

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 **D. Manuel Antonio Regueiro Seoane**, TODO-poñer os cargos que correspondan-,

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 titor,

O cotitor,

O alumno,

José Ramón Ríos Viqueira Manuel Antonio Regueiro Seoane Rubén Mosquera Varela

Índice xeral

1. Introducción	1
1.1. Contextualización	1
1.2. Motivación e Obxectivos	1
1.3. Estrutura da memoria	1
2. Xestión do proxecto	3
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	15
2.6.2. Control dos entregables	16
3. Análise de requisitos	17
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

4. Deseño de software	27
4.1. Arquitectura do sistema	27
5. Implementación e probas	29
6. Conclusións e traballo futuro	31
A. Manual técnico	33
B. Manual de usuario	35
Bibliografía	37

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

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

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.

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.

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 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* 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 [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 anteprojecto, que se representa na figura 2.1.

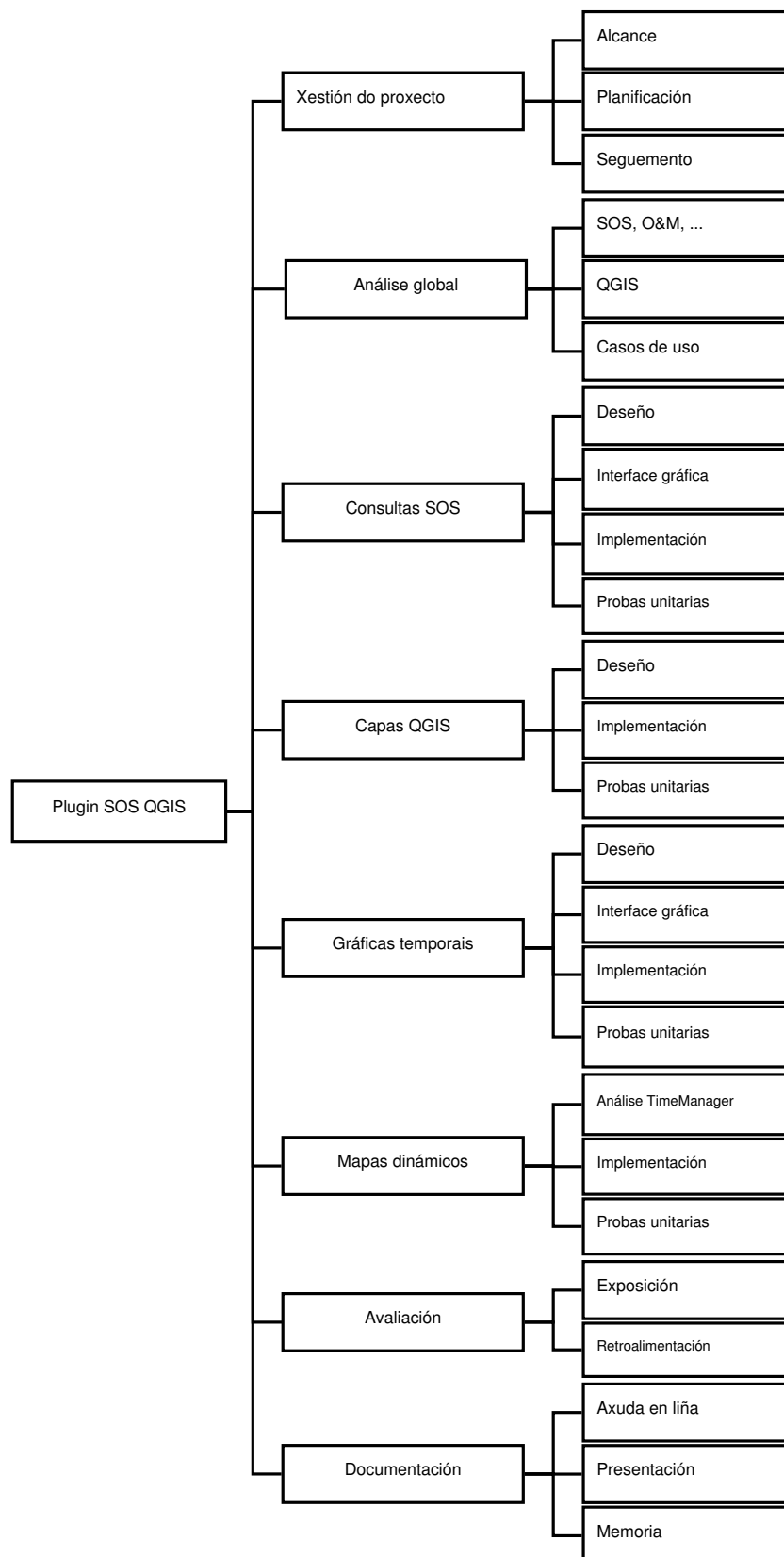


Figura 2.1: Estructura de Descomposición do Traballo

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, ó tratarse dun equipo de desenvolvemento de unha soa persoa, e 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*.

