



Ferramenta de simulação do consumo de energia na iluminação pública

Cláudio Henrique dos Santos Alegria

*Tese Submetida à Universidade da Madeira para a
Obtenção do Grau de Mestre em Engenharia Informática*

Funchal - Portugal

Novembro 2008

AGRADECIMENTOS

Ao orientador, o Professor Doutor Duarte Nuno Jardim Nunes, Professor Associado do Dep. de Matemática e Engenharias da Universidade da Madeira, por ajudar a tomar diversas decisões importantes, e guiar o processo de elaboração do projecto.

Ao Eng. Filipe Oliveira da Agência Regional da Energia e Ambiente da Região Autónoma da Madeira (AREAM), por ter-se mostrado sempre disponível a dar a sua opinião, e ter contribuído com um texto que foi adaptado neste relatório, relativo à motivação do projecto.

RESUMO

A iluminação pública é uma área importante de consumo energético. Actualmente existem uma variedade de soluções tecnológicas que permitem diminuir esse consumo e o correspondente impacte ambiental. No entanto, estas soluções tecnológicas nem sempre são utilizadas devido à falta de suporte informático que permita o planeamento para instalação de redes de iluminação, e/ou actualização das tecnologias utilizadas na rede de iluminação existente. Este relatório apresenta uma ferramenta de simulação do consumo de energia na iluminação pública, através da definição de cenários pelo utilizador, nos quais são simuladas a escolha de lâmpadas, luminárias e sistemas de controlo para cada ramal de electricidade, representados sobre um plano geográfico, que permita o cálculo de indicadores de apoio à decisão. A solução apresentada neste relatório, desenvolvida sobre o sistema de web mapping Google Maps, e sobre a plataforma de desenvolvimento para a web Ruby on Rails, permite o desenho sobre o mapa de uma rede de iluminação pública, e o cálculo em tempo de execução do custo e consumo de energia dos cenários de iluminação simulados pelo utilizador. Através de expansões futuras esta ferramenta poderá contribuir para a eficiente optimização de redes de iluminação pública.

Palavras-chave: iluminação pública, consumo de energia, web mapping, aplicação web

ABSTRACT

Public lighting is an important area of energy consumption. There are several technological solutions that allow minimizing this consumption, and the corresponding environmental impact. However, these technological solutions are not always used due to the lack of software support that enables the planning for installing public lighting networks, and/or updating the technologies used in the existing lighting network. This report presents a tool for the simulation of energy consumption in public lighting, through the definition of scenarios by the user, in which the choice of light bulbs, lights and control systems are simulated for each electricity branch, represented over a geographical plan, allowing the calculation of decision support indicators. The solution presented in this report, developed over the web mapping system Google Maps, and over the web application framework Ruby on Rails, allows the design of a public lighting network over a map, and the calculation in execution time of the cost and energy consumption amount of the lighting scenarios simulated by the user. Through future expansion this tool can contribute to the efficient optimization of public lighting networks.

Keywords: public lighting, energy consumption, web mapping, web application

ÍNDICE

Lista de figuras	8
Glossário	9
1. Introdução	10
1.1. Motivação	12
2. Estado da arte	15
2.1. Plataforma de informação geográfica.....	17
2.2. Tecnologias de desenvolvimento para a web.....	22
2.2.1. Arquitectura do Ruby on Rails	24
3. Análise de requisitos.....	27
3.1. Processo de análise de requisitos	30
3.1.1. Lista de requisitos funcionais e não-funcionais.....	30
3.1.2. Mapa de desempenho	31
3.1.3. Mapa de actores e tarefas.....	32
3.1.4. Mapa de actividades.....	32
3.1.5. Perfis das actividades.....	33
3.1.6. Mapa de actividades e tarefas.....	34
3.1.7. Descrição das tarefas	35
3.2. Desenho da interface.....	36
3.2.1. Modelos Wisdom.....	36
3.2.2. Protótipos Abstractos Canónicos.....	37
3.3. Base de dados.....	39

4. Processo de desenvolvimento e arquitectura da aplicação	42
4.1. Abordagem inicial sobre PHP	43
4.2. Abordagem inicial sobre Ruby on Rails.....	45
4.3. Modelos e controladores utilizados, e o seu funcionamento	47
4.3.1. Modelos.....	47
4.3.2. Controladores.....	47
5. Interacção com o utilizador.....	49
5.1. Secção “Tarifário”	52
5.2. Secção “Cenários”	53
5.3. Sub-secção “Rede actual”	55
5.3.1. Tabulador “Segmento”	56
5.3.2. Tabulador “Sistemas de controlo”	58
5.4. Sub-secção “Um cenário”	59
5.5. Secção “Registo de utilizadores”	60
5.6. Secção “Controlo de utilizadores”	61
6. Conclusões.....	63
Anexo I: Documento de requisitos da aplicação.....	65
Lista de requisitos funcionais	65
Lista de requisitos não-funcionais.....	68
Mapa de desempenho.....	70
Mapa de actores e tarefas	71
Mapa de actividades	72
Perfis das actividades.....	73
Mapa de actividades e tarefas.....	78
Descrição das tarefas.....	79
Diagramas Wisdom.....	86

Protótipos abstractos canónicos	92
Diagrama de entidade-relacionamento da base de dados.....	106
Anexo II: Ficheiros da aplicação.....	107
Diagramas de pastas e ficheiros	107
Modelos e controladores	110
Vistas	110
Outros ficheiros.....	112
Anexo III: Manual de instalação da aplicação.....	116
Anexo IV: Bibliografia	128

LISTA DE FIGURAS

Figura 1. Imagens de satélite do serviço de web mapping Google Maps (maps.google.com).....	17
Figura 2. Mapa baseado em informação vectorial do serviço de web mapping Sapo Mapas (mapas.sapo.pt).....	19
Figura 3. Arquitectura do Ruby on Rails	25
Figura 4. Mapa de desempenho da aplicação	31
Figura 5. Mapa de actores e tarefas da aplicação.....	32
Figura 6. Mapa de actividades da aplicação.....	33
Figura 7. Mapa de actividades e tarefas da aplicação	34
Figura 8. Modelo Wisdom de navegação.....	36
Figura 9. Protótipo abstracto canónico da secção “plano actual” (tab ramais)	37
Figura 10. Diagrama de entidade-relacionamento da aplicação	39
Figura 11. Interface experimental inicial para o redimensionamento de GroundOverlays e linhas poligonais.....	44
Figura 12. Página inicial da aplicação.....	49
Figura 13. Mapa de navegação simplificado (sem sub-espacos de interacção).....	50
Figura 14. Secção "Tarifário" da aplicação	52
Figura 15. Secção "Cenários" da aplicação	53
Figura 16. Sub-secção "Rede actual" da aplicação.....	55
Figura 17. Janela de pesquisa por lâmpadas	57
Figura 18. Sub-secção detalhando "um cenário" da aplicação	59
Figura 19. Secção de registo de novos utilizadores na aplicação	60
Figura 20. Secção de controlo de utilizadores por parte do administrador do sistema	61

GLOSSÁRIO

API – Abreviatura de Application Programming Interface. Consiste num conjunto de métodos, funções, procedimentos e pedidos estandardizados de uma aplicação particular, acessíveis a outras aplicações. [1]

Drag-and-drop – Acção de arrastar um objecto numa interface e largá-lo noutra local, usualmente através de um rato.

Framework – Um conjunto prefabricado de software e/ou uma arquitectura de software que pode ser reutilizada por programadores para situações específicas, e que leva os programadores a seguir uma determinada metodologia. [2] [3]

GIS – Sistema de informação geográfica, um sistema que integra software e eventualmente hardware com dados para capturar, gerir, analisar e representar todo o tipo de formas de informação geograficamente referenciada. [4]

Open Source – Software distribuído livremente, cujo código-fonte é também acessível ao público, em que a sua filosofia de desenvolvimento não se opõe a alterações personalizadas pelos utilizadores, desde que certas regras de copyright sejam respeitadas. [5]

Plugin – Módulo de software que pode ser acrescentado a uma aplicação, aumentando as suas funcionalidades. [6]

Popup – Janela com conteúdos que aparece sobreposta sobre os conteúdos usuais de uma interface, e que usualmente pode ser arrastada sobre a interface, e fechada apenas por iniciativa do utilizador. [7]

Web mapping – Processo de apresentar e gerar mapas com o objectivo de serem acedidos através de páginas web. Uma aplicação de web mapping oferece estes serviços a um programador web.

Widget – Um componente numa GUI (Interface gráfica do utilizador).

1. INTRODUÇÃO

Este relatório apresenta uma ferramenta de ajuda ao planeamento e re-estruturação da iluminação pública, através da simulação de cenários e da representação da rede de iluminação sobre um plano geográfico, desenvolvida no âmbito do projecto final do Mestrado em Engenharia Informática.

O projecto surgiu da necessidade da Agência Regional da Energia e Ambiente da Região Autónoma da Madeira (AREAM) em adquirir ou desenvolver uma ferramenta deste tipo, originalmente como parte integrante de um projecto de colaboração internacional a nível da União Europeia. É no seu interesse a optimização dos gastos investidos todos os anos na iluminação pública, assim como a diminuição do impacte ambiental associado à produção de energia. Pretende-se que a aplicação desenvolvida no âmbito deste projecto (ou futuras evoluções desta aplicação) contribua na obtenção destes objectivos – ao fornecer ao utilizador a possibilidade de simular cenários de instalação de novos equipamentos, de substituição de equipamentos actualmente utilizados, ou de alteração do comportamento dos equipamentos já instalados. Esta aplicação permite assim fazer um planeamento sobre uma área alargada como a de uma cidade.

Durante o período de elaboração do projecto, foram efectuadas diversas tarefas: um período inicial foi dedicado à pesquisa das soluções e tecnologias que poderiam ser consideradas no desenvolvimento da aplicação; assim como uma pesquisa sobre o estado da arte no campo dos sistemas geográficos e sistemas para análise de redes de electricidade. Também num período inicial, foi efectuada uma análise de requisitos para compreender em pormenor as necessidades dos utilizadores e as funcionalidades pretendidas para a aplicação. Seguiu-se uma fase em que foram efectuados testes às limitações e capacidades das tecnologias web consideradas. Finalmente, após tomar uma opção definitiva quanto às tecnologias a utilizar, partiu-se para a implementação da aplicação idealizada durante a fase de análise de requisitos.

Este relatório pretende apresentar todo o trabalho efectuado, detalhando as particularidades de todas estas fases referidas, justificando as opções tomadas,

apresentando os problemas encontrados ao longo do processo e as soluções encontradas para ultrapassá-los.

Outro objectivo deste relatório é apresentar o estado final da aplicação à data da entrega, focando-se particularmente na forma como a aplicação é estruturada internamente, na forma como a aplicação é apresentada aos utilizadores, e o comportamento que assume de cada vez que o utilizador interage com ela.

1.1 MOTIVAÇÃO

A utilização da energia de origem fóssil, como o petróleo, está referenciada como a principal responsável pelas emissões de Gases com Efeito de Estufa (GEE) decorrentes da actividade humana. O aumento da concentração destes gases na atmosfera é usualmente assumido como origem do aquecimento global do Planeta, o que resulta nas alterações climáticas, que poderá ter consequências prejudiciais nos ecossistemas naturais e para a Humanidade. [8]

Reconhecendo a gravidade deste problema, foram assumidos compromissos pelos signatários do Protocolo de Quioto que obrigam a União Europeia e o conjunto dos seus Estados-membros a reduzir, no período de 2008 a 2012, as suas emissões de origem humana de gases com efeito de estufa em 8%, em relação aos níveis de 1990.

De acordo com a distribuição estabelecida na União Europeia, Portugal fica obrigado a respeitar, em 2012, um nível máximo de emissões anuais de CO₂ 27% superior às emissões do ano de 1990, valor que já foi ultrapassado em 2000 e dificilmente será cumprido sem uma forte participação das fontes de energia renováveis isentas de emissões de gases com efeito de estufa. [9]

A eficiência energética, entre outras medidas, constitui uma peça basilar da estratégia nacional consagrada no Programa Nacional para as Alterações Climáticas (PNAC). Este programa tem por objectivo controlar e reduzir as emissões de gases com efeito de estufa, de modo a respeitar os compromissos de Portugal no âmbito do Protocolo de Quioto e da partilha de responsabilidades no seio da União Europeia, bem como noutros instrumentos estratégicos de política ambiental e energética.

Tendo em vista evitar ou minimizar o incumprimento das obrigações assumidas por Portugal no âmbito do Protocolo de Quioto, que acarreta pesadas multas, foram estabelecidas políticas sectoriais para promover a redução da procura de combustíveis fósseis, designadamente na produção de electricidade.

Neste sentido, através da Directiva 2003/87/CE, que estabelece o Comércio Europeu de Licenças de Emissão, os produtores de electricidade ficam sujeitos a licenças de emissão de dióxido de carbono, que constituem limites que não podem ser ultrapassados, sob pena de pesadas multas. Desta forma, o Comércio Europeu de

Licenças de Emissão constitui mais um instrumento de promoção da eficiência energética, uma vez que traduz as emissões de Gases com Efeito de Estufa evitadas em activos que podem ser transaccionados. Por outras palavras, o investimento na poupança de energia de origem fóssil, para além da redução de custos de aprovisionamento de energia, pode resultar em vantagens económicas com a transacção das licenças de emissão disponíveis.

Neste contexto, a Comissão Europeia adoptou um plano de acção no âmbito da energia cujo objectivo é reduzir 20% dos consumos de energia até 2020. O plano de acção inclui medidas que visam melhorar a eficiência energética global da União Europeia, através do desenvolvimento e aplicação de novas tecnologias de produção e utilização da energia. Na sequência deste plano, foi também recentemente lançado o Plano Nacional de Acção para a Eficiência Energética.

Na Região Autónoma da Madeira, a eficiência energética é também uma orientação do Plano Regional da Política de Ambiente (PRPA) e do Plano de Política Energética da Região Autónoma da Madeira (PPERAM), onde são reconhecidas e sublinhadas as importantes mais-valias ambientais, económicas e sociais resultantes da redução da utilização da energia, das importações e da dependência do exterior.

A iluminação pública constitui um sector onde há um grande potencial de melhoria da eficiência energética, não só através da instalação de equipamentos de iluminação eficientes, designadamente lâmpadas e luminárias, mas também através de soluções de controlo adequadas, que assegurem a optimização entre as necessidades e os consumos de energia.

Uma das principais barreiras para a implementação de soluções eficientes, que até são economicamente mais vantajosas a médio prazo, é a falta de informação integrada que permita avaliar os custos e os benefícios para diversos cenários possíveis de introdução de soluções eficientes num sistema de iluminação pública.

Sendo assim, pretende-se que a aplicação informática desenvolvida no contexto deste projecto seja uma ferramenta de simulação para construção de cenários alternativos e comparação dos resultados, em termos energéticos, económicos e ambientais, para apoio à decisão.

Ao desenvolver uma aplicação informática com estas características, esta poderá oferecer uma visão mais clara sobre os custos e benefícios de determinadas combinações de tecnologias de iluminação sobre uma área geográfica. Desta forma, esta aplicação informática poderá contribuir para um melhor planeamento de uma

rede de iluminação pública, oferecendo uma visualização fácil de assimilar de grande parte da informação necessária às decisões sobre as tecnologias a utilizar na rede.

Desta forma, a existência desta aplicação poderá incentivar e facilitar a procura de soluções de iluminação optimizadas, que permitam aumentar a eficiência energética, prevenir os custos monetários a longo prazo da utilização de tecnologias caras que possam existir na rede e diminuir os custos ambientais da emissão de gases pela rede de iluminação, contribuindo assim para o cumprimento dos compromissos ambientais assumidos por Portugal (e por conseguinte a Região Autónoma da Madeira), face à comunidade internacional.

2. ESTADO DA ARTE

A primeira preocupação no desenvolvimento deste projecto foi tentar compreender quais as tecnologias existentes no mercado e qual a melhor opção dentro dessas tecnologias, nomeadamente em duas frentes particulares: a escolha de uma plataforma de representação de dados geográficos adequada ao contexto (a representação de ramais de electricidade), e a escolha de uma tecnologia de desenvolvimento de software que também se adequasse à aplicação que se pretendia desenvolver.

Na primeira frente, ficou desde logo claro que se pretendia uma plataforma geográfica que pudesse representar mapas com detalhe elevado sobre a Ilha da Madeira. Para além da Ilha da Madeira, a disponibilidade de mapas a uma escala internacional seria sempre uma mais-valia, dado se ter colocado inicialmente a possibilidade de este projecto ser adoptado pela União Europeia para um uso a nível internacional.

Outro aspecto importante seria a flexibilidade na representação de linhas sobre os mapas, associadas a coordenadas geográficas, que representassem os ramais de electricidade. Era necessário que a tecnologia permitisse não só uma representação dos ramais que se tornasse fácil de assimilar pelo utilizador, mas também que permitisse todo o tipo de modificações em tempo de execução a esta representação, porque seria muito útil que a aplicação a desenvolver permitisse a edição destes dados por parte do utilizador.

Para além destes dois pontos essenciais (representação de mapas, e dados sobre os mapas) outros factores naturalmente foram ponderados na escolha: o custo, privilegiando-se as opções *open source* sobre opções comerciais; a existência de informação vectorial na plataforma geográfica sobre as ruas (nas quais a simulação de remais de electricidade seria baseada); a existência da possibilidade de aceder à plataforma geográfica através da Internet ou de uma rede local; a dificuldade de desenvolvimento, e o nível de suporte ao desenvolvimento, de uma aplicação sobre a plataforma geográfica em questão – assim como as perspectivas futuras de melhoramento desta plataforma.

Na outra frente, pretendia-se uma tecnologia de desenvolvimento de software que sobretudo se adequasse à plataforma geográfica escolhida. Se tivéssemos optado por um sistema GIS *stand-alone*, não faria sentido adoptar uma tecnologia de desenvolvimento de software para a *web*, como veio a acontecer.

2.1. PLATAFORMA DE INFORMAÇÃO GEOGRÁFICA

Após um período inicial de pesquisa, constatou-se que existiam várias alternativas a estudar como possíveis escolhas para a plataforma de informação geográfica do projecto.

Desde o início que a utilização das APIs do Google Maps foram vistas como potencialmente uma escolha acertada para este efeito. O Google Maps é uma aplicação de mapas para a web, estando disponível uma API para desenvolvimento de aplicações web personalizadas sobre o Google Maps [10]. A principal vantagem da sua utilização neste projecto é a cobertura global de mapas de satélite, o que facilita a possível expansão futura da ferramenta. Outra grande vantagem é a existência de um grande número de exemplos de aplicação, sítios na internet dedicados ao desenvolvimento de aplicações sobre Google Maps, documentação oficial e pequenas aplicações que facilitam a interacção com a API. Com este material é possível ter uma curva de aprendizagem mais curta que noutras alternativas, e encontrar soluções para os problemas que aparecem naturalmente durante a fase de desenvolvimento.

