

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



# Resumo

TODO



# Índice xeral

<b>1. Introducción</b>	<b>1</b>
1.1. Contextualización . . . . .	1
1.2. Motivación e Obxectivos . . . . .	1
1.3. Estrutura da memoria . . . . .	1
<b>2. Xestión do proxecto</b>	<b>3</b>
2.1. Alcance do proxecto . . . . .	3
2.1.1. Definición do alcance . . . . .	3
2.1.2. Estrutura de Descomposición do Traballo . . . . .	4
2.2. Metodoloxía . . . . .	6
2.2.1. Aplicación da metodoloxía Scrum . . . . .	6
2.3. Xestión do tempo . . . . .	7
2.3.1. Planificación temporal . . . . .	7
2.3.2. Sprint 0 . . . . .	9
2.3.3. Sprints 1, 2 e 3 . . . . .	10
2.3.4. Sprint release . . . . .	10
2.4. Xestión do custo . . . . .	10
2.4.1. Custo de recursos humanos . . . . .	11
2.4.2. Custo de recursos materiais . . . . .	11
2.4.3. Custo de recursos software . . . . .	11
2.4.4. Presuposto . . . . .	11
2.5. Xestión de riscos . . . . .	12
2.5.1. Especificación de riscos . . . . .	13
2.6. Xestión da configuración . . . . .	15
2.6.1. Control do código fonte . . . . .	16
2.6.2. Control dos entregables . . . . .	16
<b>3. Análise de requisitos</b>	<b>17</b>
3.1. Casos de uso . . . . .	17
3.2. Catálogo de requisitos . . . . .	19
3.2.1. Requisitos funcionais . . . . .	20
3.2.2. Requisitos non funcionais . . . . .	23
3.3. Matriz de trazabilidade . . . . .	25

<b>4. Deseño de software</b>	<b>27</b>
4.1. Arquitectura do sistema . . . . .	27
4.1.1. Patrón de arquitectura . . . . .	27
4.2. Estrutura de clases . . . . .	29
4.3. Interacción entre compoñentes . . . . .	32
<b>5. Implementación e probas</b>	<b>35</b>
5.1. Sprint 1 . . . . .	35
5.2. Sprint 2 . . . . .	37
5.3. Sprint 3 . . . . .	39
<b>6. Conclusións e traballo futuro</b>	<b>41</b>
6.1. Conclusións . . . . .	41
6.2. Traballo futuro . . . . .	42
<b>A. Manual de usuario</b>	<b>43</b>
A.1. Instalación . . . . .	43
A.2. Crear capa de observacións . . . . .	44
A.3. Gráficos en dúas dimensións . . . . .	50
<b>Bibliografía</b>	<b>53</b>



# Índice de figuras

2.1. Estrutura de Descomposición do Traballo . . . . .	5
2.2. Ciclo da metodoloxía Scrum . . . . .	7
2.3. Diagrama de Gantt . . . . .	8
3.1. Diagrama de casos de uso . . . . .	19
4.1. Diagrama de compoñentes . . . . .	28
4.2. Diferencias entre MVC e Model/View . . . . .	29
4.3. Diagrama de clases da compoñente SOS Client . . . . .	30
4.4. Diagrama de clases da compoñente SOS Plot . . . . .	31
4.5. Diagrama de secuencia para os casos de uso CU.01 e CU.02 . . . .	33
4.6. Diagrama de secuencia para o caso de uso CU.03 . . . . .	33
4.7. Diagrama de secuencia para o caso de uso CU.04 . . . . .	34
A.1. Pantalla de instalación do plugin . . . . .	43
A.2. Barra de ferramentas do plugin . . . . .	44
A.3. Diálogo de conexión co servidor, sen datos . . . . .	45
A.4. Diálogo de conexión co servidor, con datos . . . . .	46
A.5. Ferramenta de selección espacial . . . . .	48
A.6. Engadir capa SOS . . . . .	49
A.7. Executar ferramenta SOS Plot . . . . .	50
A.8. SOS Plot: Gráfica de varias series . . . . .	51



# Índice de cadros

2.1. Pila inicial do produto . . . . .	9
2.2. Previsión de historias a incluír en cada sprint . . . . .	10
2.3. Custos totais . . . . .	12
2.4. Nivel de exposición dun risco en base a Probabilidade e Impacto .	12
3.1. Niveis de relevancia dun requisito . . . . .	20
3.2. Matriz de trazabilidade de requisitos . . . . .	25
A.1. Lapelas do formulario de consulta ó SOS . . . . .	47



# Capítulo 1

## Introdución

a)Introdución: composta por Obxectivos Xerais, Relación da Documentación que conforma a Memoria, Descrición do Sistema, Información Adicional de Interese (métodos, técnicas ou arquitecturas utilizadas, xustificación da súa elección, etc.).

### 1.1. Contextualización

### 1.2. Motivación e Obxectivos

### 1.3. Estrutura da memoria

Este documento estrutúrase en 6 capítulos, un apéndice 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 desde o punto de vista estático e dinámico.

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

No capítulo 6 expóñense as conclusións extraídas de todo o proxecto e detállanse varias propostas de traballo futuro.

O apéndice A 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 só 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, sin 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* empacquetado 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 [12].

#### 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.



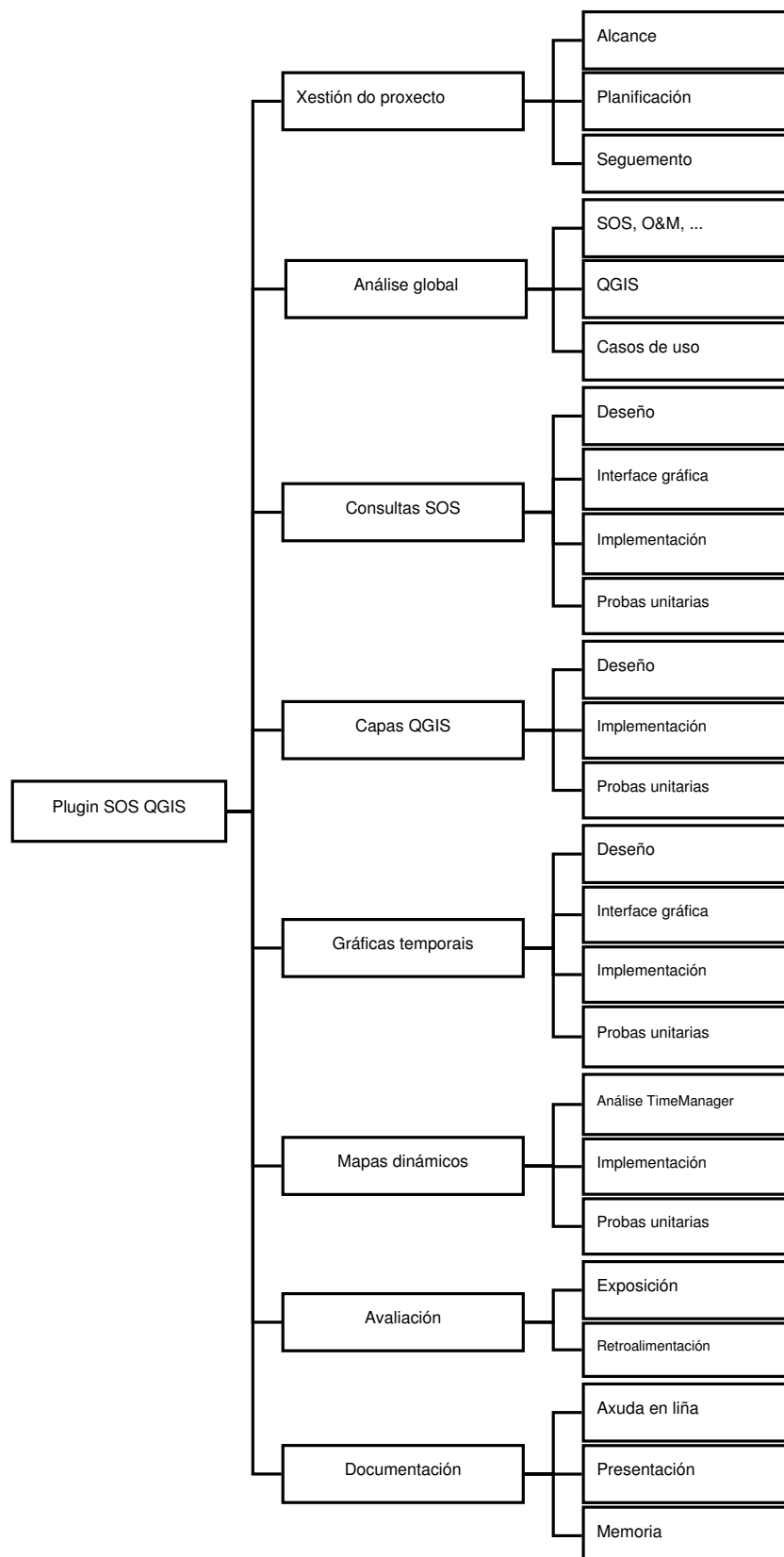


Figura 2.1: Estructura de Descomposición do Trabalho

## 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*[9], por ser unha das máis utilizadas e porque o equipo de desenvolvemento xa está familiarizado con ela.

### 2.2.1. Aplicación da metodoloxía Scrum

So 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 si 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*.

