

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Integração de mapas em visualizador de redes eléctricas.

Eduardo Pedro Filipe Ramalho

Mestrado Integrado em Engenharia Informática e Computação

Orientador: Doutor António Fernando Vasconcelos Cunha Castro Coelho, Professor Auxiliar do Departamento de Engenharia Informática da Faculdade de Engenharia da Universidade do Porto

Janeiro de 2011

Integração de mapas em visualizador de redes eléctricas

Eduardo Pedro Filipe Ramalho

Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo Júri:

Presidente: Raul Vidal (PhD)

Vogal Externo: César Analide (PhD)

Orientador: António Coelho (PhD)

Janeiro de 2011

Resumo

Os Sistemas de Informação Geográfica possibilitam a gestão e análise de informação que tem associado uma localização geográfica. Estes sistemas podem ser aplicados a qualquer sector de actividade. Neste trabalho foi estudada a sua aplicação à gestão de redes eléctricas.

O âmbito deste trabalho está associado a um sistema SCADA/DMS comercializado pela empresa EFACEC, com o nome Scatex+. O objectivo principal é expandir as suas funcionalidades de gestão e visualização de informação (específicas à rede eléctrica) integrando qualquer tipo de informação geográfica genérica. Um segundo objectivo, que funciona por restrição, é o recurso a *software* livre de forma a não elevar os custos de licenciamento aos clientes.

Para cumprir os objectivos propostos o trabalho foi dividido em duas fases. Primeiro o estudo do estado da arte nestes sistemas de informação. O segundo, a intergação de um protótipo de visualização de mapas integrado e a pesquisa de informação à base de dados.

A solução desenvolvida consiste em adicionar ao sistema SCADA existente, a base de dados Postgres com extensões geoespaciais Postgis. A geração de imagens e pesquisa é realizada através da interface WMS/TMS disponibilizada pelo GeoServer. A optimização da geração de imagens por mosaicos é realizada pelo uso do TileCache. O cliente WMS a integrar no visualizador existente é a biblioteca GDAL.

O resultado final foi a integração com sucesso de mapas de teste no visualizador SCADA e foi estabelecida uma infraestrutura para desenvolvimento de mais aplicações no futuro.

Abstract

Geographic Information Systems enable the management and analysis of information that is associated with a geographic location. These systems can be applied to any industry. In this work we have studied their application to the management of electrical networks.

The scope of this work is connected to a SCADA / DMS system marketed by EFACEC, named ScateX. The main objective of this thesis is to expand its advanced functionality management and visualization of information (specific to the electricity grid) incorporating any type of geographic data base. A second objective, which works by restriction, is the use of open source software in order to keep licensing costs low to the customers.

To meet the objectives proposed this work was divided into two phases. First the study of the state of the art in these information systems. Second, the integration of a prototype for the visualization of maps and query of information from the database.

The developed solution is based on the integration on the existing SCADA system, of a Postgres database with the Postgis geospatial extension. The visualization and query are conducted through the WMS / TMS interface provided by GeoServer and the optimization of tile images generation is made using TileCache. The WMS client integrated into the existing viewer is the GDAL library.

The end result was the successful integration of test maps on the SCADA viewer and we established an infrastructure to develop more applications in the future.

Agradecimentos

Gostava de agradecer antes de mais à empresa EFACEC pela oportunidade concedida e tempo disponibilizado para a realização deste trabalho. Agradeço em particular ao engenheiro Pedro Silva pelo acompanhamento mais próximo do projecto na empresa.

Gostava de agradecer ao meu orientador Dr. António Fernando Vasconcelos Cunha Castro Coelho, pela orientação na escrita e como expor os temas desta dissertação .

Por fim especial agradecimento à namorada Soraia e à minha família pela paciência devido aos meses de afastamento e clausura.

Eduardo Ramalho

Índice

1	Introdução.....	1
1.1	Contexto de realização.....	1
1.2	Problema.....	1
1.3	Motivação e Objectivos.....	4
1.4	Estrutura da Dissertação.....	4
2	Revisão Bibliográfica.....	5
2.1	O que é um SIG ?.....	5
2.2	SIG e distribuição de energia.....	6
2.3	Tipos de mapas.....	7
2.3.1	Mapa raster.....	8
2.3.2	Vectorial.....	9
2.4	Sistemas de Coordenadas	10
2.4.1	Coordenadas Geográficas.....	11
2.4.2	Projecção.....	11
2.5	Bases de dados espaciais.....	14
2.6	Gerar mapas.....	15
2.7	Pré-geração de imagens.....	16
2.8	Resumo	17
3	Metodologia.....	18
3.1	Problema.....	18
3.2	Proposta de arquitectura.....	20
3.2.1	WMS.....	22
3.2.2	GDAL.....	23
3.2.3	TMS.....	24
3.2.4	Estilização de mapas.....	25
3.3	Resumo e Conclusões.....	27
4	Implementação.....	28
4.1	Integração de mapas de teste.....	28
4.1.1	Integração de MDT.....	28
4.1.2	Integração do CAOP.....	29
4.1.3	Integração da rede viária.....	30

4.2	Optimização de mapas via TileCache.....	31
4.3	Draw.....	32
4.4	SLD.....	34
4.5	Resultados obtidos.....	37
4.5.1	CAOP.....	37
4.5.2	Estradas.....	38
4.6	Resumo.....	40
5	Conclusões e Trabalho Futuro.....	41
5.1	Objectivo e resultados.....	41
5.2	Trabalho Futuro.....	42
	Referências.....	43
	Anexo A: Draw e GDAL.....	44
A.1	Configuração XML.....	44
A.1	gdal_utils.hxx.....	45
A.2	gdal_utils.cxx.....	45
A.1	view.hxx.....	50
A.1	view.cxx.....	50
	Anexo C: OpenStreetMaps SLD.....	55

Lista de Figuras

Figura 1: Visualização da rede alta tensão do norte de Portugal.....	2
Figura 2: Visualização da rede da área do Grande Porto.....	2
Figura 3: Visualização da rede média tensão na área do Porto de Leixões.....	3
Figura 4: Freguesias.....	10
Figura 5: Estradas.....	10
Figura 6: Altitude.....	10
Figura 7: Latitude e Longitude.....	11
Figura 8: Projecção Mercator.....	12
Figura 9: Três tipos de projecções.....	13
Figura 10: Pirâmide de imagens.....	17
Figura 11: Arquitectura.....	21
Figura 12: Coordenadas TMS.....	25
Figura 13: Rua e Casas.....	26
Figura 14: CAOP.....	37
Figura 15: Autoestradas Algarve.....	38
Figura 16: Ruas urbanas.....	39
Figura 17: Nomes de ruas.....	39

Lista de Tabelas

Tabela 2.1: Tipos de dados e possibilidades.....	7
Tabela 2.2: WKT de Portugal.....	14
Tabela 2.3: Lista de estradas	15
Tabela 2.4: Restrição por vista.....	15
Tabela 3.1: Stringuificado.....	19
Tabela 3.2: Áreas resolvidas pela arquitectura.....	22
Tabela 3.3: Exemplo URL de pedido WMS.....	22
Tabela 3.4: Parâmetros de um pedido WMS.....	23
Tabela 3.5: Identificação de cada mosaico em TMS.....	24
Tabela 3.6: Especificação do SLD via WMS.....	26
Tabela 3.7: Exemplo SLD.....	27
Tabela 4.1: Dados MDT.....	28
Tabela 4.2: Execução do comando “gdaldem”.....	29
Tabela 4.3: Dados CAOP.....	29
Tabela 4.4: Instruções preparar o Postgres para o CAOP.....	29
Tabela 4.5: Parâmetros para importação da CAOP.....	30
Tabela 4.6: Dados OSM.....	30
Tabela 4.7: Importação OSM para Postgres.....	31
Tabela 4.8: Conversão da projecção OSM.....	31
Tabela 4.9: Configuração do TileCache.....	32
Tabela 4.10: Pedido via GDAL.....	33
Tabela 4.11: Funções de leitura por banda.....	34
Tabela 4.12: Identificações usadas pelo OpenStreetMaps.....	35
Tabela 4.13: Definição SLD do elemento a desenhar.....	36

Abreviaturas e Símbolos

API	Application Programming Interface
GDAL	Geospatial Data Abstraction Library
SIG	Sistema de Informação Geográfica
SCADA	Supervisory Control And Data Aquisition
TMS	Tile Map Service
WMS	Web Map Service
WWW	<i>World Wide Web</i>
SLD	Styled Layer Descriptor
URL	Uniform Resource Locator
GPS	Global Positioning System
PT	Posto de Transformação
CAOP	Carta Administrativa Oficial de Portugal

1 Introdução

Este capítulo apresenta o ambiente e o contexto onde foi desenvolvido este trabalho, o problema solucionado existente e os objectivos desta dissertação.

1.1 Contexto de realização

Esta dissertação foi proposta pela empresa nacional EFACEC Engenharia. Esta desenvolve e comercializa um leque alargado de sistemas (*hardware*, *software*) na área das energias (produção, transmissão, e gestão de redes).

Um dos sistemas existente é o Scatex+. Este é usado para gestão e controlo da rede eléctrica. Esta gestão engloba as seguintes funcionalidades:

- monitorização das medidas de unidades remotas no terreno
- gestão de alarmes de supervisão
- controlo de disjuntores e seccionadores
- editor de diagramas (representação lógica da rede)
- visualizador de redes e diagramas

Este sistema tem vários clientes pelo mundo entre quais se destaca a EDP e o Metro do Porto em Portugal e o CERN na Suíça.

1.2 Problema

No Scatex+ a visualização da rede eléctrica é apresentada como está disposta no terreno. Linhas de alta/média tensão, subestações e postos de transformação estão geo-referenciados. Esta informação é complementada com as fronteiras do território português, distritos e as áreas de operação do sistema. É a partir deste visualizador que o operador interage com a rede.

Introdução

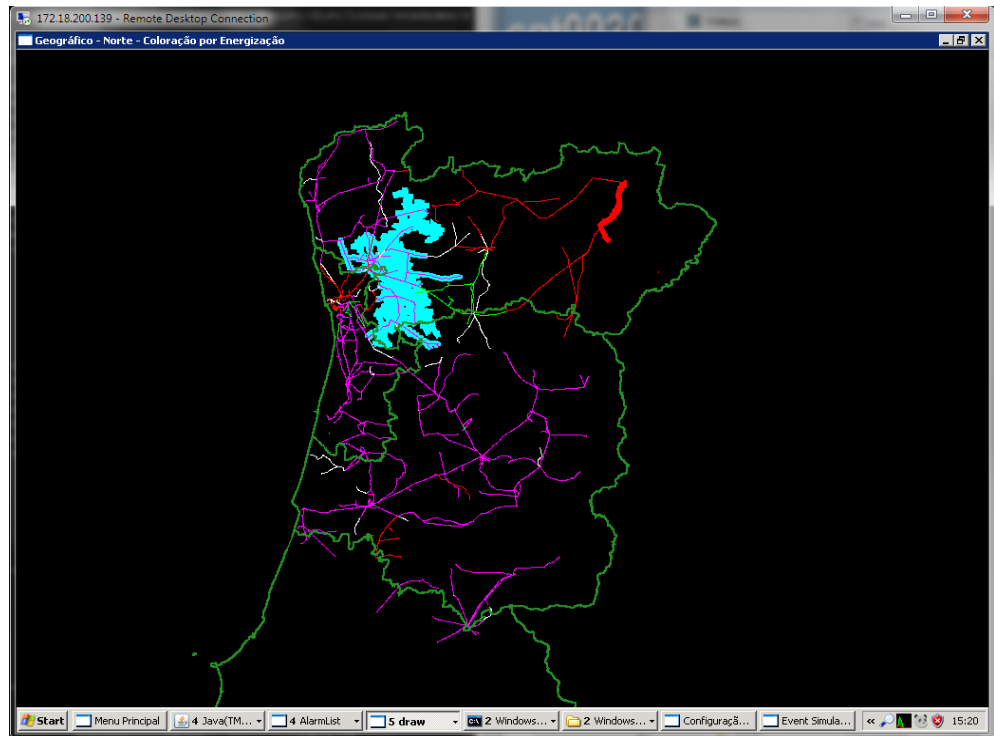


Figura 1: Visualização da rede alta tensão do norte de Portugal

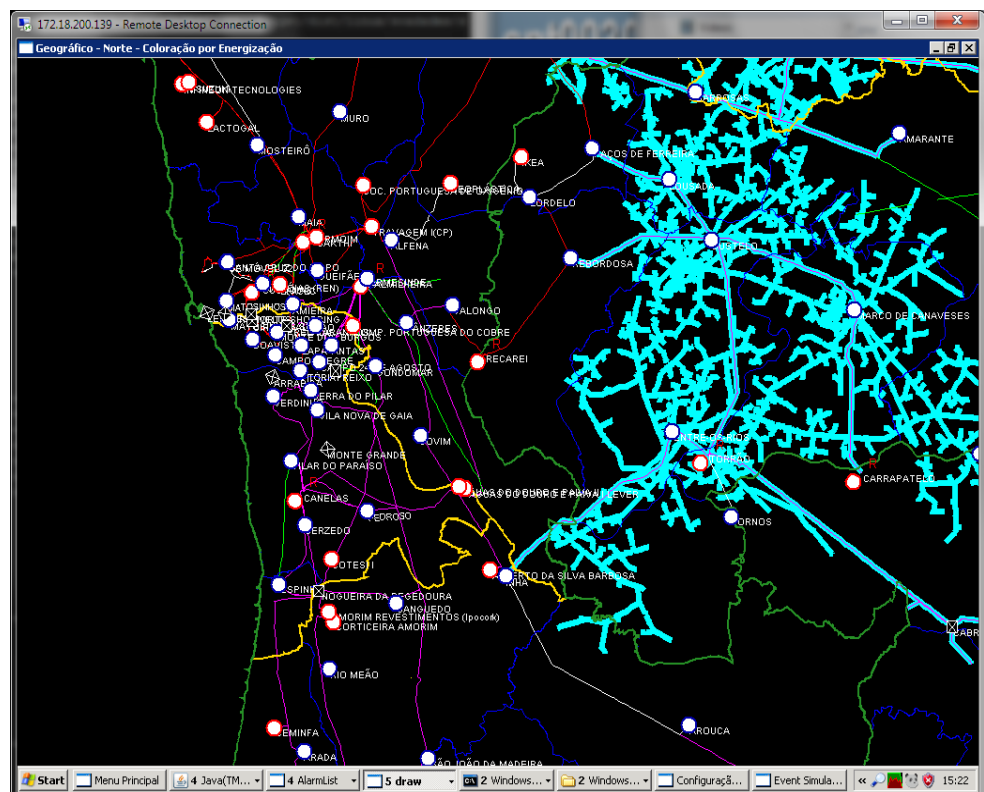


Figura 2: Visualização da rede da área do Grande Porto

Introdução

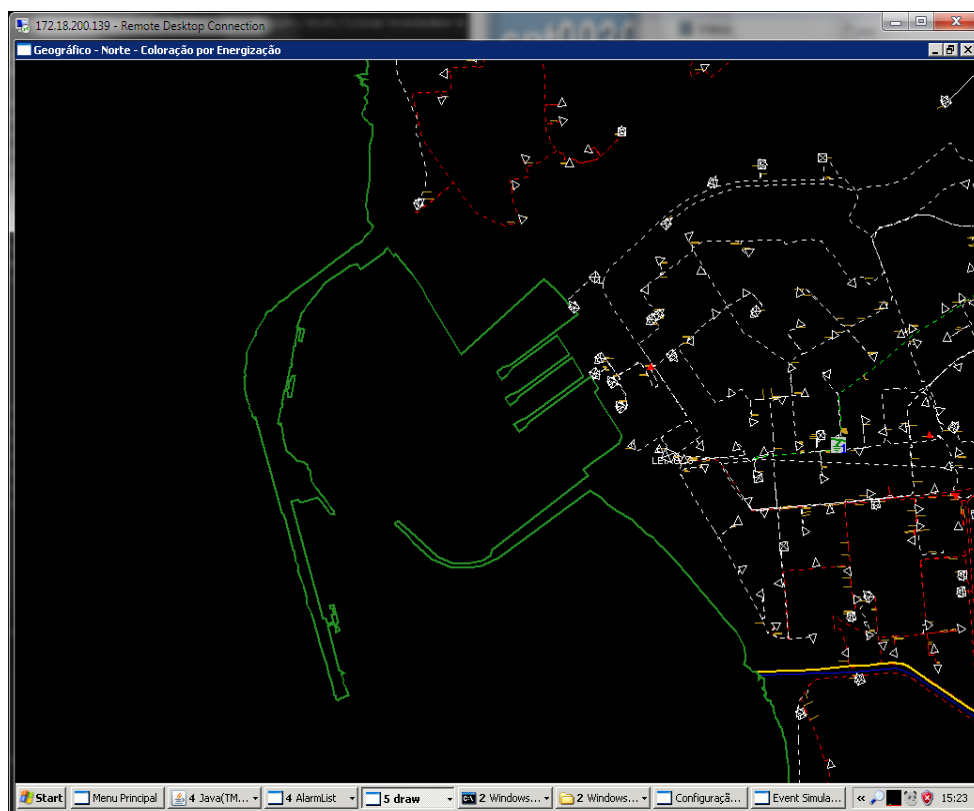


Figura 3: Visualização da rede média tensão na área do Porto de Leixões

As figuras 1, 2 e 3 mostram o visualizador dessa rede. Cada imagem representam aproximações a uma área geográfica cada vez mais pequena. Em contrapartida, cada imagem acrescenta mais pormenor à anterior. Na primeira está representado em cores diversas as linhas de alta tensão. As linhas verdes representam as fronteiras do território. A segunda é uma aproximação à região do Grande Porto. Nesta é acrescentado o pormenor das subestações (pontos). Por fim, a última imagem representa o porto de Leixões. Nesta é visível linhas de média tensão e postos de transformação.

