

T2: Combining OSS: Certification of Open Source Software Stacks

Relatório Técnico

Ref^a BI1-2011_PTDC/EIA-CCO/108995/2008_UMINHO

Neste relatório reportam-se detalhadamente os desenvolvimentos referentes à bolsa de investigação em epígrafe. O trabalho decorreu com toda a normalidade e foram cumpridos todos os pontos do plano de trabalhos.

Após o estudo de várias stacks de software open source, seleccionaram-se um conjunto de aplicações para a construção de um serviço W3DS por composição. Foram apenas desenvolvidas as entradas do serviço, isto é, foram definidas as operações que o serviço exporta.

No desenvolvimento do serviço, para a concretização de todas as manipulações necessárias foram desde logo definidas as interligações entre as diversas composições. Foi utilizado um conjunto de dados de teste, cedidos pela PT Comunicações para avaliar o serviço criado.

Índice

Introdução.....	2
Web 3D Service - W3DS.....	3
Descrição.....	3
Arquitectura.....	3
Operações.....	5
GetCapabilities.....	5
GetScene.....	6
GetTile.....	6
GetLayerInfo.....	7
GetFeatureInfo.....	7
Implementação em GeoServer.....	8
Arquitectura Geral.....	8
Principais Tarefas da Implementação.....	8
Preparação dos Dados.....	10
Geração do 3D das Infraestruturas da PT.....	10
Terrenos 3D.....	10
Resultados.....	13
Servidor para teste, WAR e vídeos.....	13
Pormenor de um terreno tiled.....	13
Tile com textura.....	13
Prédios e infraestruturas da PT posicionados sobre um terreno.....	14
Câmaras, condutas e tampas.....	14
Conclusão.....	15

Figuras

Figura 1: Arquitectura típica de um W3DS.	4
Figura 2: Arquitectura geral da implementação do W3DS no GeoServer.	9
Figura 3: Vista de algumas infraestruturas da PT em 2D e 3D.....	10
Figura 4: Vista de um terreno tiled com vários LODs.....	11
Figura 5: Pormenor de um traçado enterrado.....	12

Figura 6: Terreno com diferentes texturas.....	12
Figura 7: Pormenor de um terreno tiled. Na figura pode-se ver que a junção das tiles é perfeita.....	13
Ilustração 8: Tile obtida através da operação GetScene.....	13
Ilustração 9: Prédios e infraestruturas da PT posicionados sobre um terreno, cena obtida através de um GetScene.....	14
Ilustração 10: Pormenor de uma cena gerada pelo W3DS mostrando câmaras, condutas e tampas..	14

Tabelas

Tabela 1: Parâmetros da operação GetCapabilities.....	6
Tabela 2: Parâmetros obrigatórios da operação GetScene.....	6
Tabela 3: Parâmetros obrigatórios da operação GetTile.....	6
Tabela 4: Parâmetros obrigatórios da operação GetFeatureInfo.....	7

Introdução

Há alguns anos atrás era difícil de acreditar que os sistemas de informação geográfica (*SIG*) poderiam correr na web. Hoje em dia é comum usar sofisticados *web map services* que permitem a visualização, gestão de acessos concorrentes, edição em tempo real, etc ... sem necessitarem de um cliente com *hardware* topo de gama.

Nos últimos tempos têm surgido uma quantidade enorme de tecnologias que permitem a visualização de gráficos 3D na web. O grande salto para o *Web3D* deu-se com o aparecimento da norma *WebGL* que permite usar o standard *OpenGL* através do código Javascript de uma página web. Recentemente o *WebGL* passou a ser suportado pelo *HTML5* tornando possível visualizar nativamente nos browsers mais recentes cenas 3D de elevada complexidade¹.

Neste contexto começaram a surgir vários *WebSIG* 3D a correr na web. Os exemplos existentes vão desde globos 3D², nos quais é possível mapear informação vinda de *web map services* (*WMS*), á visualização de cidades inteiras em 3D³. Á semelhança do que aconteceu para os *WebSIG* 2D em que foi especificado o *WMS* para harmonizar a comunicação entre os clientes e os servidores, para os *WebSIG* 3D foi criado o *Web 3D Service* (*W3DS*)⁴.

Um *W3DS*⁵ permite gerir informação geográfica em três dimensões. A informação é disponibilizada sobre a forma de uma cena 3D que pode ser estática ou conter animações. As cenas geradas são optimizadas para uma visualização em tempo real a *frames rates* interactivos. Os modelos 3D retornados podem ser interactivamente renderizados e explorados nos navegadores web com ou sem *plugins* ou serem descarregados por outro tipo de clientes, por exemplo globos 3D.

Actualmente apenas existe uma implementação do *W3DS* feita para o projecto *OSM-3D*⁶, esta implementação é apenas um prototipo pelo que não está aberta ao público. Nesta âmbito foi implementado um serviço *W3DS* sobre o *GeoServer* que seguiu a versão 0.4.0 da especificação.

1 <http://www.web3d.org>

2 <http://readymap.com/>

3 <http://maps.nokia.com/3D>

4 <http://w3ds.org>

5 A versão contemplada neste documento é a 0.4.0 (http://w3ds.org/lib/exe/fetch.php?media=09-104r1_web_3d_service-0.4.0.pdf).

6 <http://www.osm-3d.org>

Web 3D Service - W3DS

Descrição

Um W3DS⁷ permite gerir informação geográfica em três dimensões como modelos digitais de terrenos, modelos 3D de cidades, vegetação, infraestruturas, etc ... A informação é disponibilizada sobre a forma de uma cena que pode ser composta por vários *layers*. Uma cena 3D pode ser estática ou conter animações. Estas cenas devem ser optimizadas para uma visualização em tempo real a *frames rates* interactivos. Os modelos 3D retornados podem ser interactivamente renderizados e explorados nos navegadores web com ou sem *plugins* ou serem descarregados por outro tipo de clientes, por exemplo globos 3D. O W3DS deve ser capaz de lidar com dados de diferentes escalas, desde o modelo 3D de uma cidade aos candeeiros de rua. Para poder lidar com objectos de escalas tão diferentes deve ser capaz de usar vários *levels of detail* para cada objecto, mantendo o desempenho sem sacrificar a qualidade. A representação de um objecto pode variar entre um modelo 3D extremamente complexo a uma simples representação abstracta. As cenas são obtidas através de pedidos ao serviço que definem a área que deve ser considerada, de onde vêm a informação e o aspecto da cena retornada. O formato usado para representar a cena 3D deve ser escolhido tendo em conta uma capacidade de banda média entre o cliente e o servidor. O formato deve permitir a reutilização de um modelo ao longo da cena, a inclusão de modelos exteriores, estar preparado para lidar com imagens e vídeos, permitir o uso de algumas tecnologias 3D, por exemplo *shaders*, a execução de animações, etc ... Deve ainda ser possível uma interacção entre o utilizador e a cena 3D.

O W3DS é muito semelhante ao WMS, ambos retornam uma vista sobre um conjunto de dados georreferenciados. Enquanto que no WMS é retornada uma imagem no W3DS é retornada uma cena em três dimensões que pode ser explorada de várias formas. Uma cena pode ser vista como uma árvore em que cada nodo pode representar uma geometria, um material, uma textura, uma transformação, etc. Uma cena pode ser obtida por duas operações um *GetScene* ou um *GetTile*, enquanto que a primeira permite aceder e juntar informação de várias fontes de tamanhos variáveis a segunda só permite aceder a informação *grided*. A cena retornada não deve conter informação acerca dos elementos que a compõe, as operações *GetFeatureInfo* e *GetLayerInfo* permitem obter essa informação. O operação *GetCapabilities* permite obter a descrição do serviço.