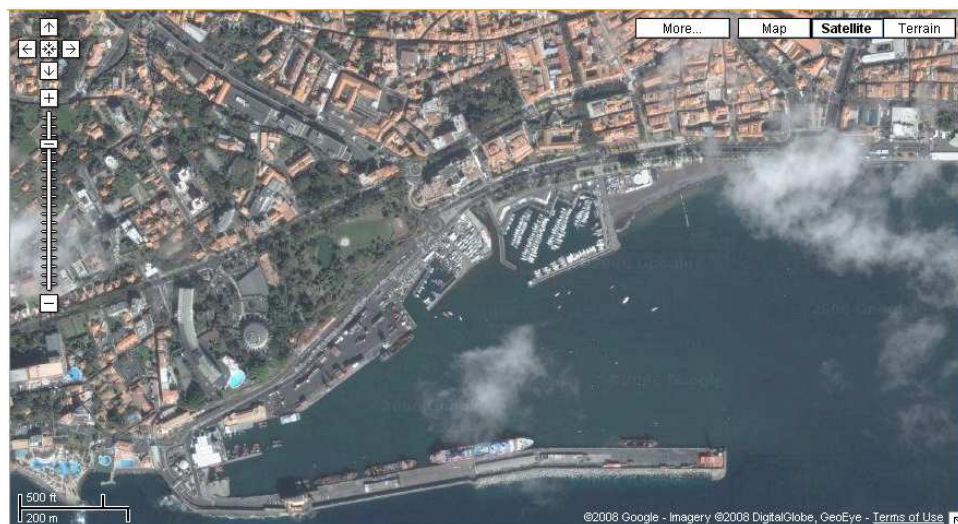


Figura 1. Imagens de satélite do serviço de web mapping Google Maps (maps.google.com)

A API do Google Maps disponibiliza uma série de funcionalidades importantes para a aplicação que se pretendia desenvolver:

- a possibilidade de definir linhas, marcadores e overlays sobre os mapas em qualquer altura de execução da aplicação;
- a possibilidade de poder alterar os atributos destas linhas, marcadores e overlays em tempo de execução, como a cor e a sua posição no mapa;
- a possibilidade de poder atribuir eventos a estes objectos, como por exemplo a execução de um bloco de código a partir da interacção do utilizador ao clicar com o rato sobre o objecto, ou a passar com o rato sobre o objecto (embora este último tipo de evento tenha alguns problemas com certos browsers em certas circunstâncias);
- o suporte existente já por defeito para marcadores arrastáveis (sendo desnecessário estar a implementar rotinas para tal efeito);
- a possibilidade de interagir com os controlos do mapa, tais como o nível de zoom ou o ponto no qual o mapa está centrado, oferecendo assim o controlo da navegação no mapa não só ao utilizador como também à aplicação caso seja necessário;
- outras funcionalidades que não seriam utilizadas no projecto em questão, mas que poderiam eventualmente ser necessárias em futuros desenvolvimentos da aplicação, como a interacção com dados carregados a partir de ficheiros XML, técnicas de geocoding (conversão de endereços em coordenadas geográficas), interacção com formatos compatíveis com o Google Earth, etc.

Finalmente, é importante considerar que o Google Maps é bastante conhecido e utilizada por milhões de pessoas em todo o mundo – ao utilizar como base do projecto o Google Maps tiramos partido da familiaridade do utilizador com a aplicação, pois este provavelmente já conhecerá a forma de interagir com os controlos e com o mapa em si.

Todavia mesmo partindo do Google Maps existiam duas possibilidades de desenvolvimento: a versão livre da API, e a versão comercial, “Google Maps for Enterprises”. A versão comercial engloba suporte dedicado extra, e permite outras funcionalidades como possibilitar a execução da aplicação numa Intranet fechada. No entanto, segundo a FAQ disponível no site [11], os custos são na ordem dos \$10.000 por ano, o que é uma verba muito para além do que se poderia considerar realista para este projecto, sem oferecer vantagens suficientemente atractivas para trazer benefício considerável.

O Google Earth foi outra alternativa considerada, por também dispor de uma API [12]. Todavia foi rapidamente descartada esta possibilidade porque, para além da API estar pouco desenvolvida, esta serve sobretudo para fornecer comandos de navegação à aplicação do Google Earth, não permitindo definir espaços de interacção complexos a incluir dentro da aplicação. A maior vantagem do Google Earth sobre o Google Maps é o suporte tridimensional. Contudo esta possibilidade é pouco relevante para o projecto, dado que a dimensão altura não traz qualquer tipo de benefício aparente para a representação dos ramais da iluminação pública. No futuro poderá ser interessante considerar o Google Earth para a criação de ficheiros KML (com informação geográfica) que possam posteriormente ser importados para a aplicação em Google Maps que está a ser desenvolvida – o que aliás é mais um motivo para a escolha da plataforma Google Maps.

Um outro serviço de web mapping, mas a nível nacional é o Sapo Mapas [13]. Este serviço foi considerado também como uma possibilidade, por conter informação vectorial detalhada de todo o território nacional, salientando-se a inclusão da ilha da Madeira com este tipo de detalhe, o que não acontece no Google Maps, que de momento apresenta apenas imagens de satélite. Os mapas baseados em informação vectorial (mapas que se focam sobretudo na representação das estradas e ruas) seriam mais adequados a este projecto, pois é esta a informação essencial para o utilizador, e tal permite uma melhor identificação da rua que está a ser visualizada, comparando com uma imagem em satélite. No entanto, a utilização destes mapas restringiriam a aplicação ao território nacional (impedindo a utilização futura da aplicação noutros países, que é um dos requisitos fundamentais do projecto). Acresce que o Sapo Mapas não oferece qualquer tipo de API de desenvolvimento livre.

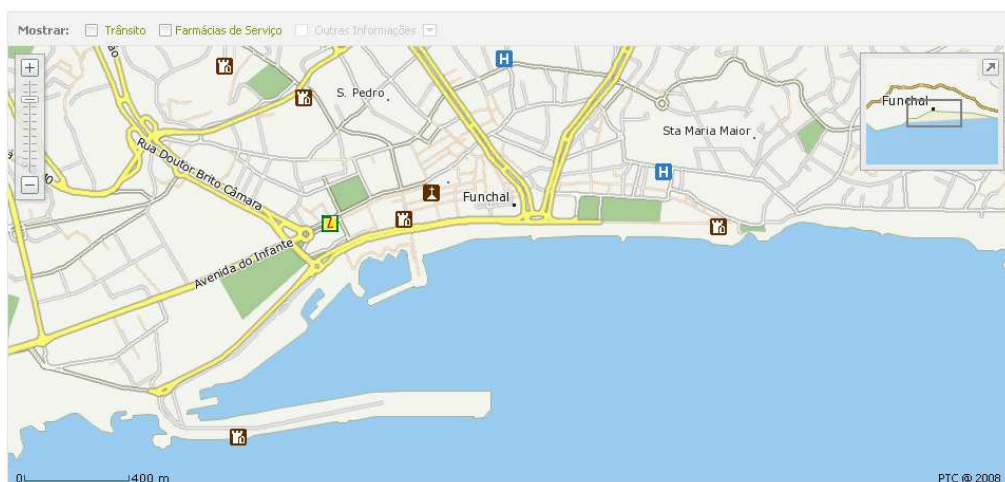


Figura 2. Mapa baseado em informação vectorial do serviço de web mapping Sapo Mapas (mapas.sapo.pt)

A nível internacional, também existem outros serviços de *web mapping* como o Yahoo Maps [14], Live Search Maps [15] e o MapQuest [16]. No entanto, apesar destes dois primeiros oferecerem APIs de desenvolvimento, estas opções teriam de ser descartadas devido à falta de um nível de detalhe suficiente dos mapas da Ilha da Madeira disponíveis nestes serviços.

Fora do âmbito do *web mapping*, foram considerados sistemas GIS *open source*, tais como o GRASS [17], o QGIS [18] e o gvSIG [19], que contam com uma comunidade considerável na *web*, e que oferecem grande nível de personalização a possíveis aplicações desenvolvidas sobre si. A principal desvantagem face ao *web mapping* é de estes sistemas serem orientados mais à introdução de mapas por parte do utilizador e/ou programador, não contando com a grande cobertura de imagens de satélite (e eventualmente imagens vectoriais) dos serviços de *web mapping*. Também têm a desvantagem de não serem tão familiares ao utilizador comum quanto os serviços *web mapping*, com os quais ocasionalmente interagem no dia-a-dia na navegação comum pela Internet.

Para tomar uma decisão definitiva sobre a tecnologia a utilizar, foi necessário conhecer as aplicações utilizadas pelas entidades responsáveis pela iluminação pública. Neste sentido, foi feita uma visita às instalações da Empresa de Electricidade da Madeira (EEM), fornecedora da iluminação pública na Região, onde foi apresentado o sistema utilizado - constatou-se que o GIS utilizado pela EEM é uma aplicação proprietária, e que tal situação ocorre na maior parte das outras entidades fora da Região. Sendo assim, não é possível utilizar esse sistema no projecto, mas ficou em aberto a possibilidade de a ferramenta a desenvolver poder importar informação geográfica proveniente de tal sistema, através de um formato de ficheiro genérico, utilizado globalmente - este sistema em particular utiliza o formato DXF, um formato desenvolvido pela AutoDesk para possibilitar a interoperabilidade de dados entre o AutoCAD e outros programas. [20]

Uma outra eventual solução que seria teoricamente possível, mas que foi imediatamente descartada, seria a possibilidade de criar um sistema geográfico de raiz. Isto permitiria um nível de personalização muito maior no desenho da aplicação, mas aumentaria a complexidade e o tempo de desenvolvimento muito para além do âmbito deste projecto.

No final, a opção tomada acabou por ser de utilizar a API livre do Google Maps como a plataforma de informação geográfica para este projecto. Esta decisão baseou-se nas vantagens desta plataforma:

- o custo livre;
- a existência de um nível de detalhe dos mapas da Região Autónoma da Madeira suficiente para a interação pretendida;
- a existência de mapas com detalhe em larga escala a nível mundial, o que permitiria extensões futuras da aplicação a outras áreas geográficas fora da Região;
- o grande número de funcionalidades de manipulação e adição de conteúdos ao mapa existentes, e disponíveis através da API (conforme referido anteriormente);
- a interface de navegação do mapa, que tem qualidade, e é familiar para grande parte dos potenciais utilizadores da aplicação a desenvolver;
- a grande comunidade de programadores sobre esta API existente na web, que torna possível recorrer a um extenso número de exemplos, tutoriais e documentação de ajuda ao desenvolvimento de aplicações sobre Google Maps.

2.2. TECNOLOGIAS DE DESENVOLVIMENTO PARA A WEB

A partir do momento que se considerou a API do Google Maps como a base para a tecnologia de informação geográfica, foi necessário proceder ao trabalho de desenvolvimento sobre a API, inicialmente apenas de forma experimental. As primeiras tentativas de desenvolver a aplicação começaram apenas com código HTML e Javascript, necessário para a interação com a API. À medida que a aplicação foi alargada para novas funcionalidades, tornou-se fundamental gravar dados numa base de dados (também ela experimental) e a primeira opção foi utilizar a base de dados MySQL, assim como a linguagem de programação PHP, tanto para a geração de código dinamicamente no lado do servidor, como também para tratar dados enviados por parte do cliente. A escolha de MySQL e PHP baseou-se no facto de serem as tecnologias livres mais utilizadas actualmente para o efeito, com uma grande comunidade online e muitos recursos de ajuda ao programador, como documentação oficial disponível, diversos tutoriais e exemplos práticos.

Após mais algum desenvolvimento efectuado sobre PHP/MySQL, constatou-se que era necessário optar por alguma tecnologia que permitisse a troca de dados remotamente entre cliente e servidor, porque tal tecnologia iria melhorar muito a experiência de utilização da aplicação - a troca remota de dados evita que seja necessário recarregar toda a página cada vez que há uma comunicação entre o cliente e o servidor. A tecnologia escolhida para o efeito foi Ajax, uma tecnologia que funciona através da troca de dados codificados em formato XML, com o cliente a correr linguagem de programação Javascript. A razão da escolha de Ajax, foi mais uma vez por esta ser a tecnologia livre mais utilizada globalmente, com as mesmas vantagens já referidas sobre PHP e MySQL. No entanto, após várias tentativas frustradas de tentar englobar Ajax na aplicação PHP existente, verificou-se que não havia permissões no servidor para tal e seria necessário fazer configurações especiais no servidor Apus, ou arranjar um servidor específico para o desenvolvimento da aplicação.

Nesse ponto surgiu a ideia de utilizar Ruby on Rails, uma framework de desenvolvimento para a web otimizada para websites que necessitam de bases de dados, que utiliza a linguagem de programação Ruby, e é estruturada através da arquitectura Model-View-Controller (MVC). A principal vantagem desta framework

é, por um lado, ser estruturada numa forma que pretende aumentar a velocidade de desenvolvimento das aplicações, através de conceitos que reduzem o número de linhas de código necessárias para definir funcionalidades, comparando com outras abordagens (como a abordagem PHP/MySQL utilizada anteriormente). Por outro lado, a framework Ruby on Rails não tem um desempenho tão bom como o PHP – mas isto não era um factor particularmente relevante pois não é esperado que a aplicação venha a ser utilizada por muitos utilizadores ao mesmo tempo, nem que tenha um nível de processamento de informação elevado no lado do servidor (processado pela linguagem Ruby).

Em relação a funcionalidades específicas necessárias a este projecto, o Ruby on Rails facilita a troca de informação por pedidos remotos uma vez que tem métodos e funções específicas para trabalhar com Ajax. Neste contexto, o Ruby gere os pedidos remotos através de ficheiros RJS, ficheiros na linguagem de programação Ruby que geram Javascript no lado do servidor. Esta arquitectura torna a interacção cliente-servidor mais simples de implementar do que em PHP.

Além disso, outra das vantagens do Ruby on Rails é a forma como é efectuada a conexão à base de dados – ao contrário da abordagem PHP, é apenas necessário colocar as informações de conexão num único ponto de um único ficheiro, o que mais uma vez simplifica bastante a implementação do código. Deixa de ser necessário estar constantemente a escrever linhas de código com informações sobre o servidor cada vez que é feito um acesso a registos da base de dados – a arquitectura do Ruby on Rails faz com que os registos da base de dados estejam disponíveis automaticamente como objectos em qualquer ficheiro em Ruby da aplicação.

Inicialmente foi definido como objectivo o desenvolvimento sobre Ruby on Rails de todas as funcionalidades previamente implementadas em PHP. Para tal, como ponto de partida foi escolhido o pacote BitNami Rubystack [21] (mais conhecido pelo nome da versão anterior, Instant Rails), que simplifica bastante a instalação do Ruby on Rails, estando disponível para diversos sistemas operativos, e incluindo desde logo um servidor Apache e uma base de dados MySQL (mesma tecnologia de base de dados utilizada anteriormente na abordagem em PHP). Este pacote traz como vantagem a facilidade de instalação e execução da aplicação no lado do servidor. A aplicação desenvolvida a partir daqui, corre localmente sem problemas, desde que o computador tenha ligação à Internet para poder aceder ao Google Maps.

Desta forma, a aplicação, embora corra num browser, pode trabalhar praticamente como se tratasse de uma aplicação standalone, com a mesma máquina a funcionar

como servidor e como cliente. Isto não impede, no entanto, que possa ser acedida por qualquer outra máquina. Também, caso se pretenda que a aplicação seja acedida apenas por uma intranet, podem ser definidas configurações mais avançadas a nível de segurança de rede, que limitem o acesso à aplicação aos computadores desta intranet. Isto permite contornar uma das desvantagens da API livre do Google Maps, que não é tão orientada à utilização em intranets quanto a API comercial.

Após seguir os primeiros tutoriais e ter a primeira aplicação Ruby on Rails a correr localmente, era necessário incluir a aplicação Google Maps. Embora fosse possível efectuá-lo normalmente como na abordagem PHP, foi utilizado um plugin designado YM4R/GM[22] que permite definir várias funções que facilitam a interacção com a API do Google Maps. Este plugin necessita de ser instalado sobre a instalação de Ruby on Rails existente. No entanto, conforme o processo de desenvolvimento da aplicação decorreu, chegou-se eventualmente à conclusão que este plugin apresentava poucas vantagens, e este acabou por ser retirado. As funções que o YM4R/GM oferece para interagir com a API do Google Maps facilitam a introdução de conteúdos no mapa, mas limitam posteriormente o acesso a estes conteúdos, impedindo a sua manipulação – o que é uma funcionalidade importante para a aplicação.

Um dos problemas observados durante as primeiras semanas de desenvolvimento em Ruby on Rails foi a dificuldade em arranjar material de apoio na Internet, devido a esta abordagem ser bem menos popular que a linguagem PHP. A este facto está associada uma certa dificuldade que advém da linguagem de programação Ruby ser consideravelmente diferente das linguagens mais usuais, utilizando convenções e abreviaturas difíceis de prever para um programador com pouca experiência na linguagem. Numa fase ainda de adaptação ao Ruby, estes dois problemas anularam a suposta velocidade de desenvolvimento superior que o Ruby on Rails possibilita – não foram raras as vezes que o desenvolvimento esbarrou com “becos sem saída”, problemas difíceis de ultrapassar pois todas as opções óbvias de contornar o problema esgotam-se, e havia pouca ajuda disponível na web.

Eventualmente foi necessário tomar uma decisão definitiva sobre qual das linguagens a usar, e dado que a aplicação em Ruby on Rails estava já mais avançada que a aplicação em PHP, optou-se por continuá-la.

2.2.1. Arquitectura do Ruby on Rails

Visto que se decidiu avançar com a utilização da plataforma Ruby on Rails, é aqui apresentada uma introdução à sua arquitectura, o que permite uma melhor

compreensão do trabalho efectuado sobre esta arquitectura, apresentado em capítulos posteriores.

Como foi referido anteriormente, o Ruby on Rails é baseado numa arquitectura Model-View-Controller (MVC), um padrão de desenvolvimento de software que consiste na separação do código da aplicação em três diferentes entidades:

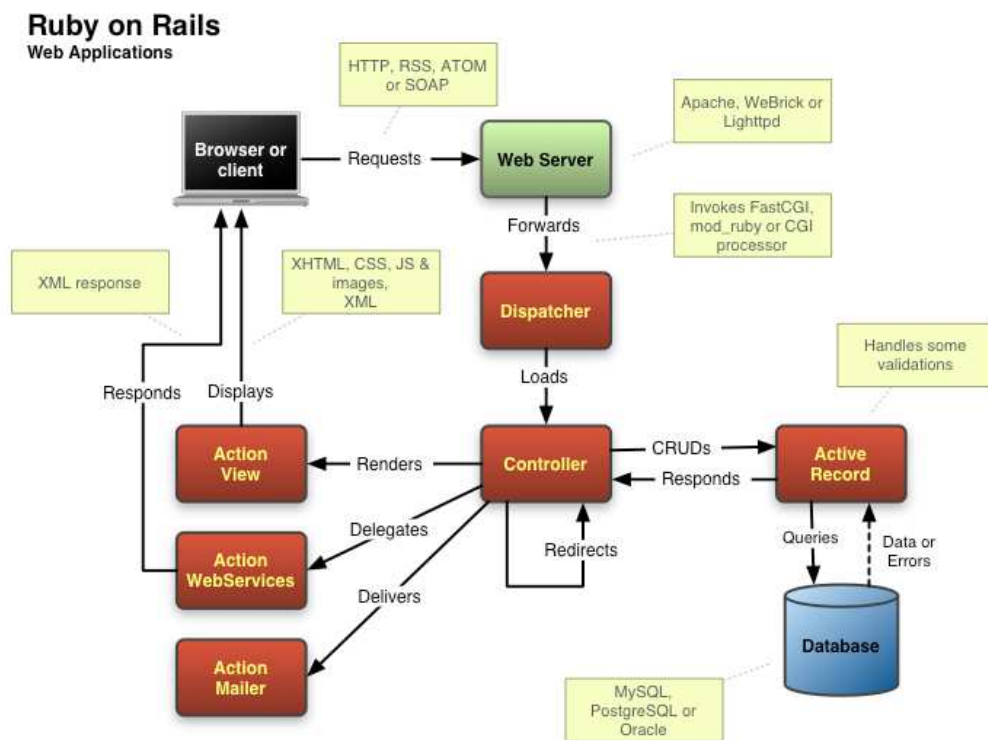


Figura 3. Arquitectura do Ruby on Rails [23]

- O **modelo (model)** existe sobretudo para definir os conceitos existentes na base de dados e as relações entre eles. No Ruby on Rails, não é necessário definir aqui explicitamente os detalhes da ligação à base de dados – a ligação é feita utilizando apenas um único ficheiro separado, o que é uma das grandes vantagens do Ruby on Rails face a outras plataformas de desenvolvimento para a web. Esta plataforma simplesmente assume que, na maior parte dos casos, as tabelas da base de dados têm o mesmo nome que um ficheiro de modelo, reduzindo assim a necessidade de definir explicitamente as correspondências entre cada lado desta ligação.

- A **vista (view)** define os detalhes da apresentação das informações ao utilizador, neste caso a aparência das páginas no browser. O Ruby on Rails utiliza para este efeito ficheiros RHTML, que são ficheiros HTML com Ruby embebido, sendo este o principal mecanismo que permite a apresentação de páginas de forma dinâmica.

- O principal objectivo da utilização da arquitectura MVC é a separação entre as vistas e os modelos (o que traz grandes benefícios de reusabilidade, e de legibilidade do código). Sendo assim, é necessário um nível de abstracção que faça a comunicação entre a vista e o modelo – neste caso o **controlador (controller)**. Um ficheiro de controlador normalmente está associado directamente a um ficheiro de modelo (excepto no caso de um controlador que controle outros controladores), e utiliza os dados abstraídos pelo ficheiro de modelo em acções, sendo que estas acções são posteriormente chamadas pelos ficheiros de vistas.