Foram identificados oportunidades de melhoria no visualizador da rede. O desenho podia ser complementado com outros elementos geoespaciais fora do âmbito das redes eléctricas. Por exemplo, não existe informação das estradas ou a localidades mais próximas de uma subestação. A partir do momento que se tem vários elementos referenciados também não existe um processo para interrogar relações entre esses elementos (por ex: qual o piquete de manutenção mais próximo de uma subestação). Finalmente, o suporte actual de informação georeferenciada é baseada em ficheiros de texto num formato proprietário. Isto dificulta e implica custos na importação de dados diversos de fontes externas.

1.3 Motivação e Objectivos

A motivação principal deste trabalho é ajudar o operador do sistema, a tomar decisões relacionadas com a gestão da rede eléctrica. Algumas destas decisões poderão ser facilitadas com a percepção da localização onde ocorrem problemas.

O objectivo deste trabalho será investigar a forma de integrar essa informação gráfica com o visualizador de redes eléctricas. Para isso é necessário investigar os seguintes pontos:

- que conteúdos geográficos existem
- formatos desses conteúdos
- que sistema usar para persistir e interrogar essa informação
- como desenhar esses conteúdos.

Por fim, uma restrição colocada é que o sistema a propor deverá ser baseado em aplicações *open source*. Desta forma os custos de licenciamento a clientes do sistema serão minimizados.

A metodologia a usar será investigar o estado da arte nestes sistemas e demonstrar com exemplos que o sistema proposto de aplicações e procedimentos vão funcionar na prática.

1.4 Estrutura da Dissertação

Para além da introdução, esta dissertação contém mais 4 capítulos. No capítulo 2, é descrito o estado da arte e são apresentados trabalhos relacionados. No capítulo 3, é indicado o problema actual e a solução encontrada. No capítulo 4 encontra-se pormenores de implementação. Por fim, no capítulo 5, são apresentadas as conclusões desta dissertação.

2 Revisão Bibliográfica

Neste capítulo é descrito o estado da arte e são apresentados trabalhos relacionados de forma a mostrar o que existe no mesmo domínio e quais os problemas encontrados.

2.1 O que é um SIG ?

Desde que o Homem ganhou consciência e convive em grupos, houve sempre a necessidade de transmitir localizações. Há 35.000 anos, caçadores *Cro-Magnon* desenhavam imagens de animais nas cavernas de Lascaux em França. Associado a estas imagens estavam desenhos de linhas e pontos. Acredita-se que estes estejam associados a indicações de caminhos de migração de animais e a posições das estrelas de forma a indicar períodos específicos do ano [Rappenglück99]. Este pode ser considerado uma forma rude de Sistema de Informação Geográfica, ou seja uma representação em duas dimensões (na parede da caverna) de diversos fenómenos da natureza.

De regresso ao presente, e numa era de avanço tecnológico sem precedentes na Humanidade, podemos ter a seguinte definição de Sistema de Informação Geográfica:

“Conjunto de sistemas de hardware, software, dados georeferenciados e procedimentos com os objectivos de capturar, gerir, analisar, prever e visualizar mais facilmente a informação” [GIS].

A parte visível de um SIG é a visualização de mapas. Na verdade o seu poder está na capacidade que pode facultar de forma automática, a capacidade de estabelecer relações entre estados ou fenómenos diversos. A partir desse ponto podem-se inferir conclusões.

Podemos dar um exemplo de um estudo [Naganobu03] da população de baleias num ambiente marinho. A informação para o estudo foi capturada via diferentes fontes: observação de satélite (temperatura da água, concentração de zooplacton, placas de gelo), amostras capturadas¹

¹ O estudo não é explícito se foram capturadas baleias para o estudo. Mas o navio referido no estudo é reconhecido como baleeiro. Este autor não aprova a caça às baleias.

de *krill*, observações directas de baleias. Todos estas variáveis estão georeferenciadas. Os dados foram analisados e no fim foi sintetizado visualmente um mapa que demonstra a relação directa entre as concentrações de *krill*, *temperaturas* e as diferentes espécies de baleias.

2.2 SIG e distribuição de energia

O uso de sistemas gráficos na indústria da gestão de redes eléctricas já está estabelecido desde os anos 70[ERSI]. Começou com sistemas de desenho de topologia de rede associados a atributos diversos não gráficos. Estes diagramas contem a informação de como cada equipamento de rede está ligado fisicamente a outro. Este tipo de sistemas foram denominados por “automated mapping/facilities management” (AM/FM), no contexto das redes.

Com passar dos anos associou-se essa topologia a coordenadas geográficas. Adicionando a possibilidade de usar funções interrogação espaciais entre elementos (ex: equipamentos como pontos dentro de um distrito representado por uma área) surgiu naturalmente a associação aos SIG.

A utilização mais básica dada neste contexto de operação é a de registo de infraestruturas eléctricas (cabos eléctricos, equipamentos). Estes sistemas permitem algumas funcionalidades:

- inventários por área geográfica
- planos de manutenção de equipamentos.
- indicação às equipas do terreno da localização precisa (mesmo em profundidade) de equipamentos.

Um Centro de Comando de uma empresa de distribuição de energia é apoiado por um sistema SCADA/DMS (Supervisory Control and Data Acquisition/Distribution Management System). Estes sistemas englobam duas funções interligadas. Uma é a capacidade de monitorizar estados de equipamentos ou grandezas físicas (temperatura, corrente eléctrica, etc), através de sensores geograficamente espalhados por grandes áreas. Também está incluído a capacidade de controlar/operar remotamente esses mesmos equipamentos. O termo DMS, indica a disponibilidade de funcionalidades relacionadas com o conhecimento que se está a operar uma rede eléctrica. Alguns exemplos de funcionalidades adicionais são:

- cálculo de cargas (*load balancing*)
- encontrar caminhos alternativos de transmissão face a uma falha num caminho ou nó da rede.
- visualização em tempo real do estado de energização da rede
- indicação da conectividade entre dois pontos da rede
- ligar/desligar automaticamente a iluminação pública dependente da incidência solar.
- diagnóstico de cascata de eventos de falha [Wendou]

Revisão Bibliográfica

Se adicionar a estes sistemas, dados georeferenciados fora do âmbito da redes eléctricas, abre novas possibilidades de estudos e gestão (ver Tabela 2.1)

Tipo de dados	Possibilidades
Topologia do terreno, áreas protegidas	planeamento de localização de postes de longo curso para transporte de energia[Jiaju].
Índice de vegetação	planeamento de corte de ramos de árvores perto de linhas de transmissão.
Densidade populacional, crescimento esperado futuro	planeamento da localização de novas subestações.
Estradas e número postal das habitações	planeamento de rotas de equipas de manutenção a diferentes pontos da rede. diminuir tempos de resposta das equipas de manutenção mais próximas (em tempo e não distância) a uma falha de equipamento. aviso prévio de áreas afectadas pela manutenção da rede.

Tabela 2.1: Tipos de dados e possibilidades

A maioria dos exemplos dados anteriormente podem ser automatizados por processos matemáticos. Mas uma não menos importante funcionalidade é permitir a um operador humano da rede, ter uma referência visual do ambiente geográfico onde estão inseridos os equipamentos que controla, os quais podem ajudar para decisões críticas no momento.

2.3 Tipos de mapas

O objectivo deste trabalho é permitir visualizar conteúdos geográficos sobrepostos com diagramas de redes eléctricas. Alguns exemplos de conteúdos geográficos são: mapa da rede rodoviária e ferroviária, mapa de divisões administrativas e fronteiras territoriais, mapa geográficos (altitude, rios, lagos), pontos a especificar cidades e outros aglomerados populacionais, pontos a indicar a localização de recursos móveis (ex: frota automóvel), fotografias áreas, estatísticas relacionadas com uma área geográfica (ex: densidade populacional).

O que todos estes exemplos tem em comum é a associação da informação a localizações precisas no planeta.

Existem empresas e organismos públicos especializados na captura, tratamento e disponibilização destes conteúdos. Normalmente implicam custos elevados. Estes são em função da área pedida ou da resolução pretendida para essa mesma área. Por exemplo, o Instituto Geográfico do Exército², comercializa o Modelo de Elevação do Terreno a um custo de 20.000€ para uma resolução de 90metros. Para uma resolução de 30m, o preço sobe para 100.000€.

Esta informação pode ser disponibilizada em dezenas de formatos. A razão da existência destes inúmeros formatos pode ser devido a três factores: às diferentes entidades e respectivas aplicações que capturaram esse conteúdo, informação específica ao domínio e finalmente devido a aplicações de gestão e análise dessa informação.

Os conteúdos geográficos podem ser agrupados em dois tipos: “raster” ou “vectorial”. A escolha pode depender dos seguintes factores: facilidade na captura da informação, necessidade de falta de espaço ocupado, capacidade de processamento do sistemas e flexibilidade na interrogação de dados.

2.3.1 Mapa raster

É um formato em que se representa como uma matriz com um número de linhas (y) e com outro número de colunas (x). Cada posição (x,y) representa uma localização geográfica relativa a uma coordenada absoluta guardada com o mapa, segundo um sistema de coordenadas pré-definido. O valor numérico dessa posição (x,y) é denominado por *pixel* e representa alguma característica física ou estatística desse ponto (por exemplo: código de cor, valor da altitude, densidade populacional). Para a mesma área terrestre, quantos mais pontos forem usados para a representar, melhor a resolução da informação porque cada ponto só se refere a uma pequena área.

Exemplos de conteúdos tipo raster:

- Fotografias áreas.
- Modelo de Elevação Terreno (MET) em que o valor de cada pixel representa a altitude média da área correspondente.
- Mapas temáticos em que cada valor de pixel pode representar uma determinada temática.

A vantagem deste formato é a relativa facilidade de capturar informação. Uma câmara fotográfica captura instantaneamente a imagem de uma área a partir de uma localização bem conhecida. A técnica de "*Synthetic aperture radar*"³ permite obter imediatamente por satélite as

2 <http://www.igeoe.pt/produtos/precario.asp>

3 <http://southport.jpl.nasa.gov/desc/imagingradarv3.html>

características de elevação do terreno de grandes superfícies do planeta (denominado MET). Câmaras de infravermelhos⁴ por sua vez conseguem capturar índices de vegetação existente.

O inconveniente deste formato é ocupar grande espaço em disco. O MET de Portugal Continental, para uma resolução de 30 metros por pixel, ocupa um ficheiro 350GB (ver Figura 6: Altitude). Nesta imagem foi atribuído níveis de cinzento mais claros em função da altitude crescente que representam.

2.3.2 Vectorial

Neste formato, características geográficas são expressas por intermédio de formas geométricas: pontos, linhas e polígonos. Alguns exemplos para cada tipo são: os pontos podem representar cidades ou locais de interesse, linhas podem representar: ruas, rios, linhas de transmissão eléctrica e os polígonos podem ser usados para regiões, lagos e habitações.

As coordenadas são expressas num sistema de coordenadas pré-definido para esse mapa. Esta é composta por dois eixos (latitude e longitude) em coordenadas geográficas ou se a área for projectada podem ser usadas coordenadas cartesianas. Também pode ser adicionado mais um eixo que representa a altitude.

Para além destas primitivas também podem ser associadas características ou propriedades relacionadas com o âmbito do mapa. Um ponto que representa uma localidade pode ter associado uma característica com nome "Classificação". Esta pode ter os valores "cidade", "vila" ou "aldeia". Uma linha eléctrica pode ter a característica "Tensão" e pode conter os valores 220V, 400MV e 20KV. Um polígono pode representar as fronteiras de uma freguesia. Este pode ter como característica "Nome". Este nome pode ser "Porto", "Maia" ou "Valongo".

Pode ser adicionado a dimensão de data e hora aos dados. Desta forma consegue-se descrever fenómenos que variam ao longo do tempo.

Este formato é vantajoso em mais situações que o formato raster . As seguintes qualidades são lhes atribuídas:

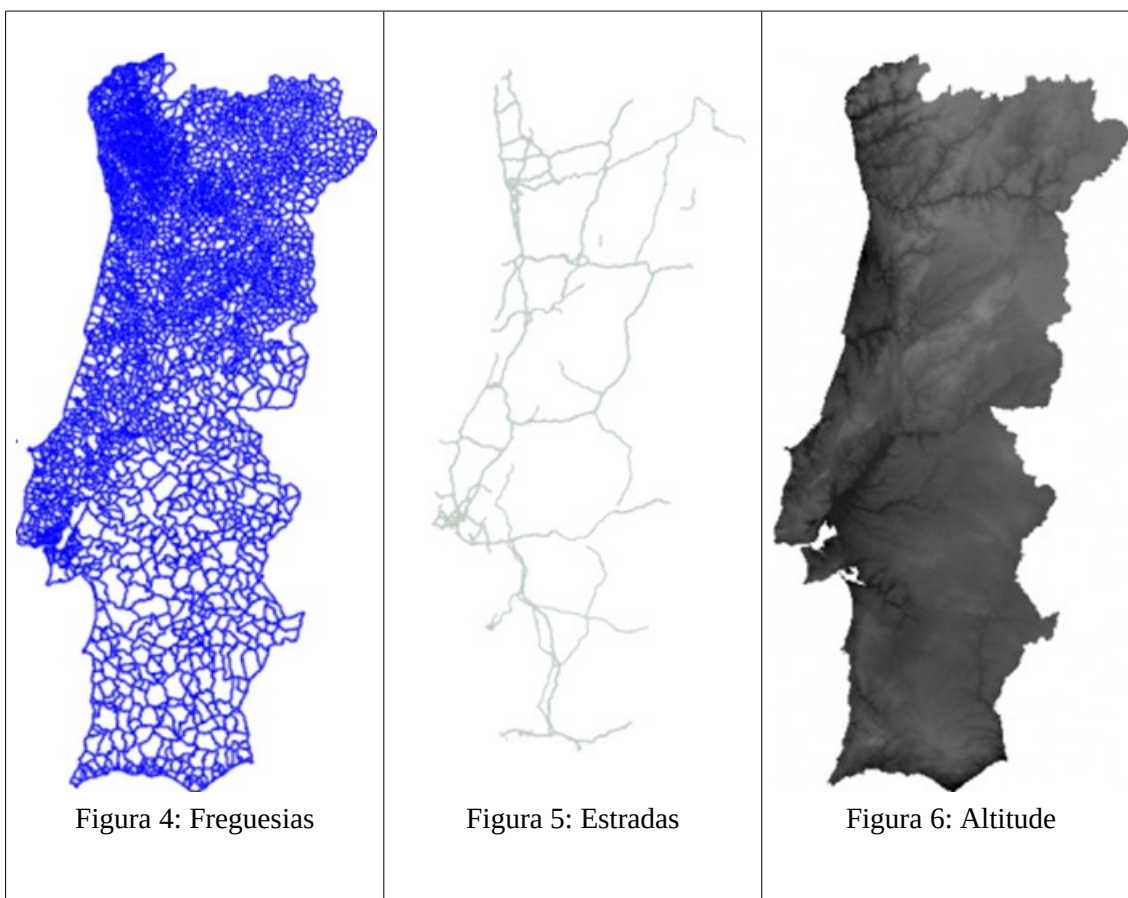
- Ocupam menos espaço em disco. Para definir uma linha basta especificar dois pares de coordenadas (x,y). Num raster teria que ser definido todos os pontos (x,y) por onde a linha passasse.
- É fácil fazer interrogações às geometrias. Operações do tipo área ocupada por um polígono, coordenadas de intercepção de duas linhas análise de grafos, são bem conhecidas da matemática.
- Operações de aproximação (*zoom*) ao mapa não provocam a degradação de imagem como num raster que fica com aspecto de blocos.
- É mais fácil actualizar mapas vectoriais para pequenas alterações no ambiente. Num raster é obrigado a actualizar logo toda uma região de uma vez.

4 <http://science.hq.nasa.gov/kids/imagers/ems/infrared.html>

As desvantagens deste formato são:

- é mais demorado capturar a informação inicial porque normalmente envolve um trabalho manual
- exige recursos de processamento de CPU para cálculos complexos de projecção de pontos num plano.

