxina web da Escola Técnica Superior de Enxeñaría ¹ .
Relevancia	Esixido
Criterio de validación	-

3.3. Matriz de trazabilidade

A matriz de trazabilidade relaciona cada requisito co caso de uso que o orixina, de xeito que se poda facer un seguimento dos mesmos. A matriz de trazabilidade para este proxecto é a representada no cadro 3.2.

	RF.01	RF.02	RF.03	RF.04	RF.05	RF.06	RF.07	RF.08	RF.09	RF.10	RF.11
CU.01	•	•	•								
CU.02				•	•	•	•				
CU.03								•	•	•	
CU.04											•

Cadro 3.2: Matriz de trazabilidade de requisitos

¹http://www.usc.es/etse/files/u1/datasdefensa14_15GREI.pdf

Capítulo 4

Deseño do software

Neste capítulo descríbese a arquitectura do sistema a desenvolver desde un punto de vista global, identificando as distintas compoñentes que o conforman e a forma de relacionarse, tanto dende o punto de vista estrutural como de interacción entre elas.

4.1. Arquitectura do sistema

Dende a definición dos obxectivos do proxecto identifícanse dúas partes diferenciadas dentro do sistema a desenvolver. Por un lado a comunicación co servidor SOS, para a obtención das súas capacidades e dos datos de observacións, e por outro a visualización e explotación das observacións descargadas. Esta división trasládase directamente á estrutura da aplicación, que se amosa na figura 4.1. Tamén se inclúen as compoñentes externas coas que se relacionan.

A compoñente **SOS Client** é a responsable da comunicación co servidor SOS tanto á hora de xerar as peticións necesarias a partir da información proporcionada polo usuario, como para almacenar e interpretar as respostas do servizo.

A compoñente **SOS Plot** é a responsable de, a partir da capa vectorial xerada, visualizar as observacións segundo as preferencias indicadas polo usuario.

As compoñentes externas incluídas no diagrama son:

- QGIS: Representa á aplicación na que se integra o *plugin*.
- TimeManager¹: É un *plugin* para QGIS que engade controles de tempo ó mesmo, de xeito que se podan animar as capas vectoriais directamente no visor de mapas, en base a un atributo de tempo.
- matplotlib²: É unha librería de Python para debuxar gráficos en 2D que permite producir figuras de alta calidade, tanto para publicacións como en entornos interactivos.

¹<https://plugins.qgis.org/plugins/timemanager/>

²<http://matplotlib.org/>

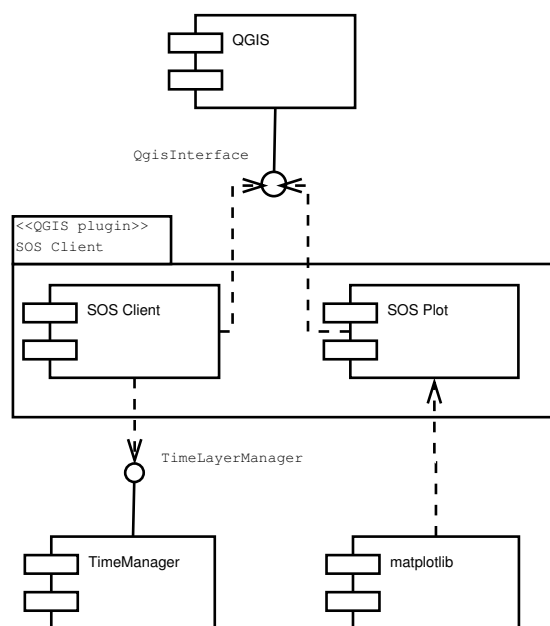


Figura 4.1: Diagrama de compoñentes

4.1.1. Patrón de arquitectura

O uso dun patrón de arquitectura axeitado para o sistema a desenvolver facilita os procesos de implementación e probas, ó proporcionar un esquema de organización estrutural dividindo o sistema en partes segundo a súa responsabilidade.

Os patróns de arquitectura máis amplamente utilizados no desenvolvemento de aplicacións son o MVC (*Model-View-Controller*) e os seus derivados. O obxectivo principal destes patróns é separar o modelo de datos, a súa visualización e a lóxica de negocio facilitando de xeito moi significativo o mantemento e evolución das aplicacións.

Dadas as características concretas desde desenvolvemento optouse por unha simplificación do patrón MVC combinando a lóxica de negocio coas vistas, dando lugar a unha arquitectura denominada *Model/View*³. Os motivos principais para levar a cabo esta simplificación son:

- A lóxica de negocio da aplicación e moi sinxela.
- As librerías para o desenvolvemento das vistas veñen impostas pola aplicación na que se integrará o *plugin*. As propias librerías están deseñadas para facilitar este patrón.

É importante destacar que aínda que se incorpore a lóxica de negocio nas vistas, si que se mantén a separación entre a creación e configuración dos com-

³<http://doc.qt.io/qt-4.8/model-view-programming.html>

poñentes gráficos do resto de lóxica da vista. A creación dos compoñentes gráficos faise a través dunha factoría que os xera directamente desde a súa definición XML.

Na figura 4.2 móstrase de xeito gráfico a diferenza entre o patrón MVC e o *Model/View*. O *Model/View*, amais da vista e o modelo, inclúe o concepto *Delegate*, que representa a un mediador entre a vista e o modelo para facilitar a personalización de como se mostran e editan determinados datos.

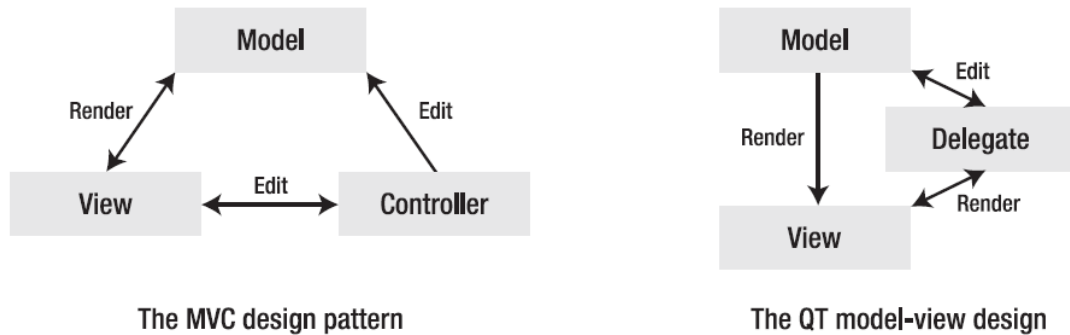


Figura 4.2: Diferencias entre MVC e Model/View

4.2. Interacción entre compoñentes

Para describir o comportamento do sistema desde un punto de vista dinámico empréganse os diagramas de secuencia, nos que se describe a interacción entre as distintas compoñentes do sistema para cumprir cada caso de uso definido.

Para conseguir un nivel de detalle suficiente para describir o comportamento do sistema divídense as compoñentes segundo a arquitectura *Model/View* proposta. A compoñente SOSClient divídese en SOSClientDialog, que representa a vista, e Sensor Observation Service, que representa o modelo. No caso da compoñente SOSPlot, a vista é representada pola compoñente SOSPlotDialog e o modelo non se representa de xeito independente pois está incluído dentro da compoñente matplotlib.

No diagrama 4.3 inclúense os caso de uso CU.01 e CU.02, xa que o CU.02 estende ó CU.01, polo que é máis claro representalos xuntos.

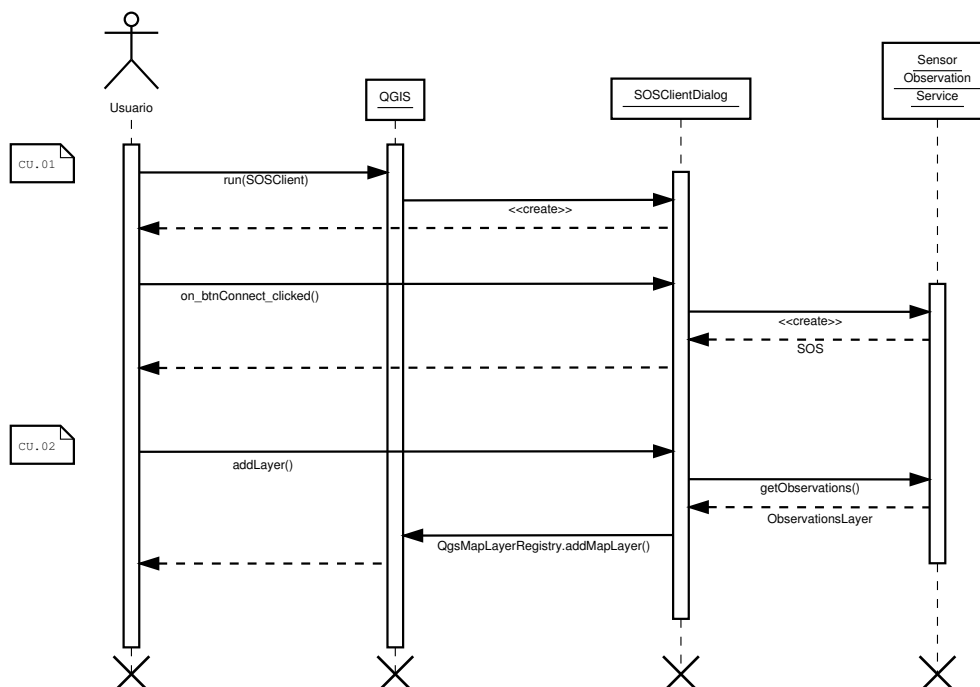


Figura 4.3: Diagrama de secuencia para os casos de uso CU.01 e CU.02

O diagrama 4.4 representa o comportamento para o caso de uso CU.03.