3. ANÁLISE DE REQUISITOS

A análise de requisitos é uma fase fundamental de qualquer projecto de desenvolvimento de software, pois a percepção errada dos requisitos pode levar ao desenvolvimento de aplicações com funcionalidades inadequadas, e de aplicações com atrasos em fases posteriores com a redefinição de funcionalidades. Durante toda a formação académica na Universidade da Madeira, os alunos são sempre alertados para a necessidade de haver esta análise para todo o tipo de aplicações informáticas complexas, e logicamente considerou-se adequado efectuar-la neste projecto, com um método baseado nos conceitos aprendidos durante o curso, nomeadamente nas disciplinas de Engenharia de Requisitos, Interação Homem-Máquina e sobretudo em Desenvolvimento Centrado nos Utilizadores.

A familiaridade adquirida ao longo do curso com estes conceitos facilitou a compilação de um documento de requisitos na fase inicial do projecto, servindo de base para o desenho das interfaces e para a definição de quais as funcionalidades que irão ser desenvolvidas. A abordagem, ao ser centrada completamente nas necessidades do utilizador, tem como objectivo fundamental criar uma interface que se adequa o mais possível às tarefas que este terá de utilizar.

Optou-se por começar a análise fazendo uma lista de requisitos, porque trata-se de uma técnica que permite uma discussão em linguagem comum, com o cliente ou destinatário da aplicação (neste caso, a AREAM), com vista a acordar quais as funcionalidades necessárias a implementar, e quais as funcionalidades com maior prioridade. Desta forma, consegue-se obter um bom ponto de partida para todo o resto da análise de requisitos.

De seguida, utilizou-se técnicas como a elaboração de mapas de desempenho, de actores e tarefas, de actividades, definindo também perfis para as actividades envolvidas no processo de interacção com a aplicação. Estas técnicas permitem perceber a forma como os utilizadores irão utilizar o sistema, partindo da identificação dos actores e papéis intervenientes na interacção, passando pela enumeração das tarefas (ou casos de utilização) e actividades, e a forma como se

relacionam entre si. Finalmente foram descritas as tarefas e actividades em detalhe, para que se tenha uma percepção aprofundada de todos os passos da interacção.

Sendo assim, numa primeira fase foram efectuados os seguintes métodos de análise de requisitos:

- Lista de requisitos funcionais e não-funcionais;
- Mapa de desempenho;
- Mapa de actores e tarefas;
- Mapa de actividades;
- Perfis das actividades;
- Mapa de actividades e tarefas;
- Descrição das tarefas.

Note-se que esta ferramenta não vem substituir um sistema existente, mas sim ocupar um espaço que se encontra vazio, ou seja, há actualmente apenas utilizadores potenciais e não casos concretos de interacção actual que possam ser estudados e analisados. Por esta razão, foi necessário assumir certas características e comportamentos, sobretudo nos perfis das actividades, baseados no que foi comunicado durante as reuniões e durante a visita à Empresa de Electricidade da Madeira (onde trabalham eventualmente potenciais utilizadores do sistema, embora se tenha chegado à conclusão que a ferramenta será mais útil para entidades reguladoras do que aos próprios fornecedores de energia).

Posteriormente, após a elaboração do documento de requisitos com os métodos de análise referidos anteriormente, foram elaborados diagramas do método Wisdom [24], nomeadamente um modelo de análise e um modelo de navegação, como fase intermédia entre a análise de requisitos e o desenho das interfaces. Estes diagramas, apesar de se terem tornado bastante complexos, foram elaboradas devido a permitirem estabelecer um paralelo directo entre as tarefas que os utilizadores necessitam realizar e os espaços de interacção que irão formar a interface final.

Finalmente, o desenho das interfaces foi efectuado através de Protótipos Abstractos Canónicos [25], que é um método de desenho que utiliza notações formais para todos os elementos de uma interface, dando uma percepção espacial da colocação destes

elementos, mas mantendo um certo nível de abstracção, o que dá margem de manobra para na fase de implementação se poder escolher qual o *widget* específico a utilizar em cada elemento – mas definindo desde já todos os elementos e a funcionalidade geral que cada um deles deverá ter.

Todo este trabalho foi realizado ao mesmo tempo que o trabalho de implementação, apesar de ser mais usual começar com a fase de implementação após a definição dos requisitos e desenho das interfaces. Isto deve-se à necessidade de experimentar desde logo certas funcionalidades das tecnologias, testando se as tecnologias que foram ponderadas seriam adequadas ao projecto. Assim que os Protótipos Abstractos Canónicos foram concluídos, a aplicação implementada foi adaptada para começar a tomar a forma da interface pretendida.

3.1. PROCESSO DE ANÁLISE DE REQUISITOS

Este sub-capítulo pretende explicar as técnicas usadas durante o processo de análise de requisitos, e como os modelos elaborados derivam uns dos outros. Desta forma, é apresentado um único exemplo de cada técnica ou modelo. Para consultar todos os modelos e todo o detalhe do trabalho desenvolvido, consulte o Anexo I: Documento de requisitos da aplicação.

3.1.1. Lista de requisitos funcionais e não-funcionais

Esta lista de requisitos, ponto de partida para toda a análise de requisitos da aplicação, caracteriza-se pela enumeração de todas as condições que devem ser satisfeitas pela aplicação, conforme as expectativas do cliente e pelos utilizadores da aplicação. A lista efectuada não só enumera os requisitos, como também apresenta uma breve explicação das suas particularidades, e um nível de prioridade a considerar na fase de implementação.

Note-se que esta lista de requisitos foi dividida em requisitos funcionais e não-funcionais. Requisitos funcionais definem funcionalidades específicas a serem produzidas pela aplicação, enquanto requisitos não-funcionais definem características ou restrições à aplicação. [26]

Exemplo de um requisito funcional da aplicação:

R01: O sistema deverá armazenar a informação relevante sobre a iluminação já existente numa determinada área geográfica.

Descrição: Para a área geográfica a analisar, o sistema deverá armazenar a informação sobre os ramos de iluminação, os segmentos em que estes se subdividem, e o tipo e quantidade de lâmpadas, luminárias, reguladores e sensores utilizados em cada segmento.

Prioridade: Muito alta

E um exemplo de um requisito não-funcional:

R12: O sistema deverá ser fácil de operar e aprender.

Descrição: Na maior parte dos casos, os estudos efectuados com este sistema serão feitos por pessoas que não irão dar um uso continuado ao sistema. Sendo assim, torna-se muito importante que a curva de aprendizagem da interacção com o sistema não seja acentuada.

Prioridade: Muito alta

A distinção entre estas duas categorias permite focar o desenvolvimento da aplicação primeiro nas funcionalidades a produzir, e segundo, na sua qualidade. [26]

3.1.2. Mapa de desempenho

O mapa de desempenho da aplicação pretende identificar os *players*, actores e papéis que participam no processo de interacção com a aplicação.

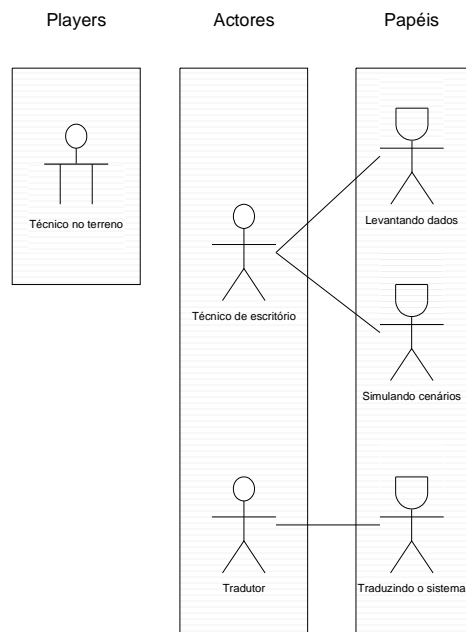


Figura 4. Mapa de desempenho da aplicação

Os **actores** são os utilizadores que interagem directamente com a aplicação. Estes utilizadores podem, no entanto, desempenhar vários **papéis** diferentes no processo de interacção, ou seja, desempenhar relações diferentes com o sistema consoante determinados contextos. [27]

Para além destes, os *players* são pessoas que não interagem directamente com a aplicação, mas que podem influenciar o processo de interacção com a aplicação. Por exemplo, comunicando com um actor enquanto este insere dados no sistema.

3.1.3. Mapa de actores e tarefas

Este mapa vem no seguimento do diagrama anterior, pois pretende associar cada papel desempenhado por um actor, a cada tarefa específica relacionada com a sua interacção com o sistema.

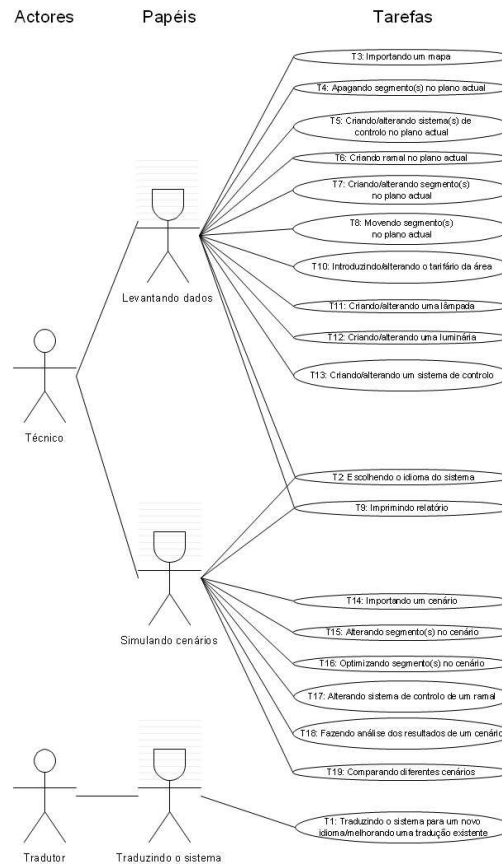


Figura 5. Mapa de actores e tarefas da aplicação

Desta forma, é assim possível listar desde já todas as tarefas para as quais o sistema deve estar preparado, oferecendo funcionalidades que permita aos utilizadores efectua-las.

3.1.4. Mapa de actividades

Este diagrama pretende representar actividades relevantes à aplicação a desenvolver, assim como representar as relações entre estas mesmas actividades. As actividades são entendidas neste contexto como uma forma de agregar várias tarefas em colecções maiores e tenuemente interligadas, por um propósito comum. [28]

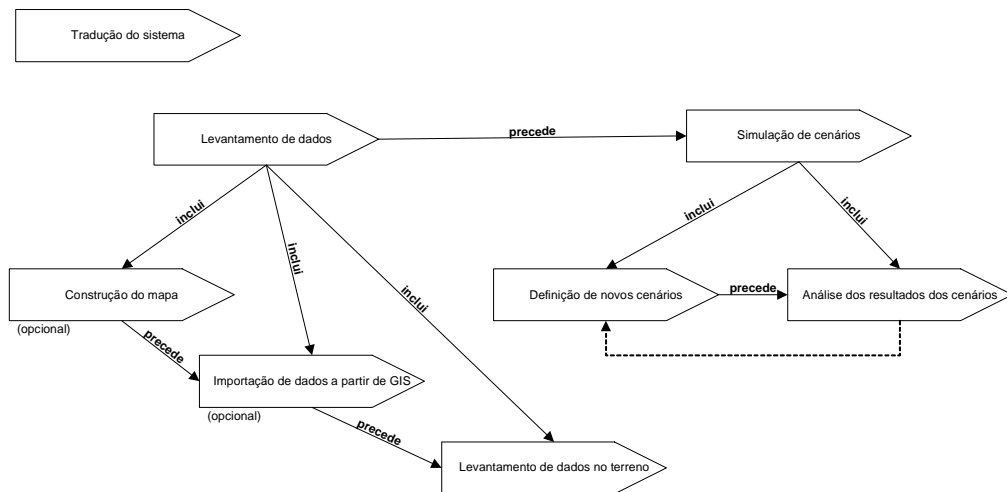


Figura 6. Mapa de actividades da aplicação

Note-se que as actividades podem incluir outras actividades, e que as actividades podem seguir uma determinada sequência, ou serem concorrentes umas com as outras. É importante perceber estas relações para desenvolver uma aplicação que se adequa às necessidades do utilizador.

3.1.5. Perfis das actividades

Identificadas as actividades, é necessário perceber em que elas consistem, em mais pormenor. O objectivo da elaboração de perfis de actividades é organizar em forma compacta os seus aspectos essenciais. Larry Constantine, no seu trabalho sobre Modelação de Actividades, recomenda a definição dos perfis das actividades sobre quatro diferentes categorias: propósito, tempo e lugar, participantes e desempenho. [28]

Exemplo do perfil de uma actividade:

Actividade: Análise dos resultados dos cenários

Propósito: analisar e comparar resultados dos diferentes cenários obtidos, tanto entre si, como com a actual iluminação pública existente na área.

Tempo e Lugar: esta actividade deverá ocorrer repetidamente no escritório, utilizando o sistema, sendo interligada com a definição dos detalhes de cada cenário.

Participantes: Um técnico que terá de ter conhecimentos algo avançados na área da electricidade (e conhecimentos básicos de informática de modo a poder utilizar o sistema), e que provavelmente terá de comunicar os seus resultados obtidos a pessoas com maior poder de decisão, possivelmente com cargos políticos. Note-se que esta

actividade em particular poderá ter maior envolvência destas pessoas, que deverão basear as suas decisões nos relatórios/conclusões que resultam desta actividade.

Desempenho: O sistema deverá tentar apontar ao utilizador os principais dados relevantes na comparação entre vários cenários, sem negar a possibilidade de o utilizador verificar todas as diferenças de forma detalhada. Os resultados deverão ser apresentados de forma bastante elucidativa, recorrendo sempre que possível a representações gráficas, permitindo também a geração de relatórios extensos.

3.1.6. Mapa de actividades e tarefas

Este diagrama aprofunda o diagrama apresentado em 3.1.4. (mapa de actividades), e faz uma ligação ao diagrama apresentado em 3.1.3. (mapa de actores e tarefas), definindo que tarefas correspondem a que actividades.

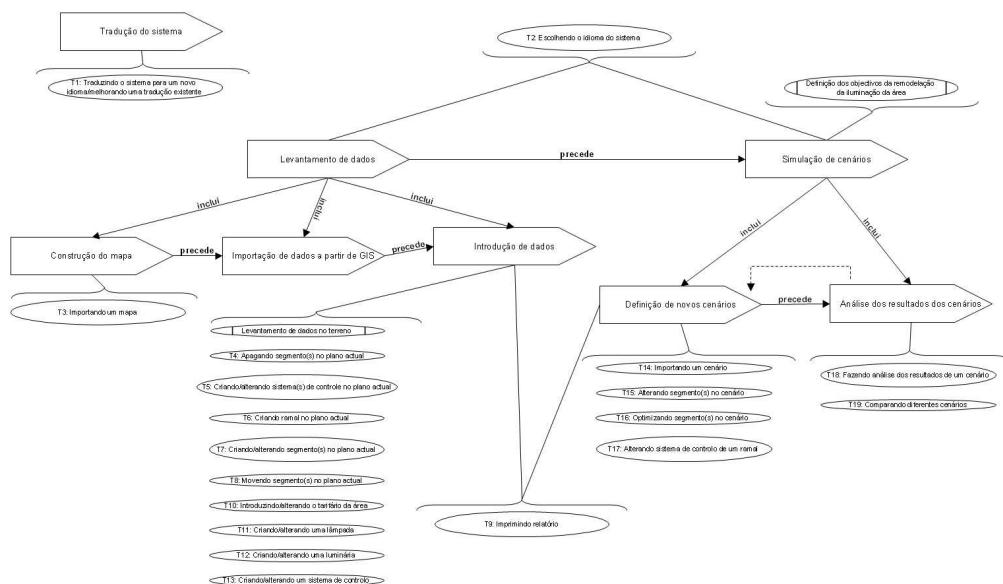


Figura 7. Mapa de actividades e tarefas da aplicação

De referir que, para além das tarefas apresentadas anteriormente, são aqui apresentadas outras tarefas, não relacionadas directamente com a interacção com o sistema, mas relacionadas com as actividades. Mais uma vez, a representação destas tarefas destina-se à melhor compreensão de todo o processo de interacção, permitindo caracterizar melhor as actividades e identificar tarefas não relacionadas com o sistema que interfiram com tarefas relacionadas com o sistema.

3.1.7. Descrição das tarefas

Identificadas não só as tarefas, mas também as relações entre si, através do nível de abstracção superior representado pelas actividades, é agora possível descrever as tarefas de uma forma mais detalhada.

Na realidade, apesar de ter sido utilizado a designação “tarefa” até agora, a “descrição das tarefas” efectuada aqui acaba por se aproximar mais de um típico caso de utilização com narrativa dividida entre intenções do utilizador e responsabilidades do sistema, do que propriamente do conceito de “tarefa” e de *task cases* apresentado por Larry Constantine, que foi seguido mais de perto nos modelos anteriores. [29] Esta opção deveu-se a uma maior familiaridade com esta notação.

Exemplo da descrição de uma tarefa:

T8: Movendo segmento no plano actual	
Intenções do utilizador	Responsabilidades do sistema
	1. Fornecer lista ou visualização dos segmentos
2. Escolher segmento	
3. Escolher nova localização geográfica do segmento	
	4. Gravar as alterações
	5. Actualizar toda a informação e visualizações afectadas

cenário. Outra ideia importante é permitir a navegação do utilizador a qualquer altura entre as diversas secções, daí o modelo de navegação ter sido dividido em dois diagramas, um dedicado a esta navegação entre as várias secções, e outro para todo o resto.

3.2.2. Protótipos Abstractos Canónicos

A elaboração dos Protótipos Abstractos Canónicos [25] assentou nos diagramas Wisdom, tendo no entanto sofrido algumas alterações, sobretudo com o intuito de uniformizar a interacção entre os vários espaços de interacção. A ideia principal é dividir o espaço em três: o menu na esquerda, em que o utilizador pode navegar entre as várias secções; o espaço maior a meio que deverá ser ocupado pelo mapa nas secções de edição do plano actual e do cenário; e o espaço na direita reservado para todos os detalhes, com vários espaços sobrepostos correspondendo ao nível de interacção (opções, ramal, segmento). No fundo, trata-se um pequeno resumo dos resultados da simulação, evitando que o utilizador tenha constantemente de mudar para a secção dos resultados enquanto altera o cenário.