Formatos como CityGML, KML, VRML e X3D⁸ permitem representar informação georreferenciada em três dimensões. Vários factores tornaram o X3D o formato principal para a representação das cenas produzidas pelo W3DS. O X3D suporta praticamente tudo o que necessitamos para representar cenas 3D como as primitivas geométricas de base, modelos de iluminação, interacção com o utilizador, animações, mecanismos de reutilização, *shaders*, etc ... O X3D têm vindo também a impor-se como um dos principais formatos para o *Web3D* fazendo com que seja possível renderizar as cenas produzidas pelo W3DS em todos os *web browsers*.

Arquitectura

O *pipeline* de visualização de um W3DS é composto por quatro níveis:

1. No primeiro nível trabalho-mos com dados georreferenciados em 3D armazenados em base de dados, *shapefiles*, etc ... Através de uma selecção obtemos os objectos que nos interessam.
2. Os objectos seleccionados no nível anterior são convertidos para uma representação gráfica, i.e. pontos, linhas, polígonos, etc ... e agregados numa cena. A representação gráfica vai estar dependente do formato de saída seleccionado. Por exemplo no caso do X3D uma linha será

⁷ A versão contemplada neste documento é a 0.4.0 (http://w3ds.org/lib/exe/fetch.php?media=09-104r1_web_3d_service-0.4.0.pdf).

⁸ <http://www.web3d.org/x3d/>

transcrita para um nodo do tipo *IndexedLineSet*.

3. Neste nível são renderizadas as cenas produzidas no nível anterior. Por exemplo no caso da cena estar codificada em X3D e for visualizada num web *browser* o render poderá ser feito recorrendo ao *X3DOM*⁹.
4. Por fim a cena produzida é apresentada ao utilizador através de um cliente.

Estes quatro níveis não residem no mesmo sistema, estando distribuídos pela rede numa arquitectura do tipo cliente-servidor. Na *Figura 1* está esquematizada a arquitectura típica de um W3DS. Os passos 1 e 2 são executados no servidor enquanto que os passos 3 e 4 são executados no cliente. Um paço extra por vezes necessário é a conversão de dados 2D em dados 3D, na *Figura 1* podemos ver que as infraestruturas 2D da PT (*RoutesPT* e *NodesPT*) são convertidas em 3D e armazenadas numa base de dados. Neste esquema a comunicação entre o cliente e o servidor é efectuada através das operações *GetScene* e *GetTile*.

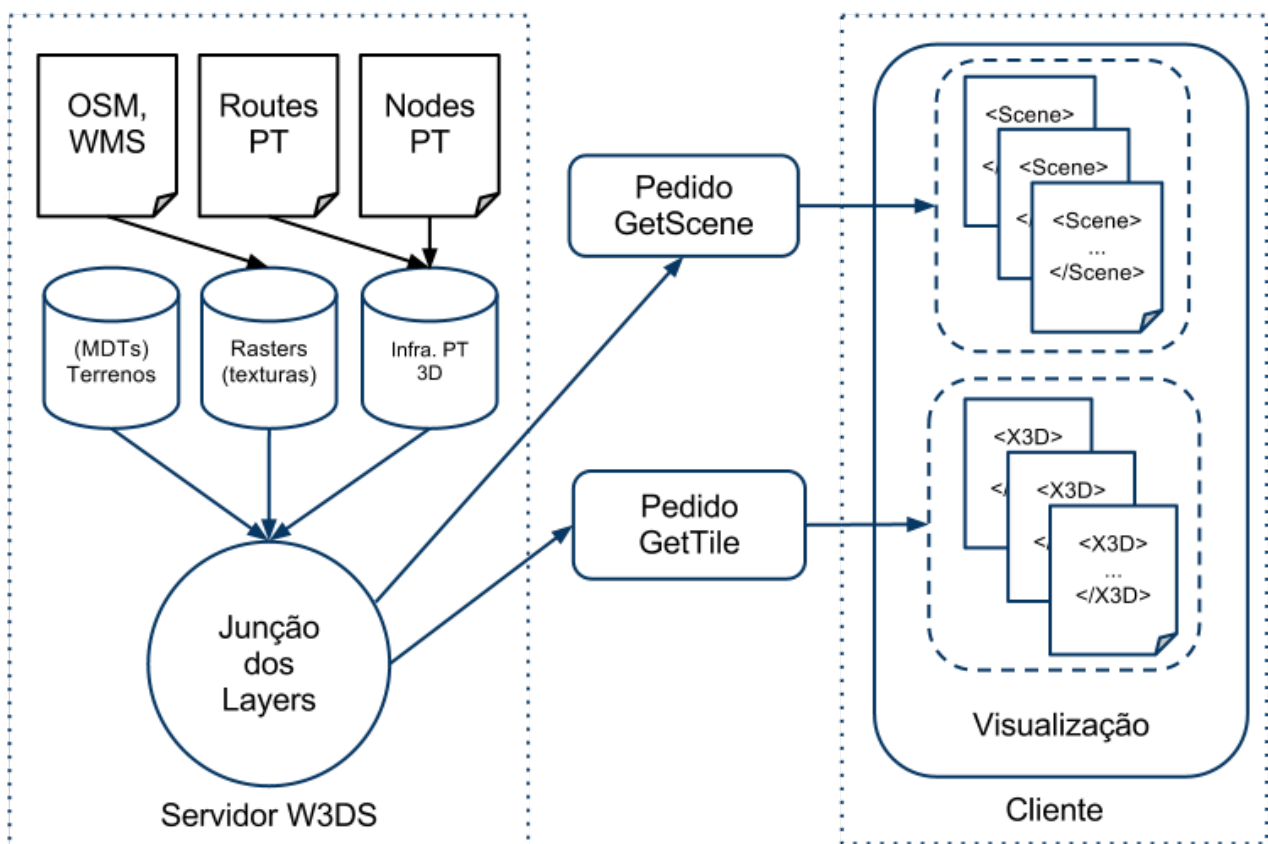


Figura 1: Arquitectura típica de um W3DS.

No servidor estão disponíveis três fontes de dados uma que contém modelos digitais de terrenos (MDTs), uma que possui as texturas dos MDTs e uma com as estruturas da PT. A informação é acedida pelo cliente recorrendo as operações *GetScene* e *GetTile*. O cliente faz o render das scenes e das tiles retornadas pelo servidor permitindo a sua exploração.

Um cliente intermédio terá que apresentar ao utilizador o que está disponível no servidor, obtendo essa informação com a operação *GetCapabilities*. Deixar o utilizador através de *GetScenes* e *GetTiles* obter os dados que pretende visualizar. Permitir obter informação sobre um *layer* através da operação *GetLayerInfo*. Obter informação acerca dos elementos de uma cena através da operação *GetFeatureInfo*. Ser capaz de renderizar as cenas geradas pelo servidor e providenciar mecanismo para a sua exploração.