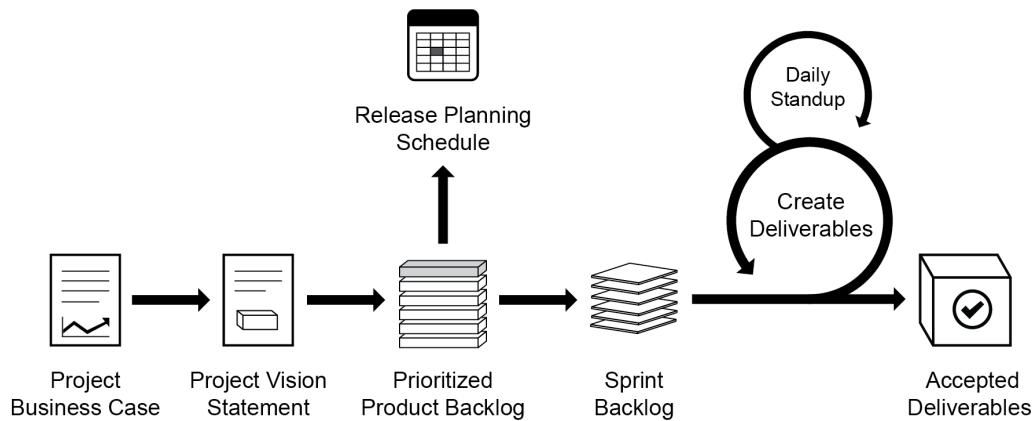


Figura 2.2: Ciclo da metodoloxía Scrum

Fonte: [https://commons.wikimedia.org/wiki/File:Scrum\\_Flow\\_for\\_one\\_Sprint.png](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.

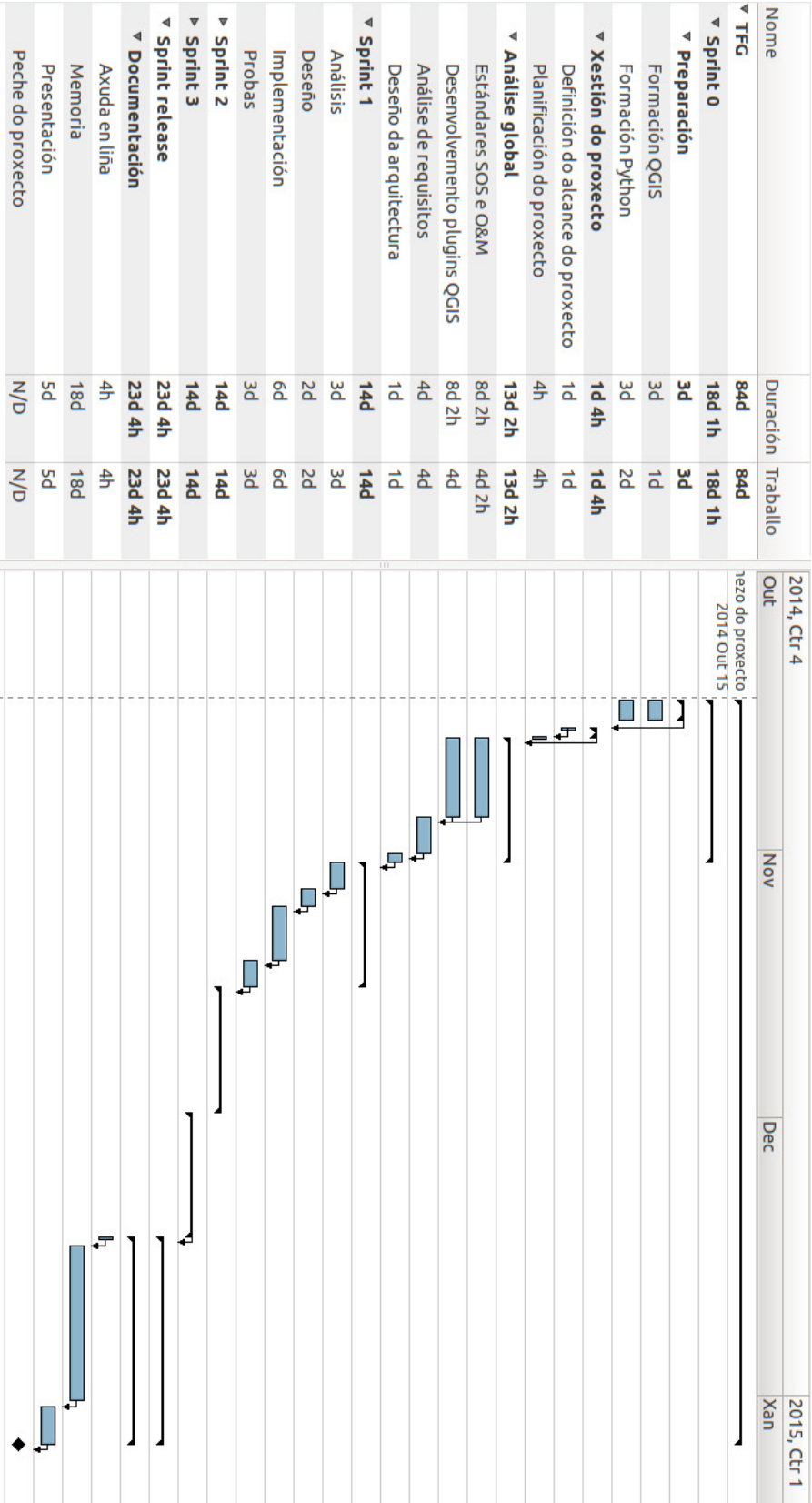


Figura 2.3: Diagrama de Gantt

#	Historia 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 do estándar SOS e outros relacionados (Observations&Measurements e SensorML) 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 requisitos e casos de uso, así como o deseño preliminar da arquitectura do software a desenvolver.

#	Historia de usuario	Horas
<b>Sprint 1</b>		
1	Conexión con servidor SOS	25
2	Procesamento do xml de capacidades do servizo	30
3	Xeración de peticións básicas	15
<b>Sprint 2</b>		
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
<b>Sprint 3</b>		
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 4).

## 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[11] 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[10] 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 IDE 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 €
<b>Total</b>			<b>7489,40 €</b>

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 clasificándoo en base á fonte do risco.

#### Riscos técnicos

##### R.TEC.01.- COMPLEXIDADE DO ESTÁNDAR SOS

<b>Descrición</b>	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.
<b>Probabilidade</b>	Moi probable
<b>Impacto</b>	Alto
<b>Nivel de exposición</b>	Alto
<b>Resposta ao risco</b>	Só se comprometerá como imprescindible soportar as operacións básicas para obter os datos de observacións provistos pola implementación SOS realizada polo CiTIUS <sup>1</sup> .

##### R.TEC.02.- ESCASEZA DE SERVIDORES SOS

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

---

<sup>1</sup><https://citius.usc.es/>

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

<b>Descrición</b>	Implementación nativa dentro do QGIS do soporte para SOS ou a publicación de algún plugin que implemente dito soporte.
<b>Probabilidade</b>	Moi probable
<b>Impacto</b>	Alto
<b>Nivel de exposición</b>	Alto
<b>Resposta ao risco</b>	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 a implementación realizada polo CiTIUS.

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

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

**R.PER.02.- DESCOÑECEMENTO DA LINGUAXE PYTHON**

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

<b>Resposta ao risco</b>	No plan de traballo do proxecto inclúese capacitación na linguaxe de programación Python.
--------------------------	---

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

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

#### Riscos na xestión do proxecto

#### R.XES.01.- ERROR NA ESTIMACIÓN TEMPORAL

<b>Descrición</b>	A estimación temporal inicial para a execución das tarefas é pouco precisa a causa da inexperiencia en proxectos similares.
<b>Probabilidade</b>	Case seguro
<b>Impacto</b>	Medio
<b>Nivel de exposición</b>	Alto
<b>Resposta ao risco</b>	A estimación inicial realizase de forma pesimista. A metodoloxía <i>Scrum</i> minimiza o impacto ó permitir a refinación de requisitos e estimacións o 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 do 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*<sup>2</sup>. As carpetas dos repositorios locais replícanse en *Dropbox*<sup>3</sup> 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.

---

<sup>2</sup><https://github.com/>

<sup>3</sup><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 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 extraense 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 sistema para o actor, entendendo por actor calquera entidade externa o sistema que lle demanda 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 co cliente, e que se resumen graficamente na figura 3.1.

#### CU.01.- CONSULTAR AS CAPACIDADES DUN SERVIDOR SOS

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

<b>Poscondicións</b>	Visualízanse as capacidades do SOS nun formato amigable para o usuario.
<b>Escenario</b>	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

<b>Propósito</b>	Xerar unha capa vectorial cos datos das observacións rexistradas no SOS.
<b>Actores</b>	Usuario
<b>Relacións</b>	Estende o CU.01
<b>Precondicións</b>	Ter consultado as capacidades do servizo.
<b>Poscondicións</b>	Capa vectorial coas observacións obtidas.
<b>Escenario</b>	O usuario crea 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

<b>Propósito</b>	Visualizar nunha gráfica en dúas dimensións a información das observacións a partir da capa xerada. Débese poder visualizar a evolución de unha propiedade o con respecto ó tempo, e tamén sería de utilidade en algúns caso poder enfrontar unha propiedade contra outra.
<b>Actores</b>	Usuario
<b>Relacións</b>	-
<b>Precondicións</b>	Ter xerada unha capa con datos de observacións.
<b>Poscondicións</b>	Gráfico en dúas dimensións dos datos seleccionados.
<b>Escenario</b>	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

<b>Propósito</b>	Facer visualizacións animadas dos datos da capa sobre o propio visor de mapas de QGIS.
<b>Actores</b>	Usuario
<b>Relacións</b>	-
<b>Precondicións</b>	Ter xerada unha capa con datos de observacións.
<b>Poscondicións</b>	Visualización animada das observacións.
<b>Escenario</b>	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.