Os formatos mais encontrados de informação vectorial são: o KML, AutoCAD DXF e ESRI Shapefile. Exemplos destes podem ser visualizados na Figura 4: Freguesias e na Figura 5: Estradas.



2.4 Sistemas de Coordenadas

Os formatos raster e vectorial descritos na secção anterior precisam de ter uma forma de especificar a localização no planeta a que esses mapas se referem. Existe muitos sistemas em uso. Podem ser definidos como coordenadas esféricas em semelhança com a forma do planeta, ou em cartesianas segundo eixos (x,y,z).

2.4.1 Coordenadas Geográficas

As coordenadas geográficas permitem identificar uma localização precisa no planeta. Na Figura 7, está representado o globo terrestre. A coordenada do ponto (P) é obtido pelo ângulo (φ) denominado por Latitude e pelo ângulo (λ) denominado por Longitude.

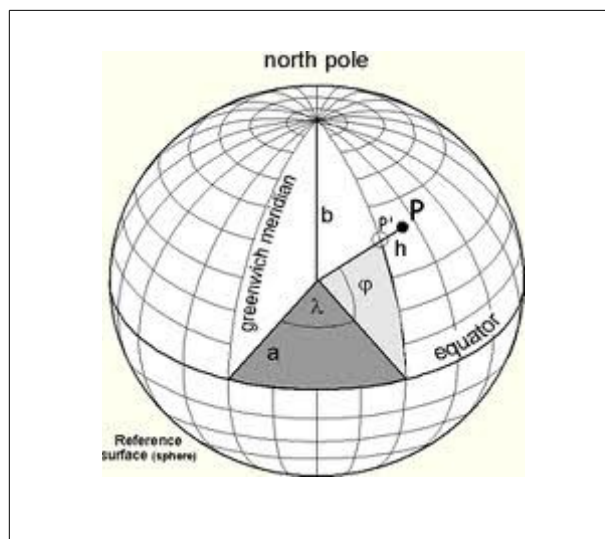


Figura 7: Latitude e Longitude

Latitude é o nome dado à coordenada vertical de um ponto do planeta Terra em relação equador. É uma medida angular em graus, dada pelo ângulo formado entre duas linhas. Ambas têm a mesma origem no centro de massa do planeta. Os outros dois pontos de destino estão separados na vertical. Um deles está localizado na superfície sobre o equador. A outra linha passa pelo ponto na superfície do planeta que se deseja medir. O ângulo varia num intervalo de 0° a 90°. Se estiver acima do equador acrescenta-se a letra (N) de North e a baixo a letra (S) de South.

Longitude é o nome dado à coordenada horizontal de um ponto no planeta em relação ao meridiano de Greenwich. Em contraponto com a Latitude, os pontos são estão separadas na horizontal e um deles está convencionado nesse meridiano. O ângulo total varia entre o 0° e 180°. Pontos à esquerda do meridiano acrescenta-se são identificados com a letra (W) do inglês West, e à direita com a letra (E) de East.

2.4.2 Projecção

O planeta Terra é uma superfície esférica. Coordenadas geográficas são perfeitamente adequadas a essa geometria. Na prática, as pessoas representam mapas em superfícies planas, desde uma folha de papel até a um ecrã de computador. A passagem de uma representação

esférica para plana é conseguida por métodos de projecção. Um novo mapa é criado e este terá um novo sistema de coordenadas.

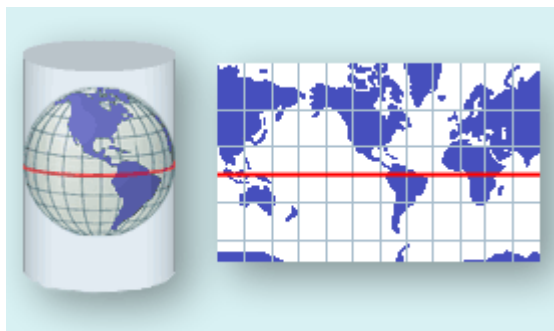


Figura 8: Projecção Mercator

Essa transformação implica alguma distorção da realidade, por isso existem critérios na selecção do tipo de projecção. Estes critérios dependem da superfície total a representar a localização no planeta dessa área ou à aplicação pretendida. Na Figura 8, vemos um exemplo de projecção de Mercator. Uma vista do planeta Terra a partir do Equador é projectado num cilindro. Este cilindro após ser “desenrolado” obtemos um mapa projectado num plano. As coordenadas dos pontos projectados passam a ser cartesianas (x,y).

O planeta terra não é uma circunferência perfeita mas sim uma forma elipsóide. O planeta é mais achatado nos seus pólos que no equador. Ao longo do tempo os valores de referência dessa forma foram sendo sucessivamente refinados. Os valores mais recentes estão definidos na norma GRS-80 (1979) são um raio equatorial de 6.378.137,0000, um raio polar de 6.356.752,3141. Estes valores são importantes para o algoritmo de transformação de coordenadas. A alteração a essas medidas também faz com que se passe a usar outro sistemas de coordenadas.

Na Figura 9 é apresentado uma a imagem dos Estados Unidos⁵ em três projecções diferentes. Nela podemos observar diferenças significativas nas fronteiras à medida que se desloca do centro.

Em Portugal foram definidas ao longo do tempo várias normas de projecção de mapas [PTGRID]. Eles foram aparecendo ao longo da história associado à continua melhoria de processos de captura de informação e de conhecimentos geofísicos. Diferentes partes do território nacional também podem ter as suas coordenadas: Portugal Continental pode ter como referência o castelo de São Jorge como centro do mapa, enquanto o arquipélago da Madeira e dos Açores tem cada os seus em particular. Se fosse necessário desenhar estes três territórios no mesmo mapa, haveria muitas características geométricas que se seriam profundamente distorcidas por estes territórios se situarem distantes entre si no globo.

5 <http://www.colorado.edu/geography/gcraft/notes/mapproj/mapproj.html>

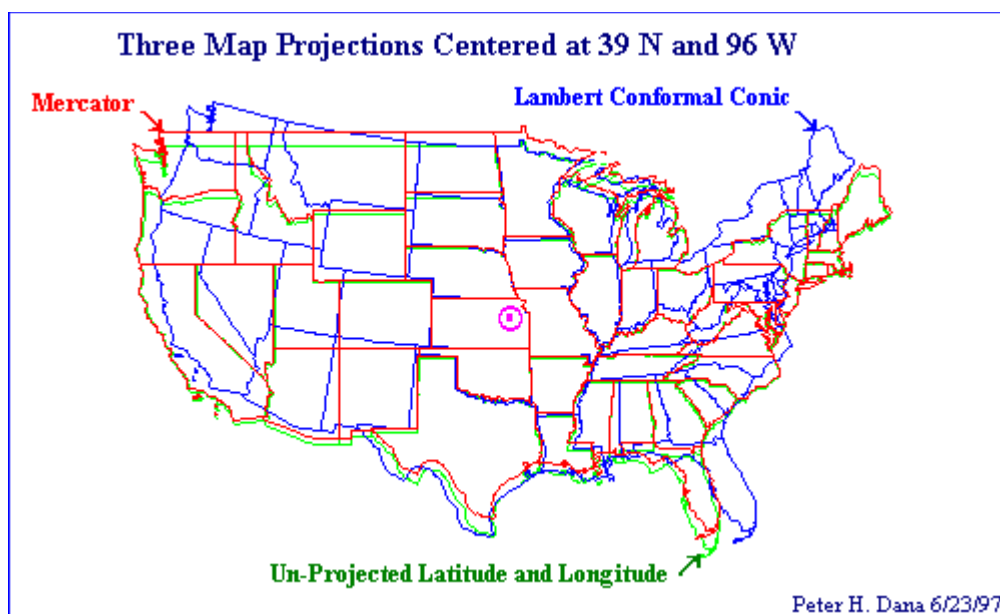


Figura 9: Três tipos de projecções

A projecção usada neste projecto é a mais recente definida para Portugal Continental. Denomina-se por PT-TM06/ETRS89⁶ ou por EPSG:3763. É com base nela que se convertem as coordenadas geográficas para cartesianas segundo a projecção Transverse Mercator.

O organismo International Association of Oil & Gas Producers⁷ define um documento com o nome ESPG que contém o catálogo de todos os sistemas de coordenadas existentes no mundo e suas formulas de conversão. Na Tabela 2.2 é apresentado o formato Well Know Text (WKT) segundo especificado pela OGC, para transmissão desses parâmetros.

Os sistemas de projecção estão sempre presentes em qualquer ferramenta de sistemas georeferenciados.

6 http://www.igeo.pt/produtos/Geodesia/Inf_tecnica/sistemas_referencia/Datum_ETRS89.htm

7 <http://www.ogp.org.uk/>

```

PROJCS["ETRS89 / Portugal TM06",
  GEOGCS["ETRS89",
    DATUM["European_Terrestrial_Reference_System_1989",
      SPHEROID["GRS 1980",6378137,298.257222101,
        AUTHORITY["EPSG","7019"]],
        AUTHORITY["EPSG","6258"]],
        PRIMEM["Greenwich",0,
          AUTHORITY["EPSG","8901"]],
          UNIT["degree",0.01745329251994328,
            AUTHORITY["EPSG","9122"]],
            AUTHORITY["EPSG","4258"]],
            UNIT["metre",1,
              AUTHORITY["EPSG","9001"]],
              PROJECTION["Transverse_Mercator"],
              PARAMETER["latitude_of_origin",39.66825833333333],
              PARAMETER["central_meridian",-8.133108333333334],
              PARAMETER["scale_factor",1],
              PARAMETER["false_easting",0],
              PARAMETER["false_northing",0],
              AUTHORITY["EPSG","3763"],
              AXIS["X",EAST],
              AXIS["Y",NORTH]]

```

Tabela 2.2: WKT de Portugal

2.5 Bases de dados espaciais

Uma base de dados espacial permite guardar os conteúdos vectoriais e rasterizados num tipo de sistema consistente (modelo relacional) e de gestão bem conhecido. As principais vantagens de o usar num sistema geográfico são:

- As geometrias e informação diversa associada ficam juntas no mesmo sistema e são acedidas pela mesma linguagem (SQL)
- Possibilidade de usar índices espaciais que tornam as pesquisas muito mais rápidas.
- Estão disponíveis logo junto aos dados funções matemáticas para análise dos dados e cruzamento de informação de natureza diferentes.
- A projecção de coordenadas para outros sistemas pode ser realizado logo na fonte.
- Guardar imagens georeferenciadas neste sistema.

Actualmente praticamente todas as principais bases de dados suportam extensões espaciais. Por exemplo a Smallworld da GE é própria para redes de energia, a Oracle Spatial/Locator é mais genérica mas com custos elevados e finalmente a Postgres com a extensão Postgis é grátis.

Para melhor compreender o que uma base de dados espacial pode fazer, temos o exemplo 2.3. Um SQL “select” retorna as ruas guardadas na tabela “roads”. Cada rua tem associada as coordenadas da linha que compõe esse segmento da estrada. Uma operação habitual em clientes de visualização de mapas é só retornar os elementos que estão dentro da área visível do ecrã. Para isso pode ser definida uma restrição via SQL, como na Tabela 2.4: Restrição por vista.

```
db=# SELECT road_id, ST_AsText(road_geom) AS geom, road_name FROM roads;
```

road_id	geom	road_name
1	LINESTRING(191232 243118,191108 243242)	Jeff Rd
2	LINESTRING(189141 244158,189265 244817)	Geordie Rd
3	LINESTRING(192783 228138,192612 229814)	Paul St
4	LINESTRING(189412 252431,189631 259122)	Graeme Ave
5	LINESTRING(190131 224148,190871 228134)	Phil Tce
6	LINESTRING(198231 263418,198213 268322)	Dave Cres
7	LINESTRING(218421 284121,224123 241231)	Chris Way

Tabela 2.3: Lista de estradas ⁸

```
SELECT ST_AsText(roads_geom) AS geom
FROM roads
WHERE roads_geom && SetSRID('BOX3D(191232 243117,191232 243119) '::box3d,-1);
```

Tabela 2.4: Restrição por vista

2.6 Gerar mapas

É necessário um componente que converta os conteúdos geográficos num mapa pronto a ser visualizado em *browsers* ou aplicações.

Existem várias ferramentas no mercado. As suas características em comuns são:

- aceder aos conteúdos guardados em base de dados ou ficheiros.
- aplicar uma projecção que o cliente pretender.

⁸ Exemplo retirado de <http://postgis.refractory.net/docs/ch04.html#id2638365>

- ter a capacidade de pintar ou desenhar os mapas segundo qualquer estilos gráficos definido.
- disponibilizar os mapas finais via interface bem definida.

Para gerar uma imagem, é normalmente aplicado uma formatação ou estilo aos conteúdos. O objectivo é sobressair alguma propriedade ou diminuir a quantidade de informação a desenhar.

Existem duas formas de colorir os elementos a desenhar. A primeira e mais simples, o utilizador pode definir a cor depende do tipo de geometria a ser desenhada. Por exemplo, todas as linhas serem desenhadas a verde e os polígonos a azul. A segunda forma mais flexível é desenhar dependente das propriedades que cada geometria tem associado. Uma linha pode representar um estrada nacional ou auto-estrada dependente da propriedade "tipo" associada a essa linha. Numa desenha um risco a vermelho e noutra desenha um três riscos paralelos com a cor amarelo, vermelho e amarelo. Imagens também podem ser estilizadas. Se cada pixel da imagem original representar uma altitude então pode ser convertido na imagem final numa rampa de cores em função desse valor. Outras opções a disponibilizar é coloração dependente do nível de detalhe pretendido. Por fim, estas ferramentas também podem desenhar frases sobre os conteúdos (por exemplo nomes de ruas ou pontos de interesse).

2.7 Pré-geração de imagens

Os geradores de imagens tem a flexibilidade suficiente para servir mapas conforme especificados em tempo real por aplicações clientes. Mas a desvantagem desse trabalho é a quantidade de recursos computacionais que exige enquanto gera os mapas. A consequência disto são principalmente dois: o cliente fica uma dezena de segundos à espera do mapa e o servidor pode ficar entupido com geração em simultâneo de mapas.

Existe uma característica dos mapas que pode ser aproveitada para colmatar esta falha. Os conteúdos geográficos são estáveis ao longo do tempo. Salvo situações excepcionais um mapa gerado ontem será igual ao mapa de hoje. Por exemplo, apesar de surgir estradas novas todos os dias, os mapas só são actualizados mensalmente.

Isto pode ser aproveitado para pré-gerar os mapas em tamanhos de imagem pequenos, normalmente blocos de 256x256 pixels, que depois são servidos em função do pedido realizado.

A estratégia de responder com blocos é gerar uma pirâmide de imagens, em que cada nível corresponde a uma escala pré-definida de aproximação ao mapa. Por exemplo, numa escala grande, representar Portugal Continental basta dois blocos de imagens na vertical. Se a aplicação cliente pedisse um mapa com o tamanho 256 por 512 pixels, era servidor esses dois blocos. Mas se o cliente já pedisse uma imagem 1024x2048 pixels, então já era servido por com imagens a um nível a baixo da pirâmide e com 32 blocos de imagens.

Caso o cliente dos mapas pedisse uma região de 100x100 quilómetros e com um tamanho de mapa de 1000x1000 pixels os blocos de imagens que seriam servidas eram os que mais se aproximassem das resoluções pré-geradas. Os blocos finais terão que ser escalados no final para as dimensões indicadas pelo cliente.

Na Figura 10, podemos visualizar esse crescente nível de detalhe para a representação de todo o globo. Os números são a identificação do bloco da imagem segundo o sistema de cache da Microsoft Bing Maps.