Operações

A interface do W3DS contempla cinco operações:

- GetCapabilities
- GetScene
- GetTile
- GetLayerInfo
- GetFeatureInfo

Apenas as duas primeiras são obrigatórias sendo as outras três opcionais. Os pedidos ao servidor podem ser feitos por HTTP GET ou HTTP POST, o primeiro método é opcional e usa a codificação KVP o segundo é obrigatório e usa o formato XML¹⁰. Cada uma das operações possui um conjunto de parâmetros que lhe são próprios sendo alguns obrigatórios e outros opcionais. Apenas os seguintes parâmetros são obrigatórios e comuns a todas as operações:

- SERVICE: identifica o serviço (por exemplo *w3ds*)
- REQUEST: identifica a operação a que se destina o pedido (por exemplo *GetScene*)
- VERSION: a versão da operação que deve ser usada, i.e a versão da especificação que deve ser tida em conta (por exemplo *0.4.0*)

Vamos agora descrever em detalhe cada uma das operações¹¹:

GetCapabilities

Esta operação é fundamental para o cliente, permite saber quais os recursos que estão disponíveis no servidor. A [Tabela 1](#) descreve os parâmetros da operação. O resultado de um *GetCapabilities* é um documento XML. Uma resposta completa¹² é composta por quatro secções:

- *ServiceIdentification* : Meta-data acerca do servidor.
- *ServiceProvider* : Meta-data sobre o dono do servidor.
- *OperationsMetadata*: Informação sobre as operações disponíveis.
- *Contents* : Descreve todos os dados presentes no servidor.

A secção *Contents* é a mais complexa e importante, pode ser vista como uma lista dos layers disponíveis no servidor. Para cada layer é fornecida a informação necessária para lhe aceder, por exemplo o seu estado, o sistema de coordenadas em que está representado, o seu tamanho, os estilos que lhe estão associados, se é tiled ou se possui lods.

Nome Parâmetro	Multiplicidade
Sections	0 ou 1 (Opcional)
UpdateSequence	0 ou 1 (Opcional)
AcceptedFormats	0 ou 1 (Opcional)

Tabela 1: Parâmetros da operação *GetCapabilities*.

¹⁰ Na operação *GetCapabilities* a transferência por HTTP XML é opcional e transferência por HTTP GET usando a codificação KVP é obrigatória.

¹¹ Para cada operação vai ser omitida a descrição dos parâmetros SERVICE, REQUEST e VERSION uma vez que esta já foi feita e é comum a todas as operações.

¹² O parâmetro *Sections* permite seleccionar apenas uma das cinco secções disponíveis.

GetScene

A operação *GetScene* permite criar cenas 3D a partir dos dados disponíveis no servidor¹³. Esta é a operação mais característica do W3DS. Na [Tabela 2](#) estão descritos os parâmetros obrigatórios da operação. Como parâmetro opcional pode ser fornecida uma lista dos estilos que devem ser usados. Outro parâmetro opcional relevante é o *offset* que deve ser aplicado as coordenadas. O resultado de um *GetScene* é uma cena 3D codificada no formato pedido.

Nome Parâmetro	Descrição	Exemplo	Multiplicidade
CRS	Sistema de coordenadas em que deve ser retornada a cena.	EPSG:4326	1 (Obrigatório)
BoundingBox	Rectângulo que vai ser usado na selecção do <i>dataset</i> .	-8.339,41.448,- 8.315,41.459	1 (Obrigatório)
Formato	Formato de saída da cena.	model/x3d	1 (Obrigatório)
Layers	Lista de <i>layers</i> que vão ser incluídos no <i>dataset</i> .	dem,predios,co ndutas	1 (Obrigatório)

Tabela 2: Parâmetros obrigatórios da operação *GetScene*.

GetTile

Esta operação permite aceder a informação *tiled* usando os índices de nível, linha e coluna da *tile* pretendida. O nível de uma *tile* está relacionado com o detalhe da mesma, quanto maior o nível mais detalhada é uma *tile*. Na [Tabela 3](#) estão descritos os parâmetros obrigatórios desta operação. Como parâmetro opcional pode ser fornecido o estilo que deve ser usado. O resultado deve estar codificado de acordo com o formato de saída pedido.

Nome Parâmetro	Descrição	Exemplo	Multiplicidade
CRS	Sistema de coordenadas usado.	EPSG:4326	1 (Obrigatório)
Layer	Nome do <i>layer tiled</i> .	dem_tiled	1 (Obrigatório)
Formato	Formato de saída da cena.	model/x3d	1 (Obrigatório)
TileLevel	Nível da <i>tile</i> pretendida.	0	1 (Obrigatório)
TileRow	Índice de linha da <i>tile</i> pretendida.	3	1 (Obrigatório)
TileCol	Índice de coluna da <i>tile</i> pretendida.	2	1 (Obrigatório)

Tabela 3: Parâmetros obrigatórios da operação *GetTile*.

GetLayerInfo

Esta operação permite obter informação da tabela de atributos de um *layer*. Apenas o parâmetro *layer* é obrigatório e permite identificar o *layer* sobre qual queremos obter informação. O parâmetro opcional *ColumnNames* permite especificar o nome das colunas para as quais queremos obter informação. O resultado é um documento do tipo *text/xml*¹⁴.

GetFeatureInfo

Muito semelhante a operação *GetLayerInfo*, esta operação permite obter informação acerca de uma

¹³ A operação *GetScene* não permite aceder a *layers tiled*.

¹⁴ O *schema* da resposta está disponível em <http://w3ds.org>.

feature. A sua principal utilidade é permitir ao utilizador obter informação sobre um elemento de um cena 3D que esteja a explorar. Na [Tabela 4](#) estão descritos os parâmetros obrigatórios da operação. O parâmetro opcional *FeatureCount* permite definir o numero de *features* que devem ser processadas. O resultado deve estar codificado de acordo com o formato de saída pedido.

Nome Parâmetro	Descrição	Exemplo	Multiplicidade
CRS	Sistema de coordenadas que deve ser usado no cálculos.	EPSG:27491	1 (Obrigatório)
Layers	<i>Layers</i> que devem ser contemplados na pesquisa de informação.	camaras,postes,t racados	1 (Obrigatório)
Formato	Formato de saída da resposta.	application/json	1 (Obrigatório)
Coordinate	Posição usada para procurar as <i>features</i> .	-851008.16,-72 1549.53,345.43	1 (Obrigatório)

Tabela 4: Parâmetros obrigatórios da operação GetFeatureInfo.

Implementação em GeoServer

Arquitectura Geral

O GeoServer pode ser visto como uma *stack de software* aberto. A sua implementação assenta sobre vários projectos autónomos como o *GeoTools*, *Maven*, *Spring FrameWork*, *Apache Wicket*, etc ... O próprio GeoServer é constituído por vários módulos que estão distribuídos por três grupos, *Core Modules*, *Extesions* e *Community Modules*. Alguns dos *core modules* possuem pontos de entrada que garantem a extensibilidade do GeoServer. Esses pontos de entrada são conseguidos recorrendo as propriedades da linguagem *Java* e a alguns dos *FrameWorks* usados.