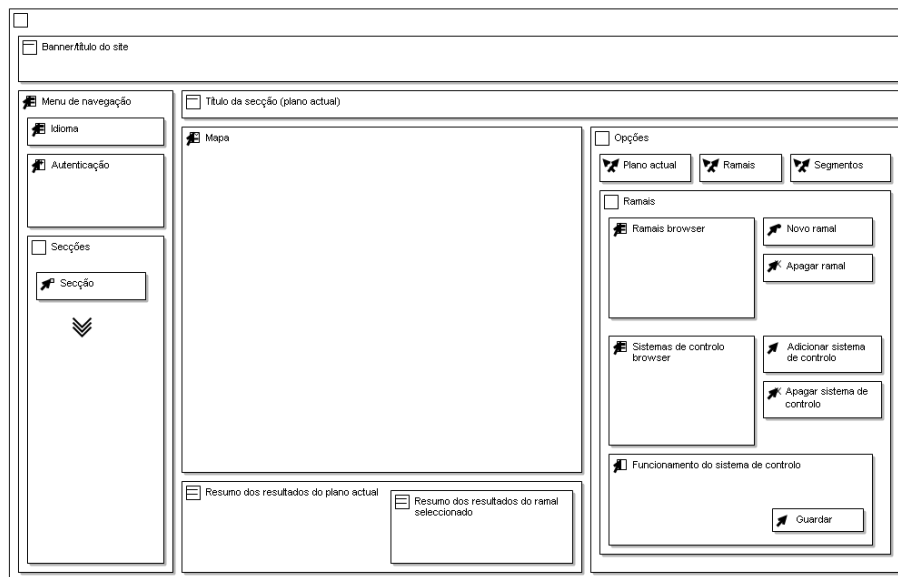


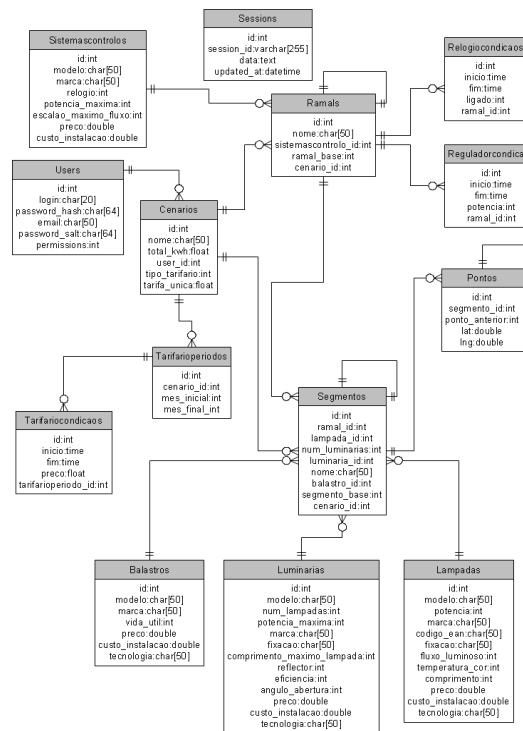
Figura 9. Protótipo abstracto canónico da secção “plano actual” (espaço ramais)

Outra particularidade deste desenho é a escolha dos equipamentos ser feita através de espaços de interacção que são lançados quando o utilizador carrega em “adicionar sistema de controlo” ou “escolher lâmpada”. Isto deve-se à falta de espaço no espaço de interacção para apresentar todos os detalhes necessários, evitando recarregar o mapa, que será sempre uma operação que levará algum tempo.

Os espaços de interacção da edição de equipamentos, e da visualização de resultados são relativamente simples, com as funcionalidades básicas necessárias, mais opções de filtragem para ajudar a que o utilizador encontre os equipamentos necessários e visualize os resultados pretendidos.

3.3. BASE DE DADOS

Como consequência da análise de requisitos efectuada anteriormente, e no decurso do desenvolvimento da aplicação, foram identificadas as seguintes entidades a serem representadas na base de dados desenvolvida em MySQL:



- **Ramal:** representa um conjunto de lâmpadas e luminárias dispostas sobre o plano geográfico, que são todas elas controladas pelo mesmo sistema de controlo.
- **Segmento:** representa uma subdivisão de um ramal em que todas as lâmpadas e luminárias são idênticas. No mapa, é representado por uma linha.
- **Ponto:** representa um ponto no mapa que faz parte de um segmento. Na base de dados, foi definida uma relação unária na tabela referente a esta entidade, com o objectivo de guardar a sequência entre os pontos necessária para representá-los pela forma de linha sobre o mapa.
- **Cenário:** representa um cenário de simulação de alterações à rede de iluminação – excepto o cenário com id 1, que representa a rede de iluminação existente actualmente. Todos os ramais e segmentos são associados a um cenário, ou seja, tanto

podem representar um ramal/segmento existente na rede actual, como uma cópia desse ramal/segmento no contexto do cenário. É devido a esta razão que as tabelas de ramais e segmentos têm relações unárias (para identificar qual o ramal/segmento a que correspondem na rede actual).

- **Luminária:** define as características de um suporte para uma lâmpada.
- **Lâmpada:** define as características de uma determinada lâmpada.
- **Balastro:** define as características de um determinado balastro.
- **Sistema de controlo:** representa um sistema de controlo que pode ser utilizado num ramal. A tabela na base de dados inclui um campo “relógio” que funciona como um valor booleano para definir o tipo de sistema de controlo, que pode ser relógio ou regulador de fluxo – caso em evoluções futuras do sistema seja necessário definir outros tipos de sistemas de controlo, provavelmente este campo deverá ser substituído por uma relação com uma nova entidade que defina um tipo de sistema de controlo.
- **Condição do relógio:** representa uma condição do funcionamento de um relógio atribuído a um ramal, ou seja, define um determinado intervalo de tempo em que o relógio liga ou desliga a electricidade no ramal.
- **Condição do regulador de fluxo:** representa uma condição do funcionamento de um regulador de fluxo atribuído a um ramal. A diferença em relação ao relógio é que neste caso é atribuído uma percentagem de potência no intervalo de tempo representado.
- **Período do tarifário:** representa um determinado período de meses no qual se aplicam determinadas condições de preço da iluminação, no caso de a área geográfica a analisar tiver um tarifário sazonal.
- **Condição do tarifário:** representa uma condição do tarifário sazonal, associada a um período de meses. Define o preço da electricidade num intervalo de tempo.
- **Utilizador:** representa um utilizador da aplicação.
- **Sessão:** representa uma sessão de um utilizador na aplicação.

A nomenclatura utilizada para as entidades da base de dados foi feita de modo a coincidir facilmente com as convenções de nomenclatura do Ruby on Rails, que

geralmente reconhece nomes de tabelas terminadas em “s” como o plural da entidade a ser representada por um ficheiro de modelo. Sendo assim, a tabela que corresponde à entidade segmento foi designado de “segmentos” na BD, ramal de “ramals”, etc.

Os relacionamentos entre as tabelas da base de dados podem ser consultados em mais pormenor no Anexo I: Documento de requisitos da aplicação, no diagrama de entidade-relacionamento.

4. PROCESSO DE DESENVOLVIMENTO E ARQUITECTURA DA APLICAÇÃO

Neste capítulo pretende-se descrever o processo de desenvolvimento experimental que a aplicação atravessou nas fases iniciais, que levou à escolha da tecnologia Ruby on Rails sobre a tecnologia PHP, assim como a arquitectura da aplicação no final do processo de desenvolvimento.

Escolhida a plataforma de informação geográfica (Google Maps), e mesmo antes desta escolha ser definitiva, foi necessário testar as capacidades de interacção com a API do Google Maps nas duas tecnologias web consideradas – os dois primeiros sub-capítulos descrevem o trabalho efectuado durante este período experimental.

Posteriormente, escolhida a tecnologia web, a aplicação foi desenvolvida com o objectivo de tornar a análise de requisitos feita anteriormente numa aplicação utilizável e confiável, tendo em grande parte sido respeitada a análise de requisitos excepto em alguns pormenores em que se revelou haver benefícios na alteração de elementos da interface. Algumas destas alterações resultaram da introdução de novas funcionalidades, ou de uma melhor ponderação sobre a forma de apresentar as funcionalidades previstas, que se constaram que podiam melhorar a interacção com o utilizador; outras alterações deveram-se a um excesso de complexidade na implementação de certos elementos previstos inicialmente, obrigando a que estes fossem descartados ou substituídos por outros elementos.

Sendo assim, este capítulo contém também uma descrição da arquitectura final da aplicação, já com todas estas alterações efectuadas, descrevendo-se tanto a base de dados como a estrutura de ficheiros de código e a forma como a aplicação funciona internamente. Para tal, inclui-se também uma introdução à arquitectura MVC do Ruby on Rails, para facilitar a compreensão da arquitectura da aplicação.

4.1. ABORDAGEM INICIAL SOBRE PHP

A primeira motivação para começar a desenvolver logo de início uma aplicação sobre Google Maps foi verificar até que ponto é que a API poderia permitir a introdução de um nível de imagens que pudesse representar a informação vectorial das estradas no mapa da Ilha da Madeira (posteriormente verificou-se que, embora seja uma funcionalidade interessante, esta não seria tão crítica à aplicação como se pensava). A API do Google Maps permite dois métodos de criação de overlays sobre o mapa: através da definição de GroundOverlays (imagens “soltas” sobrepostas ao mapa terrestre, com métodos específicos disponíveis para a manipulação de cada imagem por si) ou TileOverlays (imagens ligadas entre si que podem até servir para substituir completamente o universo de imagens representadas pelo Google Maps).

As primeiras tentativas de experimentação com GroundOverlays consistiam sobretudo na sobreposição de um GroundOverlay a representar as estradas de uma zona do Funchal, e uma mini-aplicação que permitia redimensionar esse GroundOverlay através da introdução manual de novas coordenadas que delimitavam a imagem, ou arrastando markers em cima do próprio mapa que redimensionavam a imagem.

A primeira funcionalidade adicionada foi a sobreposição de vários overlays a partir da base de dados agora disponível, e posteriormente o alargamento da funcionalidade de redimensionamento a todos os overlays carregados da base de dados. O objectivo era tentar definir a base para uma interface que permitisse a um utilizador carregar uma série de imagens e construir ele próprio uma camada de imagens interactivamente.

Entretanto o trabalho com TileOverlays estava a dar melhores resultados, e o foco passou a ser na introdução de linhas poligonais no mapa que pudessem representar ramais/segmentos de electricidade – para testar até que ponto é que estes elementos da API do Google Maps eram maleáveis. A primeira funcionalidade implementada foi a sobreposição de linhas poligonais com marcadores a permitirem o seu redimensionamento, aproveitando o código existente para o redimensionamento dos GroundOverlays.

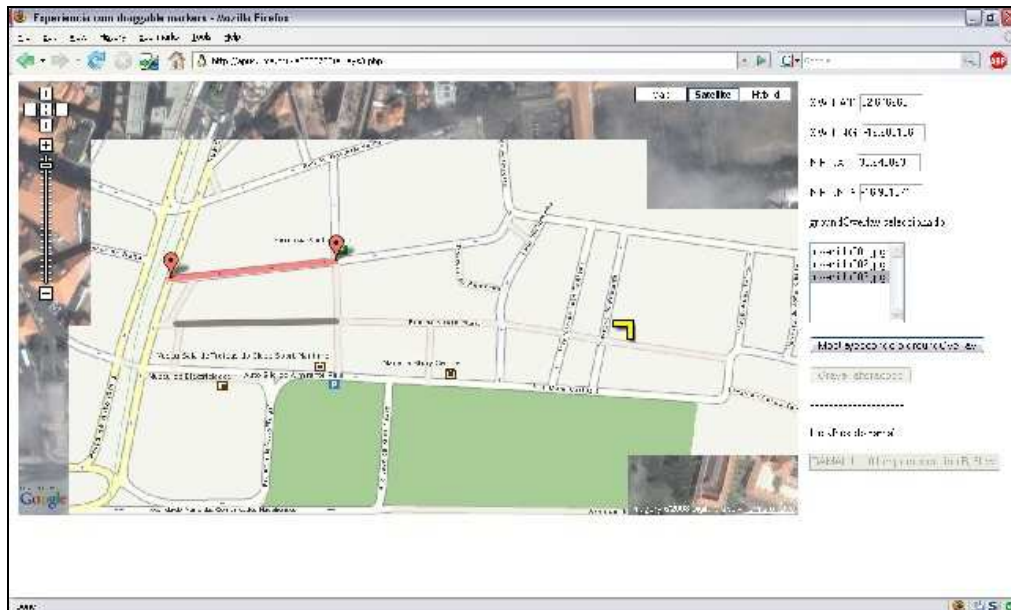


Figura 11. Interface experimental inicial para o redimensionamento de GroundOverlays e linhas poligonais

As funcionalidades adicionadas posteriormente foram a utilização de linhas poligonais com mais do que dois pontos a delimitá-las, o carregamento da informação das linhas poligonais a partir da base de dados, e a gravação da informação das linhas poligonais existentes na base de dados. A última tentativa de adicionar uma funcionalidade foi a gravação na base de dados através de uma comunicação remota. Estes foram os últimos desenvolvimentos da aplicação inicial em PHP, dado que posteriormente se optou por experimentar uma outra abordagem, conforme é descrito nas próximas secções.

4.2. ABORDAGEM INICIAL SOBRE RUBY ON RAILS

O desenvolvimento em Ruby on Rails começou com a adaptação de um tutorial disponível na web [30] a uma base de dados simples, constituída somente por uma tabela Ramal, uma tabela Segmento e uma tabela Ponto com relações de um-para-muitos, ou seja, um ramal terá vários segmentos, e um segmento (representado no mapa por uma linha poligonal) será constituído por vários pontos. A execução e posterior adaptação deste tutorial levaram a que fossem geradas várias páginas dinâmicas de introdução de dados directamente nas tabelas definidas na base de dados.

A ligação ao Google Maps deu-se inicialmente através do plugin YM4R/GM [22]. Foi definido um novo controlador externo àqueles criados especificamente para cada tabela da base de dados – e nesse controlador foi definido um método *index*, que o motor do Ruby on Rails liga automaticamente ao layout *index.rhtml*. Desta forma, e sendo que ao novo controlador foi dado o nome provisório de *MyTest*, a aplicação podia ser acedida através de http://localhost:3000/my_Test (3000 é o número da porta na qual a aplicação está configurada para correr por defeito).

Com o mapa embebido a funcionar, os próximos passos concentraram-se na tentativa de incluir todas as funcionalidades que estavam disponíveis em PHP, para as linhas poligonais (constatou-se que o trabalho sobre overlays já não era tão relevante quanto parecia no início do processo de desenvolvimento). Foi possível, após algum tempo, definir essas funcionalidades, nomeadamente o carregamento a partir da base de dados das linhas poligonais com vários pontos, o redimensionamento através de marcadores, e a apresentação de informação relativa à linha poligonal seleccionada numa área fora do mapa.

Outra funcionalidade importante que foi possível implementar foi a comunicação remota com o servidor através de ligações para funções RJS alojadas no servidor (ou seja, funções que processam código Ruby com o objectivo de gerar código javascript). Isto processa-se da seguinte forma: o utilizador carrega numa ligação, o cliente envia o pedido remoto específico para o servidor, o servidor processa a função RJS que gera um código javascript, esse código javascript é enviado para o cliente e executado na página que está a ser visualizada.

Neste ponto, verificou-se que era possível e preferível continuar com esta implementação, descartando a aplicação PHP.

4.3. MODELOS E CONTROLADORES UTILIZADOS, E O SEU FUNCIONAMENTO

4.3.1. Modelos

Todos os modelos utilizados na aplicação corresponde directamente às tabelas da base de dados, embora seja possível definir outros tipos de modelos. Sendo assim, a aplicação contém os seguintes modelos: **balastro, cenário, lampada, luminaria, ponto, ramal, condição do regulador, condição do relógio, segmento, sistema de controlo, condição do tarifário, período do tarifário e utilizador**. Para além da correspondência às tabelas da BD, há uma particularidade importante que é definida nestes ficheiros, contrariamente ao usual: os relacionamentos entre as entidades. Sendo assim, não é necessário definir chaves primárias e chaves estrangeiras na base de dados em si. Nestes ficheiros também são definidos determinados atributos que são atribuídos aos modelos para serem posteriormente utilizados na visualização das páginas, como a concatenação de certos campos da base de dados.

Note-se que, apesar de nem todos os modelos derivarem directamente dos modelos Wisdom, uma grande parte destes já estavam presentes nos modelos de análise, tendo sido acrescentado modelos para conceitos da base de dados que se revelaram posteriormente ser necessários (p. ex: os utilizadores, as condições dos sistemas de controlo), assim como foram removidos modelos que se revelaram conceitos externos à base de dados (p. ex: o mapa). Os modelos Wisdom podem ser consultados no Anexo I: Documento de Requisitos da Aplicação.

4.3.2. Controladores

Os controladores utilizados na sua maioria também derivam das tabelas da base de dados, na sua maioria sem definir acções específicas para os respectivos modelos (**balastro, lâmpada, luminária, ponto, ramal, condição do regulador, condição do relógio, segmento, sistemas de controlo, condição do tarifário, período do tarifário, utilizador**). A principal excepção é o controlador geral **fsip** (nota: FSIP = Ferramenta de Simulação da Iluminação Pública) que define acções que englobam os vários modelos da aplicação. Algumas destas acções correspondem directamente a vistas apresentadas no browser (a apresentação da página “rede actual” com os ramais existentes dispostos sobre o mapa; a apresentação da página “cenários” em que

podem ser criados cópias de cenários; a página de edição do tarifário; e a página inicial da aplicação); outras correspondem a pedidos Ajax que podem ser efectuados durante a interacção com o utilizador ou durante o próprio carregamento da página. Alguns destes pedidos Ajax são directamente sobre um modelo, mas foram definidos neste ficheiro porque necessitam de uma série de informações sobre outros modelos, que já se encontram em memória aquando do pedido de visualização da página, efectuado a partir deste controlador.

O único controlador que corresponde directamente a um modelo, e define acções sobre ele, é **cenário**, isto porque no browser existe uma página de edição de um cenário específico. Este controlador inclui certas acções semelhantes às acções definidas em *fsip*, definidas para a visualização da rede actual, mas omite uma série de acções que só fazem sentido para a rede actual, visto que a apresentação de um cenário não oferece tantas opções de edição ao utilizador.

Para além destes controladores, outros dois controladores definem especificamente elementos necessários para a navegação da página através de tabuladores (utilizando como base um *plugin* para o Ruby on Rails de seu nome Rails Widgets, em particular o componente Tabnav [31]): **tabulador da rede actual**, e **tabulador do cenário**. A utilização de tabuladores foi a alternativa escolhida à sobreposição de informação definida nos Protótipos Abstractos Canónicos (que podem ser consultados no Anexo I: Documento de Requisitos da Aplicação) para os espaços de interacção que contêm o mapa. Estes ficheiros controladores contêm acções que definem quais os modelos que devem ser carregados em memória para cada tabulador apresentada no browser, seguindo a arquitectura específica do componente Tabnav.

Um controlador especial é **aplicação**, que define parâmetros genéricos da aplicação, tais como os dados da sessão de utilizador.

5. INTERACÇÃO COM O UTILIZADOR

A aplicação encontra-se dividida em várias secções, com o objectivo de cada secção englobar casos de utilização semelhantes, ou que necessitam de elementos semelhantes na página, ou, idealmente, que correspondem a uma actividade (ver Mapa de Actividades no Anexo I: Documento de Requisitos de Aplicação).

As secções da aplicação são as seguintes:

- **Página inicial:** onde é apresentada uma pequena introdução e explicação da aplicação a todos os novos utilizadores.



Figura 12. Página inicial da aplicação

- **Tarifário:** onde é possível ao utilizador definir o tarifário em vigor, que tanto pode corresponder a uma tarifa fixa ou a uma tarifa sazonal.

- **Cenários:** onde o utilizador pode editar cenários, ou seja, criar novos cenários a partir da rede actual ou de cenários existentes. Esta secção inclui sub-secções, que são as mais importantes de toda a aplicação: a sub-secção **rede actual**, em que o utilizador tem opções para desenho sobre o mapa e representação das tecnologias de iluminação

utilizadas em toda a rede actual de electricidade; e sub-seccões relativas a cada **cenário**, em que o utilizador pode simular alterações das tecnologias de iluminação utilizadas em cada ramal e obter um comparativo rápido de consumo de electricidade versus a rede actual.

- **Relatórios**: onde são apresentados ao utilizador os resultados da simulação de consumo de energia, através de relatórios detalhados.

- **Equipamentos**: onde é possível ao utilizador adicionar, editar e apagar todos os equipamentos de iluminação disponíveis na simulação. Esta secção deverá ser dividida em 4 sub-seccões: **Lâmpadas**, **Balastos**, **Luminárias** e **Sistemas de Controlo**, cada uma qual dedicada a um tipo de equipamento.

- Para além destas secções, foram introduzidas, numa fase tardia do processo de desenvolvimento, duas secções relativas à autenticação: **Controlo de utilizadores**, com permissões exclusivas ao administrador do sistema, em que este pode gerir as permissões dos utilizadores; e **Registo de utilizadores**, em que novos utilizadores podem-se registar no sistema para ter acesso às outras secções que não a página inicial.

Note-se que estas secções resultam do trabalho de análise de requisitos efectuado inicialmente, nomeadamente dos espaços de interacção presentes no Mapa de Navegação - apesar de ter havido algumas evoluções com a introdução de novas secções, e a reestruturação da hierarquia em que certas secções se enquadram. Também foi descartado o ecrã de tradução do sistema.