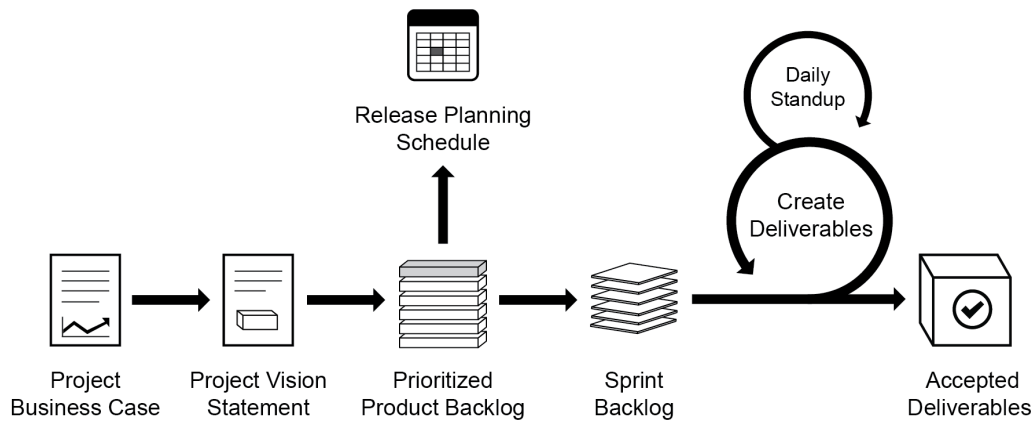


Figura 2.2: Ciclo da metodoloxía Scrum

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

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.