A implementação do W3DS no GeoServer baseia-se na arquitectura típica de um serviço *ows*. Na [Figura 2](#) está representada a arquitectura usada. Quando um pedido chega ao *dispatcher* é encontrado o serviço a que está dirigido. São encontradas todas as operações registadas para esse serviço e é seleccionada a operação a que diz respeito o pedido. Se o pedido for do tipo HTTP GET são encontrados todos os KVP Parsers registados para o serviço¹⁵. Baseado no tipo do argumento da operação é seleccionado o KVP Parser que deve ser usado. De seguida é executado o passo da leitura do pedido, nesse passo é produzido um objecto que contém toda a informação necessária para a operação ser executada. Começa a fase de execução. Esta fase corresponde a primeira etapa do pipeline de visualização do W3DS para as operações *GetScene* e *GetTile*. No fim desta fase é produzido o objecto que vai ser usado para gerar a resposta do pedido. Por fim ocorre a fase de escrita do resultado. Esta fase corresponde a segunda etapa do pipeline de visualização do W3DS para as operações *GetScene* e *GetTile*. Dependendo da operação, a codificação do resultado está predefinida ou é definida pelo pedido.

Principais Tarefas da Implementação

Num fase inicial foi estudado o suporte existente para o 3D no GeoServer. De um forma geral as várias classes estavam definidas para lidarem com objectos 3D, no entanto na maioria dos casos o valor real de *z* era ignorado ou não estava disponível. Foi portanto necessário efectuar alterações em várias camadas da aplicação para que o valor real de *z* fosse usado e conservado ao longo da execução. Uma vez garantido o suporte para o 3D no GeoServer foi implementado o W3DS. O código produzido está distribuído por sete *packages* cada um deles corresponde a uma parte importante da implementação. Segue uma breve descrição de cada um dos *packages*:

- *org.geoserver.w3ds.kvp*: contém os KVP Parsers criados para cada operação.
- *org.geoserver.w3ds.service*: neste *package* estão agrupadas todas as classes que permitem caracterizar o serviço assim como a definição das operações do W3DS.
- *org.geoserver.w3ds.responses*: contém todas as classes que permitem criar um *output* para cada uma das operações.
- *org.geoserver.w3ds.types*: agrupa os tipos de dados criados para o W3DS.
- *org.geoserver.w3ds.utilities*: contém definições de utilidade geral.
- *org.geoserver.w3ds.web*: neste *package* estão contidos todos os componentes da interface web do W3DS.
- *org.geoserver.w3ds.x3d*: contém as várias classes criadas para a geração do X3D.

Outra tarefa relevante foi a extensão dos estilos para integrarem definições próprias ao 3D, as alterações associadas foram principalmente efectuadas no *GeoTools*.

15 Os pedidos recebidos por HTTP POST não são contemplados neste documento.

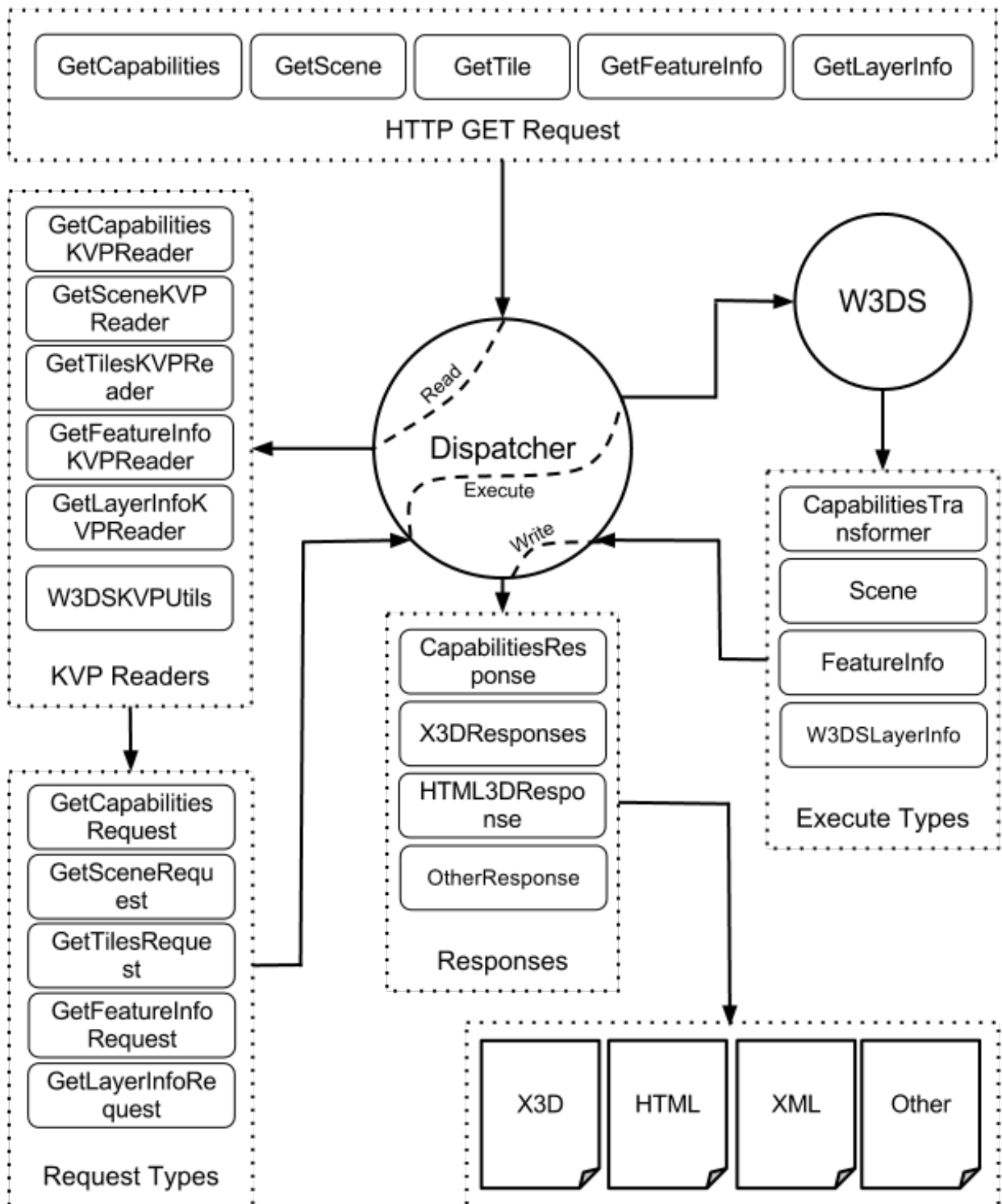


Figura 2: Arquitetura geral da implementação do W3DS no GeoServer.

O dispatcher tem um papel central assumindo o controle das três etapas principais. Leitura do pedido, execução e escrita do resultado. Neste esquema não está representado o acesso aos dados. Os dados são armazenados numa base de dados PostGIS e o acesso é feito recorrendo a biblioteca GeoTools durante o paço da execução.

Preparação dos Dados

Geração do 3D das Infraestruturas da PT

Actualmente o cadastro das infraestruturas da PT está em 2D. As infraestruturas ou são representadas sob a forma de um ponto (POINT) ou sob a forma de uma linha (LINESTRING). As linhas representam traçados que unem as várias infraestruturas que são representadas como pontos, por exemplo postes, câmaras, caixas de distribuição, etc ... Em termos de posicionamento, para cada ponto sabemos a sua localização georreferenciada, sem a componente z e a sua rotação em graus.

O 3D das infraestruturas é gerado em *PL/PGSQL* directamente na base de dados. A componente z da base de todas as geometrias geradas é igual a zero. A visualização em 2D forçou algumas imprecisões no cadastro. Na sub-imagem (a) da [Figura 3](#) o traçado aéreo em parede foi inserido para que fosse possível visualizar dois pontos que se sobrepunham. Quando geramos o 3D algumas dessas imprecisões são corrigidas. Na sub-imagem (b) da [Figura 3](#) a caixa de distribuição em poste já está posicionada em cima do poste. Note-se que por uma questão de comparação também está representado o traçado aéreo em parede.