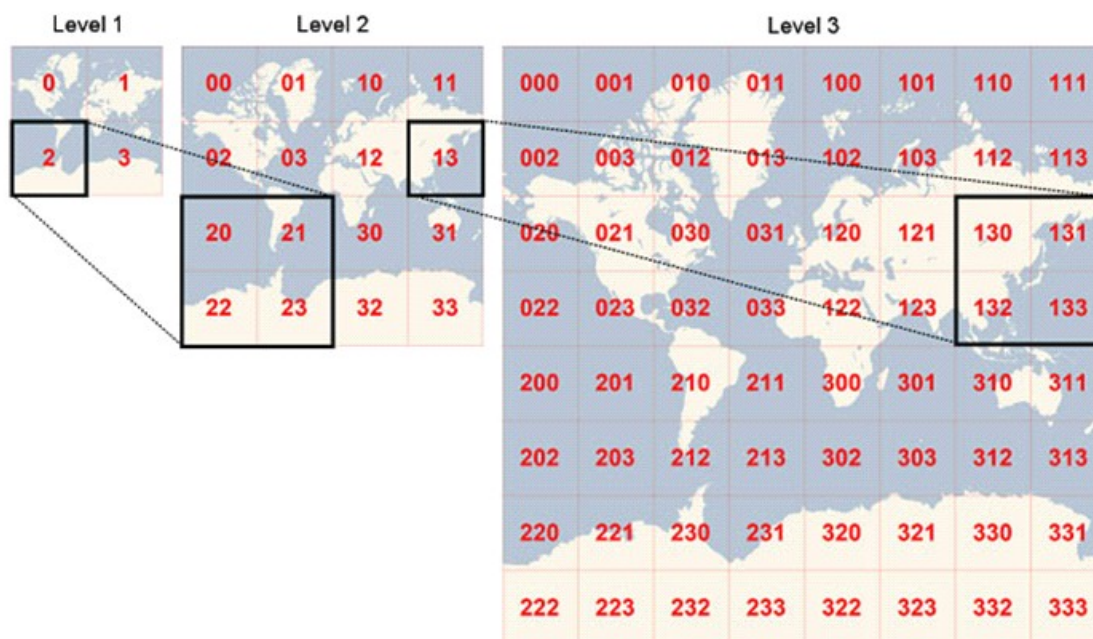


Figura 10: Pirâmide de imagens

2.8 Resumo

Neste capítulo foi introduzido o conceito de Sistema de Informação Geográfica. São apresentados alguns casos de estudo para demonstrar a sua vantagem em os integrar em sistema SCADA/DMS.

É apresentado os tipos de conteúdos existentes e como este ficam capturados num sistema. É indicado o conceito de sistemas de coordenadas e de projecção para esses conteúdos.

Como forma de persistir, manipular e interrogar a informação é apresentado o conceito de base de dados espacial.

Por fim é indicado como as imagens são geradas e as vantagens da técnica de pré-geração das imagens.

3 Metodologia

Este capítulo começa por fazer uma apresentação detalhada do problema a resolver e termina com detalhes da solução encontrada.

3.1 Problema

O sistema SCADA da EFACEC é composto por inúmeros componentes ou aplicações. No âmbito desta dissertação existem duas de interesse aos objectivos propostos. O objectivo é entender onde este trabalho irá melhorar o processo actual e tornar possível os objectivos propostos.

Actualmente com o Scatex+ podemos visualizar a topologia das redes eléctricas via um visualizador denominado Draw. As redes eléctricas visualizadas são divididas em dois tipos: geográfico e unifilares. Rede geográfica é a capacidade de representação da rede como está disposta no terreno. Esta inclui elementos como cabos de transmissão de alta e média tensão ligados a subestações ou postos de transformação (PT). Nos desenhos de unifilares, é visualizado o conteúdo dessas subestações e PT. Os desenhos representam as ligações lógicas entre equipamentos e não como estão fisicamente dispostos no local.

O Draw usa a biblioteca ILOG para implementar as funções de desenho. Esta contém primitivas gráficas suficientes para o desenho pretendido. Estas primitivas são as linhas, rectângulos, circunferências e pontos. Como complemento podemos também incluir imagens como fundo aos desenhos e escrever texto.

As coordenadas dos elementos desenhados no Draw segue a norma portuguesa de projecção EPSG:3763.

A fonte de dados que alimenta o Draw é da responsabilidade dos clientes. No contexto do desenvolvimento deste trabalho, foi usada a base de dados da EDP. O processo de importação começa com a captura e gestão da informação no sistema adquirido pela EDP

Metodologia

denominado por Smallworld da GE Energy. Todos os dias são feitas alterações como adição de equipamentos ou disposição de linhas na rede. Existe um processo que importa uma vez por dia esses dados para a base de dados do Scatex+ e para dois ficheiros de texto. São os ficheiros de texto que alimentam os visualizador de SCADA Draw.

Estes ficheiros são de dois tipos: disposição geométrica de equipamentos e o outro são os atributos não gráficos desses equipamentos. O conteúdo está num formato proprietário da EFACEC denominado por “stringuificado” (ver Tabela 3.1). Este são carregados na totalidade para a memória logo no processo de arranque do componente.

```
[PolyLine=[WorldId=0][Aerial=1][Type=line][Symbol=troco_15]
[NominalVoltage=15.000000000000000][Points[*4]=[-79774880.00000000][ -
120698400.00000000][ -79863574.00000000][ -120882032.00000000]]
[WorkingVoltage=15.000000000000000][UId=(4,26986471):2]
[Representation=troco_mt][NormalPosition=0][SiglaPath=1504L2023400.]
[ModelClass=troco_mt]]

[CIMText=[Text=1164.1 / 1 / AA 3x160][SiglaPath=1504L2023400.]
[Justification=22][Orientation=0][Y=-120765402.00000000]
[NormalPosition=0][ModelClass=troco_mt][Scale=1.0000000000000000]
[WorldId=0][Type=line][UId=(4,26986471):2]
[NominalVoltage=15.000000000000000][Representation=troco_mt]
[Angle=1.12084209918976][Symbol=anotacao_troco_mt][X=-
79807873.00000000]]
```

Tabela 3.1: Stringuificado

O objectivo desta dissertação é integrar conteúdos geográficos diversos fora do âmbito das redes eléctricas, no sistema existente. Este pode ser melhorado nas seguintes vertentes: na fonte, nos formatos e alojamento desses dados, na facilidade de acesso e análise automática e finalmente na qualidade de desenho.

O problema da **fonte de dados** é a dependência que Scatex+ tem do seus clientes para fornecer elementos geográficos complementares. O cliente só captura elementos necessários dentro da sua área de actividade. A solução passa obrigatoriamente por recorrer às entidades externas para fornecer conteúdos. Devido a isto é introduzida mais uma dificuldade, estes dados podem ser capturados em diferentes **formatos**. Estes podem ser do tipo vectorial ou raster, podem estar armazenados em diferentes tipos de ficheiros e estar com sistemas de coordenadas diversas.

Actualmente existe uma aplicação desenvolvida especificamente para o processo de importação do Smallworld para o Scatex+. A partir do momento que o leque de formatos se multiplica, deixa de ser viável gastar recursos no desenvolvimento específico de cada importador de dados. Para além disso o formato de dados proprietário da EFACEC funciona bem para guardar elementos geométricos mas não contempla as imagens. Por isso se impõe uma

solução de alojamento de dados o mais genérica e standard possível. Dessa forma só é necessário as aplicações se preocuparem com um tipo de interface de acesso a esses dados.

Existe um problema latente relacionado com a necessidade de todos os dados terem que estar presente em memória. Actualmente o arranque do Draw no ambiente de testes demora 1 minuto a carregar todos os dados e o processo passa a ocupar 300MB de memória. O sistema tem operado bem assim. Mas a partir do momento que se amplia a quantidade de conteúdos e se passa a usar imagens deixa de ser viável a estratégia actual. Por exemplo, o modelo de elevação do terreno (MDT) de Portugal Continental ocupa 300MB.

O sistema actual tem capacidades avançadas de análise das propriedades eléctricas da rede. Para isso estão disponíveis internamente funções específicas neste âmbito aplicadas a topologias (por ex: calculo de cargas eléctricas e áreas da rede energizadas). Não existem funções para **análise** de relações espaciais entre elementos. Um exemplo de relações espaciais é interrogar para um determinado ponto da rede, quais as ruas que passam mais próximas.

Por fim, falta referir a vertente de **visualização**. O Draw oferece uma interacção rica com o utilizador. Isto inclui as funcionalidades normais de selecção de elementos com o rato, aproximação em áreas do desenho com níveis de detalhe crescentes e alterações de estilos de desenho em tempo-real em função de propriedades da rede. No caso de visualização de mapas, este última componente estilo dinâmico não é importante, porque a informação é estática. Mas em contrapartida, a visualização de mapas de estradas e de imagens georeferenciadas necessita de mais opções de **estilização** de elementos. Por exemplo o Draw não suportar a aplicação de texturas dentro de elementos, várias cores na mesma linha e textos que seguem a curvatura das linhas. Estes não são importantes para a operação da rede, mas são importantes para um desenho de mapas agradável que seguem os estilos correntes.

Existem também duas restrições que afectam uma potencial solução. O Draw deve ser usado para visualizar conteúdo geográfico e todas as ferramentas necessárias deverão ser livres e grátis.

3.2 Proposta de arquitectura

O sistema do Scatex+ é eficaz na visualização da rede eléctrica. Por esta razão pretende-se que a integração das funcionalidades de um SIG tenham o mínimo impacto nas aplicações existentes.

O Draw é o visualizador de eleição no Scatex+ enquanto que os mapas são acessórios. Não há necessidade de permitir ao operador seleccionar esses elementos geográficos com o rato nem a necessidade de mudar cores ou estilos desses desenhos em tempo-real. Com base nisto optou-se por integrar no Draw a visualização de imagens de mapas. Desta forma o Draw não precisa de ser adaptado tendo em conta todas as necessidades de base de um SIG. Essa responsabilidade será passada para outros componentes. Este tipo de utilização é denominada por “*Thin Client*”.

Metodologia

Por isso, a solução passou por criar uma arquitectura semelhante para disponibilizar mapas em aplicações tipo Web.

Na figura Figura 11 encontramos a disposição dos elementos necessários. A solução passa por integrar no Draw a biblioteca GDAL de acesso a dados. Esta tem funcionalidades para aceder a imagens por protocolos de rede normalizados (TMS e WMS). O componente GeoServer é responsável por gerar as imagens usando como fonte a base de dados Postgres ou ficheiros próprios. Como a geração de imagens demora sempre alguns segundos, introduziu-se entre o Draw e o GeoServer um mecanismo de cache de imagens pré-geradas baseada na aplicação TileCache.

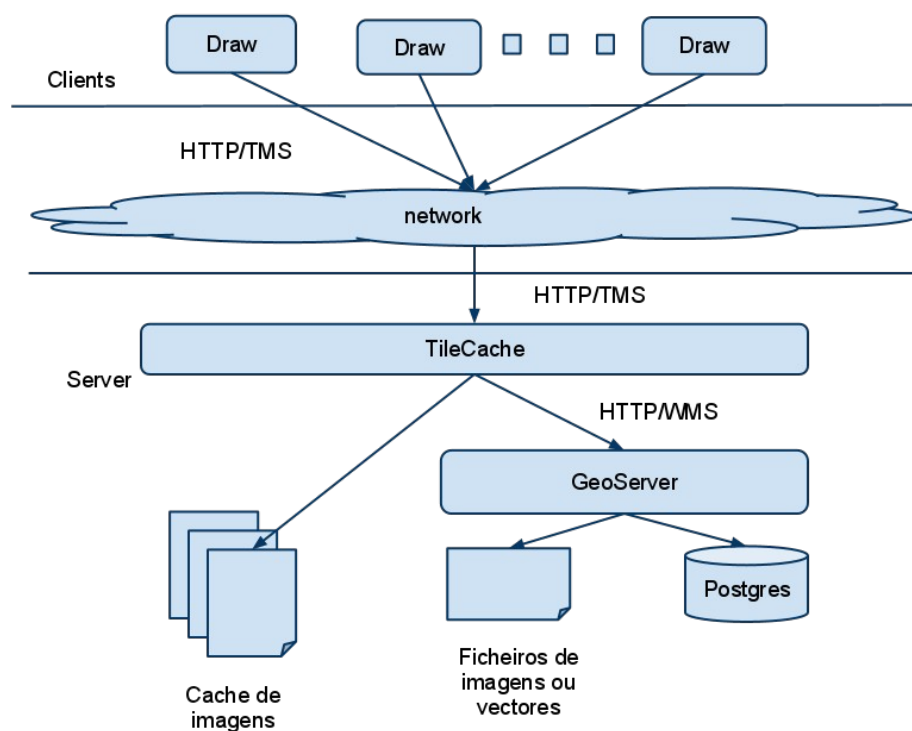


Figura 11: Arquitectura

Em relação às necessidades identificadas na secção 3.1 Problema, a Tabela 3.2 indica os componentes responsáveis por colmatar essas limitações.

Metodologia

Formatos de dados	Postgres, Geoserver e GDAL tem a capacidade de ler inúmeros formatos.
Repositório de dados	Postgres é a base de dados para conteúdos vectoriais e de raster.
Ferramentas de análise	Postgres/Postgis contém as ferramentas de base para essa análise e tratamento de dados.
Visualização	O Draw consegue visualizar mapas via biblioteca GDAL.
Estilização	O GeoServer implementa a especificação SLD de estilização de mapas.

Tabela 3.2: Áreas resolvidas pela arquitectura

No capítulos anteriores, já foi feita a referência à utilidade dos componentes 2.5 Bases de dados espaciais, 2.6 Gerar mapas e 2.7 Pré-geração de imagens. Nas secções seguintes é feita a introdução a protocolos de acesso a dados WMS e TMS, a biblioteca GDAL e o SLD.

3.2.1 WMS

A interface Web Map Service (WMS) foi definida com o propósito de aceder a mapas dinâmicos e georeferenciados via *web*. Este serviço pode ser invocado via *Browser* através de um endereço HTTP no formato Uniform Resource Locators (URL). As operações e configurações dos pedidos são todas definidas como parâmetros desse endereço. Este *standard* define três operações: GetMap, GetCapabilities e GetFeatureInfo.

GetMap

Para a realização deste projecto só a funcionalidade de obtenção de mapas é importante. A seguir apresentamos um exemplo de um pedido segundo a norma WMS.

```
http://localhost:8080/geoserver/wms?
service=WMS&version=1.1.0&request=GetMap&layers=portugal:caop3&styles=&bbox=-119191.408
,-300404.804,162129.081,276083.767&width=996&height=2048&srs=EPSG:3763&format=image/png
&TRANSPARENT=TRUE
```

Tabela 3.3: Exemplo URL de pedido WMS

O significado dos parâmetros está decomposto na Tabela 3.4.

Attributo	Significado
http://localhost:8080/	Indicação da localização na rede do servidor de imagens.
Request=GetMap	Indicação que o tipo do serviço é pedido de imagem.
Layers=portugal:caop	Indica o nome do mapa a devolver.
srs=EPSG:3763	Código do sistema de coordenadas a usar para projectar a imagem final.
Bbox=-119191.408, -300404.804, 162129.081, 276083.767	Define um rectângulo no qual só devolve os elementos geográfico no seu interior.
width=996&height=2048	Tamanho da imagem a retornar. A rácio de altura e largura deve ser o mesmo que o parâmetro "bbox". Caso contrário a imagem gerada irá ser distorcida e não corresponde à realidade.
format=image/png	A imagem gerada deve está segundo a codificação PNG.
TRANSPARENT=TRUE	A imagem gerada deve ter uma camada com a informação de transparência da cada pixel.

Tabela 3.4: Parâmetros de um pedido WMS

Os parâmetros "Transparent" e "Layers" são importante porque permite obter de uma só vez uma imagem com vários mapas desenhados sobrepostos.

GetCapabilities

Esta funcionalidade faz-se com o pedido "Request=GetCapabilities". O objectivo não é retornar uma imagem mas sim um documento XML com a descrição dos mapas e suas características que estão configurados no servidor na altura do pedido. O cliente depois pode usar essa informação para mostrar ao utilizador os mapas disponíveis.

GetFeatureInfo

A funcionalidade permite ao cliente dos mapas obter mais informação sobre um determinada coordenada que está a visualizar. Por exemplo, se o utilizador clicar numa linha, o cliente de mapas, constrói um pedido WMS com a coordenada dessa linha e a identificação do mapa. O servidor responde com um documento XML, sobre entidades a que esse ponto se pode referir. Esta funcionalidade é importante para futura expansão de funcionalidades para além de mostrar mapas.

3.2.2 GDAL

Metodologia

A biblioteca Geospatial Data Abstraction Library⁹ (GDAL) é uma ferramenta de manipulação de conteúdos geográficos. É escrita em C++, com código aberto (*opensource*) e livre (licença X11/ MIT) e que pode ser compilado em sistemas Linux (GCC4) e Windows (VS2005).

As funcionalidades disponibilizadas são:

- ler mapas em praticamente todos os formatos e suportes em uso corrente.
- obter informação de metadados associados aos conteúdos geográficos como sistema de projecção usado e número de bandas de informação.
- reprojectar mapas noutros sistemas de coordenadas
- cópia de conteúdos entre formatos (imagem e vectorial) e suportes informáticos (ficheiros e base de dados)
- coloração de mapas MDT com efeitos 3D de sombreado do terreno
- criação de mosaicos de imagens

Esta ferramenta está incluída em mais de sessenta aplicações de informação geográfica e com a consequente base de suporte de desenvolvimento e suporte.

Existe uma particularidade encontrada nesta biblioteca que é essencial para o protótipo. Ela permite ler mapas disponibilizados por rede pelos protocolos WMS e TMS.

3.2.3 TMS

O serviço Tile Map Service (TMS) é um protocolo de obtenção de imagens parte de mapas maiores, já pré-gravados num servidor. É implementado por cima do serviço HTTP. Mais uma vez este protocolo é definido pela OGC.