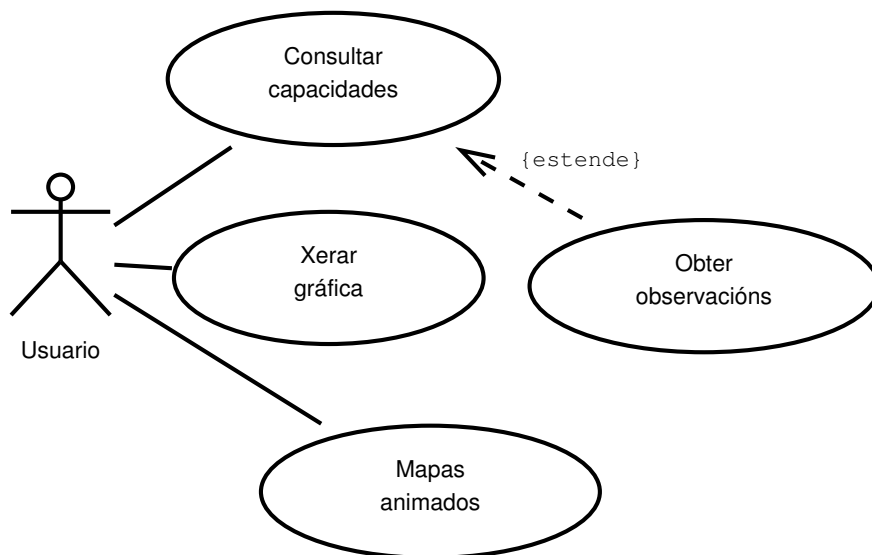


Figura 3.1: Diagrama de casos de uso

## 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 os 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

<b>Descrición</b>	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.
<b>Relevancia</b>	Esixido
<b>Criterio de validación</b>	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

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



<b>Criterio de validación</b>	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

<b>Descrición</b>	O usuario poderá visualizar en formato texto, con resaltado de sintaxe, e nun visor de árbore o XML que describe as capacidades do servidor.
<b>Relevancia</b>	Desexado
<b>Criterio de validación</b>	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

<b>Descrición</b>	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.
<b>Relevancia</b>	Esixido
<b>Criterio de validación</b>	O requisito cumprirase se a aplicación descarga os datos solicitados do servidor e os procesa XML creando 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

<b>Descrición</b>	O usuario poderá indicar o rango de tempo no que lle interesan as observacións, así como un ou máis sensores e unha ou máis entidades.
<b>Relevancia</b>	Esixido
<b>Criterio de validación</b>	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

<b>Descrición</b>	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.
<b>Relevancia</b>	Desexado
<b>Criterio de validación</b>	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

<b>Descrición</b>	O usuario poderá modificar en modo texto o XML xerado a través da interface gráfica, para obter as observacións desexadas.
<b>Relevancia</b>	Opcional
<b>Criterio de validación</b>	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

<b>Descrición</b>	O usuario poderá, a partir dunha entidade seleccionada no mapa, visualizar unha gráfica que mostre unha liña coa evolución temporal da propiedade observada.
<b>Relevancia</b>	Esixido
<b>Criterio de validación</b>	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

<b>Descrición</b>	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.
-------------------	---

<b>Relevancia</b>	Desexado
<b>Criterio de validación</b>	O requisito cumprirase se a aplicación

#### RF.10.- XERAR GRÁFICA CON VARIAS SERIES

<b>Descrición</b>	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.
<b>Relevancia</b>	Desexado
<b>Criterio de validación</b>	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

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

<b>Descrición</b>	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.
<b>Relevancia</b>	Esixido

<b>Criterio de validación</b>	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

<b>Descrición</b>	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.
<b>Relevancia</b>	Desexado
<b>Criterio de validación</b>	A aplicación programarase en linguaxe Python.

#### RNF.03.- CONSISTENCIA DA INTERFACE GRÁFICA

<b>Descrición</b>	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.
<b>Relevancia</b>	Desexado
<b>Criterio de validación</b>	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

<b>Descrición</b>	O usuario terá a súa disposición un manual de uso da aplicación.
<b>Relevancia</b>	Esixido
<b>Criterio de validación</b>	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

<b>Descrición</b>	A aplicación deberá ser deseñada de xeito que poida ser adaptada para outras linguas sen a necesidade de facer cambios a nivel de código.
<b>Relevancia</b>	Opcional

<b>Criterio de validación</b>	O requisito cumprírase se o <i>plugin</i> pode ser traducido a distintos idiomas sen modificar o código fonte.
-------------------------------	--

#### RNF.06.- DATA LÍMITE DE EXECUCIÓN DO PROXECTO

<b>Descrición</b>	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 <sup>1</sup> .
<b>Relevancia</b>	Esixido
<b>Criterio de validación</b>	-

### 3.3. Matriz de trazabilidade

A matriz de trazabilidade relaciona cada requisito co seu caso de uso de orixe. 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

<sup>1</sup>[http://www.usc.es/etse/files/u1/datasdefensa14\\_15GREI.pdf](http://www.usc.es/etse/files/u1/datasdefensa14_15GREI.pdf)



# Capítulo 4

## Deseño de software

O longo deste capítulo detállase a arquitectura do sistema a desenvolver. Mostrase primeiro a arquitectura xeral e a súa interacción con sistemas externos e posteriormente detallase máis polo miúdo a estrutura das distintas compoñentes do sistema e como cooperan entre elas para resolver cada un dos casos de uso definidos.

### 4.1. Arquitectura do sistema

Dende a definición dos obxectivos do proxecto se identifican 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, na que tamén se inclúen os compoñentes externos cos que se relacionan.

A compoñente **SOS Client** é a responsable da comunicación co servidor SOS tanto a hora de xerar as peticións necesarias a partir da información proporcionada polo usuario e 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.

#### 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

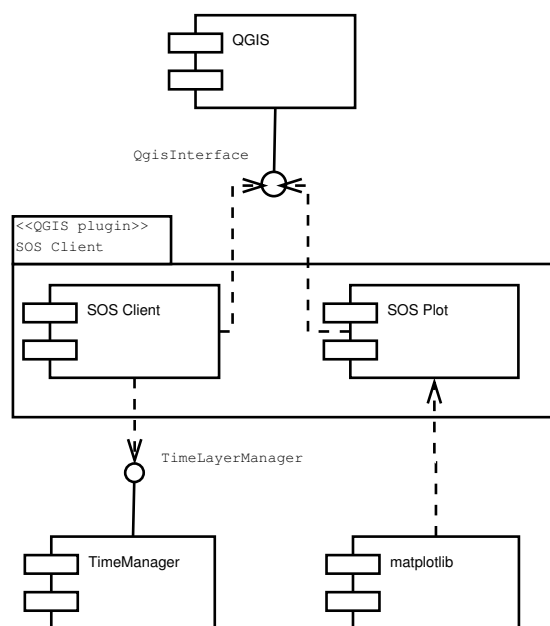


Figura 4.1: Diagrama de compoñentes

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*<sup>1</sup>. 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 se incorpore a lóxica de negocio nas vistas, si que se mantén a separación entre a creación e configuración dos compoñentes 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.

<sup>1</sup><http://doc.qt.io/qt-4.8/model-view-programming.html>



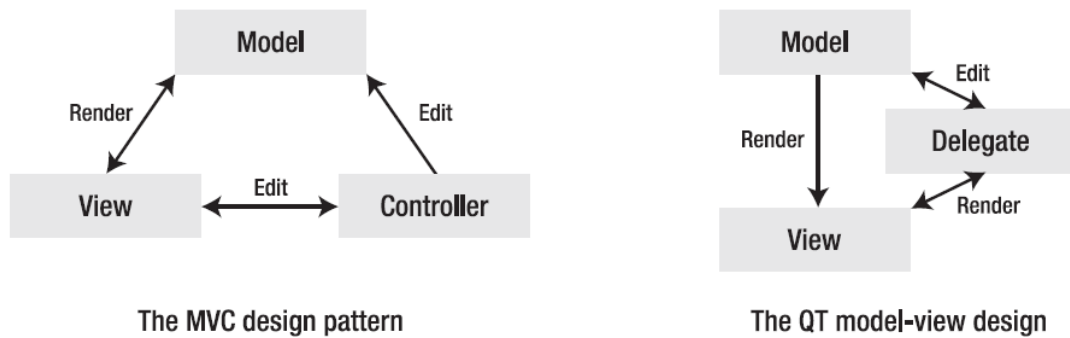


Figura 4.2: Diferencias entre MVC e Model/View

## 4.2. Estructura de clases

A continuación descríbese máis en detalle cada unha das compoñentes do sistema a desenvolver, representando as clases máis relevantes de cada unha. Nos diagramas de clase so se mostran os atributos e métodos máis relevantes para entender a función de cada unha delas.

No diagrama 4.3 represéntanse as clases correspondentes a compoñente SOS Client, separadas entre as que corresponde á vista e as que corresponden ó modelo. Represéntanse agrupadas sobre fondo escuro as clases encargadas do procesamento dos documentos XML proporcionados polo servizo SOS.