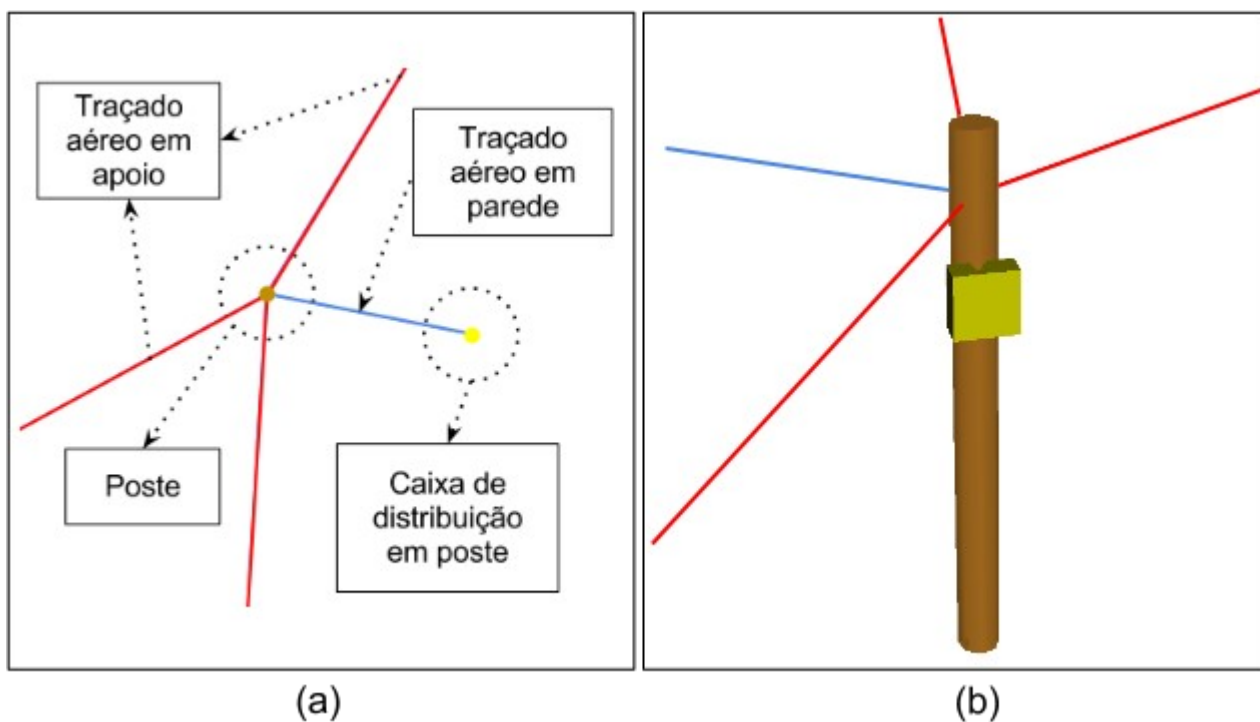


Figura 3: Vista de algumas infraestruturas da PT em 2D e 3D.

A imagem (a) é um print-screen das infraestruturas 2D da PT vistas em *Quantum GIS*. A imagem (b) mostra exactamente as mesmas infraestruturas em 3D.

Terrenos 3D

Os terrenos são um componente fundamental das cenas 3D geradas, todos os outros elementos são posicionados sobre eles. Não existia nenhum software livre que permitisse realizar as operações de que necessitávamos, por isso foi desenvolvida uma biblioteca que responde-se as nossas necessidades. A biblioteca criada está escrita em C++ e está construída sobre as bibliotecas GDAL¹⁶ e CGAL¹⁷.

¹⁶ <http://www.gdal.org/>

¹⁷ <http://www.cgal.org/>

Uma das operações que é efectuada recorrendo a nossa biblioteca é a geração de terrenos *tiled* com diferentes *levels of detail* (LODs) a partir de dados *raster*. O terreno 3D é obtido efectuando uma triangulação de *Delaunay* sobre os pontos extraídos do *raster*. De seguida é gerada uma grelha inicial onde cada célula irá corresponder a uma *tile* do nível mais baixo. Os triângulos gerados previamente são distribuídos por essa grelha. Os vários LODs são gerados juntando varias células e simplificando as geometrias usando o algoritmo *grid simplify point set*¹⁸ da biblioteca CGAL. Na [Figura 4](#) pode ser visto um terreno *tiled* produzido usando a nossa biblioteca. As *tiles* produzidas podem ser armazenadas em qualquer formato vectorial suportado pelo GDAL.

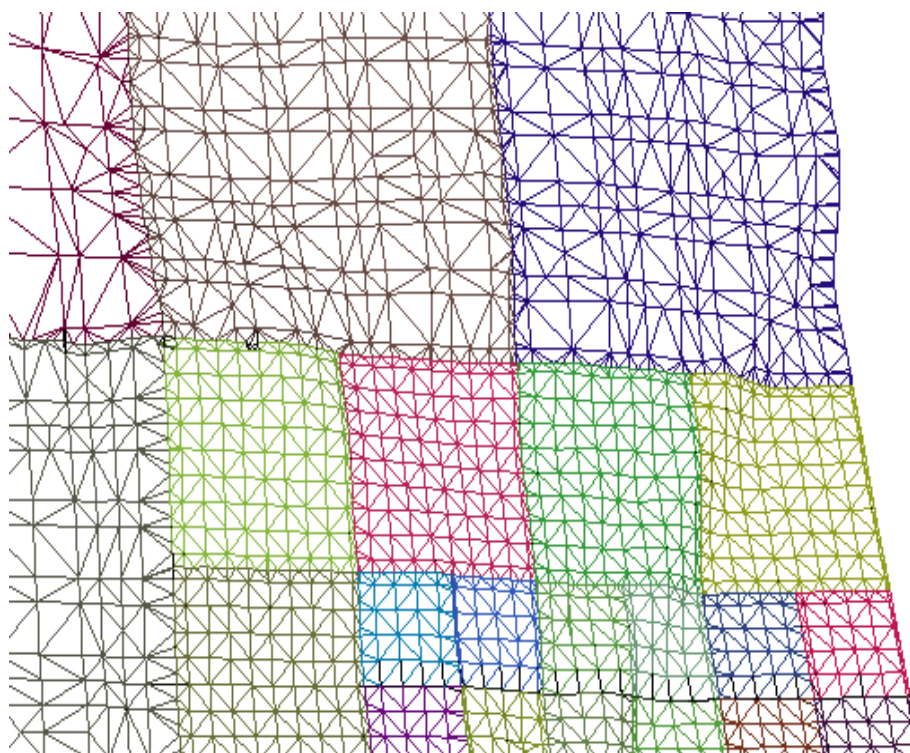


Figura 4: Vista de um terreno tiled com vários LODs.