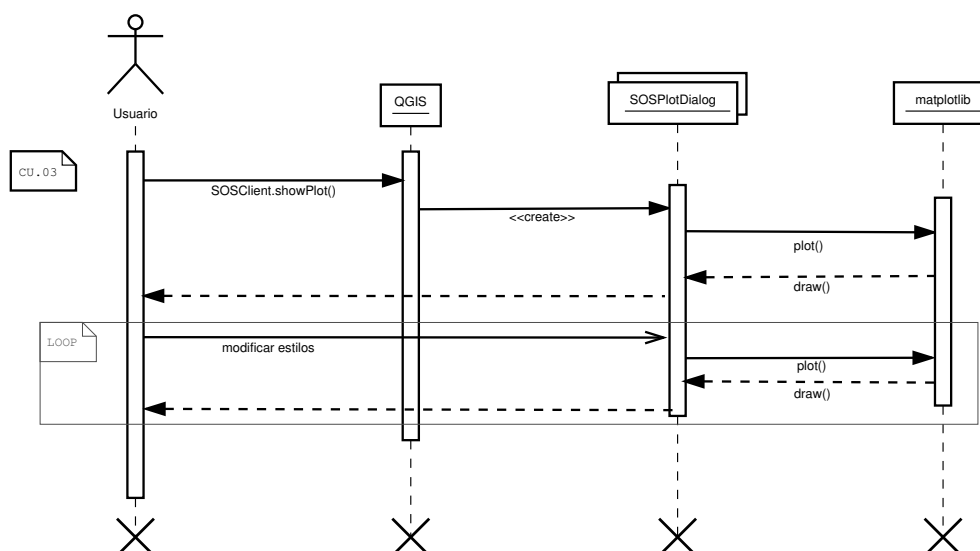


Figura 4.4: Diagrama de secuencia para o caso de uso CU.03

Para cubrir a funcionalidade requirida polo caso de uso CU.04 existe o *plugin TimeManager* para QGIS, polo que o que representa o diagrama 4.5 é o procedemento para que a capa xerada coas observacións sexa controlada polo *TimeManager*.

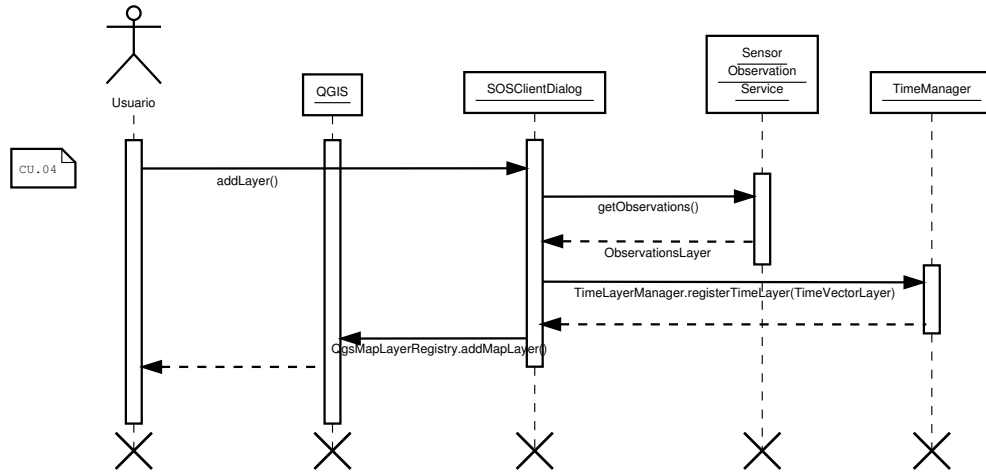


Figura 4.5: Diagrama de secuencia para o caso de uso CU.04

Capítulo 5

Desenvolvemento dos sprints

A implementación deste proxecto consta de tres ciclos de sprint nos que se desenvolven as dúas compoñentes nas que está dividido. Cada un dos sprints consta das fases de análise, deseño, implementación e probas. Dado que a análise se describiu por completo no capítulo 2, neste capítulo detallaranse o deseño, implementación e probas realizadas para validar os requisitos planificados en cada un dos sprints.

A previsión inicial do contido de cada sprint está recollida no cadro 2.2, no capítulo de Xestión do proxecto. Neste apartado describirase xa a pila de cada sprint en base ós requisitos definidos, é dicir, o resultado da planificación do sprint. Cabe destacar que para a planificación do sprint non so se ten en conta a relevancia asignada a cada requisito, se non tamén o tempo estimado para levalo a cabo e a súa repercusión no cumprimento dos requisitos non funcionais.

5.1. Sprint 1

Os requisitos funcionais que se inclúen na pila do primeiro sprint son:

- RF.01.- Conectar con servidor SOS
- RF.02.- Gardar lista de servidores SOS
- RF.04.- Xerar capa vectorial das observacións do servidor

Neste ciclo de desenvolvemento levouse a cabo o deseño inicial da interface gráfica para a ventá de conexión co servidor SOS, seguindo como guía as ventás dispoñibles de xeito nativo no QGIS para descargar datos de outros servizos similares (WFS, WMS, WCS), co obxectivo de satisfacer o requisito RNF.03.

Outra tarefa importante levada a cabo foi a definición dunha estrutura extensible para o módulo de procesamento do XML, de xeito que sexa sinxelo engadir as distintas casuísticas que vaian aparecendo, debido a gran liberdade que dan os estándares SOS e O&M. Tamén se analizaron distintas posibilidades para o

procesamento do XML en Python, sendo todas elas moi similares en canto a funcionalidade, optouse por empregar o módulo para XML das propias librerías Qt para non engadir dependencias innecesarias.

No diagrama 5.1 represéntanse o deseño de clases realizado, que se corresponde coa compoñente SOSClient. Móstranse so os atributos e métodos máis relevantes para entender a función de cada clase, que se representan separadas entre vista e modelo. Agrupadas sobre fondo escuro represéntanse as clases encargadas do procesamento dos documentos XML proporcionados polo servizo SOS.

As clases máis relevantes da figura 5.1 son:

WidgetFactory: É unha factoría de clases que crea, a partir dos XML de definición da interface, as clases cos compoñentes gráficos. A implementación real da vista farase en clases que herdarán da clase correspondente creada por esta factoría.

SOSClientDialog: É o formulario xeral de comunicación cos servidores SOS. Contén a lóxica de funcionamento da vista e as chamadas necesarias ás clases que forman o modelo.

SensorObservationService: Esta clase representa o servizo SOS. Créase a partir do XML que define as capacidades do servizo e almacena toda a información necesaria para que a vista poda presentar o usuario as opcións correspondentes para xerar as consultas.

ObservationsLayer: Esta clase é a encargada de construír a capa vectorial para o QGIS, a partir da clase SOSProvider.

SOSProvider: Simula unha clase QgsVectorDataProvider¹. É pois a clase que fai de intermediaria entre a estrutura de datos real da información da capa e a estrutura manexada polo QGIS.

XMLParserFactory: É unha factoría para crear as distintas clases para o procesamento dos XML. A partir da etiqueta XML atopada devolve a clase encargada de procesar o nodo.

XMLParser: Clase base para todas as clases de procesamento de XML.

Tamén se preparou durante este sprint o entorno para a automatización das probas unitarias a través da ferramenta *PyUnit*² e programáronse varios casos de probas para o módulo de procesamento de XML en base a ficheiros de exemplo.

Co obxectivo de validar o incremento xerado leváronse a cabo as seguintes probas:

¹<http://qgis.org/api/classQgsVectorDataProvider.html>

²<https://docs.python.org/2/library/unittest.html>

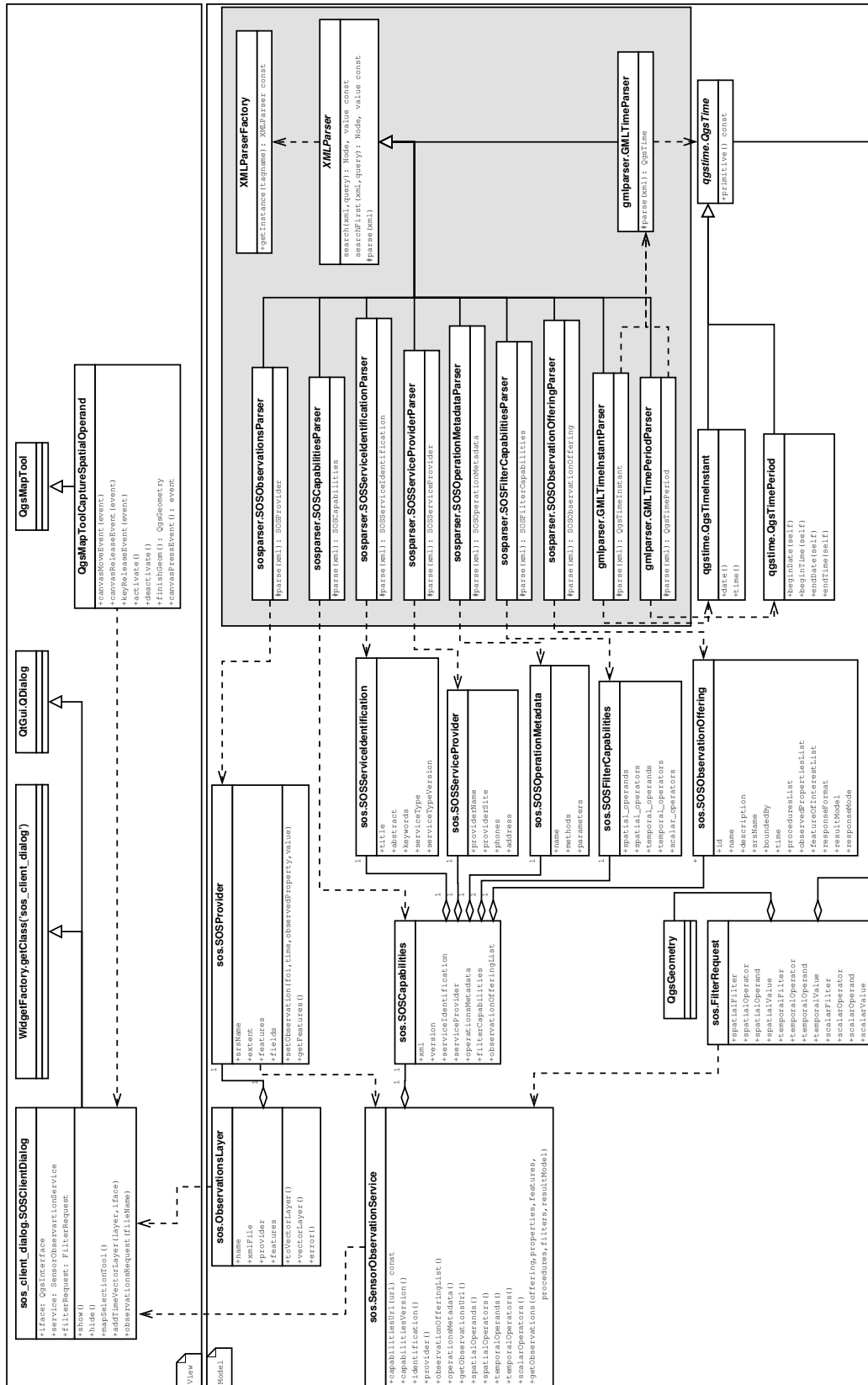


Figura 5.1: Diagrama de clases da componente SOS Client

PR.01.- CONEXIÓN CON SERVIDOR SOS

Requirimentos relacionados	<ul style="list-style-type: none"> ■ RF.01 ■ RNF.03
Descrición	Compróbase, a parte das probas unitarias automáticas, a conexión con varios servidores SOS.
Resultado	<ul style="list-style-type: none"> ■ A información das capacidades do servizo visualízanse en formato HTML cos estilos propios de QGIS. ■ As ofertas dispoñibles visualízanse nunha caixa de selección, e as propiedades e procedementos relacionados en sendas listas nas que se poden marcar e desmarcar. Estas listas actualízanse correctamente ó modificar a oferta seleccionada.

PR.02.- XESTIÓN DE SERVIDORES

Requirimentos relacionados	<ul style="list-style-type: none"> ■ RF.02 ■ RNF.03
Descrición	Próbanse as funcionalidades de alta, baixa e modificación de conexións na interface gráfica, así coma o correcto gardado entre distintas execucións da aplicación QGIS.
Resultado	Todas as operacións funcionan correctamente.

PR.03.- DESCARGAR OBSERVACIÓNS

Requirimentos relacionados	<ul style="list-style-type: none"> ■ RF.04
Descrición	Compróbase a execución da operación <i>GetObservations</i> contra varios servidores.
Resultado	<ul style="list-style-type: none"> ■ O XML descárgase correctamente. ■ O XML procesase e obtéñense as xeometrías contidas no mesmo coas que se crea unha capa vectorial en memoria no QGIS.

O RF.04 cubriuse parcialmente, pois aínda que se podían visualizar no mapa os puntos correspondentes as observacións o resto de información do XML se trata correctamente.

Tendo en conta a matriz de trazabilidade do cadro 3.2, este primeiro incremento non resolve ningún caso de uso completo.

5.2. Sprint 2

Na pila do segundo sprint inclúense os seguintes requisitos funcionais:

- RF.04.- Xerar capa vectorial cas observacións do servidor (continuación)
- RF.03.- Visualizar XML das capacidades do servidor
- RF.05.- Permitir filtrado básico das observacións a descargar
- RF.07.- Xerar petición de observacións manualmente
- RF.11.- Xerar animación no visor de mapas
- RF.08.- Xerar gráfica propiedade vs tempo

No segundo ciclo de sprint conclúese o módulo de procesamento XML para interpretar correctamente todos os datos das observacións resultado da operación *GetObservations*, tanto co modelo *Observations* como co *Measurements*. Amais comprobase que requisitos debe cumprir a capa para poder ser xestionada a través do plugin *TimeManager*, e impleméntase o proceso de creación da capa tendo en conta estes requisitos.

Tamén se implementan os filtros básicos de observacións, e por ser de utilidade para o proceso de desenvolvemento e probas tamén se implementa a visualización do XML das capacidades do servidor e a posibilidade de visualizar e editar o XML xerado para a petición de observacións.

En canto a xeración de gráficas realízanse varias probas coa librería matplotlib.

Co obxectivo de validar o incremento xerado leváronse a cabo as seguintes probas:

PR.04.- PROCESAMENTO DE XML CON OBSERVACIÓNS EN FORMATO *Observations* E *Measurements*

Requisitos relacionados	■ RF.04
Descrición	Codifícanse probas unitarias para a validación do módulo de procesamento so XML de observacións nos dous formatos, en base a ficheiros de exemplos dos servidores SOS do CiTIUS.
Resultado	<ul style="list-style-type: none"> ■ O XML procesase correctamente, detectando os campos de tipo tempo, os numéricos e as cadeas de texto. ■ A partir da estrutura de datos xerada a partir do XML crease correctamente unha capa vectorial no QGIS, con todos os campos contidos na observación e cos valores correctos.

PR.05.- XERACIÓN DE FILTROS PARA *GetObservations*

Requisitos relacionados	<ul style="list-style-type: none"> ■ RF.03 ■ RF.05 ■ RF.07
Descrición	Comprobar que os obxectos de interface conteñen os valores correctos segundo as capacidades do SOS e que o XML que se xera a través das distintas opcións de interface é válido.
Resultado	<ul style="list-style-type: none"> ■ Comprobase conectando con varios servidores que as opcións habilitadas nas distintas lapelas de filtros se corresponden coas capacidades definidas no XML. ■ A lista de sensores seleccionados e a de entidades de interese, así como o operador e operandos dos filtros de tempo inclúense correctamente no XML para a petición <i>GetObservations</i>.

PR.06.- XESTIONAR CAPA CO *TimeManager*

Requisitos relacionados	<ul style="list-style-type: none"> ■ RF.04 ■ RF.11
Descrición	Comprobar o correcto funcionamento do plugin <i>TimeManager</i> coa capa creada.
Resultado	<ul style="list-style-type: none"> ■ Comprobase que a capa se engade á lista das xestionadas polo <i>TimeManager</i> de xeito satisfactorio. ■ Verifícase que o campo de tempo se interpreta correctamente. ■ Comprobase o funcionamento xeral do <i>TimeManager</i> con varias capas de observacións.

PR.07.- GRÁFICA TEMPO *vs* PROPIEDAD

Requisitos relacionados	<ul style="list-style-type: none"> ■ RF.08
Descrición	Xerar gráfica co tempo no eixo X e a propiedade no Y.

Resultado

- A gráfica contén todas as observacións co mesmo *foi* que a entidade seleccionada no mapa.
- As observacións ordénanse polo campo tempo de xeito que se poda trazar unha liña que as una.
- Ós valores do eixo X dáselles formato de tempo.

Tendo en conta a matriz de trazabilidade do cadro 3.2, con este segundo incremento cúbrense os casos de uso CU.01 e CU.04.

5.3. Sprint 3

Na pila do terceiro sprint inclúense os seguintes requisitos funcionais:

- RF.06.- Permitir filtrado avanzado das observacións a descargar
- RF.09.- Xerar gráfica para enfrontar dúas propiedades
- RF.10.- Xerar gráfica con varias series

Neste incremento habilitase a ferramenta para introducir o filtro espacial seleccionando a xeometría directamente sobre o mapa, segundo o tipo de operando seleccionado.

Tamén se modifica a visualización de gráficas para permitir seleccionar as propiedades a visualizar, traballar con varias *foi* seleccionadas e permitir modificar os estilos e textos da gráfica. Esta tarefa implica o deseño tanto da interface gráfica como da estrutura de clases da compoñente SOSPlot, que se representa no diagrama 5.2. Móstranse agrupadas todas as clases que cumpren co rol de *Delegate*. As demais correspóndense coa vista, pois o modelo da compoñente SOSPlot está soportado por clases internas do QGIS (QgsVectorLayer) e de matplotlib (Line2D).

As clases máis relevantes da figura 5.2 son:

WidgetFactory: Factoría para a xeración das compoñentes gráficas.

SOSPlotDialog: É o formulario de visualización de gráficos. Filtra e transforma os datos da capa vectorial activa para visualizalos no QMplWidget e conecta os distintos *Delegates* co modelo.