As clases máis relevantes da compoñente SOSClient son:

**WidgetFactory:** Esta clase é unha factoría que crea a partir dos XML de definición da interface a clase 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 as 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 QgsVectorDataProvider<sup>2</sup>. É pois a clase que fai de intermediaria entre a estrutura de datos real da información da capa

<sup>2</sup><http://qgis.org/api/classQgsVectorDataProvider.html>

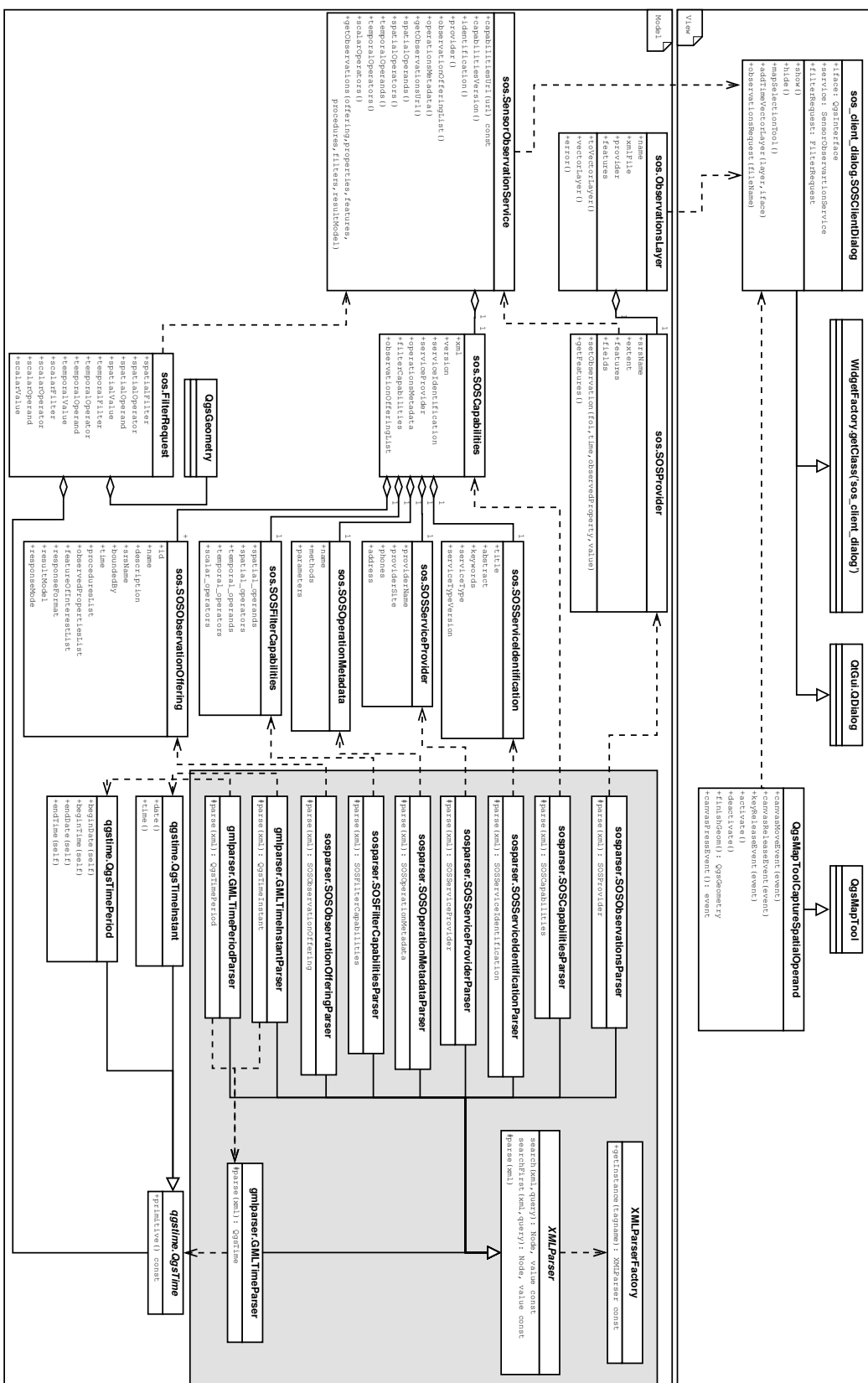


Figura 4.3: Diagrama de clases da componente SOS Client

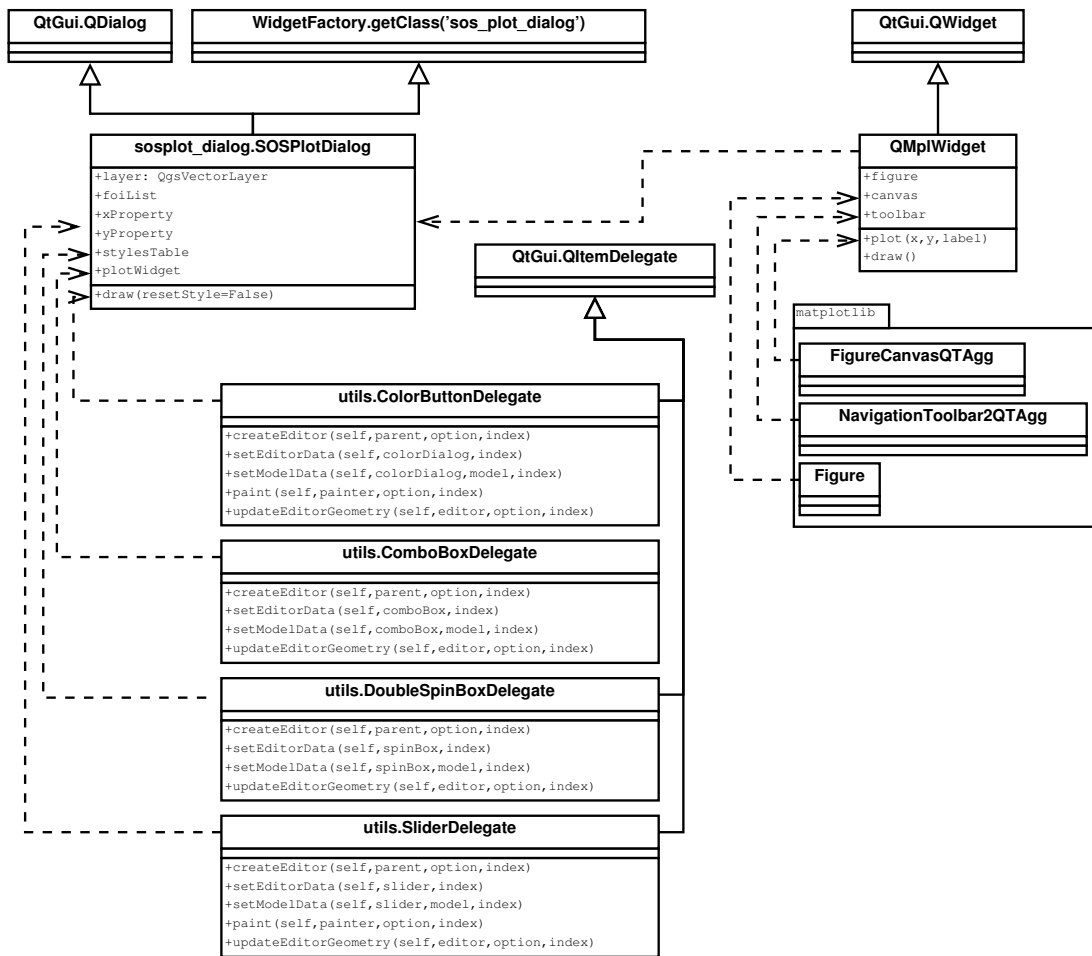


Figura 4.4: Diagrama de clases da compoñente SOS Plot

e a estrutura manexada polo QGIS.

**XMLParserFactory:** Esta clase é unha factoría para crear as distintas clase 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.

No diagrama 4.4 represéntanse as clases correspondentes a compoñente SOS Plot. Móstranse agrupadas todas as clases que cumpren co rol de *Delegate*, e todas as demais forman parte da vista. Neste módulo o modelo non está representado no diagrama pois está composto por clases internas do QGIS (*QgsVectorLayer*) e de matplotlib (*Line2D*)

As clases máis relevantes da compoñente SOSPlot son:

**WidgetFactory:** É a mesma clase xa descrita no punto anterior, que actúa como factoría das compoñentes gráficas.

**SOSPlotDialog:** Es el 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<sup>3</sup> para a visualización de gráficos.

### 4.3. 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.

No diagrama 4.5 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.

O diagrama 4.6 representa o comportamento 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 4.7 é o procedemento para que a capa xerada coas observacións sexa controlada polo *TimeManager*.