Outra funcionalidade existente é o povoamento de terrenos com objectos, i.e, dado um terreno e uma geometria posicionar essa geometria em relação ao terreno. No nosso caso essa funcionalidade foi usada para posicionar as infraestruturas da PT em relação a vários modelos digitais de terrenos. Existem várias formas de posicionamento. A forma mais usada é o posicionamento de uma objecto sobre o terreno, isto aplica-se aos postes, armários, caixas de distribuição, etc ... Outra forma consiste em posicionar os objectos no sub-solo. Os traçados enterrados foram posicionados usando um algoritmo específico. Cada segmento do traçado é interceptado com o terreno e todos os pontos de intercepção são adicionados ao traçado. Desta forma garante-se que um traçado nunca fica a vista e fica sempre a mesma profundidade, independentemente da irregularidade do terreno. Na [Figura 5](#) pode ser visto o pormenor de um traçado enterrado.

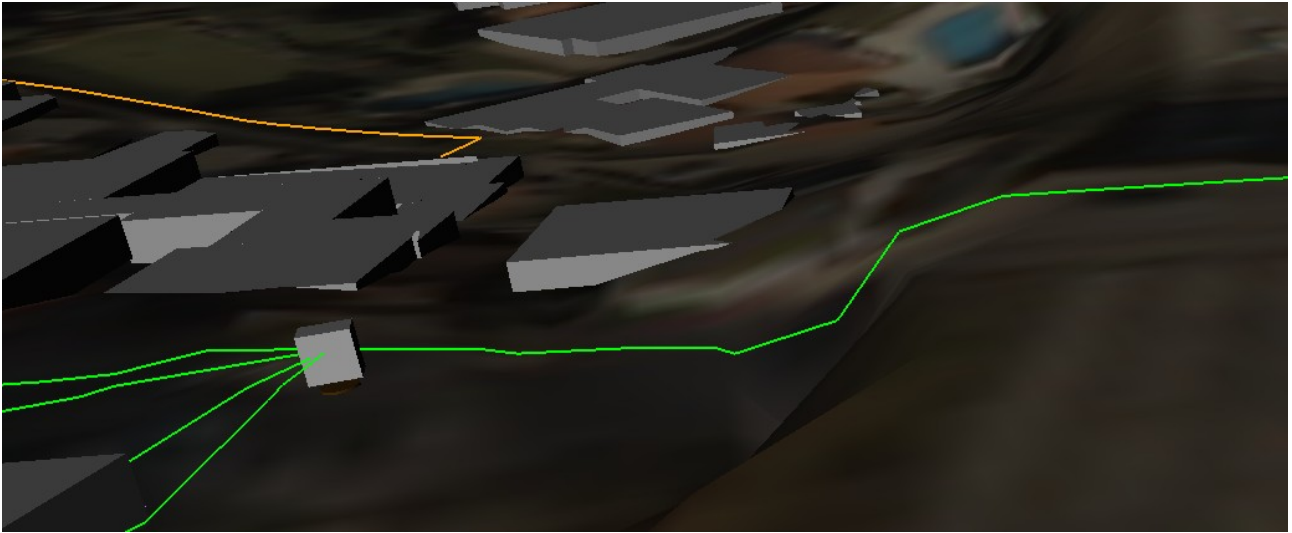


Figura 5: Pormenor de um traçado enterrado.

A verde estão representadas algumas condutas que vão seguindo as irregularidades do terreno.

Um aspecto importante dos terrenos são as texturas, estas são definidas através dos estilos. A geração das coordenadas para o posicionamento das texturas é feita *on the fly* quando a cena 3D é gerada. O caminho para a textura pode ser uma *url*, o que permite uma grande flexibilidade quanto a proveniência da textura. Na [Figura 6](#) pode ser visto um terreno com texturas vindas de um WMS.

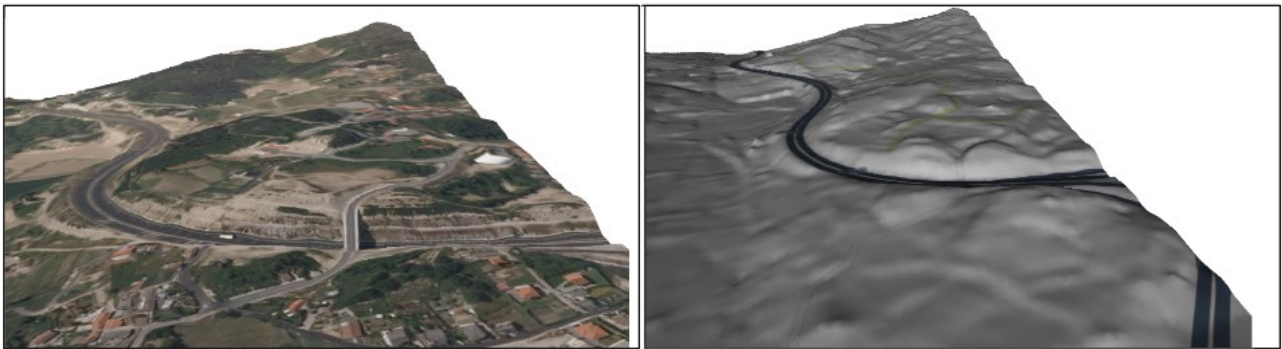


Figura 6: Terreno com diferentes texturas.

Ambas as texturas são obtidas de um WMS. A textura mostrada na imagem da esquerda é um ortofotomapa e provém do IGP. A textura do terreno da direita é obtida a partir do OSM.

Resultados

Servidor para teste, WAR e vídeos

Um servidor para testes está disponível no endereço <http://3dwebgis.di.uminho.pt/geoserver3D>¹⁹. Um WAR está disponível em <http://3dwebgis.di.uminho.pt/sources/>. Alguns vídeos podem ser descarregados neste endereço <http://3dwebgis.di.uminho.pt/videos/>.

Pormenor de um terreno tiled

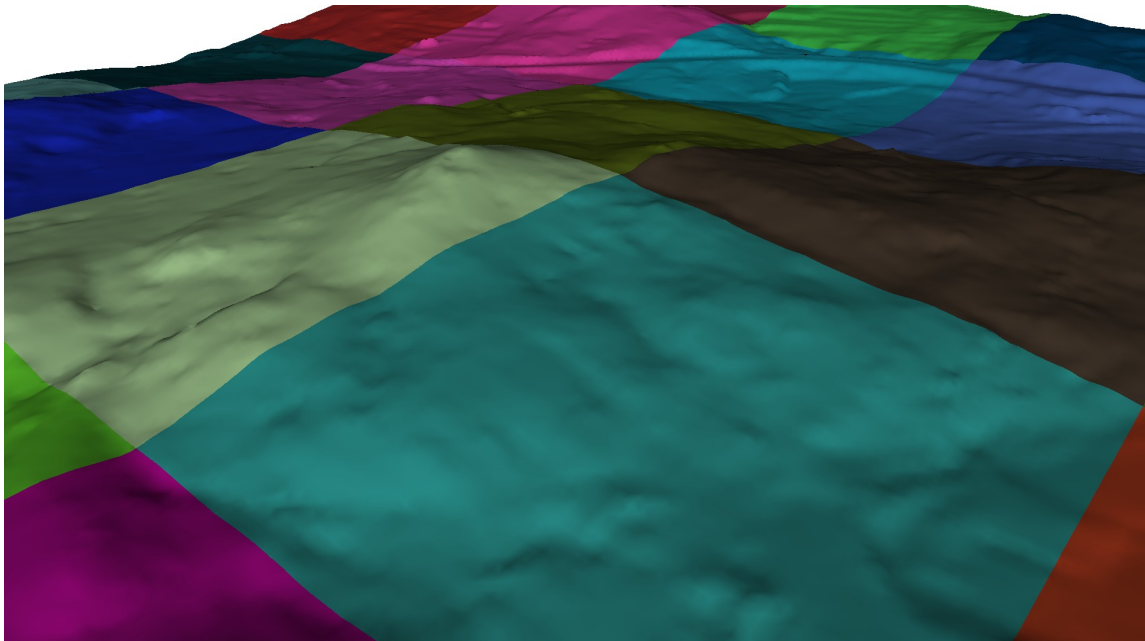


Figura 7: Pormenor de um terreno tiled. Na figura pode-se ver que a junção das tiles é perfeita.