QMplWidget: Clase que amalgama os obxectos da librería matplotlib para a visualización de gráficos.

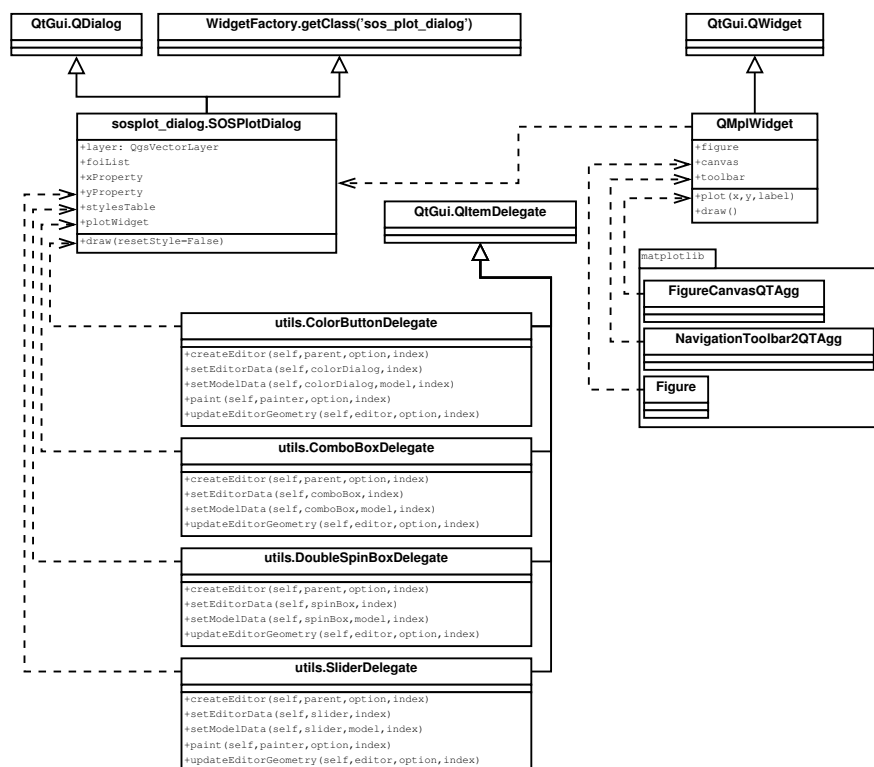


Figura 5.2: Diagrama de clases da compoñente SOS Plot

Co obxectivo de validar o incremento xerado leváronse a cabo as seguintes probas:

PR.08.- FILTROS ESPACIAIS

Requisitos relacionados

- RF.06

Descrición

Comprobar o correcto funcionamento da ferramenta de selección de xeometrías e a inclusión no XML do filtro correspondente.

Resultado

- A ferramenta de selección permite debuxar rectángulos (indicando dous puntos), polígonos, liñas e puntos.
- A xeometría debuxada no mapa transformase correctamente ó formato GML e inclúese no XML para a petición *GetObservations*.

PR.09.- XERACIÓN DE GRÁFICAS CON VARIAS SERIES

Requisitos relacionados	<ul style="list-style-type: none"> ■ RF.10
Descripción	Compróbanse a xeración de gráficas a partir de varias entidades seleccionadas, de forma que cada <i>foi</i> teña un estilo distinto.
Resultado	<ul style="list-style-type: none"> ■ Permítese xerar a gráfica con varias entidades seleccionadas. Para cada <i>foi</i> distinto ordénanse os datos e débúxase a liña con unha cor distinta. ■ Permítese modificar a cor e estilo de cada liña e dos marcadores.

PR.10.- PERSONALIZACIÓN DAS PROPIEDADES DAS GRÁFICAS

Requisitos relacionados	<ul style="list-style-type: none"> ■ RF.09
Descripción	Verificar que se poden modificar as distintas propiedades da gráfica, incluídas a propiedade a representar en cada eixo.
Resultado	<ul style="list-style-type: none"> ■ Pódense seleccionar o campo da capa a usar no eixo X e no eixo Y, así como por cal dos dous ordenar os datos para trazar liñas. ■ Permítese modificar os límites dos eixos X e Y manualmente. ■ Pódense modificar outras propiedades da gráfica como o título, as etiquetas e o formato de tempo.

Tendo en conta a matriz de trazabilidade do cadro 3.2, con este terceiro incremento a aplicación permite levar a cabo os catro casos de uso definidos.

Capítulo 6

Conclusións e traballo futuro

Para finalizar, neste capítulo descríbense as conclusións da realización do proxecto e apúntanse varias liñas de traballo futuro para a ampliación do mesmo.

6.1. Conclusións

O obxectivo deste traballo foi o desenvolvemento dun *plugin* para o sistema de información xeográfica de código aberto QGIS, que permitise obter datos de observacións de servidores que soporten o estándar Sensor Observation Service versión 1.0 definido polo Open Geospatial Consortium. O traballo xustifícase no interese de permitir explotar e analizar os datos da información recollida por unha ampla variedade de sensores a través do uso do gran abano de ferramentas que proporciona un sistema de información xeográfica de propósito xeral.

Ata o momento non existen solucións deste estilo no eido do software libre, nin ningunha solución madura entre os sistemas propietarios. Debido a esta escaseza de solucións consideramos que este traballo abre unhas posibilidades ata o momento inexistentes dentro do seu campo de aplicación.

A maior dificultade coa que nos topamos durante o proxecto foi a escasa dispoñibilidade de servidores SOS accesibles publicamente e o baixo nivel de madurez dos mesmos, e por tanto do estándar. As poucas implementacións atopadas presentan pequenas diferencias entre si que limitan a interoperabilidade do *plugin*.

Os obxectivos do traballo pódense dar por cumpridos ó producir un software completamente funcional que satisfai as necesidades plantexadas inicialmente.

A inclusión no repositorio oficial pon o *plugin* a disposición da ampla e activa comunidade de desenvolvedores e usuarios de QGIS.

6.2. Traballo futuro

En canto ás posibilidades de evolución da ferramenta, a ampliación que maior valor aportaría sería dar cobertura o maior número de implementacións posible do

estándar, e a versión 2.0 do mesmo. Unha das posibilidades para que o procesador de XML sexa capaz de soportar as pequenas diferencias entre implementacións do estándar sería permitir que sexa parametrizable polo usuario, de xeito que poda indicar, por exemplo, o nome da propiedade a usar como campo de tempo ou a etiqueta XML que contén o identificador da entidade.

Outra área de mellora na compoñente de interacción co SOS sería dotar á interface gráfica de máis posibilidades á hora de definir os filtros a aplicar para obter as observacións, permitindo, por exemplo, usar xeometrías de outras capas como filtro espacial.

Tamén sería interesante a posibilidade de que a petición *GetObservations* se puidese executar de xeito periódico cada poucos segundos, ou minutos, actualizando a capa en vez de crear unha nova. Esta funcionalidade permitiría traballar con datos en tempo case real visualizando e actualizando constantemente as últimas observacións realizadas.

En canto á xeración de gráficas, unha mellora importante sería poder visualizar varias propiedades simultaneamente en dúas gráficas que compartan un eixo, de modo que se poida ver a evolución temporal de dúas propiedades ó mesmo tempo que a relación entre elas.

A parte destas ampliacións propostas existirán outras moitas que xurdan do uso do *plugin* nun contorno real, por usuarios que o usen como ferramenta para resolver os seus problemas específicos.

Apéndice A

Manual técnico

A.1. Entorno de desenvolvemento

Se ben non é necesario o uso de ningún entorno de desenvolvemento integrado (IDE) para a modificación do código do plugin, si é recomendable para aumentar a produtividade gracias as funcionalidades como o resaltado de sintaxe, o autocompletado ou a validación automática do código.

A continuación descríbense os pasos a seguir para a instalación dun entorno de desenvolvemento integrado igual ó empregado para a realización do plugin, que se levou a cabo sobre o sistema operativo Ubuntu.

O IDE recomendado na propia documentación de QGIS para o desenvolvemento de plugins en Python é Eclipse¹ co plugin PyDev², que pode instalarse dende o propio Eclipse. Amais do plugin PyDev tamén debemos instalar o plugin EGit para descargar o código dende GitHub. Tamén debemos ter instalado no equipo o QGIS, que se pode instalar desde os propios repositorios de Ubuntu, o igual que o Eclipse.

Para crear un proxecto desde o repositorio de GitHub existen varias opcións, a máis sinxela sería a seguinte:

¹<https://eclipse.org/>

²<http://pydev.org/>

- Seleccionar la opción de Import no menú File, e posteriormente Projects from Git.