---

<sup>3</sup><http://matplotlib.org/>

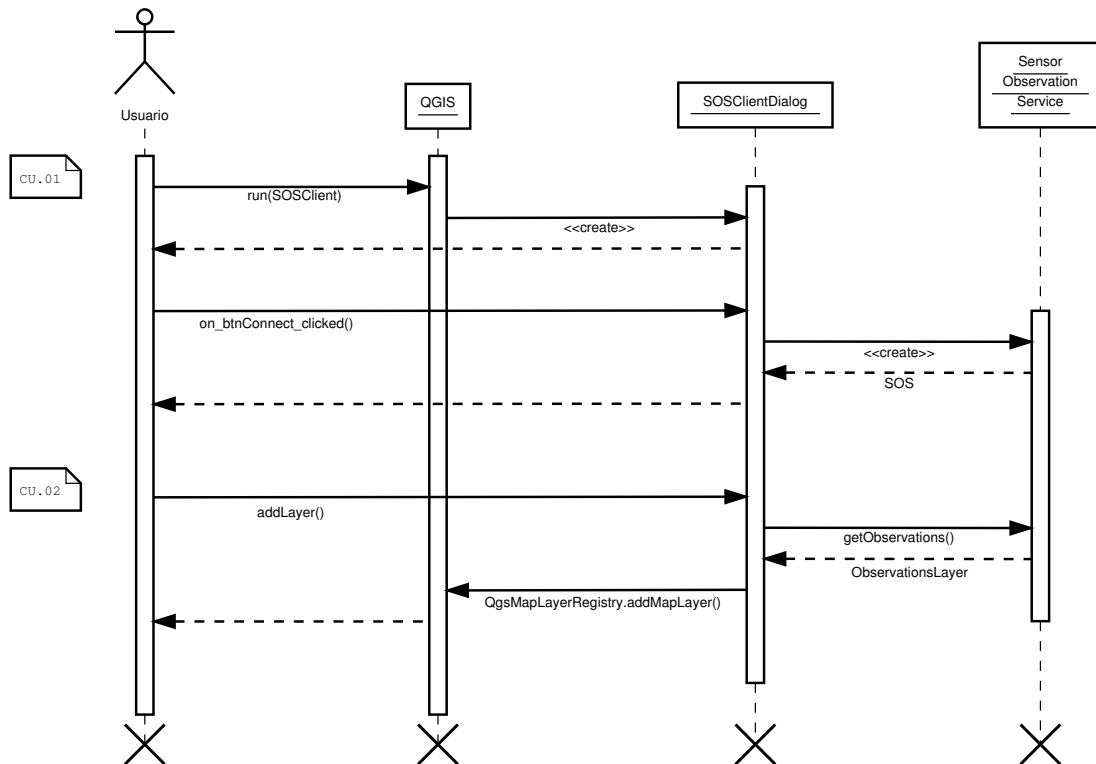


Figura 4.5: Diagrama de secuencia para os casos de uso CU.01 e CU.02

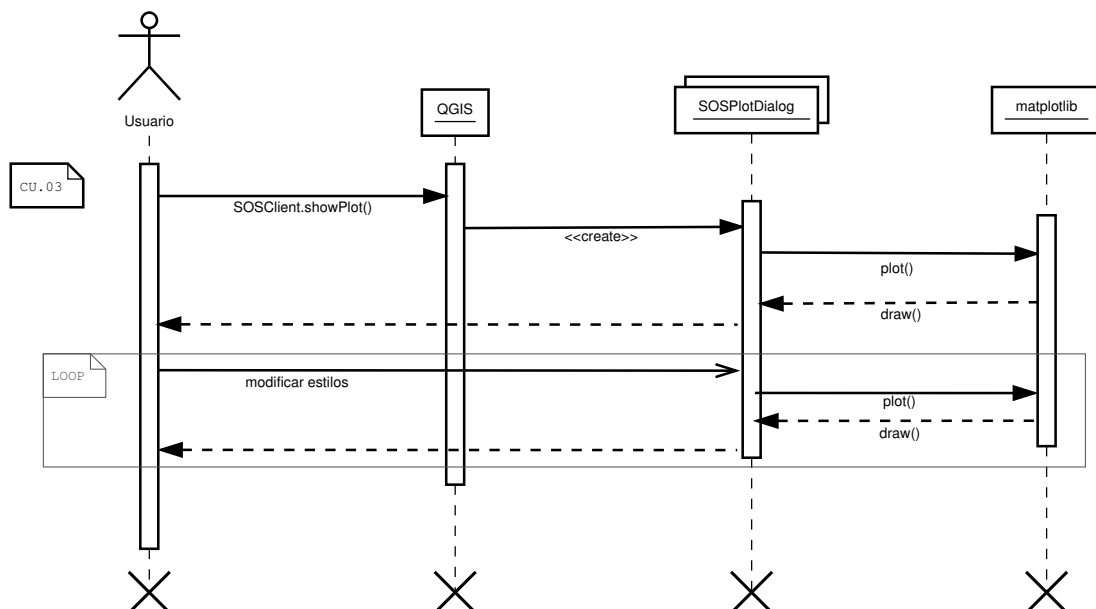


Figura 4.6: Diagrama de secuencia para o caso de uso CU.03

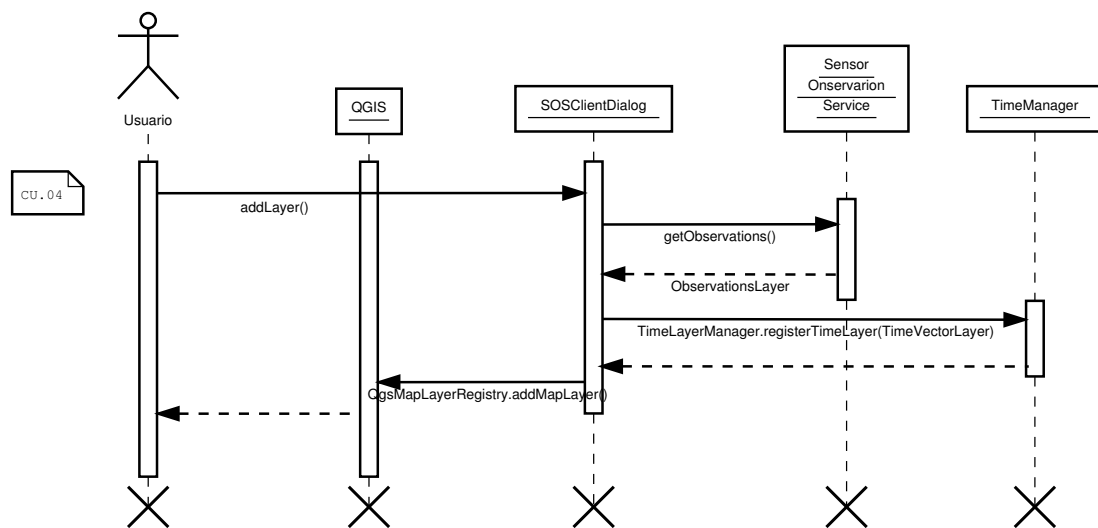


Figura 4.7: Diagrama de secuencia para o caso de uso CU.04

# Capítulo 5

## Implementación e probas

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 fase de análise, deseño, implementación e probas. Dado que a análise e o deseño se describiron por completo nos capítulos 3 e 4, neste capítulo detalláranse a 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 cas 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 que vaian aparecendo, debido a gran liberdade que dan os estándares SOS e Observations&Measurements. Tamén se analizaron distintas posibilidades

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

Tamén se preparou durante este sprint o entorno para a automatización das probas unitarias a través da ferramenta *PyUnit* 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:

#### PR.01.- CONEXIÓN CON SERVIDOR SOS

<b>Requisitos relacionados</b>	<ul style="list-style-type: none"> <li>■ RF.01</li> <li>■ RNF.03</li> </ul>
<b>Descrición</b>	Compróbase, a parte das probas unitarias automáticas, a conexión con varios servidores SOS.
<b>Resultado</b>	<ul style="list-style-type: none"> <li>■ A información das capacidades do servizo visualízanse en formato HTML cos estilos propios de QGIS.</li> <li>■ 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.</li> </ul>

#### PR.02.- XESTIÓN DE SERVIDORES

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

#### PR.03.- DESCARGAR OBSERVACIÓN

<b>Requisitos relacionados</b>	<ul style="list-style-type: none"> <li>■ RF.04</li> </ul>
<b>Descrición</b>	Compróbase a execución da operación <i>GetObservations</i> contra varios servidores.
<b>Resultado</b>	<ul style="list-style-type: none"> <li>■ O XML descárgase correctamente.</li> </ul>



- 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 do 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* e deseñase a estrutura de clases.

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

<b>Descrición</b>	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 CíTIUS.
<b>Resultado</b>	<ul style="list-style-type: none"> <li>■ O XML procesase correctamente, detectando os campos de tipo tempo, os numéricos e as cadeas de texto.</li> <li>■ 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.</li> </ul>

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

<b>Requisitos relacionados</b>	<ul style="list-style-type: none"> <li>■ RF.03</li> <li>■ RF.05</li> <li>■ RF.07</li> </ul>
<b>Descrición</b>	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.
<b>Resultado</b>	<ul style="list-style-type: none"> <li>■ Comprobase conectando con varios servidores que as opcións habilitadas nas distintas lapelas de filtros se corresponden coas capacidades definidas no XML.</li> <li>■ A lista de sensores seleccionados e a de entidades de interese, así como o operador e operandos dos filtros de tempo se xeran correctamente no XML para a petición <i>GetObservations</i>.</li> </ul>

#### PR.06.- XESTIONAR CAPA CO *TimeManager*

<b>Requisitos relacionados</b>	<ul style="list-style-type: none"> <li>■ RF.04</li> <li>■ RF.11</li> </ul>
<b>Descrición</b>	Comprobar o correcto funcionamento do plugin <i>TimeManager</i> coa capa creada.
<b>Resultado</b>	<ul style="list-style-type: none"> <li>■ Comprobase que a capa se engada a lista das xestionadas polo TimeManager de xeito satisfactorio.</li> </ul>

- Verifícase que o campo de tempo se interpreta correctamente.
- Comprobase o funcionamento xeral do *TimeManager* con varias capas de observacións.