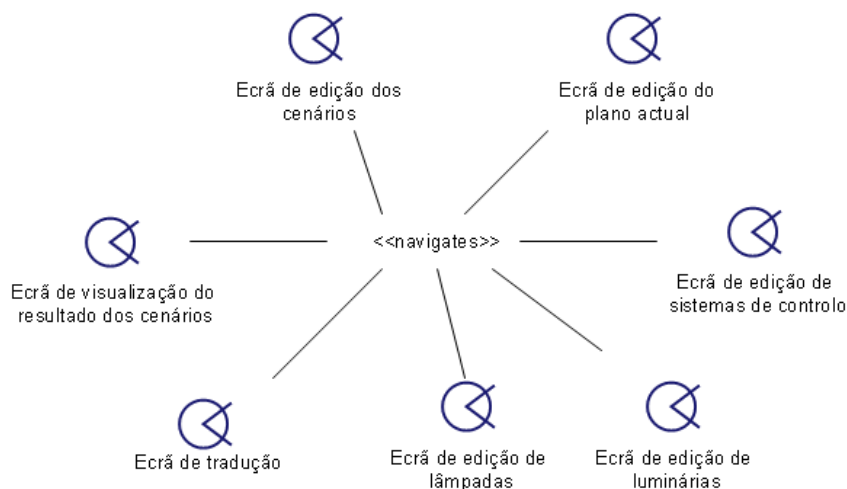


Figura 13. Mapa de navegação simplificado (sem sub-espacos de interacção)

A possibilidade de navegação fácil entre todas estas secções é essencial, e para tal, foi introduzido um menu vertical, do lado esquerdo do ecrã. Esta solução corresponde a um padrão de desenho de interfaces, por ser frequentemente utilizada num grande número de páginas/aplicações web. [32]

A razão principal da utilização deste padrão é o facto deste possibilitar um acesso imediatamente visível a todas as secções da aplicação (ao contrário de outros padrões de desenho de interfaces mais elaborados, em que parte das secções são visíveis apenas após o utilizador interagir com o menu), o que é possível devido à aplicação ter poucas secções. Contudo, o menu pode expandir caso sejam criados muitos cenários, o que pode causar um problema ao obrigar o utilizador a fazer *scroll* para aceder às últimas secções do menu – mas é de esperar que a aplicação seja utilizada em ecrãs com resolução alta, e nesse caso o utilizador poderá criar um número considerável de cenários antes que este problema ocorra.

Foi considerada inicialmente a hipótese de utilização do padrão “Breadcrumbs” [33], que consiste num pequeno espaço no topo da página, em que é mostrado todos os níveis de hierarquia do site até chegar ao nível seleccionado. Este cabeçalho foi posteriormente descartado, dado que a aplicação não deverá ter mais do que 2 níveis de secções e é facilmente perceptível ao utilizador em que secção se encontra, olhando para a secção que se encontra seleccionada no menu principal.

Para além do menu principal, acima deste encontra-se um pequeno espaço dedicado à autenticação do utilizador, que dinamicamente altera o seu aspecto dependendo de uma acção de *login* por parte do utilizador. Caso esta acção não tenha sido efectuada, o espaço permite que um utilizador insira os seus dados para se autenticar, e faz uma ligação à secção de registo de novos utilizadores. Caso o utilizador tenha efectuado o *login*, é-lhe apresentado o seu nome de utilizador como reconhecimento de que a sessão na aplicação se encontra activa, e uma ligação para terminar a sessão.

Este espaço também serve para mostrar uma mensagem de erro, caso a autenticação tenha sido mal sucedida; e fazer uma ligação à secção de controlo de utilizadores, caso o administrador do sistema tenha se autenticado.

5.1. SECÇÃO “TARIFÁRIO”

A secção “Tarifário” tem uma única área com conteúdo, com um funcionamento relativamente simples: o utilizador tem a opção de seleccionar entre utilizar uma tarifa única, que corresponde a um único valor de EUR/kWh a ser considerado pela simulação como o preço da energia em todos os casos; ou uma tarifa sazonal. Caso o utilizador selecione esta, as opções da tarifa sazonal têm uma estrutura hierárquica: a tarifa é dividida em períodos de meses, e estes por sua vez subdivididos em condições horárias.

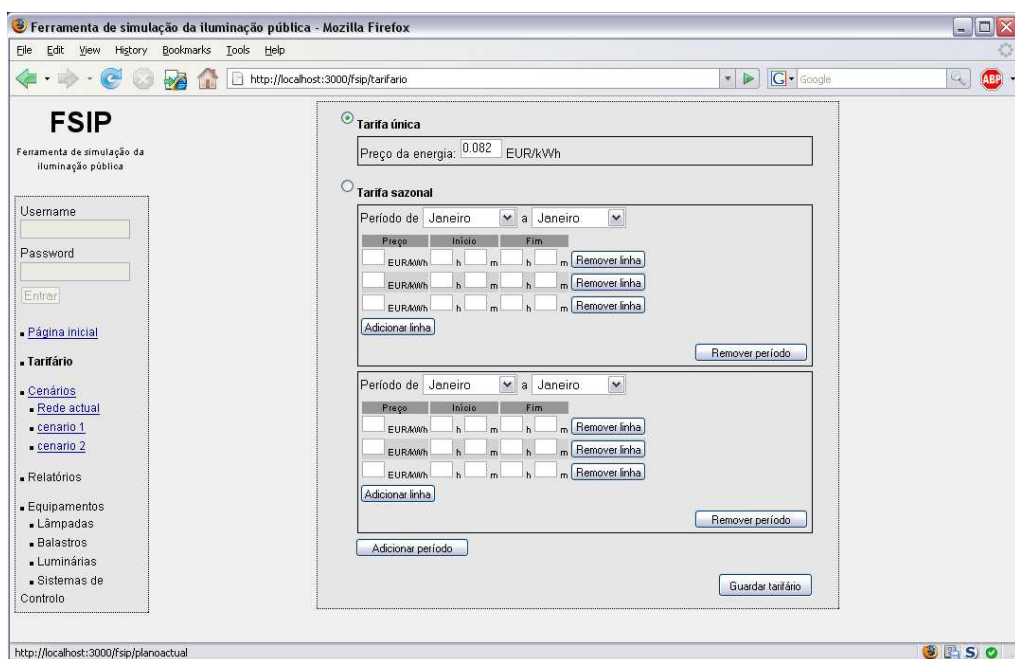


Figura 14. Secção "Tarifário" da aplicação

O utilizador tem a opção de adicionar e remover condições e períodos à tarifa sazonal, permitindo assim a flexibilidade necessária à aplicação para se adaptar a qualquer legislação existente na área na qual irá ser efectuada a simulação.

No final da página a opção “Guardar tarifário” guarda as alterações efectuadas através dum pedido Ajax à base de dados, respondendo posteriormente com uma mensagem com fundo verde de gravação efectuada com sucesso (ou com fundo vermelho caso ocorra um erro). Este sistema tem a vantagem de dar *feedback* ao utilizador da situação em que a aplicação se encontra.

5.2. SECÇÃO “CENÁRIOS”

Tal como a secção tarifário, a secção cenários tem para o utilizador uma estrutura simples, com apenas uma área com conteúdo. O primeiro cenário é um cenário “especial”, pois corresponde à rede actual de electricidade, com diferentes propriedades e que, ao contrário dos outros cenários, não pode ser apagada, nem o seu nome pode ser alterado – estes botões são desactivados sempre que a rede actual é seleccionada.

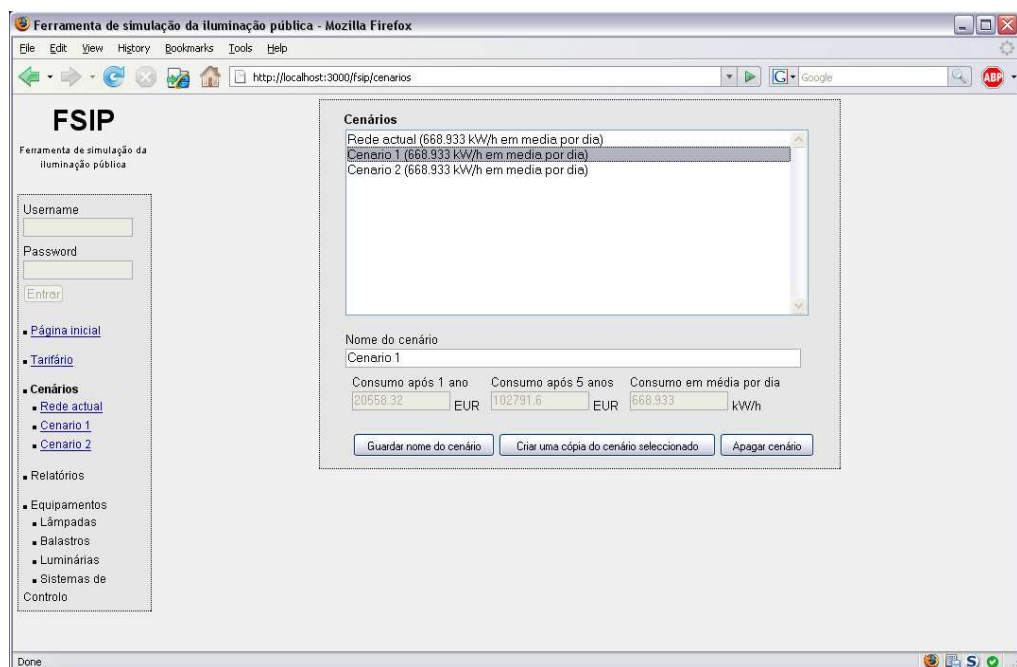


Figura 15. Secção "Cenários" da aplicação

A razão pela qual a rede actual foi considerada como parte dos cenários foi para permitir mais facilmente a criação de um novo cenário – um cenário não pode ser criado por si mesmo, tem de se basear ou na rede actual ou num outro cenário para efeitos de comparação. Sendo assim, a interface torna-se mais simples considerando a rede actual como um cenário, a criação de um cenário utiliza apenas um botão: “criar uma cópia do cenário seleccionado”.

Cada vez que os botões são pressionados, a página é recarregada. A razão deste comportamento é permitir o recarregamento do menu principal, que sofre alterações cada vez que o utilizador efectua alterações nesta interface.

Uma particularidade desta interface é limitar a edição apenas aos cenários pertencentes ao próprio utilizador, sendo que os cenários dos outros utilizadores não são listados, nem aparecem no menu. Desta forma, um utilizador ao criar um cenário pode ter a certeza que os outros utilizadores do sistema não irão interferir com o seu trabalho.

De futuro, uma funcionalidade que possivelmente poderá ser adicionada é a possibilidade de partilhar o acesso a um cenário a vários utilizadores específicos. Desta forma, seria possível a colaboração de várias pessoas na elaboração de um cenário, sem que se retire a possibilidade de assegurar que o trabalho também possa ser elaborado individualmente sem interferências, como na actual implementação da aplicação.

5.3. SUB-SECÇÃO “REDE ACTUAL”

Este espaço de interacção está subdividido em três áreas principais: o mapa, que ocupa a maior parte do espaço; a área de “resultados rápidos”, abaixo do mapa; e a área com todos os detalhes dos ramais, do lado direito do ecrã. Estas três áreas interagem todas entre si.

A interface também tem a particularidade de estar disposta sobre um código de três cores: laranja, a que corresponde o segmento seleccionado; amarelo, a que corresponde o ramal seleccionado (e todos os outros segmentos do ramal seleccionado); e branco, a que corresponde todos os outros segmentos e ramais.

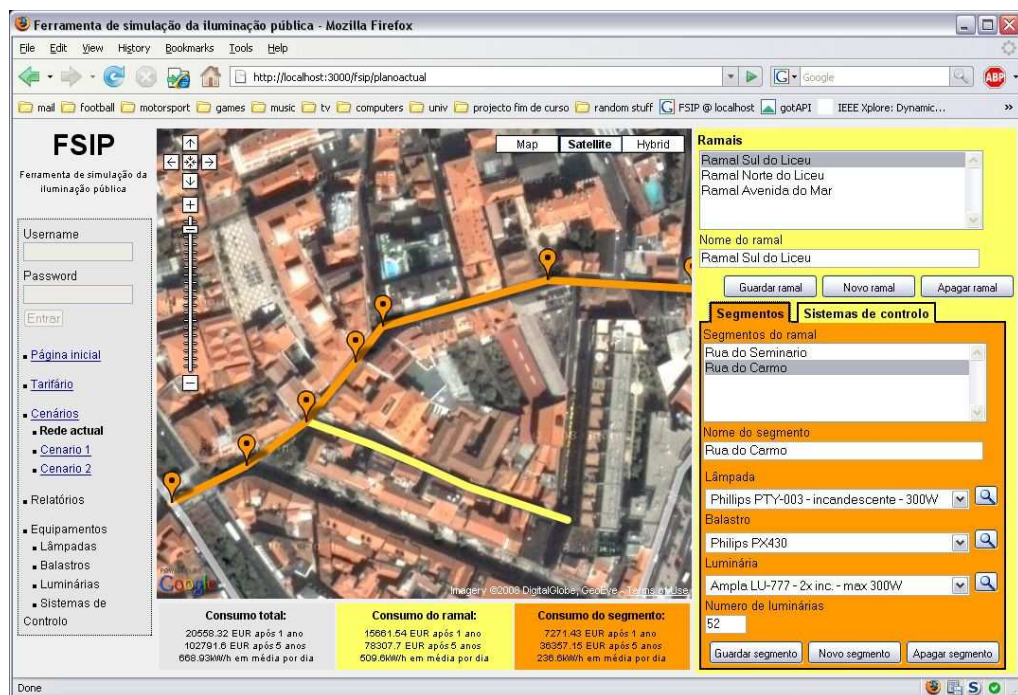


Figura 16. Sub-secção "Rede actual" da aplicação

A área dos detalhes no geral corresponde ao ramal seleccionado, e no topo contém uma lista com todos os ramais presentes no mapa. Esta lista pode ser alterada pelo utilizador, mudando o nome dos ramais, apagando ramais, ou criando novos ramais – acções que interagem com todo o resto da interface. Abaixo desta lista, encontra-se uma parte dividida em duas tabs: Segmentos e Sistemas de controlo.

A tab “Segmentos” lista todos os segmentos do ramal e permite ao utilizador tanto alterar detalhes do segmento seleccionado, como desenhar novos segmentos, ou

apagar novos segmentos. Esta tab tem cor de fundo laranja por corresponder a edição ao nível dos segmentos. Por sua vez, a tab “Sistemas de controlo” permite ao utilizador escolher o sistema de controlo de um ramal, e alterar as condições de funcionamento do mesmo. Como o sistema de controlo está associado ao ramal, esta tab é mantida a amarelo. A principal razão para a utilização de um sistema de tabs nesta parte da interface é permitir poupar espaço no ecrã, necessário para expandir o mapa o máximo possível. As tabs são agrupadas devido a ambas referirem-se a conceitos associados ao ramal (os segmentos fazem parte do ramal) – apesar de a tab Segmento poder editar em maior pormenor ao nível do segmento.

5.3.1. Tabulador “Segmento”

A edição de um segmento existente efectua-se da seguinte forma: primeiro o utilizador selecciona um segmento, através da lista de segmentos incluída na tab, ou clicando em cima de um segmento no mapa. Ao fazê-lo, a tab é actualizada com a informação do segmento: nome do segmento; lâmpada, balastro e luminária utilizados e número de luminárias que o segmento contém. Ao mesmo tempo o mapa é actualizado, seleccionando o segmento pretendido, colocando-o a laranja e activando os marcadores dos pontos do segmento.

O utilizador pode então alterar os detalhes na tab, e/ou alterar o desenho do segmento no mapa. Isto pode ser efectuado movendo os marcadores de pontos através de *drag-and-drop*, ou clicando com o botão direito sobre um ponto, o que abre um menu com as opções “Apagar ponto” e “Inserir ponto”.

Ao seleccionar a lâmpada para o segmento, pode ocorrer que existam demasiadas lâmpadas na lista e se torne difícil encontrar a lâmpada pretendida. Por essa razão, foi adicionado um botão com o ícone de uma lupa, que, ao ser clicado, abre uma janela embebida na página com opções de pesquisa por lâmpadas. As opções de pesquisa incluem todos os detalhes sobre lâmpadas disponíveis na base de dados, e, ao clicar no botão “Procurar”, todas as lâmpadas que correspondem aos elementos introduzidos nas opções de pesquisa, são apresentadas numa lista. O utilizador então pode escolher uma lâmpada, e todos os seus detalhes são apresentados numa área do lado direito da janela. Caso corresponda à lâmpada pretendida, pode clicar no botão “Escolher lâmpada seleccionada” e a janela será fechada, actualizando a tab com a lâmpada escolhida. A janela pode ser movida, para permitir que o utilizador possa ao mesmo tempo visualizar os elementos da página, caso assim pretenda.

A mesma funcionalidade está disponível para a pesquisa de balastros e de luminárias.

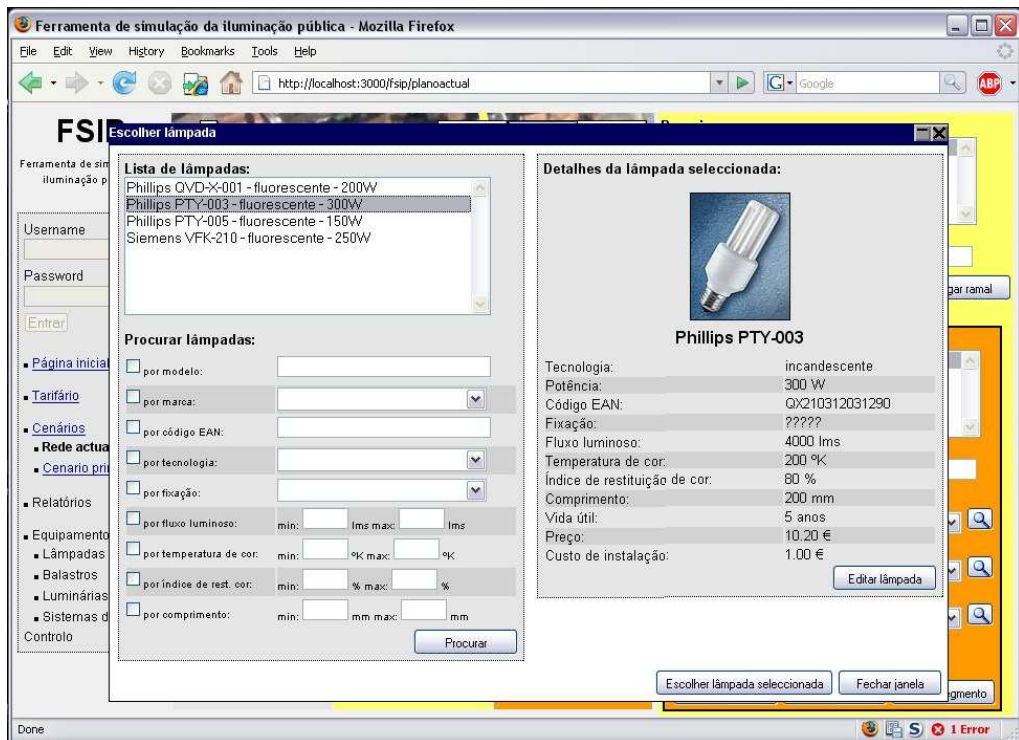


Figura 17. Janela de pesquisa por lâmpadas

No final, as alterações sobre a tab podem ser guardadas pelo botão “Guardar segmento”, que dá origem a uma mensagem ao utilizador com o *feedback* da acção de gravação na base de dados: verde caso seja bem sucedida, vermelho caso ocorra um erro. As alterações no mapa têm efeito permanente na base de dados sem necessidade de clicar no botão “Guardar segmento” – optou-se por esta alternativa dado a haver uma tendência na interacção com o mapa para ajustar o desenho de vários segmentos consecutivamente, e caso as alterações não fossem permanentes iria obrigar o utilizador a andar sempre a alternar a interacção com o mapa e com a tab.

Para criar um segmento, o utilizador tem primeiro que carregar em “Novo segmento”. Ao fazê-lo, os elementos da tab são reinicializados para os seus valores predefinidos, e aparece uma mensagem para o utilizador: “Clique no mapa para criar pontos do novo segmento”. A partir daqui, todas as acções de único clique (sem movimento) sobre o mapa criam um marcador de ponto e o segmento é desenhado pela ordem em que os cliques são efectuados. Note-se que o mapa tem todas as propriedades do Google Maps, ou seja, certas formas de interacção já estão reservadas, como o duplo clique, que aumenta o nível de zoom; ou um movimento em que o utilizador clique e arraste sobre o mapa, que move a área mostrada na direcção do movimento. Por esta razão foi escolhido a acção de único clique para a criação de um ponto. No final, quando o segmento estiver desenhado e todos os seus detalhes introduzidos, o botão “Guardar segmento” guarda o segmento na base de dados.