Um exemplo de endereços que especificam blocos de imagens é indicado na Tabela 4. Este significa que está a requerer um mapa denominado por "vmap0" no nível de zoom ou resolução 2, e o bloco que se encontra na coluna 3, linha 4.

http://tms.osgeo.org/1.0.0/vmap0/2/3/4.jpg

Tabela 3.5: Identificação de cada mosaico em TMS

A Tabela 3.5 indica como são identificados os blocos para um determinado nível da pirâmide. O nível da pirâmide é dado por um conjunto de escalas pré-estabelecidas na configuração do TileCache.

⁹ <http://www.gdal.org/>

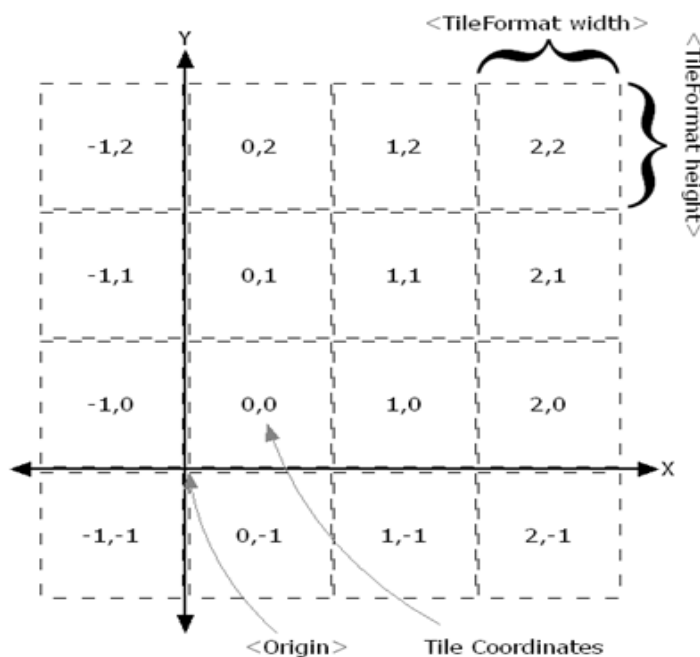


Figura 12: Coordenadas TMS

Este tipo de endereço tem a particularidade de mapear directamente para uma estrutura de directorias num sistema de ficheiros onde o ficheiro se pode encontrar. Assim o software que implementa o serviço de cache é muito mais simples e rápido porque não precisa de processar variáveis no endereço URL.

Este servidor pode ter uma funcionalidade extra que é, quando uma imagem ainda não está gerada, sabe consultar o GeoServer via protocolo WMS para gerar essa imagem em falta.

O Draw terá a capacidade de obter os mapas por TMS ou por WMS. A escolha vai depender da performance pretendida em contraponto com a versatilidade do WMS para aplicar estilos gráficos (ver 3.2.4).

3.2.4 Estilização de mapas

A vantagem do Draw simplesmente ler uma imagem é que não é necessário o trabalho implementar todo o tipo de funções de coloração de dados. O GeoServer já executa essa tarefa por ele. Assim, resta só indicar a esta aplicação como se desenham os elementos. O GeoServer implementa um protocolo denominado de Style Layer Descriptor[SLD] para essa tarefa.

O SLD é um conjunto de regras que define como cada elemento geométrico é desenhado. Existem várias estratégias de desenho. A mais simples é baseada na geometria dos dados. Todas as linhas são pintadas com cor vermelha. Todos os polígonos são pintados nas bordas de azul e no interior de verde.

Metodologia

Uma funcionalidade mais avançada é desenhar conforme o tipo de propriedades associadas as geometrias. Por exemplo, as estradas no OpenStreetMaps, tem a propriedade “highway” associado a linhas. Esta propriedade pode ter os valores “primary”, “secondary” e “others”. Se for “primary” representa auto-estradas, por isso são desenhadas com três traços paralelos (vermelho, amarelo, vermelho). Se for “secondary” representa outras estradas e pode ser desenhado como uma linha amarela de 1 pixel. Para “others” não é desenhado nada.

Outra forma de filtragem de elementos é em função do nível de escala do mapa. Se estivermos a observar o mapa de Portugal, podemos indicar só para visualizar as auto-estradas. Mas se aproximar a vista para a área do Porto, ficam visíveis as restantes estradas e vias.

Por fim, para cada regra e geometria, existem várias construções possíveis de serem usadas para estilizar os mapas. As existentes são: linhas, polígonos, circunferências, texto e imagens. Cada uma pode ser opções de preenchimento com cores, níveis de transparência, tipo de fontes, espessura do traço usado (“stroke”), etc.

Esta especificação está intimamente ligado ao protocolo WMS. No URL usado pelo Draw para obter os mapas, é possível indicar o parâmetro “Styles”. O nome atribuído a esse campo é o estilo a usar para pintar o mapa. No exemplo da Tabela 3.6, estamos a pedir para mostrar os dados “Roads” e “Houses” em que aplicamos respectivamente os estilo “centerline” e “outline”.

```
http://yourfavoritesite.com/WMS?VERSION=1.1.0&REQUEST=GetMap&
BBOX=0.0,0.0,1.0,1.0&WIDTH=400&HEIGHT=400&FORMAT=PNG
LAYERS=Roads,Houses&
STYLES=CenterLine,Out line
```

Tabela 3.6: Especificação do SLD via WMS

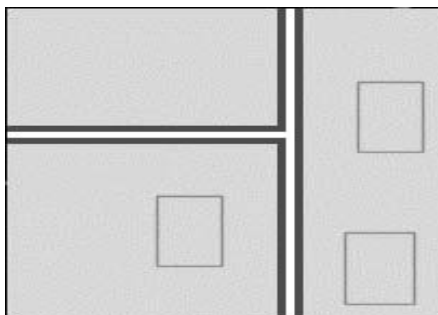


Figura 13: Rua e Casas

O resultado desta chamada WMS pode ser a imagem da Figura 13, em que os quadrados representam habitações e os riscos paralelos, as estradas.

A forma de especificação das regras de desenho é com base numa sintaxe XML, com elementos bem definidos (ver Tabela 3.7).

Esta especificação implementada no GeoServer, é muito mais abrangente do que as funcionalidades existentes no Draw, por isso não compensa trabalho suplementar de implementação do Draw.

```

<?xml version="1.0" encoding="utf-8"?>
<sld:StyledLayerDescriptor xmlns="http://www.opengis.net/sld"...
...
    <sld:Rule>
        <sld:Name>rule motorway trunk min
1000000</sld:Name>
        <sld:Title>&lt;diferente&gt;;, motorway,
motorway_link,
trunk, trunk_link, min 1000000</sld:Title>
        <ogc:Filter>
            <ogc:Or>
                <ogc:PropertyIsEqualTo>
                    <ogc:PropertyName>highway</ogc:PropertyName>
                    <ogc:Literal>motorway</ogc:Literal>
                </ogc:PropertyIsEqualTo>
            </ogc:Filter>
        <sld:MinScaleDenominator>
1000000</sld:MinScaleDenominator>
        <sld:LineSymbolizer>
            <sld:Stroke>
                <sld:CssParameter name="stroke">
#000000</sld:CssParameter>
                <sld:CssParameter name="stroke-width">
3</sld:CssParameter>
                <sld:CssParameter name="stroke-linejoin">
round</sld:CssParameter>
            </sld:Stroke>
...

```

Tabela 3.7: Exemplo SLD

3.3 Resumo e Conclusões

Este capítulo está dividido em duas secções. Começam por ser identificadas as lacunas do sistema Scatex+ que dificultam o suporte de visualização de mapas. Na secção seguinte é indicado a arquitectura proposta como resposta.

4 Implementação

Este capítulo é dedicado a expor pormenores de implementação da metodologia proposta no capítulo anterior. São apresentadas diferentes formas de importar dados geográficos de teste. São apresentadas as alterações necessárias ao Draw e como foi estilizado o mapa de estradas.

4.1 Integração de mapas de teste

O primeiro passo da validação da protótipo é importar diferentes tipos de dados. As secções seguintes indicam como se importam os três tipos de conteúdos geográficos usados em testes.

4.1.1 Integração de MDT

O Modelo Digital do Terreno (MDT) é um mapa do tipo imagem em que cada pixel ou ponto corresponde a uma área geográfica de Portugal Continental. O valor desse pixel não é um código de cor mas sim o valor da altitude média dessa área. No MDT usado cada pixel corresponde a 30m no terreno. Este mapa foi capturado e disponibilizado pela NASA gratuitamente e no formato GeoTiff. Este mapa é usado no âmbito deste trabalho para controlo de projecção no visualizador.

Fonte	http://blog.viasig.com/2010/03/mdt-30m-para-portugal/
Data	27 Novembro, 2010
Formato	GeoTIFF, raster
Projecção	Hayford-Gauss Datum 73

Tabela 4.1: Dados MDT

Implementação

O único tratamento de imagem dado ao mapa é a geração de perfil de relevo com base em sombras a partir da informação da altitude. A ferramenta usada é o **gdaldem** da biblioteca GDAL.

```
C:\portugal_terrain\mdt>gdaldem hillshade -z 4 mdt.tif pt.tif  
0...10...20...30...40...50...60...70...80...90...100 - done.a
```

Tabela 4.2: Execução do comando “gdaldem”

É gerado o ficheiro pt.tif o qual é importado para o GeoServer.

4.1.2 Integração do CAOP

O mapa Carta Administrativa Oficial de Portugal (CAOP) representa a fronteira territorial soberana, fronteiras geográficas naturais e fronteiras administrativas (freguesias e concelhos). É disponibilizada gratuitamente pelo Instituto Geográfico Português.

Este mapa é do tipo vectorial. Este mapa é importante porque é usado como controlo. A rede eléctrica actualmente desenhada, contém também linhas auxiliares que identificam as fronteiras portuguesas. Essas linhas têm que estar sobrepostas com as do CAOP.

Fonte	http://www.igeo.pt /produtos /cadastro /caop /download /CAOP2010_Shapes /Cont_troco_CAOP2010.zip
Data	27 Novembro, 2010
Formato	ERSI Shapefile, vectorial
Projecção	EPSG:3673

Tabela 4.3: Dados CAOP

As instruções seguintes indicam como importar os dados para a base de dados Postgres. Preparar a base de dados:

```
D:\soft\PostgreSQL\8.4\bin>createuser -U postgres caop  
Shall the new role be a superuser? (y/n) y  
  
D:\soft\PostgreSQL\8.4\bin>createdb -U postgres -E UTF8 -O caop caop  
  
D:\soft\PostgreSQL\8.4\bin>createlang -U postgres plpgsql caop  
  
D:\soft\PostgreSQL\8.4\bin>psql -U postgres -d caop -f  
..\share\contrib\postgis-1.5\postgis.sql
```

Tabela 4.4: Instruções preparar o Postgres para o CAOP

Implementação

Do ficheiro *Cont_troco_CAOP2010.zip* extrair o seu conteúdo para “c:\”.

Correr a aplicação **Shape File to PostGIS Importer**. Esta aplicação é instalada juntamente com o PostgreSQL. Os seguintes parâmetros são necessários para a correcta importação:

Username	= postgres
Database	= caop
SRID	= 3763
Shapefile	= c:\Cont_AAd_CAOP20090\Cont_AAd_CAOP20090.shp

Tabela 4.5: Parâmetros para importação da CAOP

4.1.3 Integração da rede viária

A OpenStreetMaps (OSM) é um projecto colaborativo de pessoas anónimas com o objectivo de captura e catalogação de informação relativa a estradas e outros pontos de interesse. O resultado é uma base de dado com conteúdo grátis. São disponibilizados num formato XML do tipo vectorial. O objectivo deste tipo de mapas é avaliar as capacidades de definição de estilos do GeoServer segundo o protocolo SLD e análise da qualidade do desenho quando colocadas no ecrã.

Fonte	http://downloads.cloudmade.com/ europe/ portugal/ portugal.osm.bz2
Data	27 Novembro, 2010
Formato	OSM, vectorial
Projecção	EPSG:90918

Tabela 4.6: Dados OSM

As seguintes instruções são necessárias para importar as estradas, para a base de dados Postgres.

Implementação

```
D:\soft\PostgreSQL\8.4\bin>createuser -U postgres osm
Shall the new role be a superuser? (y/n) y

D:\soft\PostgreSQL\8.4\bin>createdb -U postgres -E UTF8 -O osm gis

D:\soft\PostgreSQL\8.4\bin>createlang -U postgres plpgsql gis
createlang: language "plpgsql" is already installed in database "gis"

D:\soft\PostgreSQL\8.4\bin>psql -U postgres -d gis -f
"D:\soft\PostgreSQL\8.4\share\contrib\postgis-1.5\postgis.sql"

D:\transferências\osm2pgsql\osm2pgsql>osm2pgsql -c -d osm -U postgres
-W -H localhost -P 5432 D:\workspace\mapas\portugal.osm
```

Tabela 4.7: Importação OSM para Postgres

Após importação é necessário converter da projecção EPSG:90918 para a EPSG:3763. As seguintes instruções são executadas dentro da linha de comando do PLSQL.

```
psql -U postgres

psql> update planet_osm_point set way=transform(way,3763);

psql> update planet_osm_line set way=transform(way,3763);

psql> update planet_osm_polygon set way=transform(way,3763);

psql> update planet_osm_roads set way=transform(way,3763);

psql> update geometry_columns set srid=3763 where
f_table_name='planet_osm_point';

psql> update geometry_columns set srid=3763 where
f_table_name='planet_osm_line';

psql> update geometry_columns set srid=3763 where
f_table_name='planet_osm_roads';

psql> commit;
```

Tabela 4.8: Conversão da projecção OSM

4.2 Optimização de mapas via TileCache

Os primeiros testes de consulta de informação via protocolo WMS, revelaram um consumo de CPU excessivo e tempo de resposta de desenho superior a 5 segundos para obter um mapa de estradas à resolução de 1280x1024 pixels.

Procedeu-se uma pequena análise do ambiente de teste para descobrir qual era o processo mais solicitado. Foram realizadas várias operações manuais de “zoom” no mapa enquanto se observava a taxa de ocupação do CPU via ferramenta Process Explorer¹⁰. Esta ferramenta revelou que era o processo do GeoServer (o gerador de imagens) que ocupava por completo o CPU por alguns segundos. Empiricamente este comportamento não é aceitável tendo em conta que numa instalação actual do Scatex+, podem existir 20 clientes Draw a aceder a mapas em simultâneo.

Foi incorporado no sistema a ferramenta TileCache. O objectivo foi melhorar o desempenho através da cache de imagens pré-geradas. Os mesmo testes anteriores comprovaram que a taxa de ocupação global de CPU baixou para os 20%, ocupado maioritariamente pelo Draw. O intervalo de tempo entre uma acção de zoom no ecrã e o desenho do mapa ficou sempre em menos de 3 segundos.

A Tabela 4.9 indica a configuração do TileCache usada para testes.

```
[cache]
type=Disk
base=D:\workspace\tilecache-2.10\cache[pt_caopa]
type=WMS
url=http://localhost:8080/geoserver/wms?
extension=png
layers=pt_caopa
bbox=-119191.408,-300404.804,162129.081,276083.767
srs=EPSG:3763
levels=20
size=256,256
```

Tabela 4.9: Configuração do TileCache

4.3 Draw

As alterações ao nível do Draw consistiram em fazer chamadas à biblioteca GDAL para obter as imagens dos mapas. Todo o código necessário está descrito na íntegra no Anexo A: Draw e GDAL. Nesta secção só será focado as partes mais importantes.

A implementação desta parte é baseada na aplicação de teste do próprio GDAL, denominada por *gdal_translate*¹¹. Esta aplicação demonstra o uso da API em que obtém mapas via protocolo WMS/TMS, de um servidor externo e gera um ficheiro no disco local com esse

¹⁰ <http://technet.microsoft.com/en-us/sysinternals/bb896653>

¹¹ http://www.gdal.org/gdal_translate.html

Implementação

conteúdo. No caso desta integração, não é necessário criar o ficheiro. Basta gerar a imagem em memória sem qualquer tipo de codificação aplicada (PNG ou JPG).

O primeiro passo é indicar à biblioteca parâmetros sobre o mapa que pretendido. A função implementada é: `DRWView::build_xmlfile_for_gdal`. Para isso, é necessário definir um texto no formato XML essa configuração como indicado como exemplo na Tabela 4.10. Nela é definido: o endereço do servidor de mapas (o GeoServer), o nome do mapa (o CAOP), e parâmetros de definição da *bounding box* do mapa e o protocolo a usar (o TMS).

As coordenadas da área do mapa a mostrar são obtidas directamente da aplicação do Draw. Este já tem funções que convertem tamanhos de janelas em pixels para a projecção EPSG:3763, usada nos mapas.