#### PR.07.- GRÁFICA TEMPO *vs* PROPIEDAD

<b>Requisitos relacionados</b>	■ RF.08
<b>Descrición</b>	Xerar gráfica co tempo no eixo X e a propiedade no Y.
<b>Resultado</b>	<ul style="list-style-type: none"> <li>■ A gráfica contén todas as observacións co mesmo <i>foi</i> que a entidade seleccionada no mapa.</li> <li>■ As observacións ordénanse polo campo tempo de xeito que se podan trazar unha liña que as una.</li> <li>■ Ós valores do eixo X dáselles formato de tempo.</li> </ul>

Tendo en conta a matriz de trazabilidade do cadro 3.2, con este segundo incremento cúbreanse 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.

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

#### PR.08.- FILTROS ESPACIAIS

<b>Requisitos relacionados</b>	■ RF.06
--------------------------------	---------

<b>Descrición</b>	Comprobar o correcto funcionamento da ferramenta de selección de xeometrías e a inclusión no XML do filtro correspondente.
<b>Resultado</b>	<ul style="list-style-type: none"> <li>■ A ferramenta de selección permite seleccionar rectángulos (indicando dous puntos), polígonos, liñas e puntos.</li> <li>■ A xeometría debuxada no mapa transformase correctamente ó formato GML e inclúese no XML para a petición <i>GetObservations</i>.</li> </ul>

#### PR.09.- XERACIÓN DE GRÁFICAS CON VARIAS SERIES

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

#### PR.10.- PERSONALIZACIÓN DAS PROPIEDADES DAS GRÁFICAS

<b>Requirimentos relacionados</b>	<ul style="list-style-type: none"> <li>■ RF.09</li> </ul>
<b>Descrición</b>	Verificar que se poden modificar as distintas propiedades da gráfica, incluídas a propiedade a representar en cada eixo.
<b>Resultado</b>	<ul style="list-style-type: none"> <li>■ 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.</li> <li>■ Permítese modificar os límites dos eixos X e Y manualmente.</li> <li>■ Pódense modificar outras propiedades da gráfica como o título, as etiquetas e o formato de tempo.</li> </ul>

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, e adicionalmente, coa inclusión do *plugin* oficial de QGIS.

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 comparan un eixo, de modo que se poida ver a evolución temporal de dúas propiedades o mesmo tempo que a relación entre elas.

A parte de estas 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 de usuario

### A.1. Instalación

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

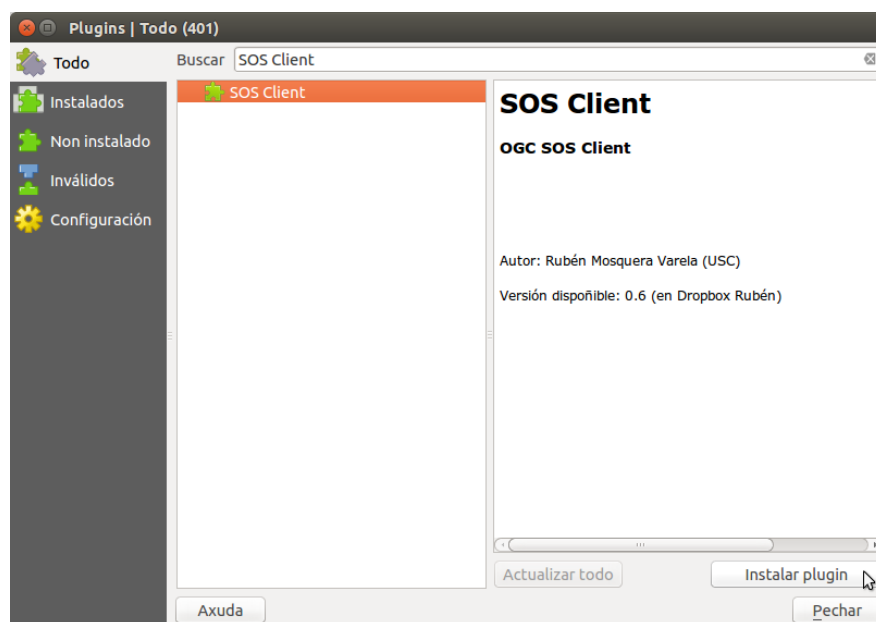


Figura A.1: Pantalla de instalación do plugin

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



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

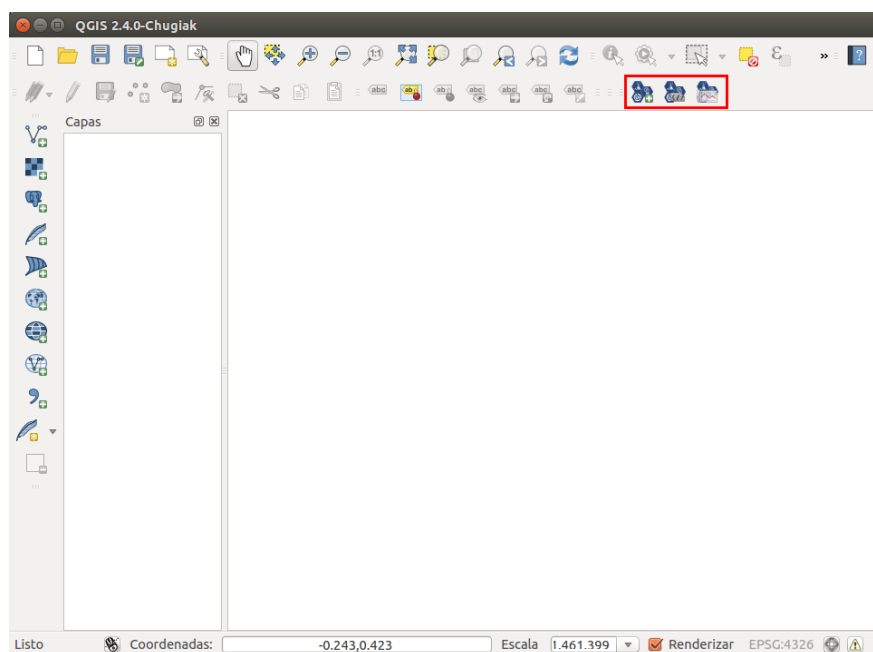


Figura A.2: Barra de ferramentas do plugin




Visualiza o ficheiro XML usado para xerar a capa activa.



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

## A.2. Crear capa de observacións

Ó pulsar a acción  mostrarase o formulario da figura TODO. 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 A.4).

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

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 A.6.