Para apagar um segmento, basta ao utilizador seleccionar o segmento pretendido, e clicar em “Apagar segmento”.

No final de cada uma destas acções (criação, gravação e apagamento de um segmento) a área de “resultados rápidos” é actualizada com o cálculo do consumo de energia.

5.3.2. Tabulador “Sistemas de controlo”

A tab de sistemas de controlo contém apenas a escolha de sistema de controlo para o ramal (através de uma lista e de uma janela de pesquisa de sistemas de controlo semelhante às referidas anteriormente para a lâmpada, balastro e luminária na *tab* Segmento), e as suas condições de funcionamento. Caso o sistema de controlo escolhido seja do tipo relógio, as linhas com as condições de funcionamento contêm uma *checkbox* ligado/desligado, um espaço para o início do intervalo de tempo e outro espaço para o fim do intervalo de tempo. Pretende-se com isto que o utilizador defina um intervalo de tempo no qual a electricidade é ligada no ramal. Caso o sistema de controlo escolhido seja do tipo regulador de fluxo, em vez de uma *checkbox* ligado/desligado, cada linha contém uma caixa de texto em que o utilizador poderá definir a percentagem de potência que o sistema de controlo aplica ao ramal. As linhas podem ser removidas ou adicionadas através de botões incluídos na interface.

Ao clicar no botão “Guardar”, todas as alterações são guardadas e a área de “resultados rápidos” é actualizada.

5.4. SUB-SECÇÃO “UM CENÁRIO”

Para o utilizador, as diferenças na interface entre a secção “rede actual” e a secção referente a um cenário não são muitas, apesar de haver diferenças substanciais no código que é executado (como se mostrou no sub-capítulo 4.5, que refere a existência de ficheiros separados para estes dois casos). A principal diferença é a remoção de funcionalidades: num cenário não é possível alterar o nome dos segmentos, nem o número de luminárias a que um segmento se refere, nem o desenho dos segmentos. Também não é possível adicionar novos segmentos nem apagar segmentos, ou seja, apenas é permitido a alteração dos detalhes do segmento e o funcionamento do sistema de controlo do ramal.

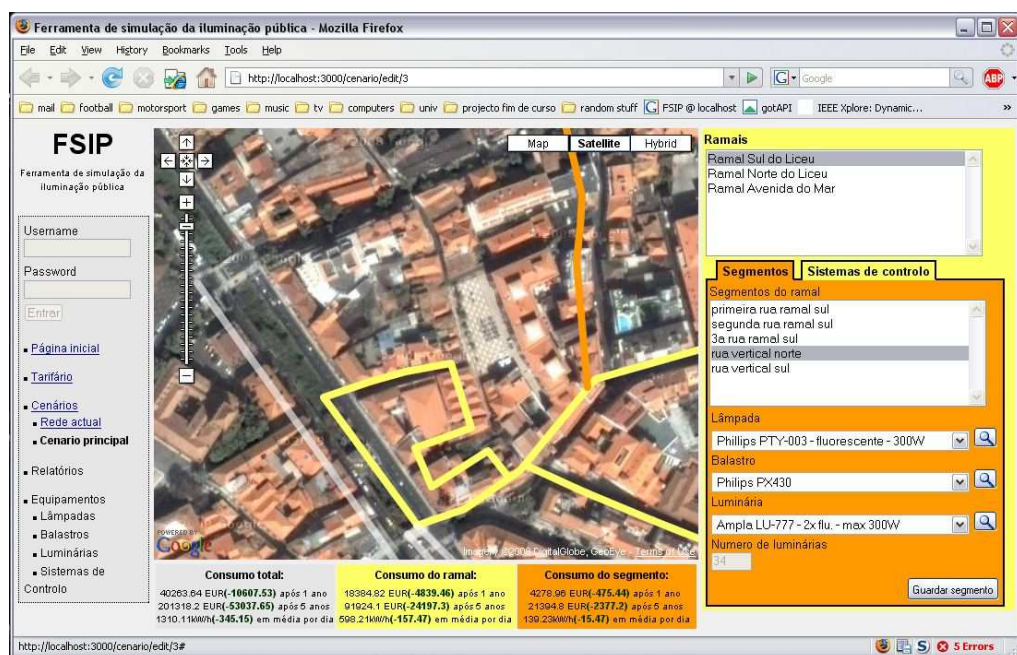


Figura 18. Sub-secção detalhando "um cenário" da aplicação

A principal funcionalidade adicional visível ao utilizador é a amostragem de um comparativo com a rede actual na secção “resultados rápidos” – caso o consumo de energia seja menor que na rede actual, aparecem valores a verde com a diferença de consumo; caso seja maior, os valores aparecem a vermelho. Considerando que a secção de relatórios não se encontra actualmente implementada, esta é a principal funcionalidade de aplicação actualmente, permitindo ao utilizador fazer a optimização de consumos de energia com base nestes resultados.

5.5. SECÇÃO “REGISTO DE UTILIZADORES”

Este espaço de interacção surgiu numa fase tardia do processo de desenvolvimento, com o objectivo de possibilitar o registo no sistema de um novo utilizador que pretenda utilizar a aplicação. Desta forma, é contornado o inconveniente de alterar directamente a tabela “Utilizador” da base de dados, e são aplicadas medidas de segurança para a encriptação das palavras-passes de utilizador, evitando assim o acesso fácil a estas.



Figura 19. Secção de registo de novos utilizadores na aplicação

O utilizador para se registar terá que introduzir o nome pretendido para a aplicação, a sua palavra-passe (de forma duplicada, para evitar erros), e o seu endereço de e-mail como eventual forma de contacto pelo administrador do sistema. Ao criar a conta, a página será recarregada e uma mensagem de confirmação ou de erro irá ser apresentada, dando feedback ao utilizador sobre se a sua conta está ou não já disponível.

Por defeito, é dado ao utilizador permissões apenas para ver a rede actual – partindo do princípio que o administrador do sistema poderá depois alterar o seu nível, caso seja um utilizador de confiança. Todo este funcionamento do sistema de autenticação pode ser facilmente alterado a nível de programação, dependendo das necessidades.

5.6. SECÇÃO “CONTROLO DE UTILIZADORES”

Este espaço de interacção destina-se exclusivamente ao administrador do sistema, que terá a seu cargo a definição de níveis de permissão para os utilizadores, sendo que os diversos níveis de permissão dão acesso (ou negam acesso) a todas as secções da aplicação.

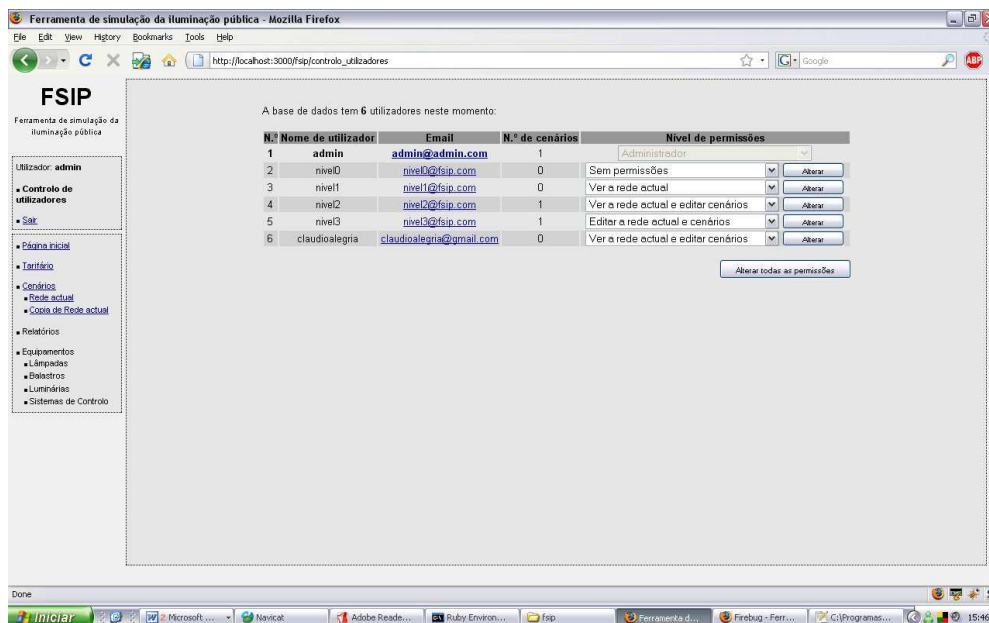


Figura 20. Secção de controlo de utilizadores por parte do administrador do sistema

O administrador neste espaço de interacção pode consultar uma lista com todos os utilizadores registados no sistema, com o seu nome, endereço de correio electrónico, número de cenários criados, e o seu nível de permissões. É-lhe dada a opção de alterar as permissões individualmente para cada utilizador, ou globalmente, gravando de uma só vez todas as permissões alteradas.

Os níveis de permissões existentes são os seguintes:

- **Administrador:** acesso a todas as secções do sistema.
- **Editar a rede actual e cenários:** acesso a todas as secções excepto a esta mesma secção de controlo de utilizadores.
- **Ver a rede actual e editar cenários:** acesso à edição de cenários, mas impossibilitado de editar a rede actual de electricidade. Note-se que para este nível de permissões,

seria necessário a implementação de uma modificação do espaço de interacção da rede actual, que permitisse apenas a visualização da mesma, sem a possibilidade de efectuar quaisquer alterações. À data de entrega deste relatório, esta modificação não se encontra implementada. Este nível de permissões também nega o acesso à alteração do tarifário (que se assume ser uma tarefa a realizar pelas mesmas pessoas que editam a rede actual).

- **Ver a rede actual:** acesso apenas à secção de visualização da rede actual, que como foi referido no ponto anterior, não se encontra ainda disponível.

- **Sem permissões:** acesso negado a todas as secções do sistema, exceptuando a página inicial.

Alguns destes níveis de permissões podem deixar de ser necessários, dependendo da forma como se pretenda publicar a aplicação. Por exemplo, caso se pretenda utilizar a aplicação apenas numa intranet em que todos os utilizadores tenham qualquer tipo de responsabilidades sobre a rede de electricidade, todos os utilizadores terão de ter permissões, no mínimo, para a edição de cenários. No entanto, caso se pretenda utilizar a aplicação para a apresentação ao público geral do estado da rede actual de electricidade, através da Internet, o nível de permissões oferecido ao utilizador comum deverá ser “Ver a rede actual”.

Da mesma forma, o nível “Sem permissões” faz apenas sentido caso se pretenda que sejam associadas permissões a um utilizador manualmente pelo administrador, e não imediatamente após o registo; ou no caso de querer retirar permissões a um utilizador que faça uso indevido da aplicação. No global, este sistema de níveis de permissões oferecido ao administrador pretende possibilitar um uso flexível da aplicação, sem que o administrador tenha necessariamente de utilizar todas estas opções.

É necessário referir que, à data de entrega deste relatório, este ecrã não se encontrava totalmente funcional, fazendo o carregamento dos dados para a página correctamente, mas sem que os botões alterassem na base de dados os níveis de permissões dos utilizadores.

Além disso, dado que este espaço de interacção não resultou directamente da análise de requisitos e dos protótipos abstractos canónicos, é possível que a interface possa ser melhorada. Por exemplo, poderia ser efectuada a distribuição da lista de utilizadores por várias páginas e implementado um sistema de pesquisa de utilizadores, caso haja um grande número de utilizadores na base de dados.

6. CONCLUSÕES

O desenvolvimento efectuado sobre esta aplicação permitiu que se estabelecesse uma base sólida para uma possível ferramenta de simulação de cenários de iluminação. A aplicação actual pode ser utilizada para o desenho e representação geográfica de redes de electricidade, e para a obtenção de resultados de cálculos sobre a simulação de cenários, facilitando a escolha de tecnologias para ramais de electricidade, que diminuam os custos de energia numa rede de iluminação pública.

O trabalho realizado, demonstrou que é possível ter uma interface simples para este efeito, e que tal aplicação pode ser desenvolvida sobre software Open Source e plataformas disponíveis livremente ao público sobre aplicações proprietárias (neste caso o Google Maps). Desta forma, foram minimizados grandemente os custos de desenvolvimento.

A aplicação desenvolvida no entanto tem os seus problemas, sendo que o principal problema é de desempenho. Mesmo com poucos segmentos sobre o mapa, a aplicação tem tendência a ocupar grandes quantidades de memória e a aumentar o tempo de reacção do browser – uma tendência que se agrava quanto maior o tempo que o utilizador trabalhar sobre a aplicação. As causas deste comportamento não são completamente conhecidas, mas poderá ser possível minimizá-lo futuramente utilizando diferentes técnicas de optimização da forma como a informação é amostrada ao utilizador (por exemplo, não carregando todos os segmentos e marcadores em memória mas apenas aqueles referentes à área do mapa actualmente visível). No entanto, é também possível que este problema seja de qualquer forma difícil de contornar e intrínseco à utilização extensiva de Javascript, necessário para controlar a API do Google Maps.

Caso seja continuado o processo de desenvolvimento desta aplicação no futuro, deverá ser possível transformá-la numa aplicação suficientemente completa para ser utilizada no dia-a-dia com maior fiabilidade e com maiores aplicações. A principal funcionalidade em falta é a possibilidade de poder alargar o cálculo de consumo de energia a outros factores, tais como o custo de instalação dos novos equipamentos a

serem adquiridos segundo um cenário, e associada a esta funcionalidade a geração de relatórios que permitam analisar em mais pormenor os benefícios de um cenário versus a rede actual de electricidade, ou outros cenários.

Uma funcionalidade que aumentaria bastante o valor da aplicação seria a implementação das janelas de pesquisa, que apesar de terem a interface já disponível, actualmente não têm qualquer comportamento dinâmico implementado. Para completar a aplicação, as secções relativas à introdução de equipamentos na base de dados, que actualmente não estão implementadas, também serão necessárias, e também são necessários alguns acabamentos no módulo de autenticação de utilizadores.

Apesar de o projecto ao nível da União Europeia no qual esta aplicação se enquadrava, não ter sido aprovado, a aplicação poderá ser posta a funcionar a nível regional. Este factor significa que futuros desenvolvimentos poderão levar ao sentido de especializar a aplicação, como por exemplo possibilitar a importação de dados sobre a rede de electricidade a partir de um sistema específico, em vez de tentar aplicar ideias genéricas que possam se adequar a vários países, e a sistemas diferentes, como foi considerado na análise inicial de requisitos.

ANEXO I: DOCUMENTO DE REQUISITOS DA APLICAÇÃO

Lista de requisitos funcionais

R01: O sistema deverá armazenar a informação relevante sobre a iluminação já existente numa determinada área geográfica.

Descrição: Para a área geográfica a analisar, o sistema deverá armazenar a informação sobre os ramais de iluminação, os segmentos em que estes se subdividem, e o tipo e quantidade de lâmpadas, luminárias, reguladores e sensores utilizados em cada segmento.

Prioridade: Muito alta

R02: O sistema deverá permitir a simulação de diversos cenários possíveis para o melhoramento da iluminação pública na área a analisar.

Descrição: Nestes cenários o utilizador terá a opção de alterar manualmente as condições existentes de iluminação pública na área, simulando o que aconteceria se fossem implementadas na vida real as alterações.

Prioridade: Muito alta

R03: O sistema deverá fazer o cálculo do consumo de energia, dos custos, e do impacto ambiental de um determinado cenário.

Descrição: Estes resultados são o principal objectivo da simulação, e deverão ser apresentados tendo em conta um período de vários anos após a implementação real do cenário.

Prioridade: Muito alta

R04: O sistema deverá poder sugerir ao utilizador qual o tipo de lâmpada ideal a utilizar para um determinado segmento, dependendo dum conjunto de restrições à escolha do utilizador.

Descrição: Esta opção deverá ser uma alternativa à definição manual do tipo de lâmpadas por parte do utilizador, fazendo assim com que a aplicação consiga otimizar o tipo de lâmpadas a utilizar, dependendo do critério que o utilizador pretenda otimizar.

Prioridade: Muito alta

R05: O sistema deverá permitir fazer a comparação entre diversos cenários.

Descrição: A comparação entre diversos cenários deverá ajudar o utilizador a tomar uma decisão sobre qual o cenário a escolher para implementar na vida real.

Prioridade: Média

R06: O sistema deverá armazenar um catálogo de toda informação relevante à simulação sobre cada equipamento.

Descrição: Todos os equipamentos que podem ser utilizados na simulação (lâmpadas, luminárias, reguladores, sensores) deverão ser catalogados pelo programa, com toda a informação relevante à simulação para os cálculos necessários, e para a correcta identificação do equipamento em questão por parte do utilizador.

Prioridade: Muito alta

R07: O sistema deverá implementar uma visualização geográfica dos ramais de iluminação na área a analisar.

Descrição: Esta visualização deverá ser interactiva e permitir ao utilizador uma mais fácil compreensão da área geográfica sobre a qual definirá cenários. Este requisito é importante também do ponto de vista do desafio tecnológico que representa, no âmbito deste trabalho se tratar dum projecto de fim de curso.

Prioridade: Alta

R08: O sistema deverá permitir definir um tarifário de electricidade comum para toda a área a analisar.

Descrição: Este tarifário de electricidade é importante para o cálculo dos custos ao longo do tempo.

Prioridade: Alta

R09: O sistema deverá permitir definir um horário de funcionamento e de intensidade de luz de cada segmento.

Descrição: Para cada segmento, o sistema deverá verificar se existe equipamento que permita desligar e ligar a electricidade num determinado horário, e/ou que permita gerir a intensidade da luz. Caso exista esse equipamento, o sistema deverá permitir que o utilizador defina o horário.

Prioridade: Média

R10: O sistema deverá estar preparado para ser disponibilizado em vários idiomas.

Descrição: Dado que é possível que o sistema venha a ser implementado noutras regiões de outros países, é necessário que este esteja preparado para que a tradução para novos idiomas seja fácil de implementar.

Prioridade: Baixa

R11: O sistema deverá permitir a importação de dados sobre a iluminação já existente na área, a partir de sistemas de informação geográfica (GIS).

Descrição: Em determinados casos, poderá já existir informação armazenada em GIS sobre a iluminação já existente, em particular o posicionamento geográfico dos ramais/segmentos. Neste caso, torna-se importante poder reaproveitar esses dados automaticamente, em vez de forçar o utilizador a ter de introduzi-los manualmente. Este requisito até certo ponto também representa um desafio tecnológico.

Prioridade: Média

Lista de requisitos não-funcionais

R12: O sistema deverá ser fácil de operar e aprender.

Descrição: Na maior parte dos casos, os estudos efectuados com este sistema serão feitos por pessoas que não irão dar um uso continuado ao sistema. Sendo assim, torna-se muito importante que a curva de aprendizagem da interacção com o sistema não seja acentuada.

Prioridade: Muito alta

R13: O sistema deverá ser eficiente e eficaz.

Descrição: O sistema tem que devolver resultados correctos, e num intervalo de tempo minimamente aceitável. Deverá também ter um comportamento estável e previsível ao utilizador.

Prioridade: Média

R14: O sistema deverá estar pronto para funcionar em diversos ambientes

Descrição: Como este sistema deverá ser implementado em várias localizações distintas, as tecnologias disponíveis tais como o hardware e o sistema operativo podem ser diferentes em cada localização, e o sistema deverá estar minimamente preparado para estas diferentes condições.