Esta configuração é gerada num ficheiro virtual. Não tem existência física no disco.

```
<GDAL_WMS>
<Service name="TMS">
<ServerUrl>http://localhost:8888/</ServerUrl>
<Layer>pt_caopa</Layer>
<Format>png</Format>
</Service>
<DataWindow>
<UpperLeftX>-119191.408000</UpperLeftX>
<UpperLeftY>276083.767000</UpperLeftY>
<LowerRightX>169052.877500</LowerRightX>
<LowerRightY>-300404.804000</LowerRightY>
<TileLevel>20</TileLevel>
<TileCountX>1</TileCountX>
<TileCountY>2</TileCountY>
</DataWindow>
<Projection>EPSG:3763</Projection>
<BlockSizeX>256</BlockSizeX>
<BlockSizeY>256</BlockSizeY>
<BandsCount>4</BandsCount>
</GDAL_WMS>
```

Tabela 4.10: Pedido via GDAL

Esse ficheiro virtual é criado e entregue ao GDAL via funções: `"VSIFileFromMemBuffer(...);"` e `"GDALOpenShared( pszSource, GA_ReadOnly )."`

As intruções da função `GDALGetDataset(...)` são uma cópia do código do `"gdal_translate"`, até ao passo de construção da imagem na memória.

As funções usadas para essa construção são:

Implementação

```
bandRed-> RasterIO(GF_Read, 0,0, nOXSize,nOYSize, imageData+0,
nOXSize,nOYSize,GDT_Byte,4,nOXSize * 4);

bandGreen-> RasterIO(GF_Read, 0,0, nOXSize,nOYSize, imageData+1,
nOXSize,nOYSize,GDT_Byte,4,nOXSize * 4);

bandBlue-> RasterIO(GF_Read, 0,0, nOXSize,nOYSize, imageData+2,
nOXSize,nOYSize,GDT_Byte,4,nOXSize * 4);

bandAlpha-> RasterIO(GF_Read, 0,0, nOXSize,nOYSize, imageData+3,
nOXSize,nOYSize,GDT_Byte,4,nOXSize * 4);
```

Tabela 4.11: Funções de leitura por banda

Um pixel é representado por 4 bytes. Cada byte corresponde a uma componente de cor RGB (vermelho, verde e azul), e nível de transparência (*alpha*). Para a biblioteca GDAL, cada pixel não corresponde obrigatoriamente a uma imagem. Cada conjunto de “bytes” para um pixel, pode corresponder a valores de altitude ou índice populacional e não um código RGB. Para a implementação foi realizada a simplificação de considerar que o GDAL recebe sempre imagens no formato RGB.

É construído uma estrutura em memória “imageData” com essa imagem, a qual é entregue à biblioteca ILOG via função `IlvBitmap(...)`. Esta é usada automaticamente pelo Draw para colocar a imagem do mapa como fundo da janela.

4.4 SLD

Foi realizada a experiência de estilização de mapas de estradas via SLD. No Anexo C, encontramos um excerto do ficheiro criado.

O teste consistiu em três partes da estilização: desenho com base na escala da imagem, desenho em função do tipo de entidade e colocação de nomes de ruas.

A base de dados da OpenStreetMaps define vários tipos de estradas¹². No SLD de teste, essas estradas foram agrupadas em três tipos para simplificação, como indicado na Tabela 4.12.

12 <http://wiki.openstreetmap.org/wiki/Key:highway>

Implementação

Tipo de Estrada	Identificações usadas no OpenStreetMaps
Auto-estradas	<code>motorway, motorway_link, trunk, trunk_link</code>
Estradas	<code>tertiary, tertiary_link, junction, rondabout, living_street, minor, path, pedestrian, planned, primary, primary_link, residential, road, secondary, secondary_link</code>
Ruas ou diversos	<code>bridleway, construction, cycleway, footway, ford, proposed, raceway, rest_area, service, staircase, steps, track, unclassified, unclassified; secondary, unsurfaced</code>

Tabela 4.12: Identificações usadas pelo OpenStreetMaps

O excerto de código SLD da Tabela 4.13 indica como é realizada essa primeira filtragem. Neste só é indicado que a regra definida só se aplica às autoestradas através dos elementos XML: `Filter` e `PropertyIsEqualTo`. O passo seguinte é definir o nível de escala para qual a regra se aplica. São usados os elementos `MaxScaleDenominator` e `MinScaleDenominator`.

Implementação

```
<sld:Rule>
  <ogc:Filter>
    <ogc:Or>
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>highway</ogc:PropertyName>
        <ogc:Literal>motorway</ogc:Literal>
      </ogc:PropertyIsEqualTo>
    </ogc:Or>
  </ogc:Filter>
  <sld:MaxScaleDenominator>1000000
</sld:MaxScaleDenominator>
  <sld:MinScaleDenominator>100000
</sld:MinScaleDenominator>
  <sld:LineSymbolizer>
    <sld:Stroke>
      <sld:CssParameter name="stroke">
        #000000</sld:CssParameter>
      <sld:CssParameter name="stroke-
        width">3
      </sld:CssParameter>
      <sld:CssParameter name="stroke-
        linejoin">round</sld:CssParameter>
    </sld:Stroke>
  </sld:LineSymbolizer>
  <sld:TextSymbolizer>
    <sld:Label>
      <ogc:PropertyName>ref
      </ogc:PropertyName>
    </sld:Label>
    <sld:Font>
      <CssParameter name=
        "font-family">Arial</CssParameter>
    </sld:Font>
    <sld:VendorOption name="followLine">
      true
    </sld:VendorOption>
    <sld:LabelPlacement>
      <sld:LinePlacement />
    </sld:LabelPlacement>
  </sld:TextSymbolizer>
</sld:Rule>
```

Tabela 4.13: Definição SLD do elemento a desenhar

Por fim é indicado como desenhar a estrada. É usado o elemento `LineSymbolizer` para indicar que será representado por uma linha. Nessa linha é indicado o tamanho do traço de três pixies e a forma do inicio e fim desse segmento deverá ser redonda. Associado à estrada é colocado uma frase com o nome dessa rua. O nome vem associado à propriedade “ref” da base de dados. O elemento XML usado é o `TextSymbolizer` na qual indicamos como é realizado a colocação do nome da estrada no ecrã.

Implementação

O resultado visual pode ser consultado no Erro: Origem da referência não encontradaErro: Origem da referência não encontrada. Neste existem várias capturas visuais da janela que representam desde da área do Algarve até uma aproximação a ruas de Faro.

Um facto curioso só evidente após os primeiros testes, é as linhas de média tensão representadas no SCATEX+ como tracejado, estarem coincidentes com as estradas. Isto acontece na realidade porque as vias são usadas para passar esses cabos no subsolo. A Figura 15 demonstra esse resultado obtido para a cidade de Faro.

4.5 Resultados obtidos

Esta secção apresenta o resultado visual final da integração dos mapas como pano de fundo para a rede eléctrica do Scatex+.

4.5.1 CAOP

Na Figura 14 é mostrado o resultado da integração do mapa CAOP (fronteiras e freguesias) com a rede eléctrica. A área observada é a da Grande Lisboa e Península de Setúbal. Nesta imagem pode ser observado que os limites de Portugal Continental coincidem com as nativas ao visualizador. Esta é a indicação que o mapa gerado está na projecção certa.

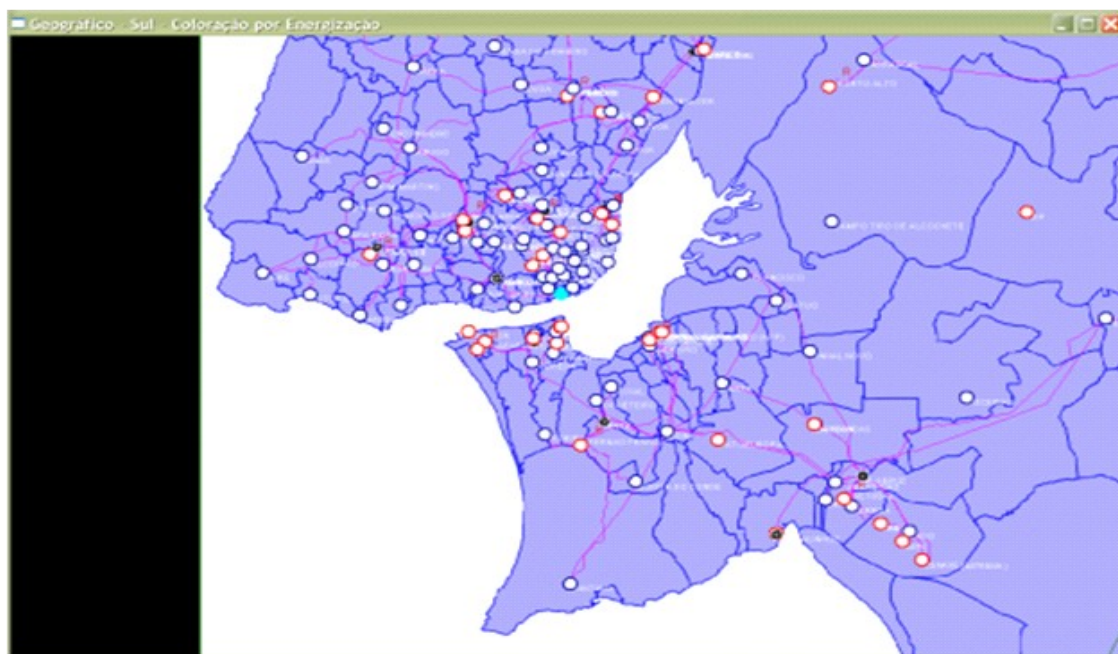


Figura 14: CAOP

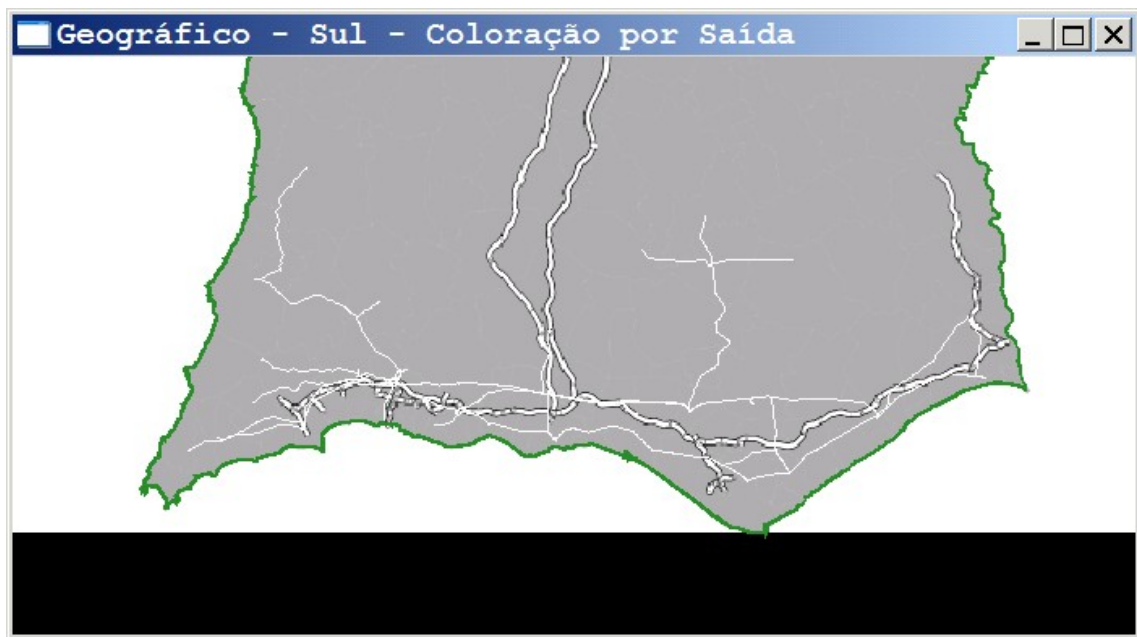
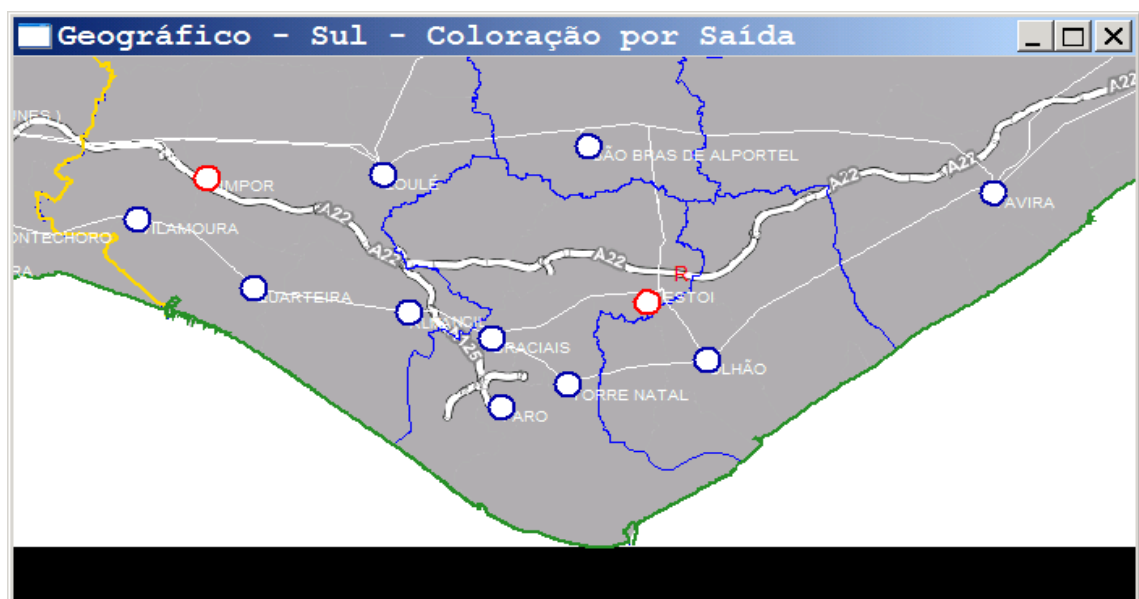


Figura 15: Autoestradas Algarve

As linhas a rosa, representam linhas de alta tensão e as circunferências representam subestações. O utilizador do sistema continua a operar nessa rede sem alterações.

4.5.2 Estradas

A sequência das quatro figuras seguintes, mostra a integração do mapa de estradas com a rede eléctrica.



Implementação

Cada imagem é uma aproximação sucessivamente crescente desde a área do Algarve até à cidade de Faro.

O objectivo deste mapa foi demonstrar as capacidades de estilização das estradas via SDL. Neste são seleccionados os elementos a desenhar em função do nível de *zoom* crescente. A partir de um determinado nível, já é possível colocar o desenho dos nomes das estradas no mapa. Imagens com linhas a tracejado e os triângulos representam respectivamente as linhas de baixa tensão e postos de transformação.

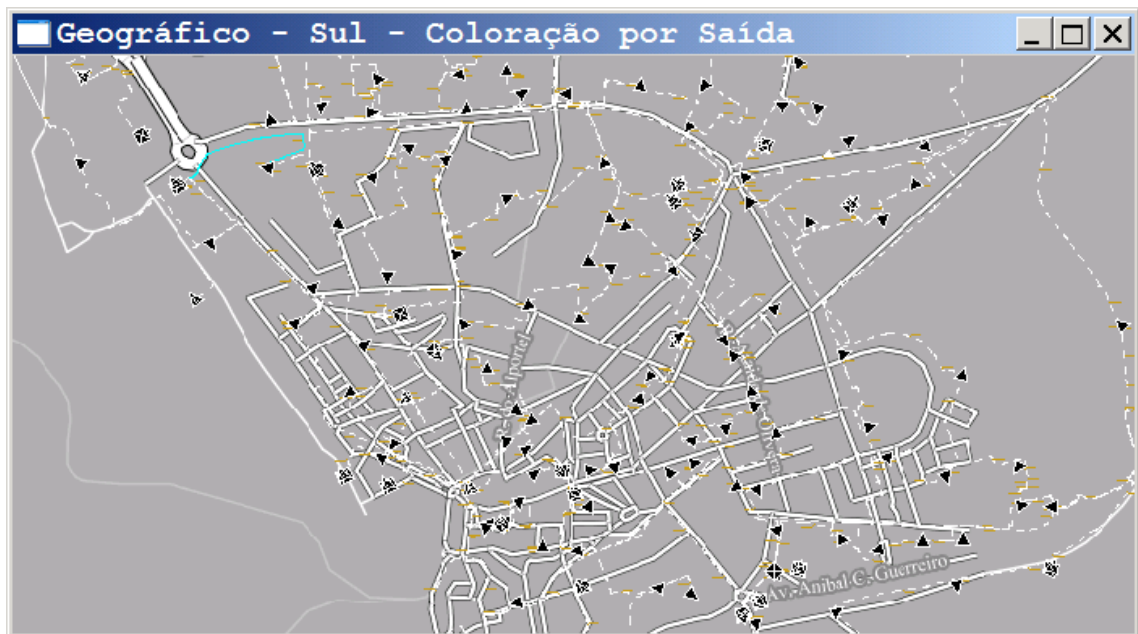


Figura 16: Ruas urbanas



Figura 17: Nomes de ruas

4.6 Resumo

Neste capítulo são referidos pormenores de implementação. As três grandes tarefas realizadas foram a importação de dados, a visualização e a estilização de mapas.