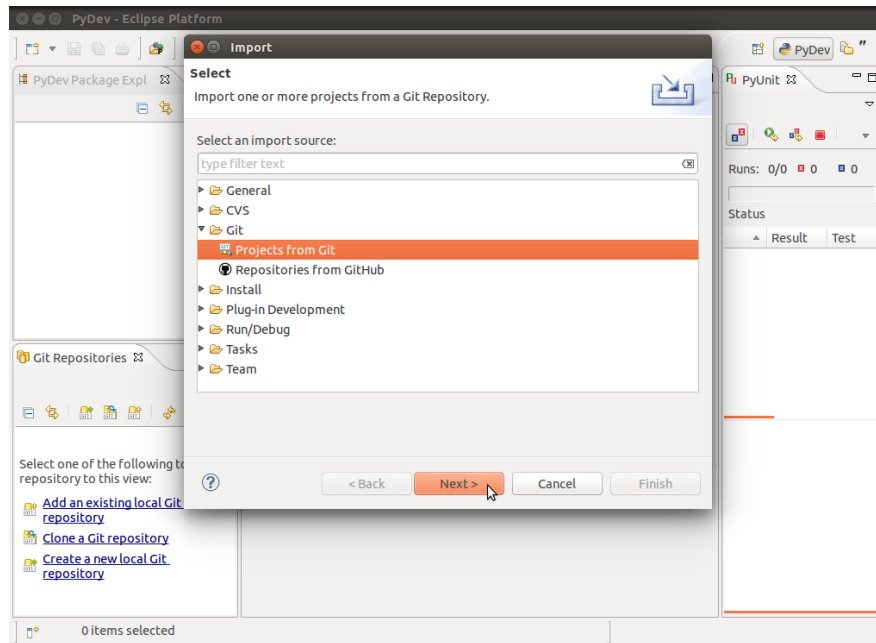


Figura A.1: Importación do repositorio en Eclipse, paso 1

- Seleccionar la opción GitHub e buscar o repositorio SOS Client.

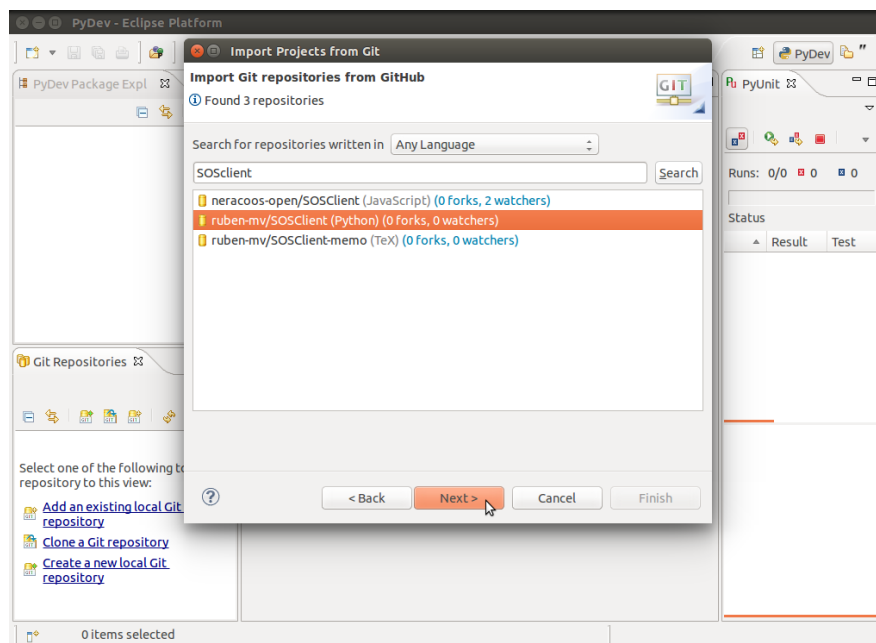


Figura A.2: Importación do repositorio en Eclipse, paso 2

- Engadir a rama master e importar coa opción Import as general project.

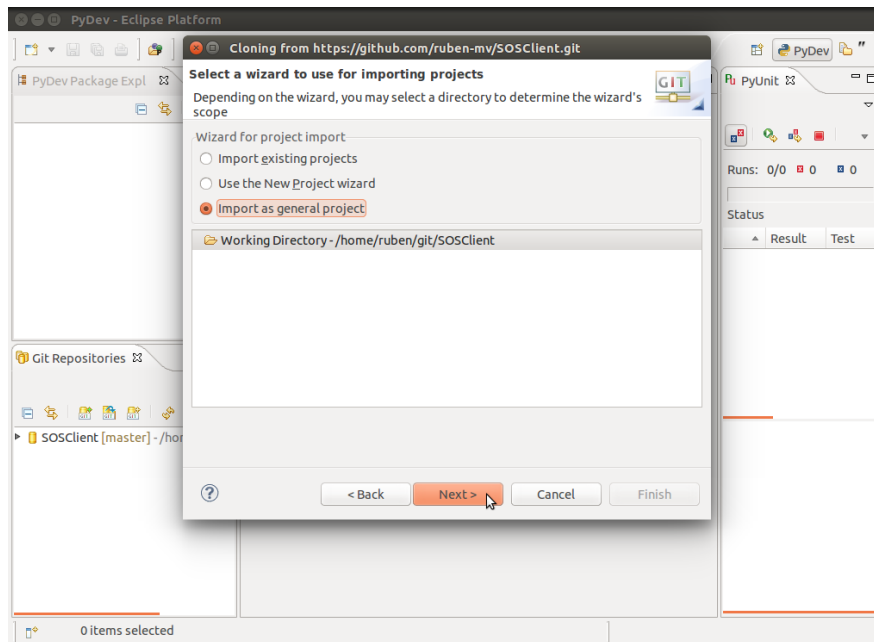


Figura A.3: Importación do repositorio en Eclipse, paso 3

- Unha vez feito importado o proxecto pódese converter nun proxecto de PyDev.

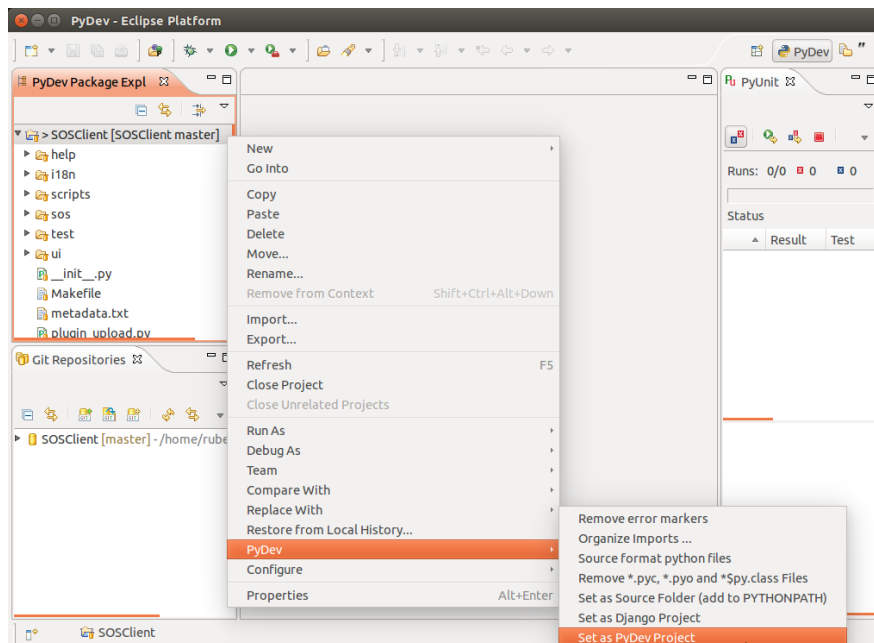


Figura A.4: Importación do repositorio en Eclipse, paso 4

A continuación débese configurar o PyDev para que reconeça a API de QGIS e de PyQt4 de forma que teñamos dispoñibles as funcións de autocompletado. Para esto, dende a ventá de Window → Preferences accedese á configuración do interprete de Python, onde haberá que engadir na lapela de librerías a carpeta `~/.qgis/python/plugins/`, e na lapela Forced builtin engadir qgis e PyQt4.

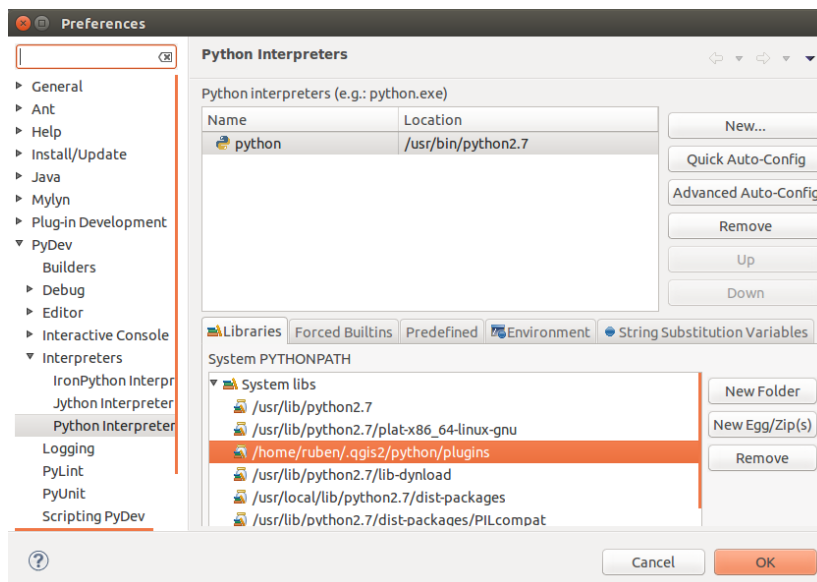


Figura A.5: Configuración de Python en Eclipse

Tamén se deben engadir as carpetas do proxecto ó PYTHONPATH, desde as propiedades do proxecto:

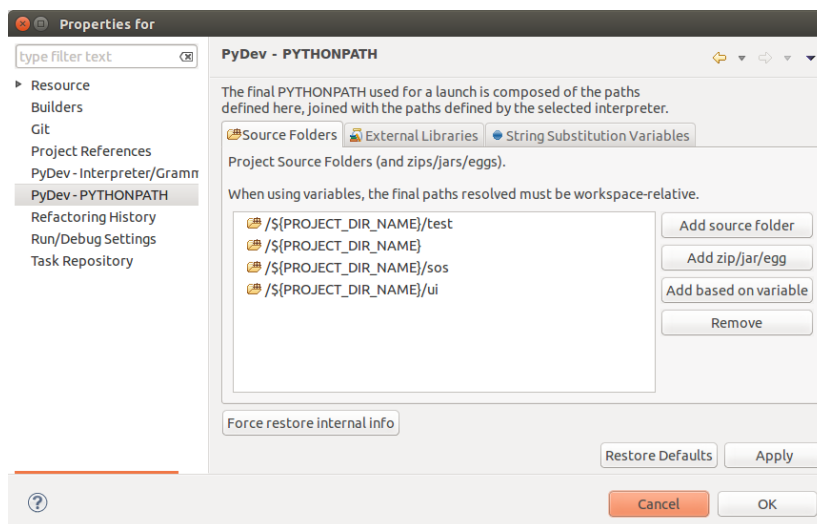


Figura A.6: Configuración do PYTHONPATH

O proxecto foi creado inicialmente co plugin 'Plugin Builder'³ de QGIS polo que contén un Makefile que nos permite usar make coas seguintes opcións, entre outras:

deploy: Desprega o plugin na carpeta correspondente.

derase: Elimina a carpeta de despregue do plugin.

transup: Actualiza os ficheiros de traducións.

zip: Desprega o plugin e crea un zip axeitado para subir ó repositorio de QGIS.

test: Executa os tests.

Tamén é de moita utilidade instalar no QGIS o plugin 'Plugin Reloader'⁴, que se pode instalar dende o propio menú do QGIS. Este plugin permite volver a cargar un plugin dende disco sen necesidade de cerrar e volver a abrir o QGIS.

Outras ferramentas que se poden necesitar para o desenvolvemento do plugin son o QtDesigner, para modificar os ficheiros de extensión .ui nos que se define a interface gráfica, e o QtLingüist, para as traducións a distintos idiomas.

A.2. Ampliación do módulo de procesamento XML

Co obxectivo de que o módulo de procesamento se poda ampliar para dar cobertura a máis implementacións do servizo SOS e do estándar O&M este foi deseñado de forma modular, con distintas clases específicas para o procesamento de cada parte do XML. Para engadir soporte para etiquetas xml non soportadas será necesario crear unha clase que herde de XMLParser co nome da etiqueta que procesa co sufixo Parser, de xeito que poda ser instanciada a través da factoría XMLParserFactory cando sexa necesaria.

A clase XMLParser descríbese na figura A.7, e forma parte do módulo xmlparser no paquete sos.

³<http://g-sherman.github.io/Qgis-Plugin-Builder/>

⁴http://plugins.qgis.org/plugins/plugin_reloader/

```

class XMLParser (object):
    """
    XML parser base class
    """
    __metaclass__ = abc.ABCMeta

    @abc.abstractmethod
    def parse (self, xml=None):
        """
        :param xml: XML to parse
        :type xml: QDomElement or str
        """

    @staticmethod
    def searchFirst (xml, query):
        """
        :param xml: XML to parse
        :type xml: QDomNode
        :param query:
        :type query: str
        :return: QDomNode, str
        """

    @staticmethod
    def search (xml, query):
        """
        :param xml: XML to parse
        :type xml: QDomNode
        :param query:
        :type query: str
        :return: (QDomNode, str) generator
        """

```

Figura A.7: Definición da clase XMLParser

Esta clase consta de tres métodos. O método *parse* é o que debe ser sobreescrito polas clases fillas, para levar a cabo as operacións que corresponda, e os métodos *searchFirst* e *search* están definidos para abstraer o procesamento do XML das librerías empregadas. Ambas funcións reciben como parámetro o xml sobre o que traballar e a consulta a executar, coa diferenza de que a primeira devolve a primeira ocorrencia, e a segunda permite iterar sobre todas as ocorrencias. O parámetro query permite interrogar por etiquetas, atributos, e atributos con un valor concreto. Por exemplo:

- `node, version = self.searchFirst (xml, "@version")`
Devolve o atributo version da etiqueta en curso.
- `for _, value in self.search (xml, "resultModel"): pass`
Itera sobre todas as etiquetas resultModel fillas do nodo xml.

- `for _, value in self.search (xml, "observedProperty@href"): pass`
Itera sobre o atributo href de todas as etiquetas observedProperty fillas do nodo xml.
- `for node, tag in self.search (member, "Observation/*"): pass`
Itera sobre todas as etiquetas fillas de todas as etiquetas Observation fillas do nodo xml.
- `opNode, _ = self.searchFirst (xml, "Operation@name=GetCapabilities")`
Devolve o primeiro nodo de etiqueta Operation cuxo atributo name vale GetCapabilites.

Apéndice B

Manual de usuario

B.1. Instalación

O plugin pode instalarse desde a opción 'Administrar e instalar plugins' no menú Plugins do QGIS.

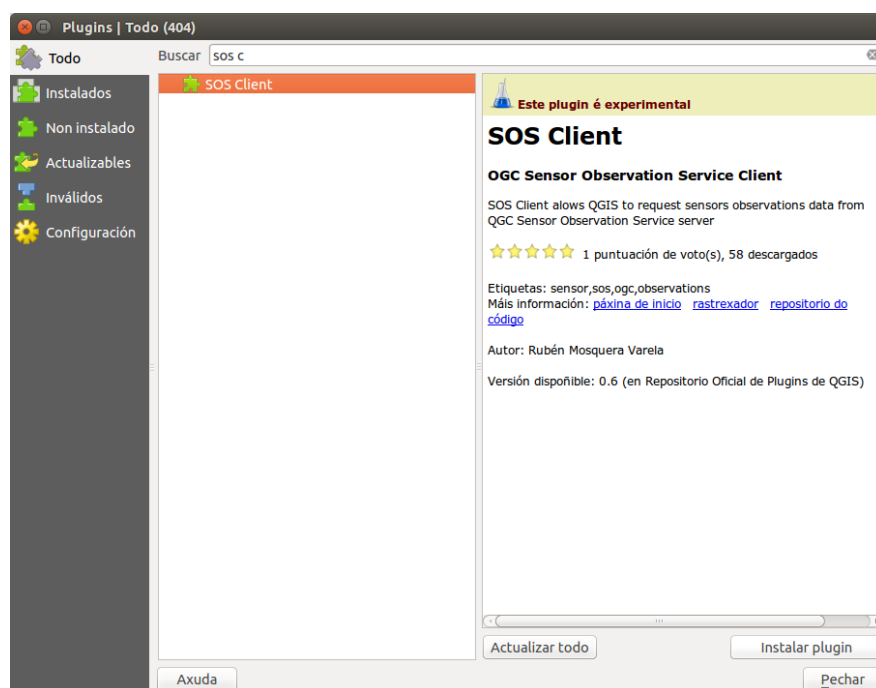


Figura B.1: Pantalla de instalación do plugin

Unha vez instalado engadirase unha nova entrada no menú Web e tres novas accións na barra de ferramentas:

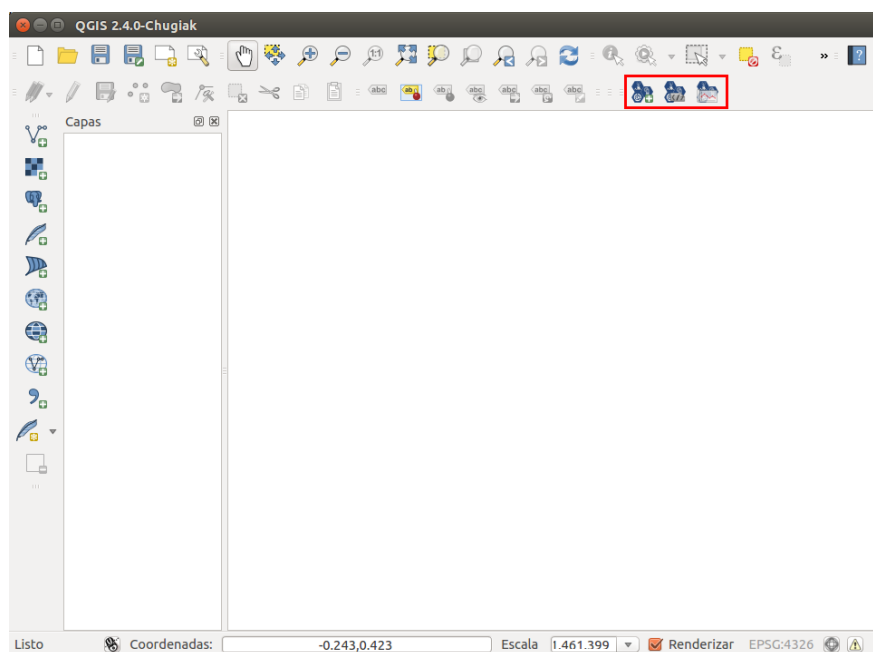


Figura B.2: Barra de ferramentas do plugin



Mostra o diálogo para conectar con un servidor SOS e engadir unha capa de observacións.




Visualiza o ficheiro XML usado para xerar a capa activa.



Xera unha gráfica coas observacións correspondentes ás entidades seleccionadas na capa activa.