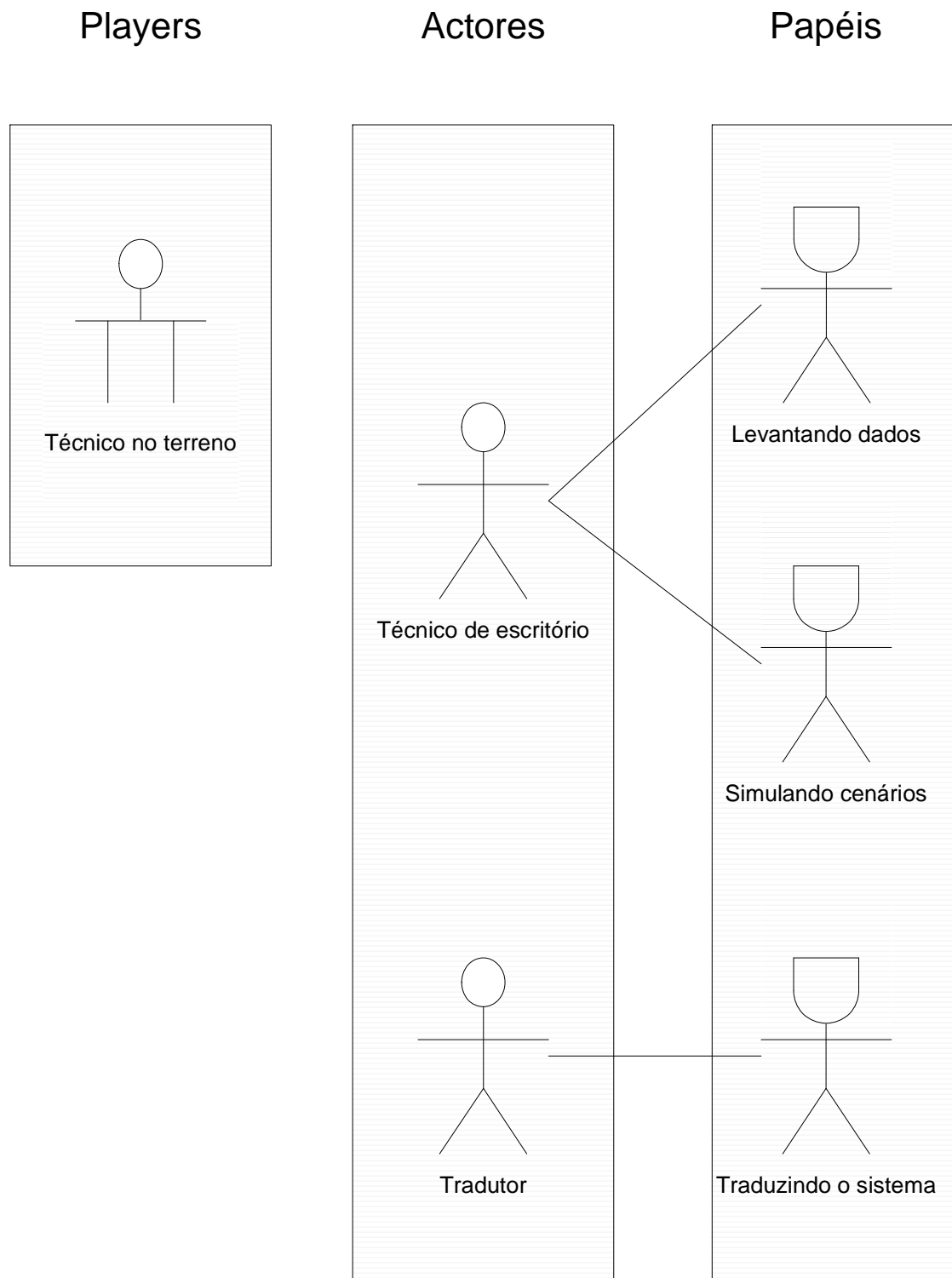
Prioridade: Baixa

R15: O sistema deverá ser desenvolvido com o intuito de proporcionar expansões futuras

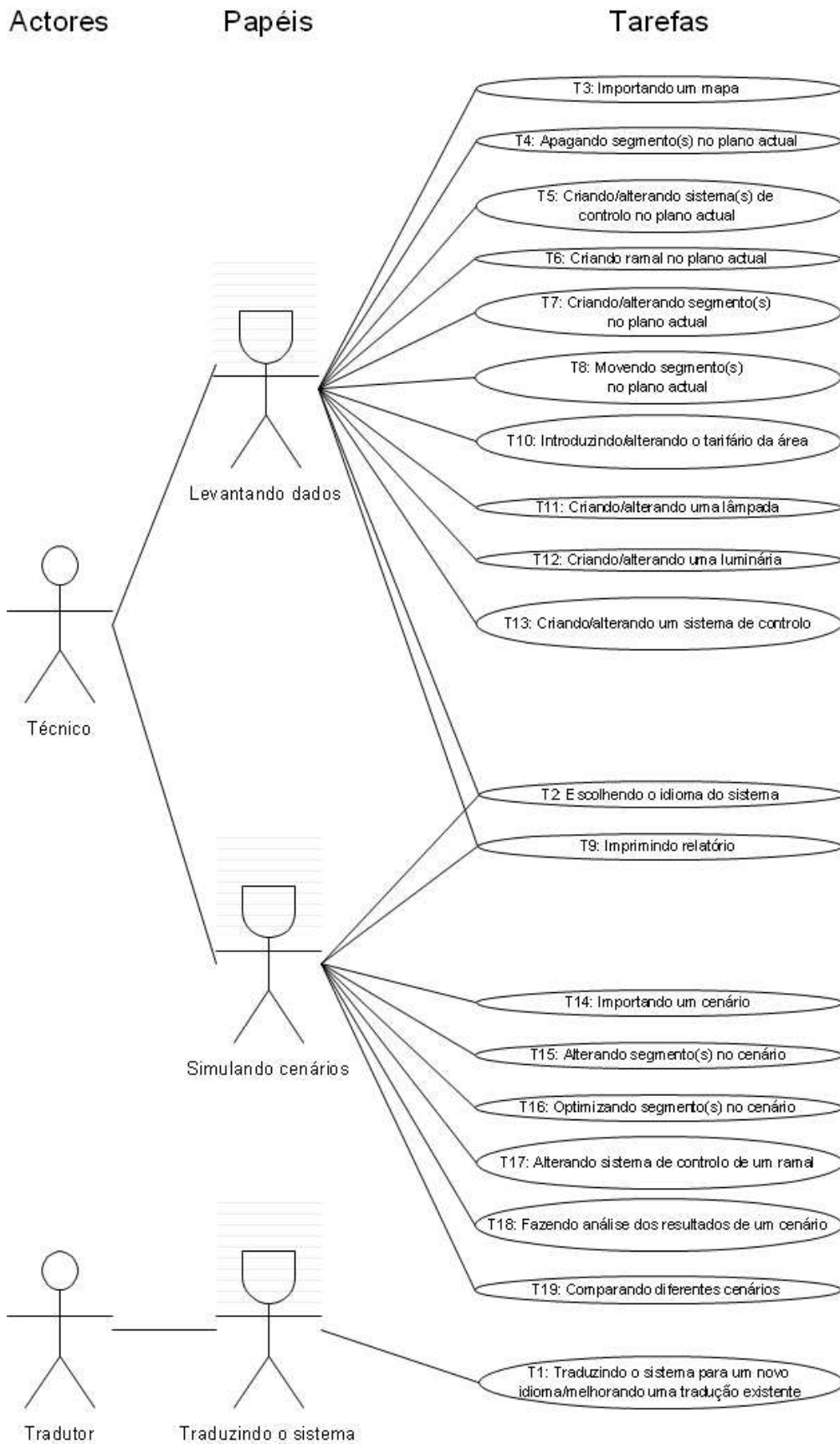
Descrição: A ideia principal é desenvolver uma base que sirva para a simulação do consumo de energia na iluminação pública, mas pode ser que no futuro esta base seja alargada para outros âmbitos, como a simulação de outros tipos de consumo de energia.

Prioridade: Alta

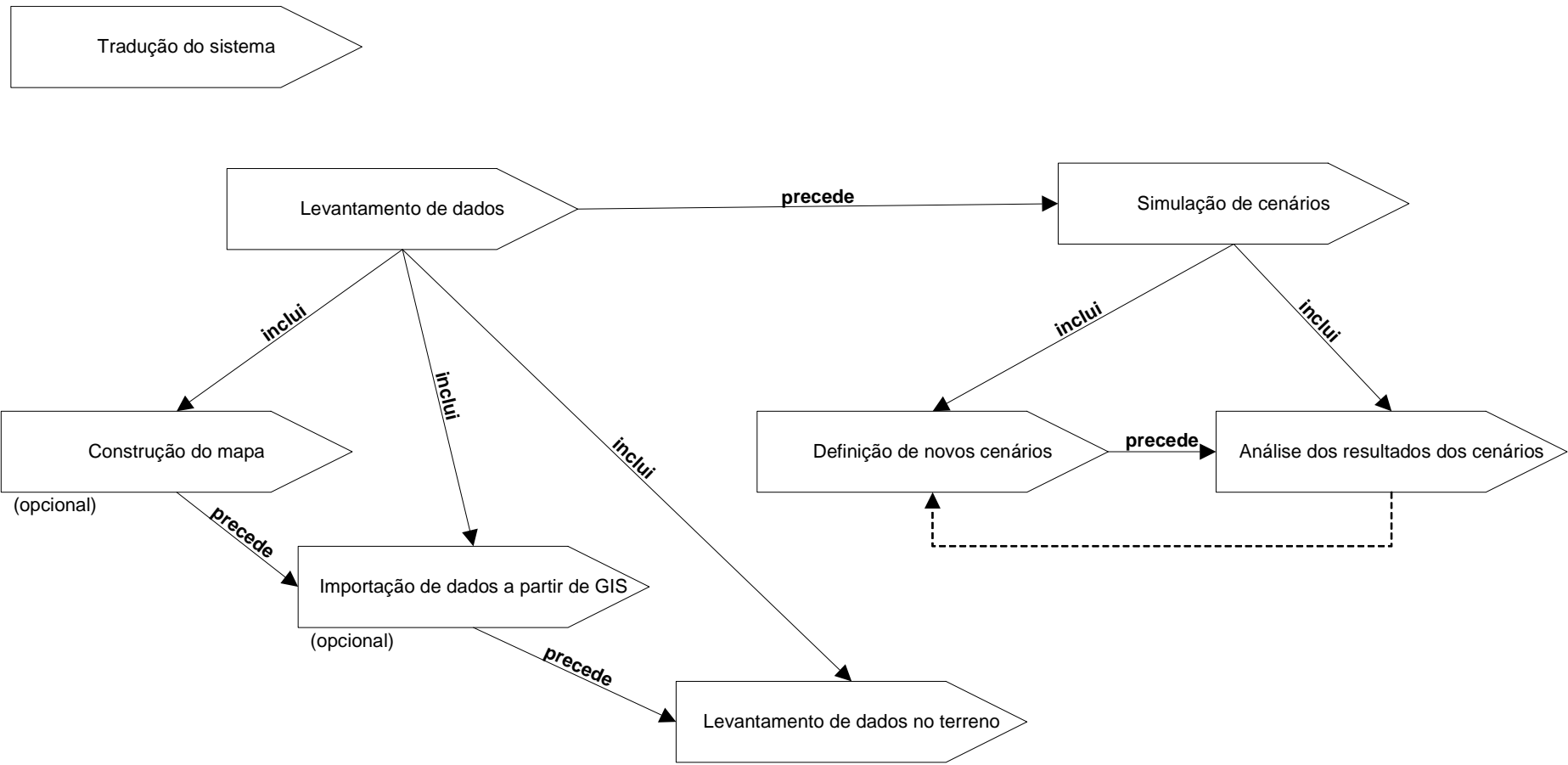
Mapa de desempenho



Mapa de actores e tarefas



Mapa de actividades



Perfis das actividades

Actividade: Levantamento de dados

Propósito: levantar todos os dados necessários à fase posterior de simulação, incluindo a localização e o tipo de equipamentos existentes no terreno, associados a toda a informação geográfica que seja possível obter.

Tempo e Lugar: possivelmente um trabalho demorado, dependendo do tamanho da área a analisar. Alguma informação provavelmente deverá ser recolhida no terreno, mas será sempre necessário agregar a informação no escritório.

Participantes: possivelmente um técnico/funcionário com conhecimentos informáticos suficientes para saber importar e agregar a informação já existente em sistemas informáticos, e um técnico/funcionário ou grupo de técnicos para fazer o levantamento dos equipamentos no terreno. Este(s) técnico(s) deverão ter conhecimentos na área da electricidade suficientes para poder reconhecer as tecnologias existentes. No entanto, pode ser irrealista esperar que a entidade que irá necessitar do estudo sobre a iluminação pública (em princípio a Câmara Municipal), tenha já técnicos e funcionários especificamente para estas funções, portanto é possível que destaquem pessoal a partir de outras áreas, sem que estes tenham os conhecimentos necessários.

Desempenho: devido a este trabalho ser feito de base apenas uma única vez, excepto eventuais pequenas actualizações conforme forem instalados novos equipamentos, este levantamento não deverá ser feito sobre grande pressão de horários – a não ser que haja alguma decisão administrativa com alta prioridade, pendente do estudo da iluminação pública na área. No entanto, como é possível que os técnicos/funcionários encarregados sejam destacados a partir de outras áreas, estes podem ter outros compromissos/tarefas a fazer.

Actividade: Construção do mapa

Propósito: importar/construir um mapa baseado em ortofotos, fotos de satélite, ou um mapa com informação vectorial da área a analisar, caso o tipo de mapa pretendido não esteja já disponível por defeito no sistema, ou o mapa disponível no sistema não tenha a qualidade mínima necessária para o trabalho a efectuar. Caso isso já aconteça, esta actividade não é necessária.

Tempo e Lugar: usualmente esta actividade deverá ocorrer uma única vez no escritório, antes da importação da informação relativa à localização geográfica dos equipamentos, embora seja possível que posteriormente se queira introduzir novos tipos de mapas no sistema.

Participantes: o técnico/funcionário com maiores conhecimentos informáticos, possivelmente com a ajuda da pessoa ou entidade que lhes fornecerá os mapas.

Desempenho: ao se tratar de uma tarefa única, a rapidez da construção do mapa não é uma grande prioridade – no entanto será sempre preferível a importação de um mapa completo de uma vez, ou peças agregadas a informação geográfica, do que a importação de peças separadas que tenham de ser encaixadas manualmente, que pode ser um processo muito moroso.

Actividade: Importação de dados a partir de GIS

Propósito: importar para o sistema a localização de parte ou todos os ramais e equipamentos existentes no terreno, a partir da informação que já exista armazenada noutra sistema, que pode ou não ser propriedade da entidade que irá realizar o estudo sobre a iluminação pública.

Tempo e Lugar: tal como a construção do mapa, esta actividade deverá ocorrer uma única vez no escritório, no entanto é possível que aconteça uma situação em que não seja possível importar a informação toda de uma vez, ou que seja mais conveniente fazê-lo aos poucos.

Participantes: o técnico/funcionário com maiores conhecimentos informáticos, possivelmente com a ajuda da pessoa ou entidade que lhes fornecerá os dados a partir de GIS.

Desempenho: esta actividade tem a condicionante de os dados originais poderem ter informação a mais da que é necessária para o estudo da iluminação pública, portanto o utilizador poderá ter a tarefa de filtrar qual a informação a utilizar, e qual a deitar fora. Os dados originais podem estar em muitos formatos diferentes, que o sistema poderá não prever completamente.

Actividade: Levantamento de dados no terreno

Propósito: percorrer a área a estudar, e levantar todos os dados dos ramais e equipamentos utilizados que não estejam já introduzidos no sistema.

Tempo e Lugar: nas ruas da área a estudar. É conveniente que o levantamento seja feito recorrendo a algum tipo de equipamento informático portátil, caso contrário o levantamento terá de ser feito recorrendo a anotações em papel, que depois têm de ser compiladas no escritório. Dependendo do tamanho da área, toda esta actividade poderá demorar vários dias.

Participantes: o(s) técnico/funcionário(s) com maiores conhecimentos na área de electricidade, sendo que esta actividade pode necessitar mesmo de alguém com um nível de conhecimento avançado.

Desempenho: Torna-se importante que esta actividade seja feita de modo rápido, e que a interacção com o sistema não prejudique esta rapidez, porque destacar uma pessoa durante vários dias para esta actividade significa retirá-la de outra actividade onde possa ser útil. É preciso também ter em atenção para que as pessoas estejam conscientes de quais as informações que já se encontram do sistema, para não desempenharem o trabalho inútil de voltar a percorrer ruas já reconhecidas, a não ser que se pretenda expressamente confirmar se a informação existente é correcta.

Actividade: Simulação de cenários

Propósito: fazer o estudo de vários cenários possíveis de modificações à iluminação pública, e escolher de entre eles a solução ideal para otimizar o consumo de energia e/ou custo e/ou impacto no meio ambiente da rede de iluminação pública na área.

Tempo e Lugar: esta actividade deverá ocorrer em várias sessões no escritório, em que a pessoa encarregada de efectuar o estudo irá aperfeiçoando os cenários existentes ou experimentando com a criação de novos cenários, até obter o resultado pretendido. O mais provável é que esta actividade seja feita em várias sessões separadas, pois o utilizador poderá ter a tendência de fazer esboços de cenários temporários, pedir opiniões a outras pessoas, e então voltar à actividade.

Participantes: Um técnico que terá de ter conhecimentos algo avançados na área da electricidade (e conhecimentos básicos de informática de modo a poder utilizar o sistema), e que provavelmente terá de comunicar os seus resultados obtidos a pessoas com maior poder de decisão, possivelmente com cargos políticos.

Desempenho: Nesta actividade algo complexa o sistema terá de permitir que o utilizador consiga definir todos os aspectos que ele pretenda alterar no cenário, assim como sugerir a utilização de equipamentos que, dado os objectivos pretendidos pelo utilizador, possam otimizar esses mesmo objectivos. O sistema terá também de permitir a análise flexível e rapidamente perceptível dos resultados, de modo a ajudar da melhor forma a decisão de qual o melhor cenário a implementar. Apesar de a definição de cenários e a análise de resultados serem considerados duas actividades separadas, elas estão ligadas, e poderá ser útil ter pequenos indicadores de como é que os resultados são afectados, à medida que se definem os cenários.

Actividade: Definição de novos cenários

Propósito: definir um cenário com uma determinada particularidade, ou abordagem à iluminação da área.

Tempo e Lugar: esta actividade deverá ocorrer repetidamente no escritório, utilizando o sistema. É esperado que o utilizador regresse a esta actividade após a análise dos resultados, de forma a tentar melhorar o cenário obtido anteriormente.

Participantes: Um técnico que terá de ter conhecimentos algo avançados na área da electricidade (e conhecimentos básicos de informática de modo a poder utilizar o sistema), e que provavelmente terá de comunicar os seus resultados obtidos a pessoas com maior poder de decisão, possivelmente com cargos políticos.

Desempenho: Outra das preocupações para além de dar ao utilizador a facilidade de alterar todos os detalhes em que consiste um cenário, é também fazer com que os cenários sejam facilmente identificados entre si, realçando as diferenças. É importante que haja ferramentas que possam acelerar o processo de definição de um cenário, como a alteração de detalhes em vários ramos semelhantes ao mesmo tempo, já que a alteração de detalhes ramal-a-ramal, apesar de muitas vezes ser necessário, pode ser muito demorada.