Foram importados com sucesso três tipos de mapas diferentes (dois do tipo vectorial e outro do tipo raster) e com diferentes projecções de origem. Os mapas vectoriais foram importados para o Postgres enquanto a imagem foi tratada directamente pelo GeoServer.

A visualização consistiu em provar que o conceito de obter mapas de um servidor remoto funciona e dentro dos tempos aceitáveis. Foi testado também a capacidade de estilização de mapas via SLD.

No fim, o prototipo implementado demonstrou que a abordagem delineada funcionou como previsto.

5 Conclusões e Trabalho Futuro

Este capítulo apresenta as contribuições deste trabalho para o problema proposto assim como propostas de desenvolvimentos futuros.

5.1 Objectivo e resultados

O objectivo desta dissertação era encontrar um processo de integrar informação geográfica diversa num visualizar da rede eléctrica de um sistema SCADA/DMS. Basicamente pretendia-se visualizar mapas diversos como pano de fundo da rede eléctrica. Foi realizado um estudo das várias vertentes que compõem um sistema SIG em geral.

Foi identificado que elementos geográficos capturados podem ser do tipo vectorial ou raster. Cada um destes é representado por dezenas de formatos ou suportes de informação. Esta informação precisa de persistir num sistema e ser facilmente manipulado e interrogada. A solução para isto consistiu em usar uma base de dados espacial baseada em Postgres/Postgis. Todos os conteúdos analisados são possíveis de serem inseridos nesse sistema. Este por sua vez disponibiliza a interface SQL como standard conhecido para manipulação e interrogação de dados.

A outra vertente identificada era como desenhar os mapas e os colocar no visualizador. Optou-se por introduzir um componente externo, denominado por Geoserver. A responsabilidade deste é interrogar a base de dados e gerar imagens a partir dos elementos guardados. A especificação de como desenhar esses elementos é definida através da linguagem SLD. A adaptação do visualizador consistiu pedir ao Geoserver para gerar uma imagem em tempo-real, para a área visível no ecrã. Esta tarefa é possível com o recurso à biblioteca GDAL e ao protocolo WMS de acesso ao Geoserver.

Por fim foi implementada uma optimização em complemento com a solução anterior. Em vez do Geoserver, responder a cada pedido com o redesenho da imagem, foi introduzido o componente TileCache para guarda imagens pré-geradas. Desta forma o tempo de obtenção de uma imagem foi minimizado de 10 segundos para 3 segundos.

O protótipo foi testado e demonstrado com sucesso tendo sido importados vários tipos de dados cuja visualização foi correctamente integrada.

Toda as partes da solução não apresentam custos de licenciamento permitindo flexibilidade na sua comercialização e na definição de margens. O único custo identificado são os custos dos conteúdos de interesse para o cliente final.

5.2 Trabalho Futuro

O problema foi identificado e os objectivos foram cumpridos. O desempenho cumpre os requisitos mas existe ainda espaço para desenvolvimentos e experiências futuras.

Na vertente do desempenho ainda se pode fazer os seguintes desenvolvimentos: pedir os mapas assincronamente e fazer cache de imagens no cliente.

A vantagem do pedidos assíncronos é o desenho da rede ser o mais imediato possível. É esse desenho que é crítico para o operador e não o mapa. Por isso o compasso de espera existente enquanto é criado e copiado o mapa do servidor pode desaparecer.

Quanto à cache de imagens o objectivo é não repetir os pedidos de blocos de imagens quando o cliente já sabe à partida que os tem em memória. Desta forma espera-se tornar a operação arrasto da rede mais fluída.

Por fim, fica por fazer a experiência de importar os dados da rede eléctrica para a arquitectura descrita neste trabalho. Isto abriria duas possibilidades futuras. Numa a rede podia ser visualizada por mais aplicações (*browsers, google earth*), para além da actual. Outra possibilidade seria o potencial de estudos da integração da rede com o ambiente usando as ferramentas de análise espacial facultadas pela arquitectura.

Referências

- [ERSI] ERSI. Utility GIS—More Than Just AM/FM, www.esri.com/library/whitepapers/pdfs/utility-gis.pdf, Consulta em 2010/Dez
- [GIS] What is GIS?, <http://www.gis.com/content/what-gis>, Consulta em 2010/Dez
- [Jiaju04] Jiaju Qiu and Xiaomeng Bian, Xiaojun Li. Intelligent Design of Power Transmission Lines Based on 3D GIS. In *Proceedings of the 5th World Congress on Intelligent Control and Automation*, 2004, pp. 5125-5128 Vol.6
- [Naganobu03] Naganobu, M., S. Nishiwaki, H. Yasuma, R. Matsukura, Y. Takao, K. Taki, T. Hayashi, Y. Watanabe, T. Yabuki, Y. Yoda, Y. Noiri, M. Kuga, K. Yoshikawa, N. Kokubun, H. Murase, K. Matsuoka, and K. Ito. Interactions between oceanography, krill and baleen whales in the Ross Sea and Adjacent Waters. In 2003.
- [PTGRID] Cliff Mugnier. The Portuguese Republic.. In *PHOTOGRAMMETRIC ENGINEERING & REMOTE SENSING*, April 2002.
- [Rappenglück99] Michael A. Rappenglück. Palaeolithic Timekeepers Looking At The Golden Gate Of The Ecliptic; The Lunar Cycle And The Pleiades In The Cave Of Lascaux. In , pp. 391-404. , 1999.
- [SLD] Open GIS Consortium Inc. Styled Layer Descriptor Implementation Specification. , 2002, <http://www.opengeospatial.org/standards/sld>, Consultado em 2010/Dez.
- [TMS] OSGeo. Tile Map Service Specification, 1.0, http://wiki.osgeo.org/wiki/Tile_Map_Service_Specification, Consulta em 2011/Dez
- [Wendou04] Wendou AN, Caixin SUN, Quan ZHOU, Ji YANG. GIs Based Multi-Level Intelligent Fault Diagnosis System on Electric Power Equipment, 2004, Pp 398 – 401, Electrical Insulation, 2004. *Conference Record of the 2004 IEEE International Symposium*

Anexo A: Draw e GDAL

Nesta secção é descrito as adaptações necessárias ao visualizador DRAW para usar a biblioteca GDAL.

A.1 Configuração XML

O seguinte texto foi usado para indicar à biblioteca GDAL, os parâmetros necessários para obter um mapa do servidor. Está no formato XML.

```
<GDAL_WMS>
  <Service name="TMS">
    <ServerUrl>http://localhost:8888/</ServerUrl>
    <Layer>pt_caopa</Layer>
    <Format>png</Format>
  </Service>
  <DataWindow>
    <UpperLeftX>-119191.408000</UpperLeftX>
    <UpperLeftY>276083.767000</UpperLeftY>
    <LowerRightX>169052.877500</LowerRightX>
    <LowerRightY>-300404.804000</LowerRightY>
    <TileLevel>20</TileLevel>
    <TileCountX>1</TileCountX>
    <TileCountY>2</TileCountY>
  </DataWindow>
  <Projection>EPSG:3763</Projection>
  <BlockSizeX>256</BlockSizeX>
  <BlockSizeY>256</BlockSizeY>
  <BandsCount>4</BandsCount>
</GDAL_WMS>
```

Os únicos parâmetros alterados dinamicamente são os relacionados com a definição da janela de visualização (UpperLeftX, UpperLeftY, LowerRightX, LowerRightY).

A.1 gdal_utils.hxx

```
#if !defined(DRW_GDAL_UTILS_HXX)
#define DRW_GDAL_UTILS_HXX

#include "gdal_priv.h"

class IlvDisplay;

CPLerr
GDALGetDataset(
    unsigned char * imageData,
    IlvDisplay* pDisplay,
    const char *pszFilename,
    const char *pszDest,
    double dfULX,
    double dfULY,
    double dfLRX,
    double dfLRY,
    int nOXSize,
    int nOYSize );

#endif
```

A.2 gdal_utils.cxx

```
#define CPL_DLL __declspec(dllimport)

#include "gdal_utils.h"
#include "gdal_priv.h"
#include "cpl_vsi.h"
#include "cpl_conv.h"
#include "cpl_string.h"
#include "ogr_spatialref.h"
#include "vrtdataset.h"
#include <bus/bus.hxx>
#include "view.hxx"
//#include "style_manager.hxx"

//CPL_CVSID("$Id: gdal_translate.cpp 18576 2010-01-17 22:32:24Z rouault $");

static void AttachMetadata( GDALDatasetH, char ** );
static int bSubCall = FALSE;

/*****
/*
/* ProxyMain()
*/
*****/
```

Draw e GDAL

```
CPLErr
GDALGetDataset(
    unsigned char * imageData,
    IlvDisplay* pDisplay,
    const char *pszFilename,
    const char *pszDest,
    double dfULX,
    double dfULY,
    double dfLRX,
    double dfLRY,
    int nOXSiz,
    int nOYSiz )
{

    const char *pszSource = pszFilename;
    //const char *pszDest = "void.mem";
    const char *pszFormat = "PNG";

    GDALDatasetH hDataset;
    int i;
    int nRasterXSiz, nRasterYSiz;
    GDALDriverH hDriver;
    int *panBandList = NULL, nBandCount = 0, bDefBands = TRUE;
    double adfGeoTransform[6];
    GDALDataType eOutputType = GDT_Unknown;
    char *pszOXSiz=NULL, *pszOYSiz=NULL;
    char **papszCreateOptions = NULL;
    int anSrcWin[4], bStrict = FALSE;
    const char *pszProjection;
    int bScale = FALSE, bHaveScaleSrc = FALSE, bUnscale=FALSE;
    double dfScaleSrcMin=0.0, dfScaleSrcMax=255.0;
    double dfScaleDstMin=0.0, dfScaleDstMax=255.0;
    char **papszMetadataOptions = NULL;
    char *pszOutputSRS = NULL;
    int bQuiet = TRUE, bGotBounds = FALSE;
    GDALProgressFunc pfnProgress = GDALTermProgress;
    int nGCPCount = 0;
    GDAL_GCP *pasGCPs = NULL;
    int iSrcFileArg = -1, iDstFileArg = -1;
    int bCopySubDatasets = FALSE;
    double adfULLR[4] = { 0,0,0,0 };
    int bSetNoData = FALSE;
    double dfNoDataReal = 0.0;
    int nRGBExpand = 0;

    anSrcWin[0] = 0;
    anSrcWin[1] = 0;
    anSrcWin[2] = 0;
    anSrcWin[3] = 0;

    CPLSetConfigOption( "CPL_DEBUG", "ON" );
    CPLSetConfigOption( "CPL_TIMESTAMP", "ON" );
    CPLSetConfigOption( "GDAL_CACHEMAX", "64" );

    dfULX = dfULX / 1000.0;
    dfULY = dfULY / 1000.0;
    dfLRX = dfLRX / 1000.0;
```

Draw e GDAL

```

dfLRY = dfLRY / 1000.0;

/* ----- */
/*      Attempt to open source file.                  */
/* ----- */
hDataset = GDALOpenShared( pszSource, GA_ReadOnly );

if( hDataset == NULL )
{
    fprintf( stderr,
             "GDALOpen failed - %d\n%s\n",
             CPLGetLastErrorNo(), CPLGetLastErrorMsg() );
    //GDALDestroyDriverManager();
    return CE_Failure;
}

/* ----- */
/*      Collect some information from the source file.  */
/* ----- */
nRasterXSize = GDALGetRasterXSize( hDataset );
nRasterYSize = GDALGetRasterYSize( hDataset );

if( !bQuiet )
    printf( "Input file size is %d, %d\n", nRasterXSize, nRasterYSize );

if( anSrcWin[2] == 0 && anSrcWin[3] == 0 )
{
    anSrcWin[2] = nRasterXSize;
    anSrcWin[3] = nRasterYSize;
}

/* ----- */
/*      Build band list to translate                    */
/* ----- */
/* ----- */
if( nBandCount == 0 )
{
    nBandCount = GDALGetRasterCount( hDataset );
    if( nBandCount == 0 )
    {
        fprintf( stderr, "Input file has no bands, and so cannot be
translated.\n" );
        //GDALDestroyDriverManager();
        return CE_Failure;
    }

    panBandList = (int *) CPLMalloc(sizeof(int)*nBandCount);
    for( i = 0; i < nBandCount; i++ )
        panBandList[i] = i+1;
}
else {
    //GDALDestroyDriverManager();
    return CE_Failure;
}

/* ----- */
/*      Compute the source window from the projected source window */
/*      if the projected coordinates were provided. Note that the */

```

Draw e GDAL

```

/*      projected coordinates are in ulx, uly, lrx, lry format,      */
/*      while the anSrcWin is xoff, yoff, xsize, ysize with the      */
/*      xoff,yoff being the ulx, uly in pixel/line.                  */
/* ----- */
if( dfULX != 0.0 || dfULY != 0.0 || dfLRX != 0.0 || dfLRY != 0.0 ) {
    double    adfGeoTransform[6];

    GDALGetGeoTransform( hDataset, adfGeoTransform );

    if( adfGeoTransform[2] != 0.0 || adfGeoTransform[4] != 0.0 )
    {
        fprintf( stderr,
            "The -projwin option was used, but the geotransform is\n"
            "rotated. This configuration is not supported.\n" );
        GDALClose( hDataset );
        CPLFree( panBandList );
        //GDALDestroyDriverManager();
        return CE_Failure;
    }

    anSrcWin[0] = (int)
        ((dfULX - adfGeoTransform[0]) / adfGeoTransform[1] + 0.001);
    anSrcWin[1] = (int)
        ((dfULY - adfGeoTransform[3]) / adfGeoTransform[5] + 0.001);

    anSrcWin[2] = (int) ((dfLRX - dfULX) / adfGeoTransform[1] + 0.5);
    anSrcWin[3] = (int) ((dfLRY - dfULY) / adfGeoTransform[5] + 0.5);

    if( !bQuiet )
        fprintf( stdout,
            "Computed -srcwin %d %d %d %d from projected window.\n",
            anSrcWin[0],
            anSrcWin[1],
            anSrcWin[2],
            anSrcWin[3] );

    if( anSrcWin[0] < 0 || anSrcWin[1] < 0
        || anSrcWin[0] + anSrcWin[2] > GDALGetRasterXSize(hDataset)
        || anSrcWin[1] + anSrcWin[3] > GDALGetRasterYSize(hDataset) )
    {
        fprintf( stderr,
            "Computed -srcwin falls outside raster size of %dx%d.\n",
            GDALGetRasterXSize(hDataset),
            GDALGetRasterYSize(hDataset) );
        return CE_Failure;
    }
}

/* ----- */
/*      Verify source window.                                         */
/* ----- */
if( anSrcWin[0] < 0 || anSrcWin[1] < 0
    || anSrcWin[2] <= 0 || anSrcWin[3] <= 0
    || anSrcWin[0] + anSrcWin[2] > GDALGetRasterXSize(hDataset)
    || anSrcWin[1] + anSrcWin[3] > GDALGetRasterYSize(hDataset) )
{
    fprintf( stderr,
        "-srcwin %d %d %d %d falls outside raster size of %dx%d\n"
        "or is otherwise illegal.\n",

```

Draw e GDAL

```

        anSrcWin[0],
        anSrcWin[1],
        anSrcWin[2],
        anSrcWin[3],
        GDALGetRasterXSize(hDataset),
        GDALGetRasterYSize(hDataset) );
    return CE_Failure;
}

/* ----- */
/*      Find the output driver.                      */
/* ----- */
hDriver = GDALGetDriverByName( pszFormat );

/* ===== */
/*      Create a virtual dataset.                      */
/* ===== */
VRTDataset *poVDS;

/* ----- */
/*      Make a virtual clone.                          */
/* ----- */
poVDS = (VRTDataset *) VRTCreate( nOXSize, nOYSize );

...

if (poVDS == NULL) {
    LOG (STATUS << "[GDAL] ERROR: VirtualDataSet is null. Return" << endl);
    GDALClose( hDataset );
    CPLFree( panBandList );
    CPLFree( pszOutputSRS );
    if( !bSubCall ) GDALDumpOpenDatasets( stderr );
    CSLDestroy( papszCreateOptions );
}

GDALRasterBand* bandRed = poVDS->GetRasterBand(1);
GDALRasterBand* bandGreen = poVDS->GetRasterBand(2);
GDALRasterBand* bandBlue = poVDS->GetRasterBand(3);
GDALRasterBand* bandAlpha = poVDS->GetRasterBand(4);

for(int b=1; b <= 4; ++b) {

    GDALRasterBand* poBandInit = poVDS->GetRasterBand(b);

    int nBandBlockXSize, nBandBlockYSize;
    poBandInit->GetBlockSize( &nBandBlockXSize, &nBandBlockYSize );

    LOG (STATUS << "[GDAL] Block=" << nBandBlockXSize << "x" <<
nBandBlockYSize <<
                                " Type=" << GDALGetDataTypeName(poBandInit-
>GetRasterDataType()) <<
                                " ColorInterp=" << poBandInit->GetColorInterpretation() <<
endl);
}

int bandSize = 4;
int length = nOXSize * nOYSize * bandSize;

CPLErr eErr = CE_None;