Tile com textura



Ilustração 8: Tile obtida através da operação GetTile..

Pedido: [http://\(...\)w3ds?](http://(...)w3ds?)

`VERSION=0.4.0&SERVICE=w3ds&REQUEST=GetTile&CRS=EPSG:27492&FORMAT=model/x3d&LAYER=dem_tiled&TILELEVEL=0&TILEROW=5&TILECOL=7&STYLE=dem_texture_igp`

¹⁹ Administração: User=admin Pass=geoserver.

Prédios e infraestruturas da PT posicionados sobre um terreno

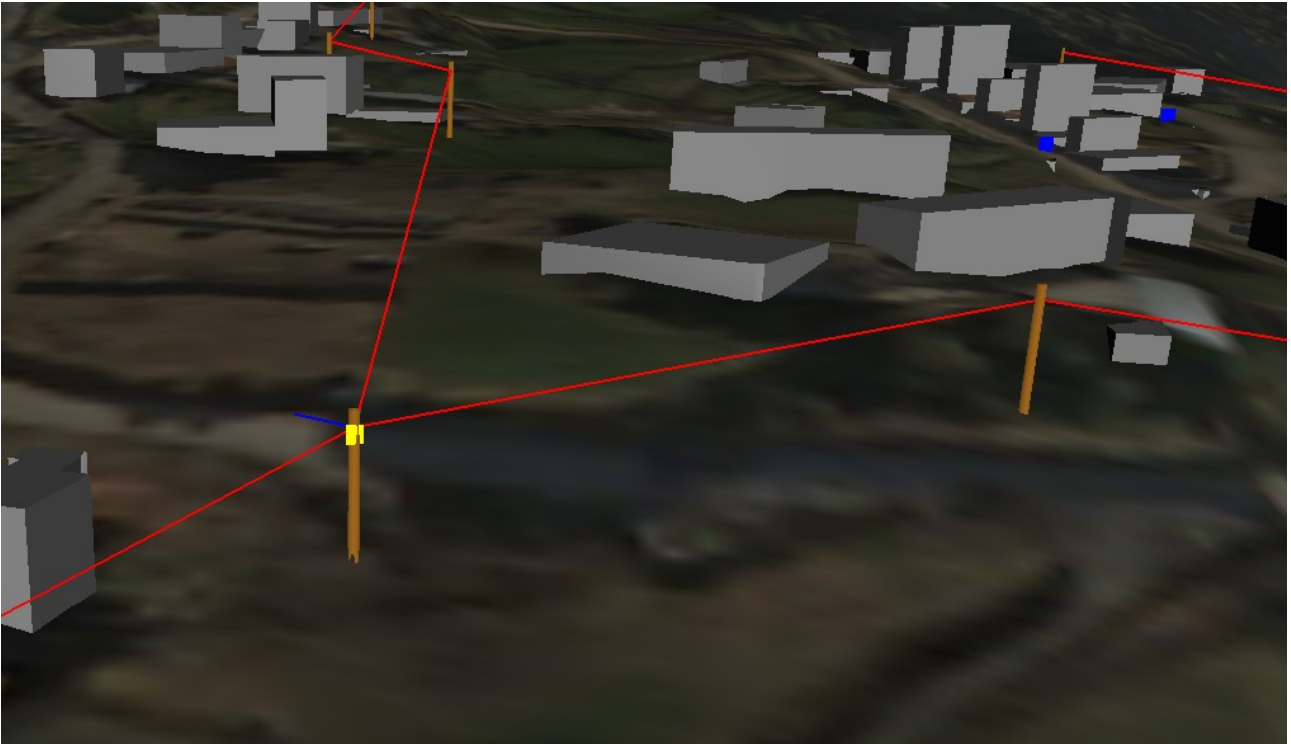


Ilustração 9: Prédios e infraestruturas da PT posicionados sobre um terreno, cena obtida através de um GetScene.

Pedido: <http://...>w3ds?

VERSION=0.4.0&SERVICE=w3ds&REQUEST=GetScene&CRS=EPSG:27492&FORMAT=model/x3d&LAYERS=postes,caixas,mdt_tiled&BOUNDINGBOX=-15269,196602,-10934,198014

Câmaras, condutas e tampas

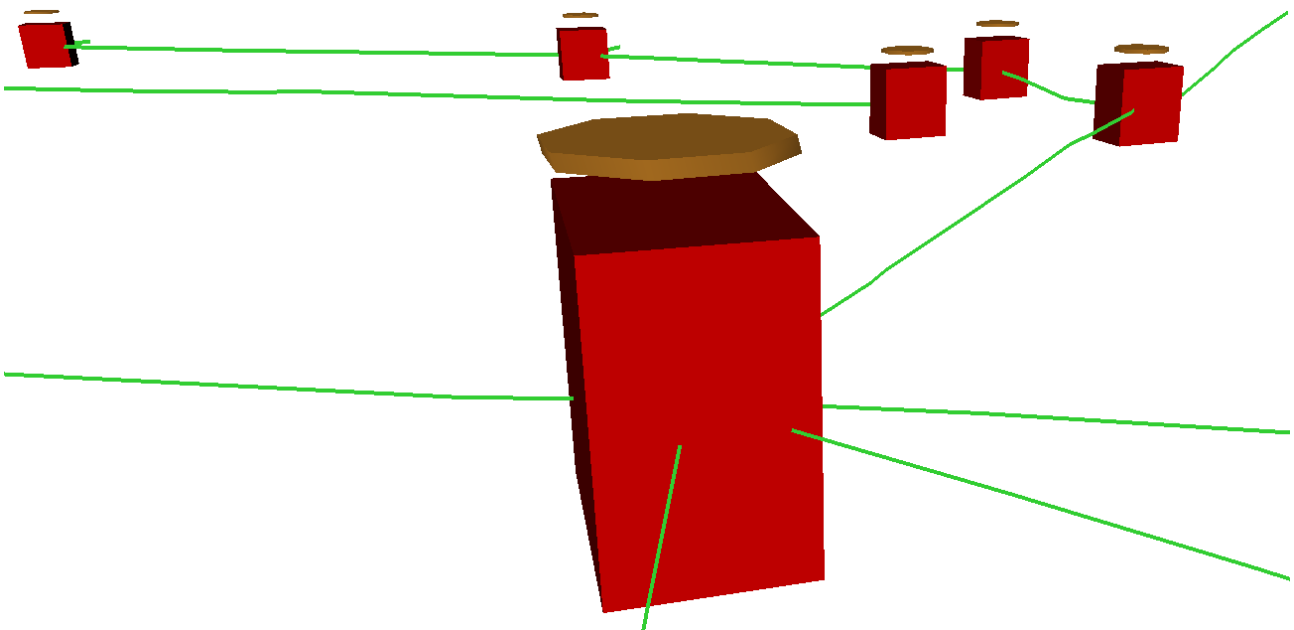


Ilustração 10: Pormenor de uma cena gerada pelo W3DS mostrando câmaras, condutas e tampas.