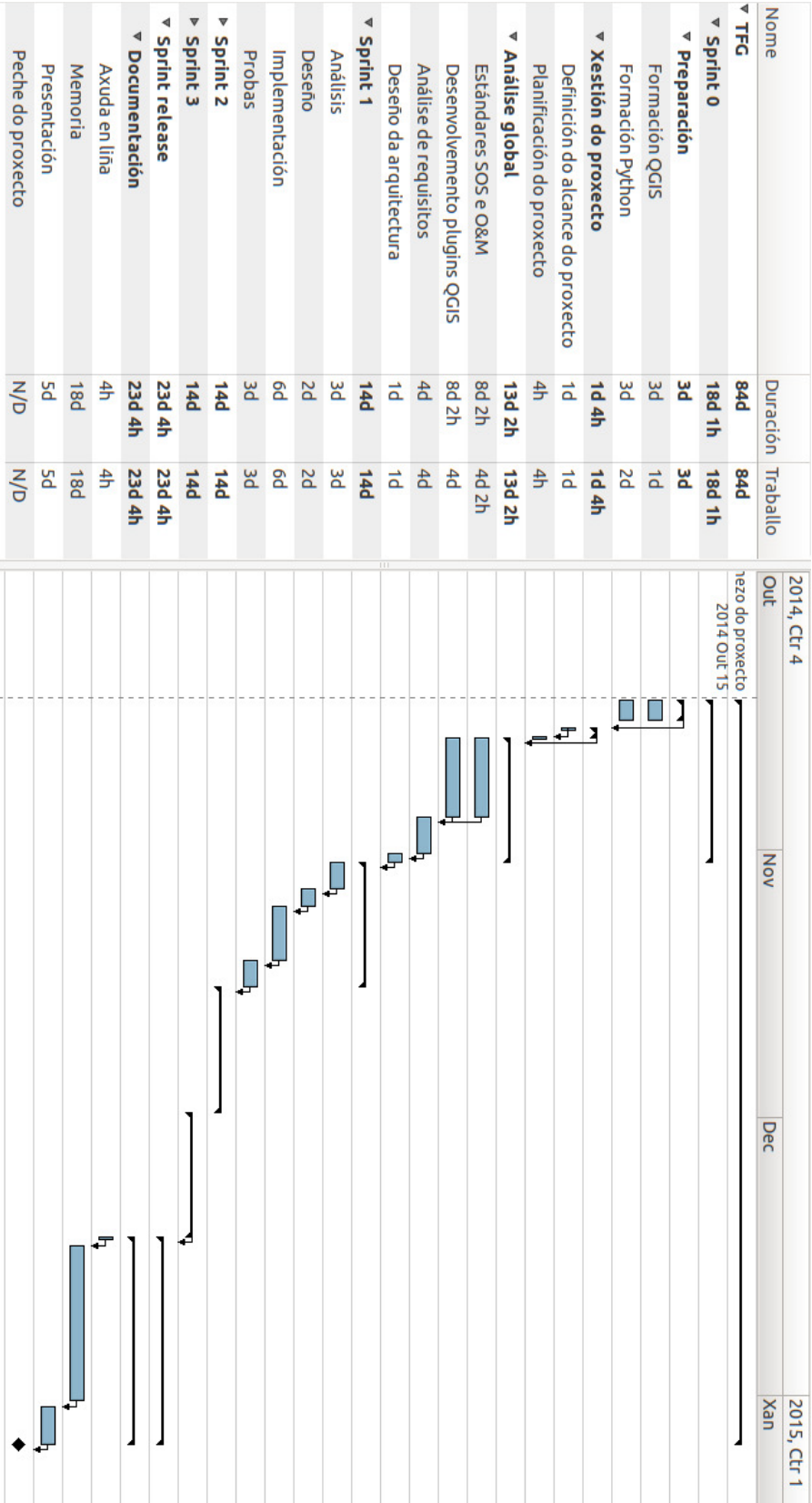


Figura 2.3: Diagrama de Gantt

#	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
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 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

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. Os dous titores do proxecto non se consideran parte do equipo ó nivel de xestión de custos pois a este efecto actúan no rol de clientes.

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 €
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 clasificándoo 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.

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.

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 a 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 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 capacitación na linguaxe de programación Python.

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

Descrición	Cambios na situación persoal ou laboral nos 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 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 da aplicación, imaxes, documentos de axuda e demais que se inclúen no *plugin* empacado, 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*¹. As carpetas dos repositorios locais replícanse en *Dropbox*² 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* empacutado de xeito que sexa moi sinxelo de instalar.

¹<https://github.com/>

²<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 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 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

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.- OBTEN 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 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

Propósito	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.
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 sobre o 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.

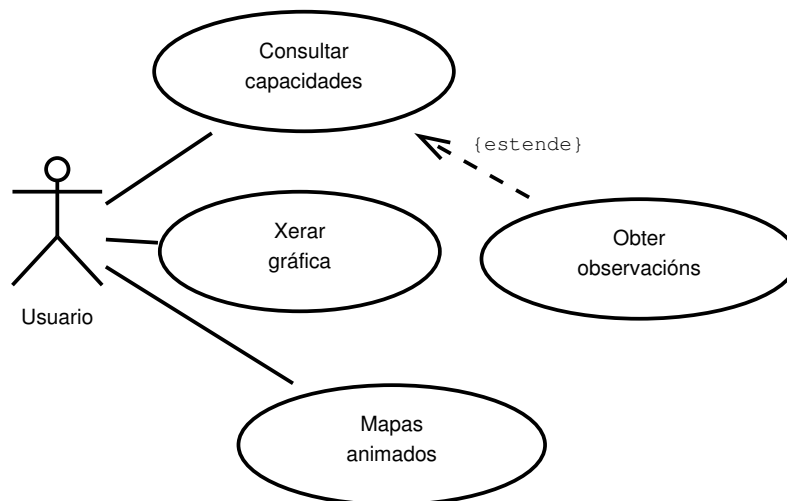


Figura 3.1: Diagrama de casos de uso

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 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

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 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

Descrición	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.
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

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 só as observacións correspondentes o tempo indicado nesta barra. Ademais disporá dun botón de reprodución que desprazará esta barra automaticamente.
Relevancia	Opcional
Criterio de validación	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

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 cumprírase 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 cumprírase 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 cumprírase 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 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

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 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

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

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, 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] 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, 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, 2014.