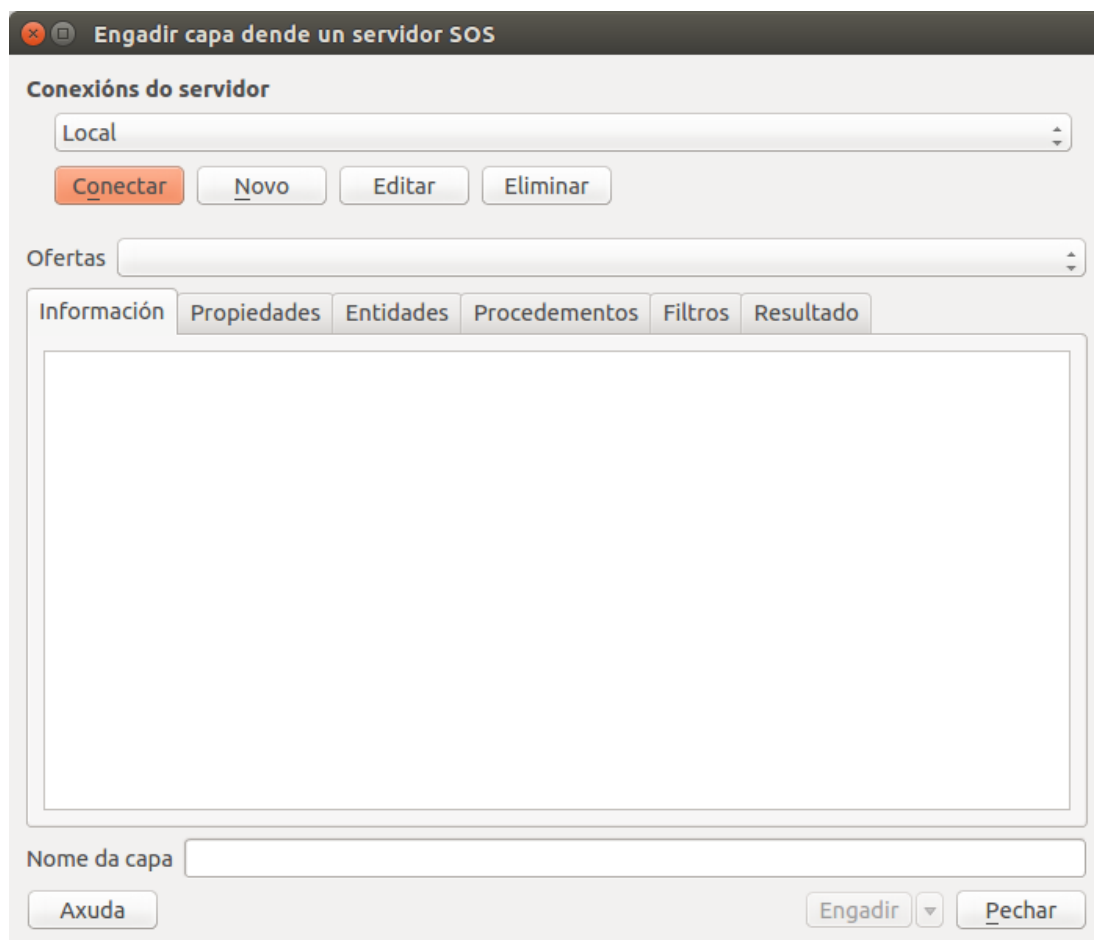


Figura A.3: Diálogo de conexión co servidor, sen datos

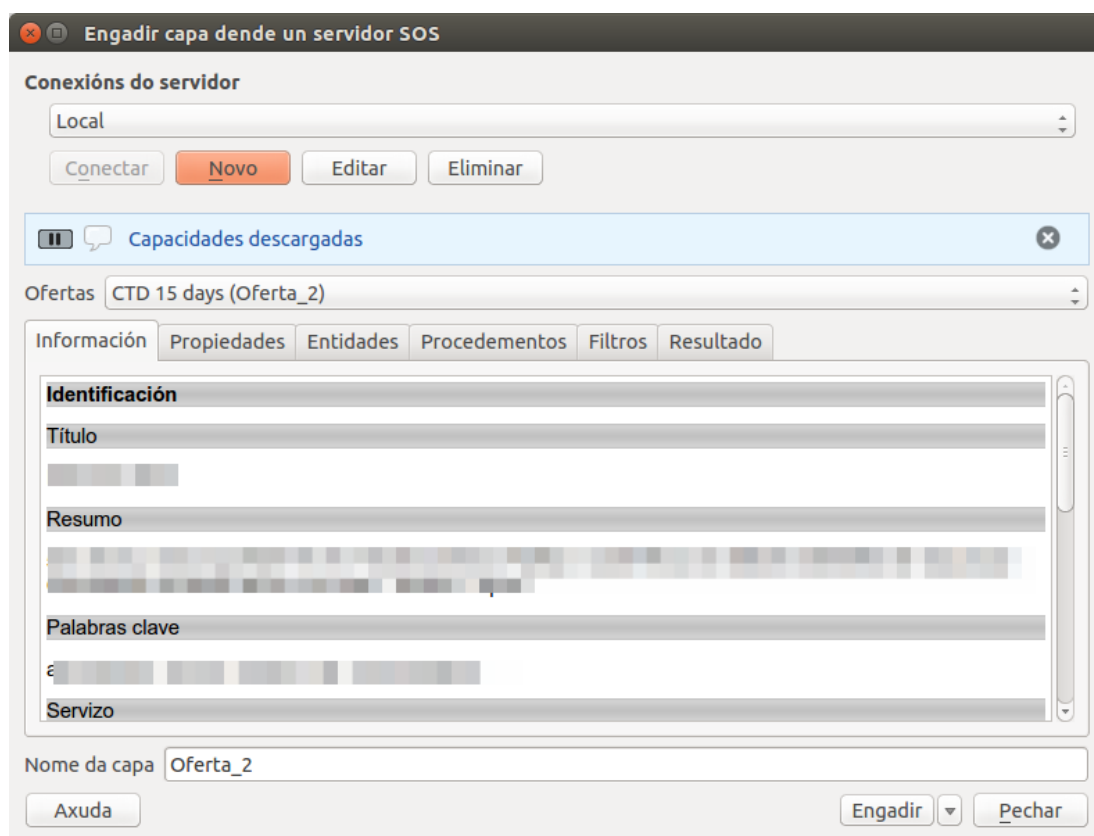
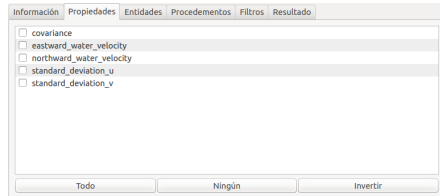
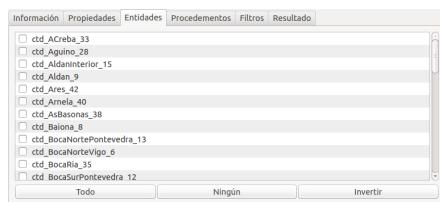


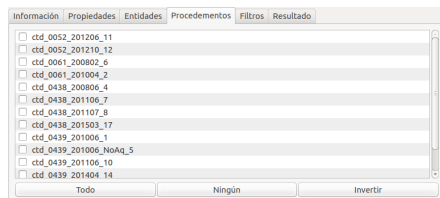
Figura A.4: Diálogo de conexión co servidor, con datos



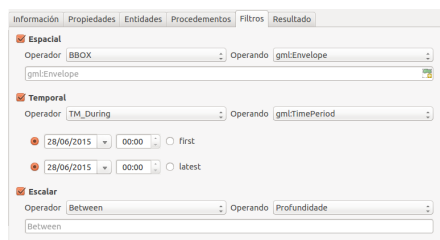
Lista de propiedades da oferta. Pode seleccionar unha ou varias.



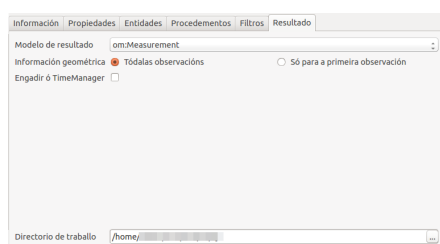
Lista de entidades da oferta. Pode seleccionar varias ou ningunha.



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 A.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 A.1: Lapelas do formulario de consulta ó SOS