B.2. Crear capa de observacións

Pulsando na acción  mostrase o formulario da figura B.3. Neste formulario pódese xestionar a lista de servidores cos botóns Novo, Editar e Eliminar, e co botón Conectar visualizar as capacidades do servidor seleccionado.

Unha vez se conectou con un servidor móstranse as súas capacidades na lapela Información (figura B.4).

Para obter as observacións débese seleccionar unha oferta no campo Ofertas e pódese modificar o Nome da capa. As distintas lapelas do formulario descríbense no cadro B.1:

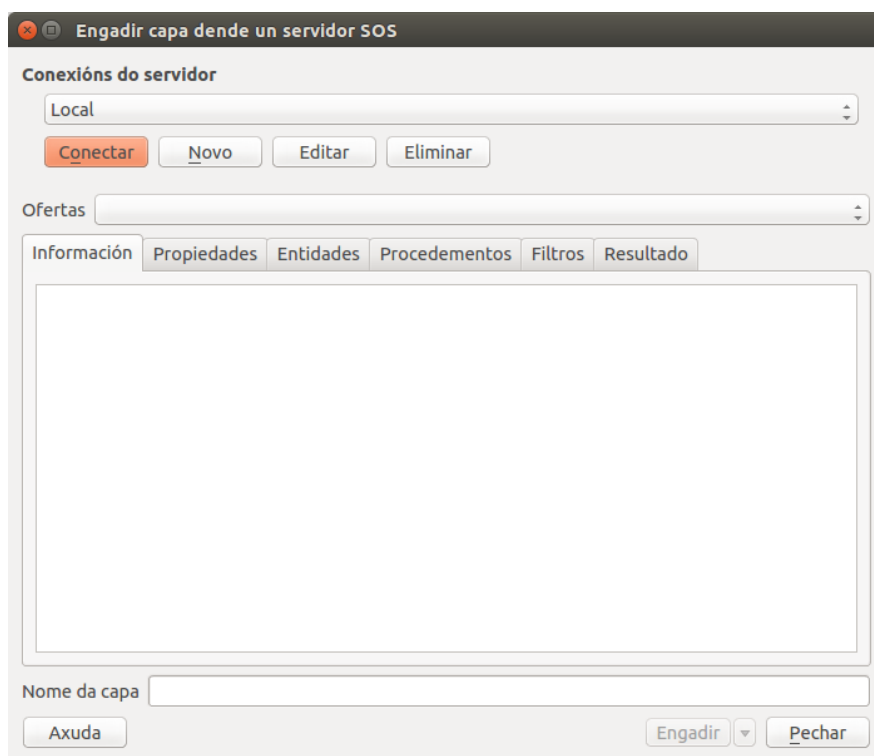


Figura B.3: Diálogo de conexión co servidor, sen datos

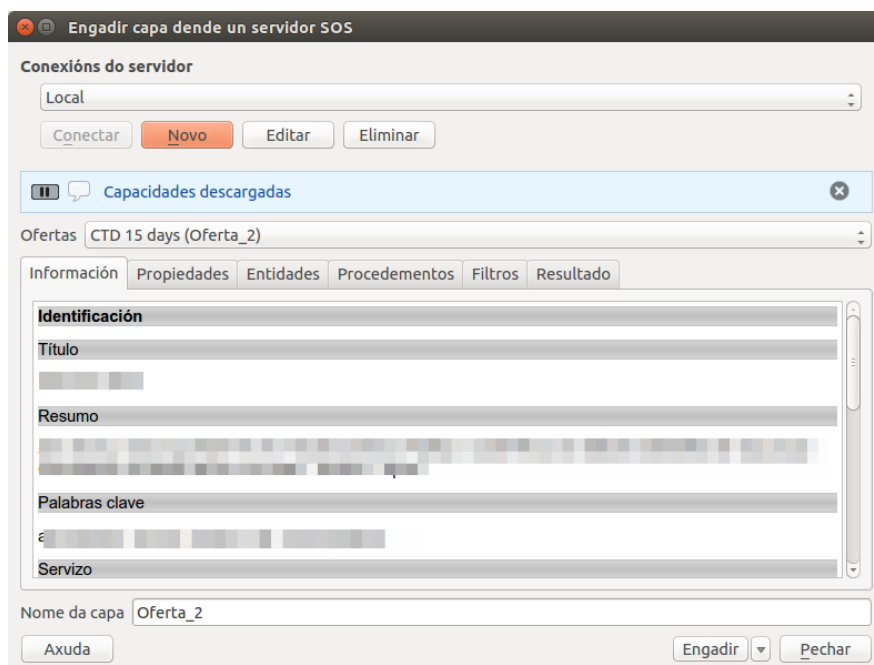


Figura B.4: Diálogo de conexión co servidor, con datos

Lista de propiedades da oferta. Pode seleccionar unha ou varias.

Lista de entidades da oferta. Pode seleccionar varias ou ningunha.

Lista de procedementos. Pode seleccionar varios ou ningún.

Filtros dispoñibles. Pódense activar varios ó mesmo tempo. No caso do espacial, pulsando na icona poderase seleccionar a xeometría a consultar debuxando no mapa, como se amosa na figura B.5.

Pódese seleccionar o modelo de resultado entre os dispoñibles para o servidor. Permite indicar se todas as entidades terán información xeométrica ou so a primeira para cada *foi* e seleccionar se engadir a capa ó TimeManager. Tamén se pode seleccionar o directorio de traballo no que se gardan os datos descargados.

Cadro B.1: Lapelas do formulario de consulta ó SOS

Unha vez seleccionadas as opcións desexadas pódese engadir a capa ó QGIS pulsando no botón Engadir, ou cambiar o XML a enviar antes de engadir a capa co botón 'Editar petición', como se ve na figura B.6.

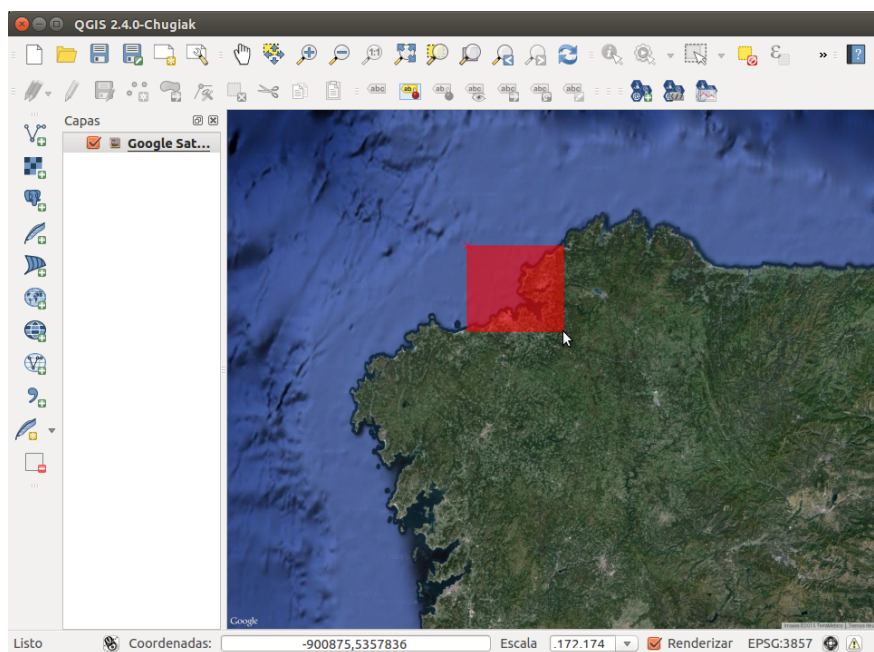


Figura B.5: Ferramenta de selección espacial

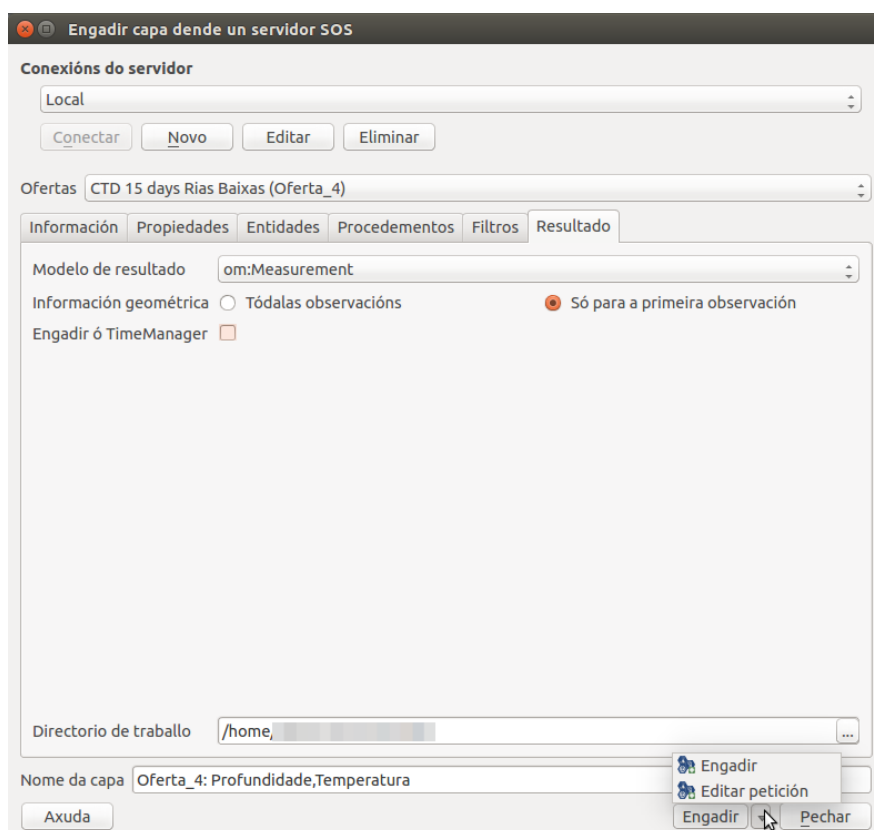



Figura B.6: Engadir capa SOS

B.3. Gráficos en dúas dimensións

Para visualizar un gráfico é necesario que a capa activa teña unha ou máis entidades seleccionadas, e despois premer no botón , como se amosa na figura B.7.