```

Draw e GDAL

```
LOG (STATUS << "[GDAL] Start read band: " << endl;);
eErr = bandRed-
>RasterIO(GF_Read,0,0,nOXSize,nOYSize,imageData+0,nOXSize,nOYSize,GDT_Byte,4,nOXSize *
4);
eErr = bandGreen-
>RasterIO(GF_Read,0,0,nOXSize,nOYSize,imageData+1,nOXSize,nOYSize,GDT_Byte,4,nOXSize *
4);
eErr = bandBlue-
>RasterIO(GF_Read,0,0,nOXSize,nOYSize,imageData+2,nOXSize,nOYSize,GDT_Byte,4,nOXSize *
4);
eErr = bandAlpha-
>RasterIO(GF_Read,0,0,nOXSize,nOYSize,imageData+3,nOXSize,nOYSize,GDT_Byte,4,nOXSize *
4);
LOG (STATUS << "[GDAL] End read band: " << eErr << endl;);

if (eErr != CE_None) LOG (STATUS << "[GDAL] Error: " << CPLGetLastErrorMsg() <<
"." << endl;);

// CPLFree( imageData );

if ( poVDS != NULL ) GDALClose( (GDALDatasetH) poVDS );
if ( hDataset != NULL ) GDALClose( (GDALDatasetH) hDataset );
CPLFree( panBandList );
CPLFree( pszOutputSRS );
if( !bSubCall ) GDALDumpOpenDatasets( stderr );
CSLDestroy( papszCreateOptions );

return CE_None;
}
```

A.1 view.hxx

```
...
DRWDrawnObject* pLastDRWRasterMap;           // EPR 2010.04.28 - Maps
static int s_nextMemFile;                     // EPR 2010.04.23 - Counter used as
a unique                                     // name for a mem file
...
// epr 2010.04.14 - start
void requestMap (const DRWViewRect& area);
DRWDrawnObject* createGeoRaster (const DRWViewRect& area);
void build_xmlfile_for_gdal (EFAString& sXmlContents, const DRWViewRect& area);
// epr 2010.04.14 - end
...
```

A.1 view.cxx

```
...
#define CPL_DLL __declspec(dllimport)         // EPR - Can't use GDAL lib without
this                                         defined
...
#include "gdal_utils.h"
...
```

Draw e GDAL

```
void DRWView::build_xmlfile_for_gdal (EFAString& sXmlContents, const DRWViewRect&
area )
{

    IlvDim w = width();
    IlvDim h = height();

    IlvFloat leftX = area.left()/1000;
    IlvFloat upperY = area.top()/1000;
    IlvFloat lowerY = area.bottom()/1000;
    IlvFloat rightX = area.right()/1000;

    double scale = getTransformer()->getScaleFactor();
    if (scale > 0.002) {
    }

    // get this at runtime
    EFAString URL = "http://localhost:8888/";
    EFAString LAYERS = "caop_and_roads";

    std::stringstream ssW;
    std::stringstream ssH;
    ssW << w;
    ssH << h;

    EFAString SizeX(ssW.str());
    EFAString SizeY(ssH.str());

    std::stringstream ssLX;
    std::stringstream ssRX;
    std::stringstream ssUY;
    std::stringstream ssLY;

    ssUY << upperY;
    ssLX << leftX;
    ssLY << lowerY;
    ssRX << rightX;

    EFAString sLeftX (ssLX.str());
    EFAString sLowerY(ssLY.str());
    EFAString sRightX (ssRX.str());
    EFAString sUpperY (ssUY.str());

    // START TMS
    sXmlContents = "<GDAL_WMS>";
    sXmlContents += "<Service name=\"TMS\">";
        //sXmlContents += "<Version>1.0.0</Version>";
        sXmlContents.append("<ServerUrl>");
            sXmlContents.append(URL);
        sXmlContents.append("</ServerUrl>");
        //sXmlContents.append("<SRS>EPSG:3763</SRS>");
        sXmlContents.append("<ImageFormat>image/png</ImageFormat>");
        //sXmlContents.append("<Transparent>TRUE</Transparent>");
        sXmlContents.append("<Layer>");
            sXmlContents.append(LAYERS);
        sXmlContents.append("</Layer>");
        sXmlContents.append("<Styles></Styles>");
```

Draw e GDAL

```
sXmlContents.append("</Service>");
sXmlContents.append("<DataWindow>");
    sXmlContents.append("<UpperLeftX>-119191.408000</UpperLeftX>");
    sXmlContents.append("<UpperLeftY>276083.767000</UpperLeftY>");
    sXmlContents.append("<LowerRightX>169052.877500</LowerRightX>");
    sXmlContents.append("<LowerRightY>-300404.804000</LowerRightY>");
    sXmlContents.append("<TileLevel>20</TileLevel>");
    sXmlContents.append("<TileCountX>1</TileCountX>");
    sXmlContents.append("<TileCountY>2</TileCountY>");
sXmlContents.append("</DataWindow>");
sXmlContents.append("<Projection>EPSG:3763</Projection>");
sXmlContents.append("<BandsCount>4</BandsCount>");
sXmlContents.append("<BlockSizeY>256</BlockSizeY>");
sXmlContents.append("<BlockSizeX>256</BlockSizeX>");
sXmlContents.append("</GDAL_WMS>");
// END TMS
}

DRWDrawnObject* DRWView::createGeoRaster (const DRWViewRect& area)
{
    LOG (STATUS << "Start request map ..." << endl);

    DRWDrawnObject* pObj = 0;

    IlvDim w = width();
    IlvDim h = height();

    IlFloat leftX = area.left();
    IlFloat upperY = area.top();
    IlFloat lowerY = area.bottom();
    IlFloat rightX = area.right();

    LOG (STATUS << "View area geo coordinates "
        << "    leftX: " << leftX
        << "    upperY: " << upperY
        << "    lowerY: " << lowerY
        << "    rightX: " << rightX
        << endl);

    IlDouble ptbbox_leftX = -119191408.0;
    IlDouble ptbbox_rightX = 162129081.0;
    IlDouble ptbbox_lowerY = -300404804.0;
    IlDouble ptbbox_upperY = 276083767.0;

    IlDouble dfULX, dfULY, dfLRX, dfLRY;
    int noXSize, noYSize;
    bool isToAdjustPixels = false;

    if (leftX < ptbbox_leftX) {
        dfULX = ptbbox_leftX;
        isToAdjustPixels = true;
    }
    else {
        dfULX = leftX;
    }

    if (rightX > ptbbox_rightX) {
        dfLRX = ptbbox_rightX;
        isToAdjustPixels = true;
    }
```

Draw e GDAL

```
}
else {
    dfLRX = rightX;
}

if (lowerY < ptbbox_lowerY) {
    dfLRY = ptbbox_lowerY;
    isToAdjustPixels = true;
}
else {
    dfLRY = lowerY;
}

if (upperY > ptbbox_upperY) {
    dfULY = ptbbox_upperY;
    isToAdjustPixels = true;
}
else {
    dfULY = upperY;
}

if (isToAdjustPixels == true) {

    // convert bbox coord do pixels
    IlvFloatPoint pointUL (dfULX, dfULY);
    getTransformer()->apply(pointUL);

    IlvFloatPoint pointLR (dfLRX, dfLRY);
    getTransformer()->apply(pointLR);

    // get pixels (pxl) width from absolute pixels position
    IlvFloat fwidthPxl = abs(pointLR.getX() - pointUL.getX());
    IlvFloat fheightPxl = abs(pointUL.getY() - pointLR.getY());

    w = (int)ceil(fwidthPxl);
    h = (int)ceil(fheightPxl);
}

nOXSize = w;
nOYSize = h;

LOG (STATUS << "View area in pixels: "
    << "  nOXSize/w: " << nOXSize
    << "  nOYSize/h: " << nOYSize
    << endl);

LOG (STATUS << "View area adjusted to bbox "
    << "  leftX/dfULX: " << dfULX
    << "  upperY/dfULY: " << dfULY
    << "  lowerY/dfLRY: " << dfLRY
    << "  rightX/dfLRX: " << dfLRX
    << endl);

/* ----- */
/*      Build xml request      */
/* ----- */

EFAString xmlGdalParams;
build_xmlfile_for_gdal (xmlGdalParams, area);
```

Draw e GDAL

```

/* ----- */
/*      Open xml request */
/* ----- */

char vfileName[64];
sprintf_s(vfileName, 64, "/vsimem/%u", s_nextMemFile++);

FILE          *fp          =          VSIFFileFromMemBuffer(vfileName,
(GByte*)xmlGdalParams.asStdString().c_str(), xmlGdalParams.length(), FALSE);

    if (fp == NULL) {
        LOG (STATUS << "[GDAL] Destroy GDAL. Reason: \"File VSIFFileFromMemBuffer
== NULL\"<< endl);
        //GDALDestroyDriverManager();
        return NULL;
    }
    VSIFCloseL (fp);

    CPLErr  eErr = CE_None;
    char * pszDest = "d:\\out.png";
    IlvDisplay* pDisplay = DRWStyleManagerManager::instance()->getDisplay ();

    int bandSize = 4;
    int length = nOXSz * nOYSz * bandSize;
    unsigned char * dataImage = (GByte *) CPLMalloc( length );

    eErr = GDALGetDataset(dataImage, pDisplay, vfileName, pszDest, dfULX, dfULY,
dfLRX, dfLRY, nOXSz, nOYSz);
    if( eErr == CE_None ) {
        LOG (STATUS << "[GDAL] Ilog bitmap" << endl);
        IlvDisplay* pDisplay = DRWStyleManagerManager::instance()->getDisplay ();

        IlvBitmap * pIlvBitmap = new IlvBitmap(
            pDisplay,
            nOXSz,
            nOYSz,
            4,
            dataImage
        );

        IlvDim dx = (IlvDim)abs(dfLRX - dfULX);
        IlvDim dy = (IlvDim)abs(dfULY - dfLRY);

        IlvRect ilvRect ((IlvPos) dfULX,
                        (IlvPos) dfULY,
                        dx,
                        dy);

        IlvPalette* pPalette = 0;
        pObj = new DRWRaster (pIlvBitmap, ilvRect, pPalette);
    }

    VSIUnlink(vfileName);
    CPLFree(dataImage);

    LOG (STATUS << "End fetch maps" << endl);
    return pObj;
}

```

Anexo C: OpenStreetMaps SLD

O texto XML desta secção define os estilos gráficos de desenho dos mapas de estradas OpenStreetMaps. Está descrito no formato XML, segundo a especificação Style Language Descriptor (SLD).

Com o objectivo de diminuir o número de páginas em anexo desta dissertação, está só representado a definição para o desenho de:

- auto-estradas
- dois níveis de escala de mapa
- escrita dos nomes das estradas

```
<?xml version="1.0" encoding="utf-8"?>
<sld:StyledLayerDescriptor xmlns="http://www.opengis.net/sld"
xmlns:sld="http://www.opengis.net/sld"
xmlns:ogc="http://www.opengis.net/ogc"
xmlns:gml="http://www.opengis.net/gml" version="1.0.0">
  <sld:UserLayer>
    <sld:LayerFeatureConstraints>
      <sld:FeatureTypeConstraint />
    </sld:LayerFeatureConstraints>
    <sld:UserStyle>
      <sld:Name>Default Styler</sld:Name>
      <sld:Title />
      <sld:IsDefault>1</sld:IsDefault>
      <sld:FeatureTypeStyle>
        <sld:Name>name</sld:Name>
        <sld:FeatureTypeName>Feature</sld:FeatureTypeName>
        <sld:SemanticTypeIdentifier>
          generic:geometry</sld:SemanticTypeIdentifier>
        <sld:SemanticTypeIdentifier>
```

Anexo C: OpenStreetMaps SLD

```
colorbrewer:custom:ylorrd</sld:SemanticTypeIdentifier>
<sld:Rule>
  <sld:Name>rule motorway trunk min 1000000</sld:Name>
  <sld:Title>&lt;diferente&gt;; motorway, motorway_link,
trunk, trunk_link, min 1000000</sld:Title>
  <ogc:Filter>
    <ogc:Or>
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>highway</ogc:PropertyName>
        <ogc:Literal>&lt;diferente&gt;</ogc:Literal>
      </ogc:PropertyIsEqualTo>
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>highway</ogc:PropertyName>
        <ogc:Literal>motorway</ogc:Literal>
      </ogc:PropertyIsEqualTo>
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>highway</ogc:PropertyName>
        <ogc:Literal>motorway_link</ogc:Literal>
      </ogc:PropertyIsEqualTo>
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>highway</ogc:PropertyName>
        <ogc:Literal>trunk</ogc:Literal>
      </ogc:PropertyIsEqualTo>
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>highway</ogc:PropertyName>
        <ogc:Literal>trunk_link</ogc:Literal>
      </ogc:PropertyIsEqualTo>
    </ogc:Or>
  </ogc:Filter>
  <sld:MinScaleDenominator>
1000000</sld:MinScaleDenominator>
  <sld:LineSymbolizer>
    <sld:Stroke>
      <sld:CssParameter name="stroke">
#000000</sld:CssParameter>
      <sld:CssParameter name="stroke-width">
3</sld:CssParameter>
      <sld:CssParameter name="stroke-linejoin">
round</sld:CssParameter>
    </sld:Stroke>
  </sld:LineSymbolizer>
  <sld:LineSymbolizer>
    <sld:Stroke>
      <sld:CssParameter name="stroke">
#FFFFFF</sld:CssParameter>
      <sld:CssParameter name="stroke-width">
2</sld:CssParameter>
      <sld:CssParameter name="stroke-linejoin">
round</sld:CssParameter>
    </sld:Stroke>
  </sld:LineSymbolizer>
```

Anexo C: OpenStreetMaps SLD

```
</sld:Rule>
<sld:Rule>
  <sld:Name>rule motorway max 1000000 min 100000</sld:Name>
  <sld:Title>&lt; diferente &gt;;, motorway, motorway_link max
1000000 min 100000</sld:Title>
  <ogc:Filter>
    <ogc:Or>
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>highway</ogc:PropertyName>
        <ogc:Literal>&lt; diferente &gt;;</ogc:Literal>
      </ogc:PropertyIsEqualTo>
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>highway</ogc:PropertyName>
        <ogc:Literal>motorway</ogc:Literal>
      </ogc:PropertyIsEqualTo>
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>highway</ogc:PropertyName>
        <ogc:Literal>motorway_link</ogc:Literal>
      </ogc:PropertyIsEqualTo>
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>highway</ogc:PropertyName>
        <ogc:Literal>trunk</ogc:Literal>
      </ogc:PropertyIsEqualTo>
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>highway</ogc:PropertyName>
        <ogc:Literal>trunk_link</ogc:Literal>
      </ogc:PropertyIsEqualTo>
    </ogc:Or>
  </ogc:Filter>
  <sld:MaxScaleDenominator>
1000000</sld:MaxScaleDenominator>
  <sld:MinScaleDenominator>100000</sld:MinScaleDenominator>
  <sld:LineSymbolizer>
    <sld:Stroke>
      <sld:CssParameter name="stroke">
#000000</sld:CssParameter>
      <sld:CssParameter name="stroke-width">
5</sld:CssParameter>
      <sld:CssParameter name="stroke-linejoin">
round</sld:CssParameter>
    </sld:Stroke>
  </sld:LineSymbolizer>
  <sld:LineSymbolizer>
    <sld:Stroke>
      <sld:CssParameter name="stroke">
#FFFFFF</sld:CssParameter>
      <sld:CssParameter name="stroke-width">
4</sld:CssParameter>
      <sld:CssParameter name="stroke-linejoin">
round</sld:CssParameter>
    </sld:Stroke>
  </sld:LineSymbolizer>
```

Anexo C: OpenStreetMaps SLD

```
</sld:LineSymbolizer>
<sld:TextSymbolizer>
  <sld:Label>
    <ogc:PropertyName>ref</ogc:PropertyName>
  </sld:Label>
  <sld:Fill>
    <CssParameter name="fill">#FFFFFF</CssParameter>
  </sld:Fill>
  <sld:Halo>
    <sld:Radius>2</sld:Radius>
    <sld:Fill>000000</sld:Fill>
  </sld:Halo>
  <sld:Font>
    <CssParameter name="font-family">Arial</CssParameter>
    <CssParameter name="font-weight">bold</CssParameter>
    <CssParameter name="font-size">12</CssParameter>
  </sld:Font>
  <sld:VendorOption name="followLine">
    true</sld:VendorOption>
  <sld:LabelPlacement>
    <sld:LinePlacement />
  </sld:LabelPlacement>
</sld:TextSymbolizer>
</sld:Rule>
```