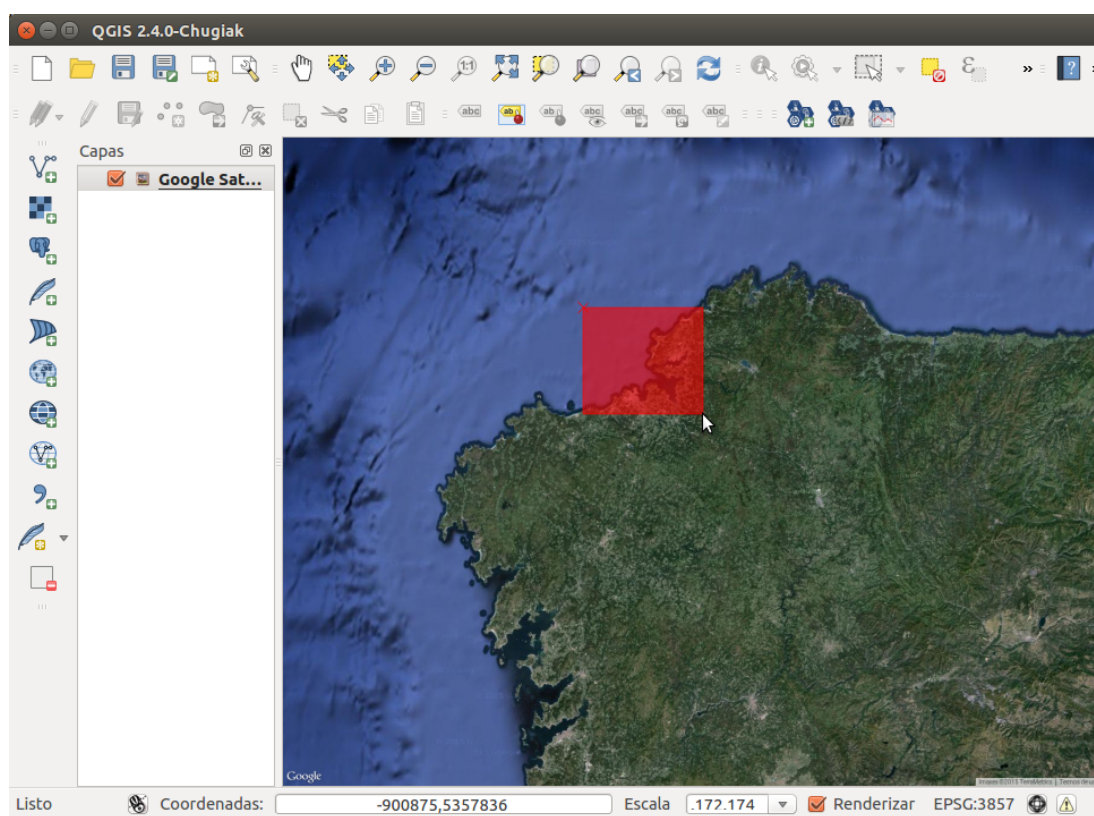


Figura A.5: Ferramenta de selección espacial

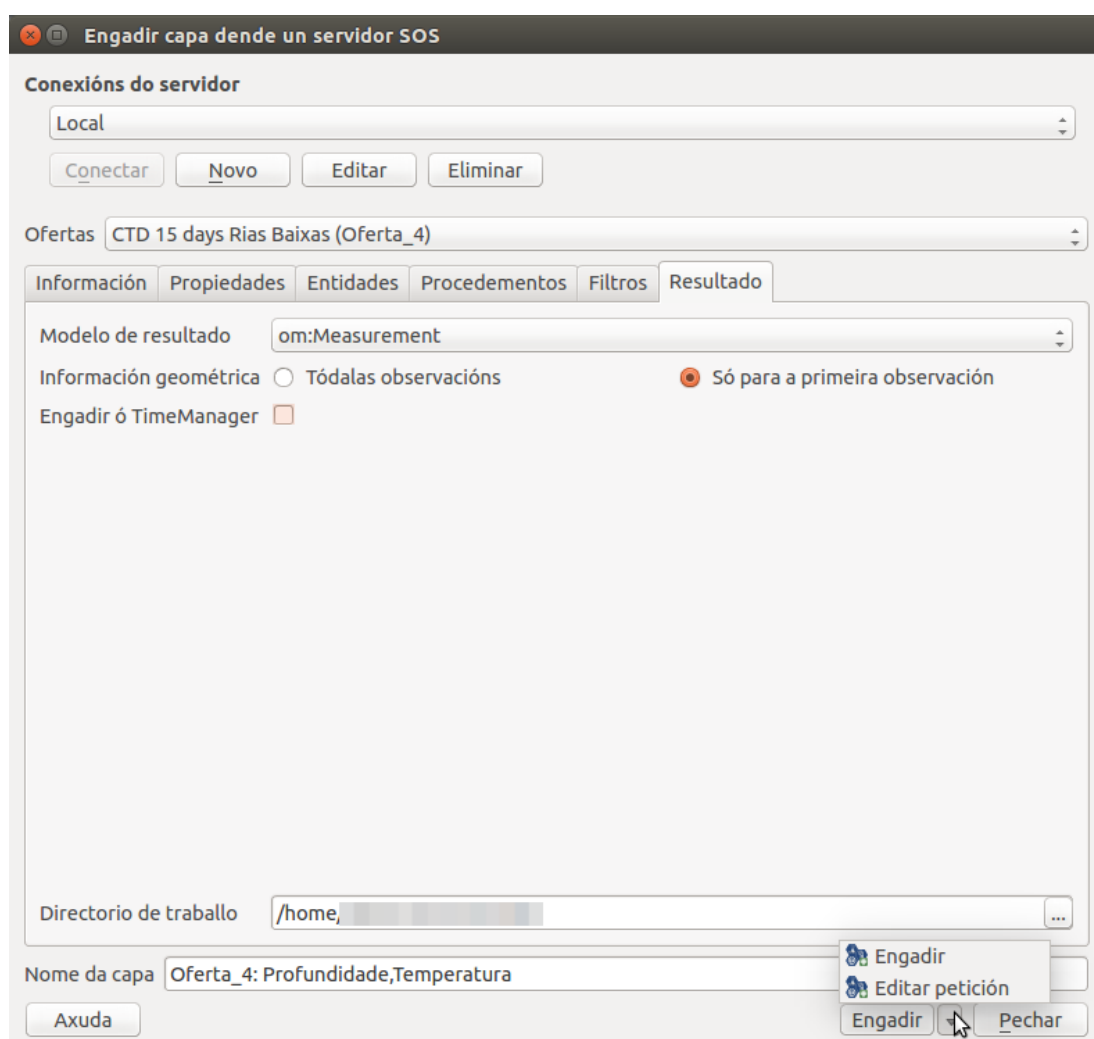



Figura A.6: Engadir capa SOS

### A.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 A.7.

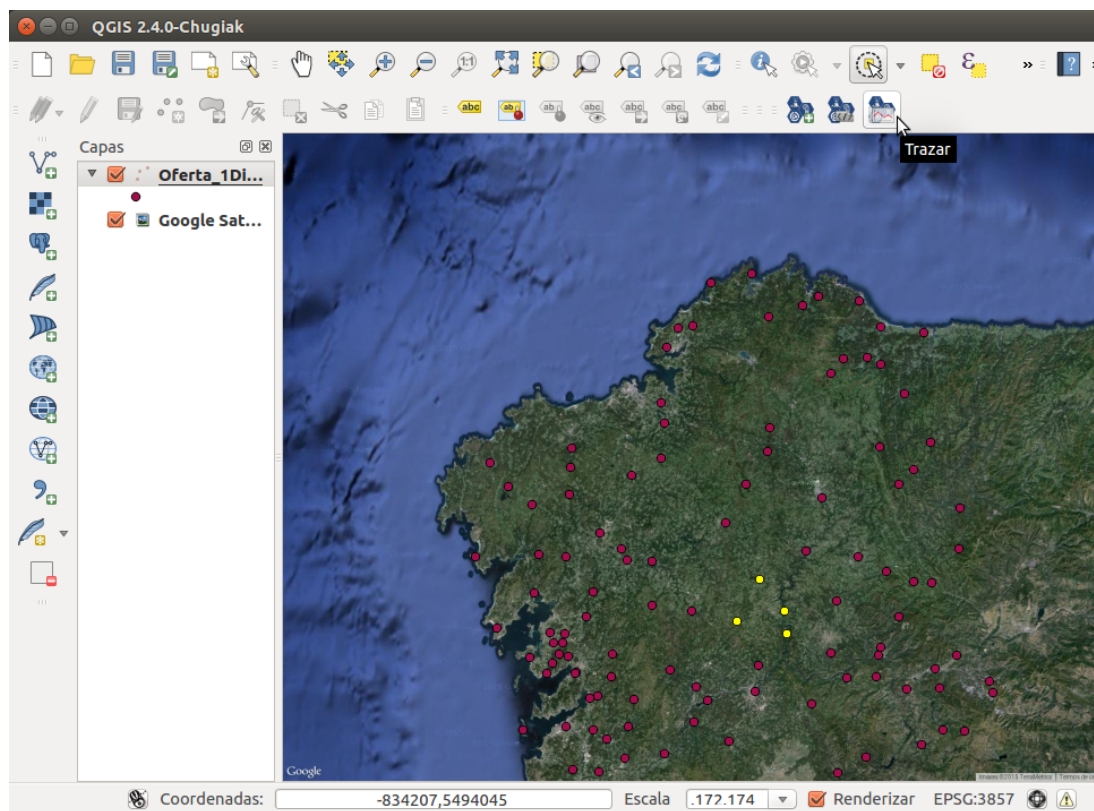


Figura A.7: Executar ferramenta SOS Plot

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

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.

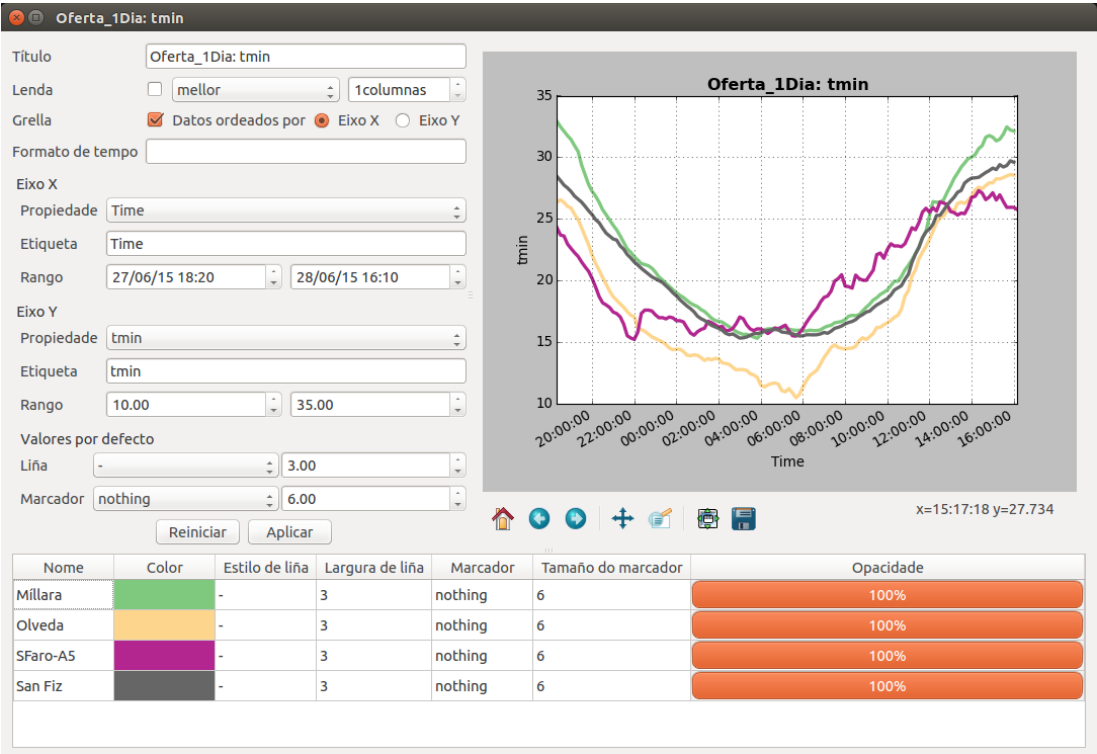


Figura A.8: SOS Plot: Gráfica de varias series





# Bibliografía

- [1] Pyqgis developer cookbook. [http://docs.qgis.org/2.2/en/docs/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<sup>a</sup> 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](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] Juan Palacio. *Gestión de proyectos con Scrum Manager*. Scrum Manager, 2014.
- [10] Seguridad Social. Bases y tipos de cotización 2015. [http://www.seg-social.es/Internet\\_1/Trabajadores/CotizacionRecaudaci10777/Basesytiposdecotiza36537/index.htm](http://www.seg-social.es/Internet_1/Trabajadores/CotizacionRecaudaci10777/Basesytiposdecotiza36537/index.htm), 2015 (consultado o 23 de Xuño).
- [11] tecnoempleo.com. Informe empleo informática. <http://www.tecnoempleo.com/informe-empleo-informatica.php>, 2015 (consultado o 23 de Xuño).
- [12] USC. Regulamento do traballo fin de grao. [http://www.usc.es/etse/files/u1/RegulamentoTFG\\_GrEI\\_CG\\_30xan2014.pdf](http://www.usc.es/etse/files/u1/RegulamentoTFG_GrEI_CG_30xan2014.pdf), 2014.