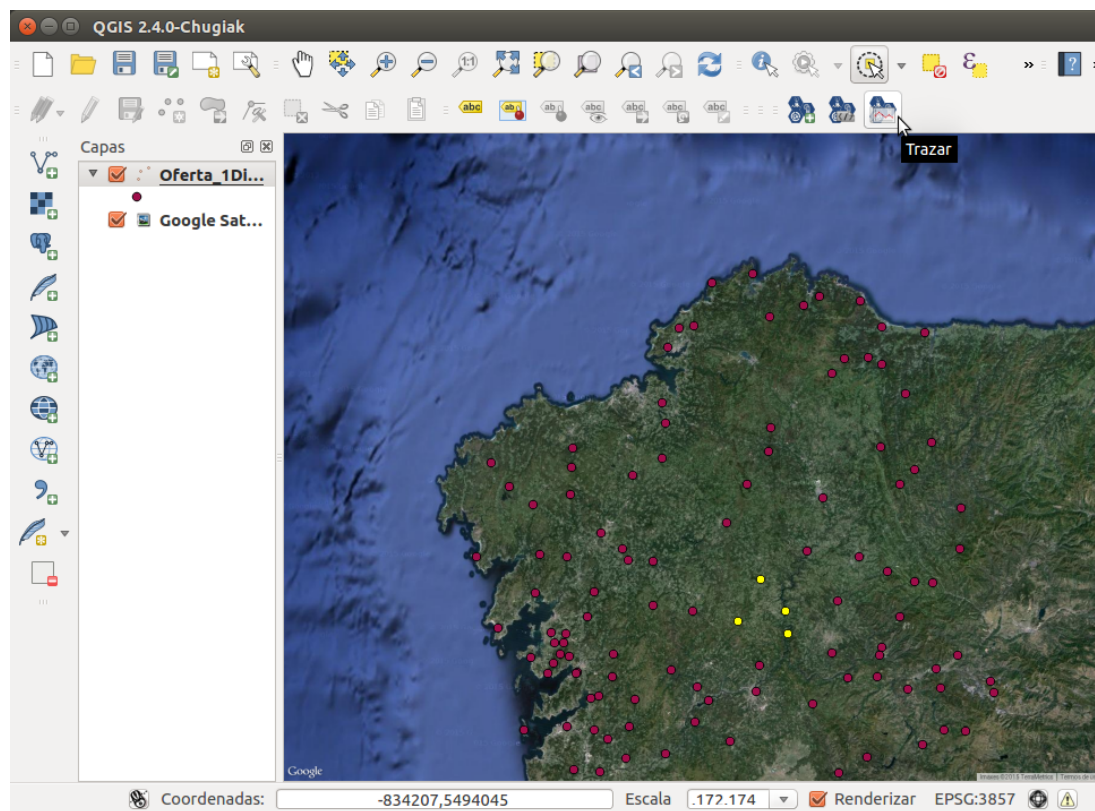


Figura B.7: Executar ferramenta SOS Plot

Esta operación abrirá unha nova ventá (figura B.8) na que se visualizará a gráfica, e na que se poden editar as opcións do mesmo e interactuar coa gráfica.

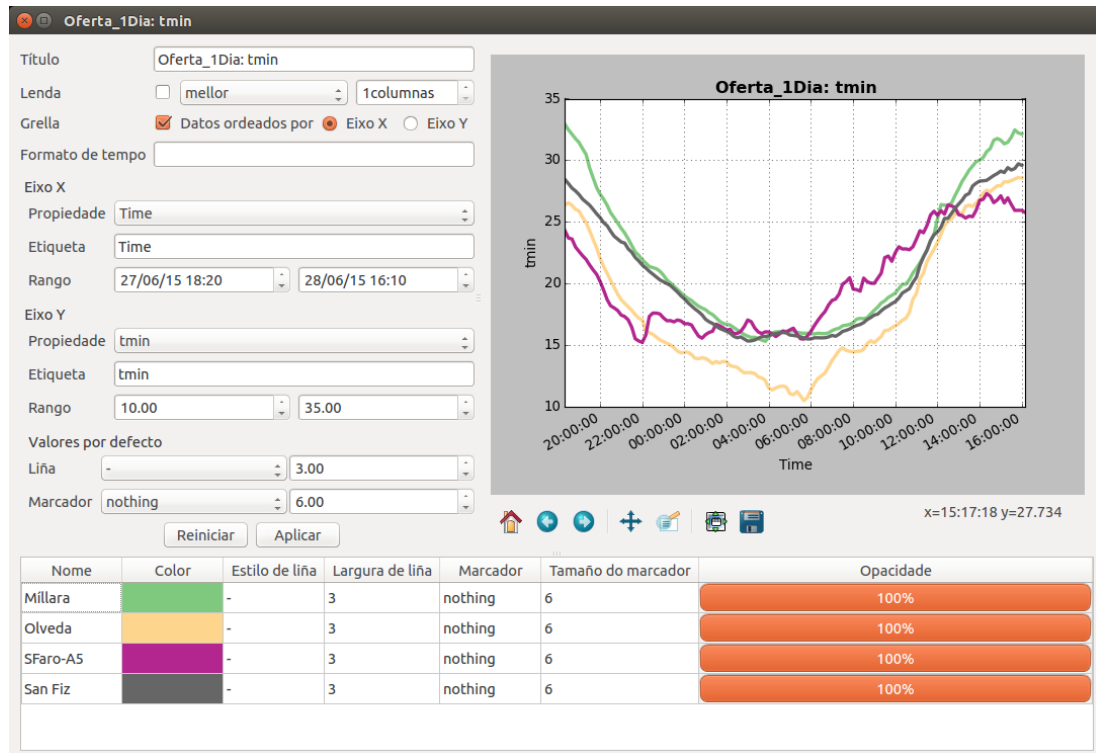


Figura B.8: SOS Plot: Gráfica de varias series

Neste formulario pódese editar o título do gráfico e dos eixos, a propiedade a representar en cada eixo e os límites dos mesmos, o formato no que representar o tempo, a inclusión dunha lenda, e o estilo e cor de liña e marcador de cada unha das series debuxadas. Amais sobre a gráfica pódese facer zoom, desprazala e gardar a imaxe.

Bibliografía

- [1] Pyqgis developer cookbook. http://docs.qgis.org/2.2/en/docs/pyqgis_developer_cookbook/.
- [2] Qgis api documentation. <http://qgis.org/api/2.2/>.
- [3] *A Guide to the Project Management Body of Knowledge*. Project Management Institute, 3^a edition, 2003.
- [4] Laboratorio Nacional de Calidad del Software. *Guía práctica de gestión de configuración*. INTECO, 2008.
- [5] Laboratorio Nacional de Calidad del Software. *Guía práctica de gestión de requisitos*. INTECO, 2008.
- [6] Ministerio de trabajo e inmigración. Xvi convenio colectivo estatal de empresas de consultoría y estudios de mercados y de la opinión pública. <http://www.boe.es/boe/dias/2009/04/04/pdfs/BOE-A-2009-5688.pdf>, 2009.
- [7] Xefatura do estado. Lei 27/2014, do 27 de novembro, do imposto sobre sociedades. https://www.boe.es/boe_gallego/dias/2014/11/28/pdfs/BOE-A-2014-12328-G.pdf, 2014.
- [8] Matt Stephens Doug Rosenberg. *Use Case Driven Object Modeling with UML Theory and Practice*. Apress, 2007.
- [9] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007.
- [10] Juan Palacio. *Gestión de proyectos con Scrum Manager*. Scrum Manager, 2014.
- [11] Harry J.W. Percival. *Test-Driven Development with Python*,. O'Reilly Media, Inc., 2014.
- [12] M. Rizwan Jameel Qureshi and Fatima Sabir. A comparison of model view controller and model view presenter. *CoRR*, abs/1408.5786, 2014.

- [13] Seguridad Social. Bases y tipos de cotización 2015. http://www.seg-social.es/Internet_1/Trabajadores/CotizacionRecaudaci10777/Basesytiposdecotiza36537/index.htm, 2015 (consultado o 23 de Xuño).
- [14] tecnoempleo.com. Informe empleo informática. <http://www.tecnoempleo.com/informe-empleo-informatica.php>, 2015 (consultado o 23 de Xuño).
- [15] Sandro Tosi. *Matplotlib for Python Developers*. Packt Publishing Ltd., Novembro 2009.
- [16] USC. Regulamento do traballo fin de grao. http://www.usc.es/etse/files/u1/RegulamentoTFG_GrEI_CG_30xan2014.pdf, 2014.