Imóveis e infraestruturas da PT

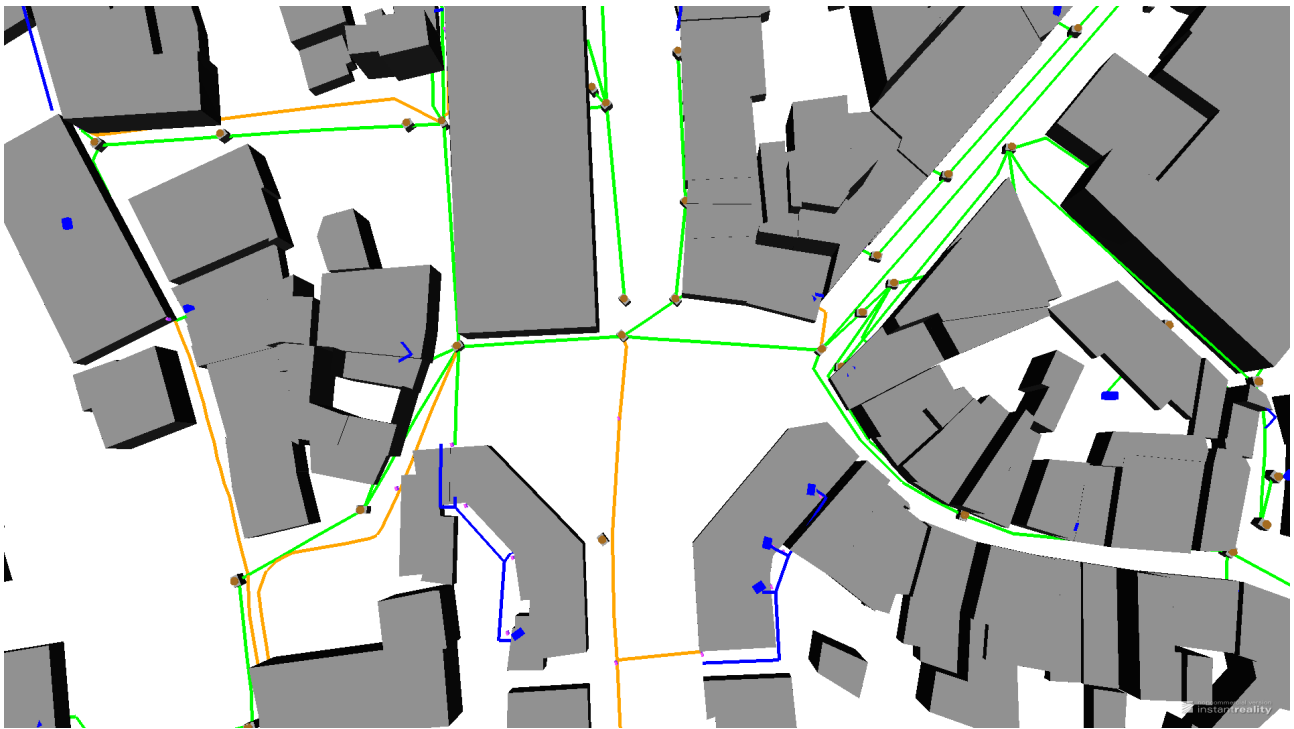


Ilustração 11: Vista aérea das infraestruturas da PT.

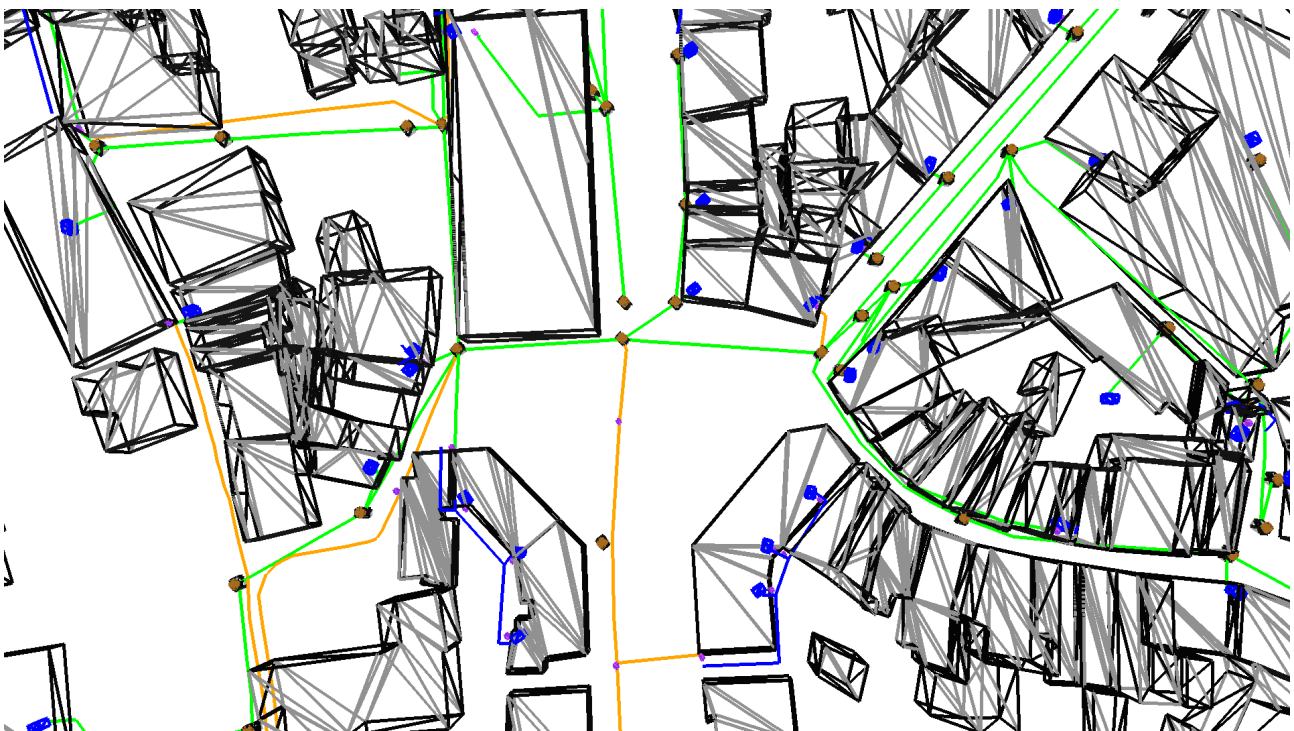


Ilustração 12: Vista aérea das infraestruturas da PT onde se vê as infraestruturas que estão dentro dos imóveis.

Conclusão

Como se pode verificar pelos resultados alcançados, foi integralmente cumprindo o objetivo de implementar um serviço W3DS por composição de componentes de software existentes. O mesmo está pronto a ser utilizado e, como foi apresentado na seção dos resultados, foi preparada uma demonstração do mesmo com dados reais. A demonstração serviu para aferir a viabilidade da utilização do software desenvolvido em futuros projetos que careçam da manipulação de Informação Geográfica em 3D em ambiente web.

Avaliando o desenrolar dos trabalhos, vemos que a preparação dos dados tomou bastante tempo do projeto, comparado com o desenvolvimento do software. Felizmente conseguiu-se conciliar as duas coisas.