Actividade: Análise dos resultados dos cenários

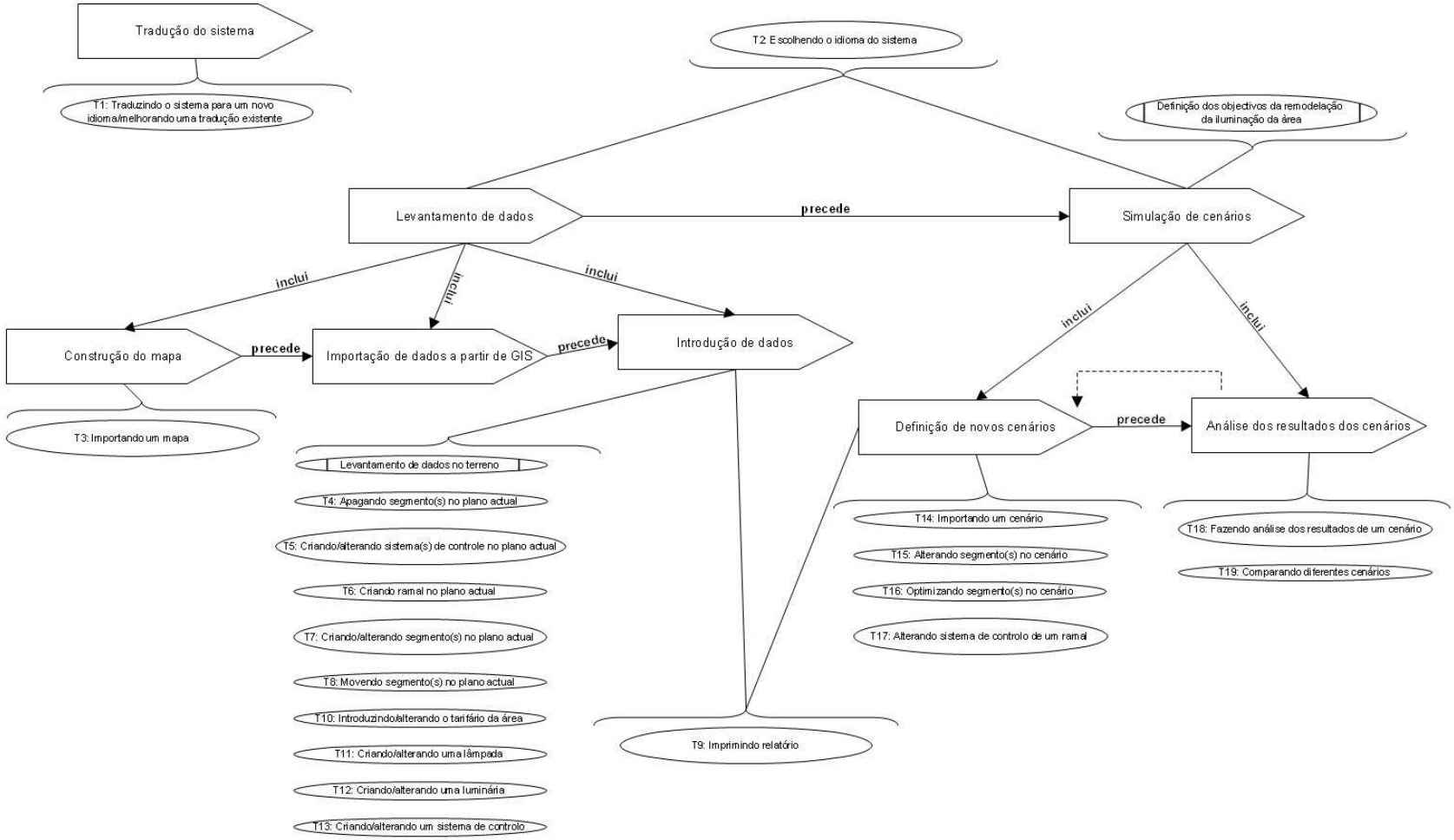
Propósito: analisar e comparar resultados dos diferentes cenários obtidos, tanto entre si, como com a actual iluminação pública existente na área.

Tempo e Lugar: esta actividade deverá ocorrer repetidamente no escritório, utilizando o sistema, sendo interligada com a definição dos detalhes de cada cenário.

Participantes: Um técnico que terá de ter conhecimentos algo avançados na área da electricidade (e conhecimentos básicos de informática de modo a poder utilizar o sistema), e que provavelmente terá de comunicar os seus resultados obtidos a pessoas com maior poder de decisão, possivelmente com cargos políticos. Note-se que esta actividade em particular poderá ter maior envolvimento destas pessoas, que deverão basear as suas decisões nos relatórios/conclusões que resultam desta actividade.

Desempenho: O sistema deverá tentar apontar ao utilizador os principais dados relevantes na comparação entre vários cenários, sem negar a possibilidade de o utilizador verificar todas as diferenças de forma detalhada. Os resultados deverão ser apresentados de forma bastante elucidativa, recorrendo sempre que possível a representações gráficas, permitindo também a geração de relatórios extensos.

Mapa de actividades e tarefas



Descrição das tarefas

T1: Traduzindo o sistema para um novo idioma/modificando um idioma existente

Intenções do utilizador	Responsabilidades do sistema
	1. Fornecer informação a traduzir
2. Traduzir informação	
2.1 Inserir nome do idioma	
2.2 Traduzir linhas*	
	3. Gravar as alterações

T2: Escolhendo o idioma do sistema

Intenções do utilizador	Responsabilidades do sistema
	1. Fornecer lista de idiomas disponíveis
2. Escolher idioma	
	3. Aplicar as alterações no sistema

T3: Importando um mapa

Intenções do utilizador	Responsabilidades do sistema
	1. Pedir informações necessárias à importação do mapa
2. Introduzir formato do mapa	
3. Introduzir localização do mapa	
	4. Adicionar o mapa ao sistema

T4: Apagando segmento(s) no plano actual

Intenções do utilizador	Responsabilidades do sistema
	1. Fornecer lista ou visualização dos segmentos
2. Escolher segmentos a apagar	
	3. Apagar os segmentos
	4. Actualizar toda a informação e visualizações afectadas

T5: Criando/alterando sistema(s) de controlo no plano actual

Intenções do utilizador	Responsabilidades do sistema
	1. Fornecer lista ou visualização dos ramais
2. Escolher ramais	
	3. Fornecer lista dos sistemas de controlo
4. Escolher sistemas de controlo	
5. Definir funcionamento do sistema de controlo	
	6. Gravar as alterações
	7. Actualizar toda a informação e visualizações afectadas

T6: Criando ramal no plano actual

Intenções do utilizador	Responsabilidades do sistema
	1. Pedir informações necessárias à criação do ramal
2. Introduzir segmentos*	
2.1 Criar novo segmento	
2.2 Introduzir localização geográfica do segmento	
	2.3 Fornecer lista das luminárias
2.4 Introduzir número de luminárias no segmento	
2.5 Escolher luminária	
	2.6 Fornecer lista das lâmpadas
2.7 Escolher lâmpada	
	3. Fornecer lista dos sistemas de controlo
4. Escolher sistema de controlo	
5. Definir funcionamento do sistema de controlo	
	6. Gravar as alterações
	7. Actualizar toda a informação e visualizações afectadas

T7: Criando/alterando segmento(s) no plano actual

Intenções do utilizador	Responsabilidades do sistema
	1. Fornecer lista ou visualização dos segmentos
2. Escolher segmentos	
	3. Fornecer lista das luminárias
4. Escolher luminária	
	5. Fornecer lista das lâmpadas
6. Escolher lâmpada	
	7. Gravar as alterações
	8. Actualizar toda a informação e visualizações afectadas

T8: Movendo segmento no plano actual

Intenções do utilizador	Responsabilidades do sistema
	1. Fornecer lista ou visualização dos segmentos
2. Escolher segmento	
3. Escolher nova localização geográfica do segmento	
	4. Gravar as alterações
	5. Actualizar toda a informação e visualizações afectadas

T9: Imprimindo relatório

Intenções do utilizador	Responsabilidades do sistema
	1. Fornecer opções da geração de relatórios
2. Escolher tipo de informação a ser apresentada no relatório	
	3. Gerar relatório
	4. Mostrar uma visualização do relatório ao utilizador
5. Confirmar que o relatório está correcto	
	6. Imprimir relatório

T10: Introduzindo/alterando tarifário da área

Intenções do utilizador	Responsabilidades do sistema
	1. Pedir informações necessárias à definição do tarifário
2. Introduzir tarifário	
	3. Guardar alterações
	4. Actualizar toda a informação e visualizações afectadas

T11: Criando/alterando uma lâmpada

Intenções do utilizador	Responsabilidades do sistema
	1. Pedir informações necessárias à introdução de uma nova lâmpada
2. Introduzir modelo e referências da lâmpada	
	3. Fornecer lista das tecnologias de lâmpadas
4. Escolher tecnologia de lâmpada	
5. Introduzir dados relativos à lâmpada	
	6. Guardar alterações

T12: Criando/alterando uma luminária

Intenções do utilizador	Responsabilidades do sistema
	1. Pedir informações necessárias à introdução de uma nova luminária
2. Introduzir modelo e referências da luminária	
3. Introduzir dados relativos à luminária	
	4. Guardar alterações

T13: Criando/alterando um sistema de controlo

Intenções do utilizador	Responsabilidades do sistema
	1. Pedir informações necessárias à introdução de um novo sistema de controlo
2. Introduzir modelo e referências do sistema de controlo	
	3. Fornecer lista dos tipos de sistemas de controlo
4. Escolher tipo de sistema de controlo	
5. Introduzir dados relativos ao sistema de controlo	
	6. Guardar alterações

T14: Importando um cenário

Intenções do utilizador	Responsabilidades do sistema
	1. Pedir localização do cenário
2. Introduzir localização do cenário	
	3. Importar as informações do cenário em memória
	4. Calcular diferenças entre o cenário e o plano actual, e actualizar as visualizações afectadas

T15: Alterando segmento(s) no cenário

Intenções do utilizador	Responsabilidades do sistema
	1. Fornecer lista ou visualização dos segmentos
2. Escolher segmentos	
	3. Mostrar informação actual dos segmentos
4. Alterar informação	
	4.1. Fornecer lista das luminárias que podem ser utilizadas
4.2 Escolher luminária	
	4.3. Fornecer lista das lâmpadas que podem ser utilizadas
4.4 Escolher lâmpadas	
	5. Guardar alterações
	6. Calcular o novo resultado das diferenças entre o cenário e o plano actual, e actualizar as visualizações afectadas

T16: Optimizando segmento(s) no cenário

Intenções do utilizador	Responsabilidades do sistema
--------------------------------	-------------------------------------

	1. Fornecer lista ou visualização dos segmentos
2. Escolher segmentos	
	3. Fornecer critérios que podem ser utilizados para a optimização
4. Escolher critérios a utilizar	
	5. Escolher melhor lâmpada que se enquadra no critério pedido
6. Aceitar a sugestão	
	7. Guardar alterações
	8. Calcular o novo resultado das diferenças entre o cenário e o plano actual, e actualizar as visualizações afectadas

T17: Alterando sistema de controlo de um ramal

Intenções do utilizador	Responsabilidades do sistema
	1. Fornecer lista ou visualização dos ramais
2. Escolher ramal	
3. Alterar sistema de controlo	
	3.1. Fornecer lista dos sistemas de controlo disponíveis
3.2. Escolher novo sistema de controlo	
3.3. Definir novo funcionamento do sistema de controlo	
	4. Guardar alterações
	5. Calcular o novo resultado das diferenças entre o cenário e o plano actual, e actualizar as visualizações afectadas

T18: Fazendo análise dos resultados de um cenário

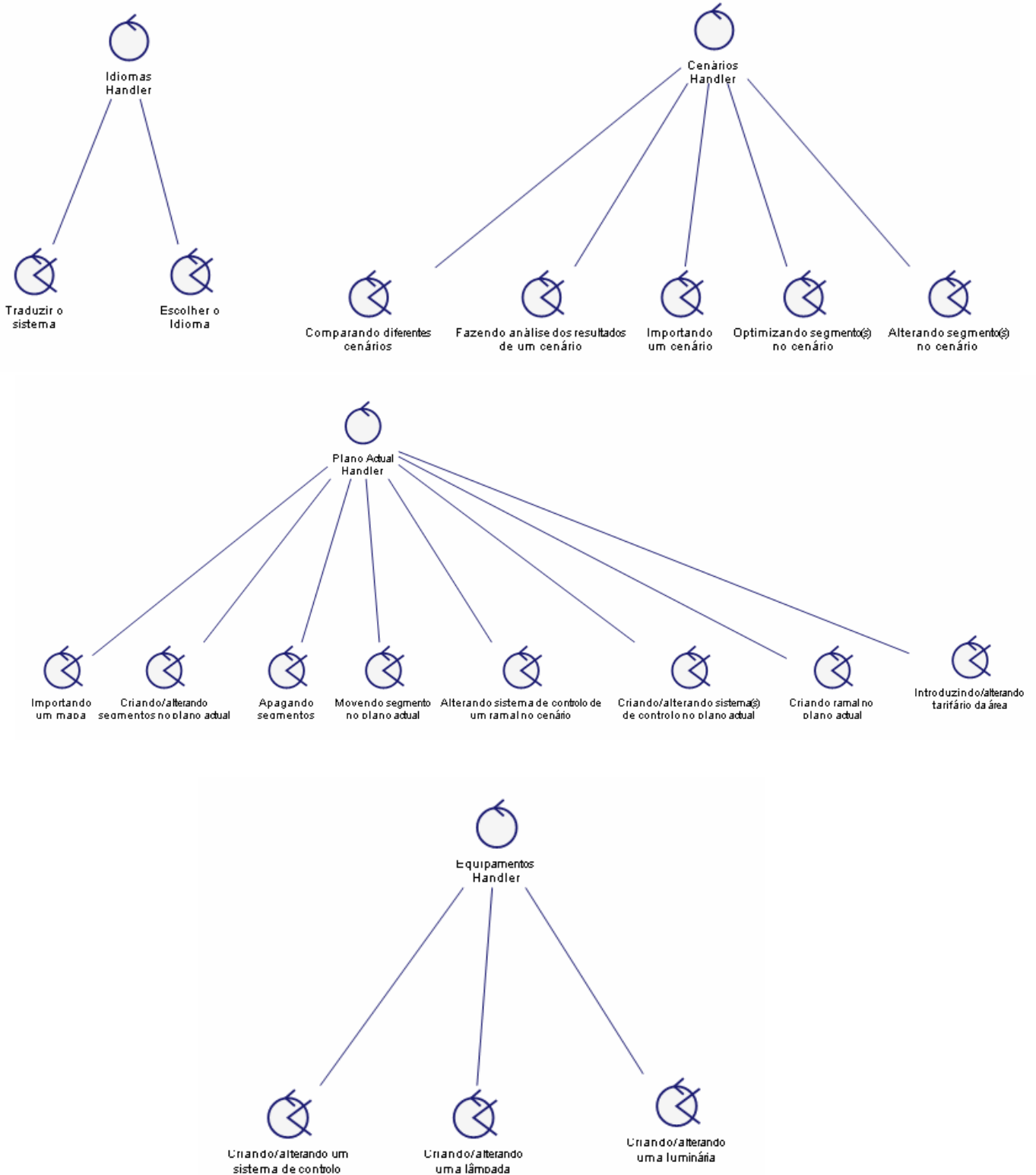
Intenções do utilizador	Responsabilidades do sistema
	1. Fornecer opções de visualização dos resultados
2. Escolher opções de visualização dos resultados	
	3. Gerar um relatório detalhado dos resultados

T19: Comparando diferentes cenários

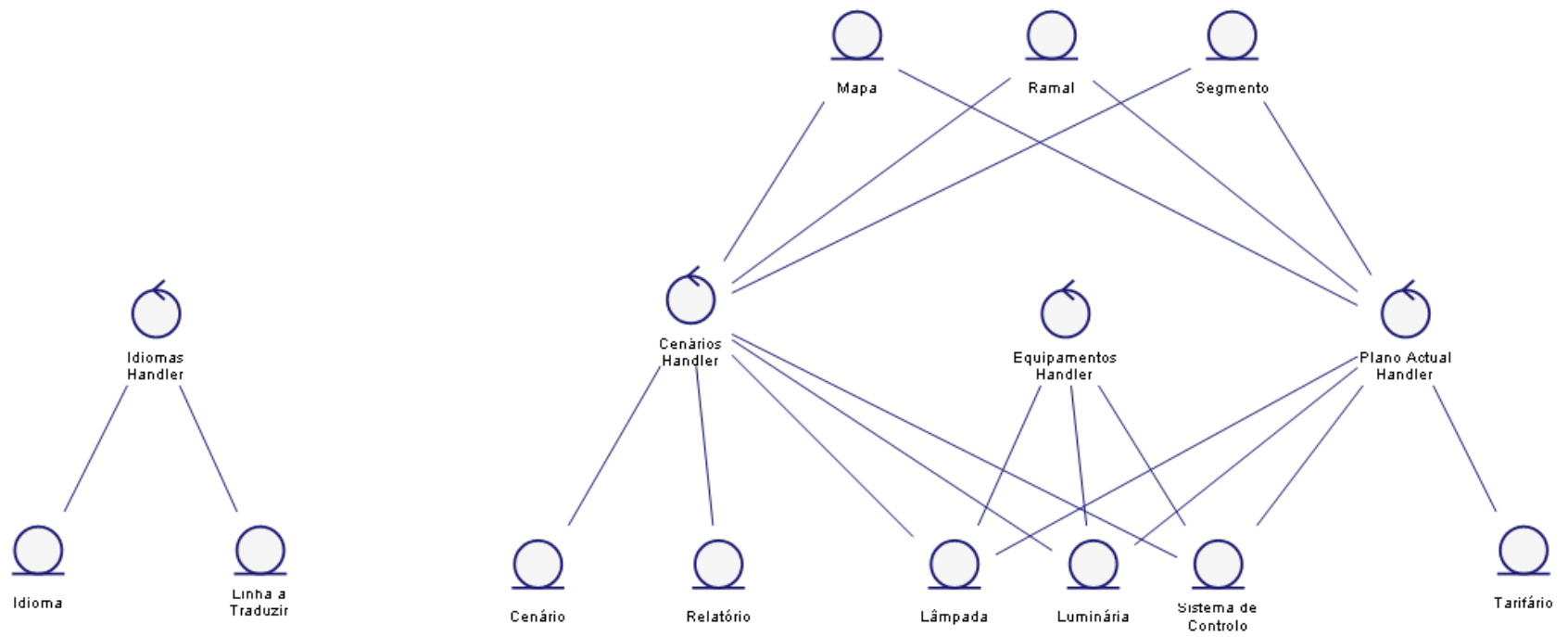
Intenções do utilizador	Responsabilidades do sistema
	1. Fornecer lista ou visualização dos cenários
2. Escolher cenários	
3. Escolher opções de visualização dos resultados	
	4. Gerar um relatório ou visualização da comparação

Diagramas Wisdom

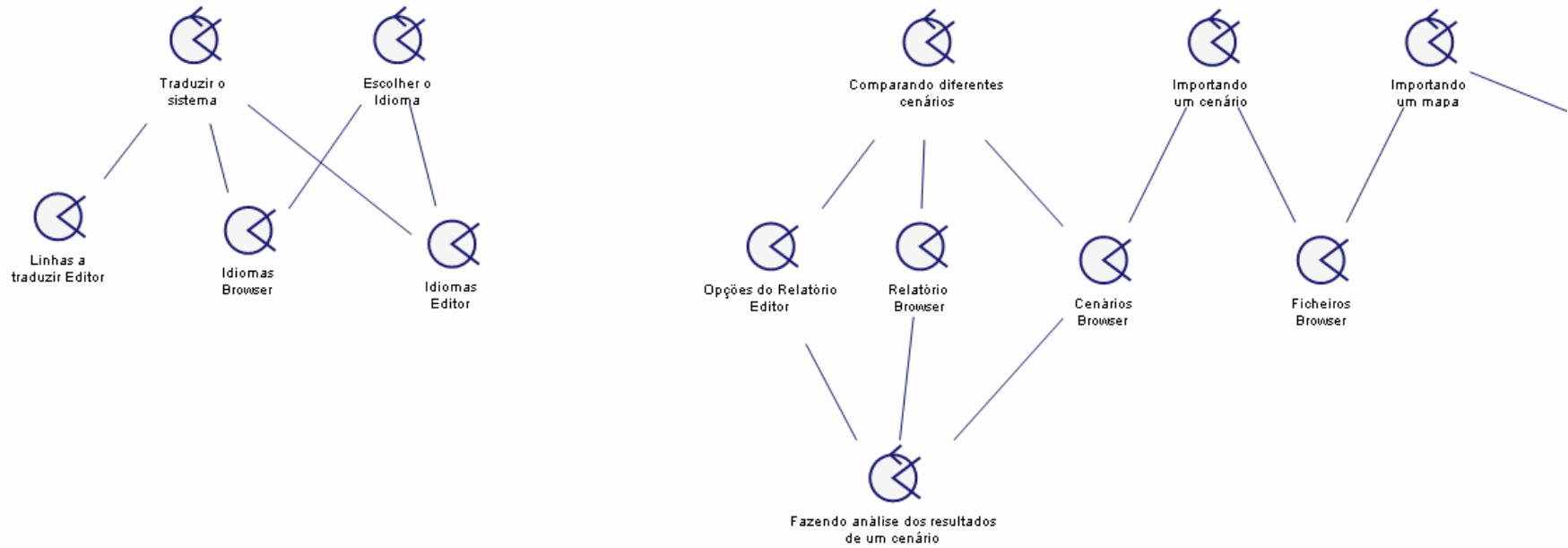
Modelo de análise subdividido (1)



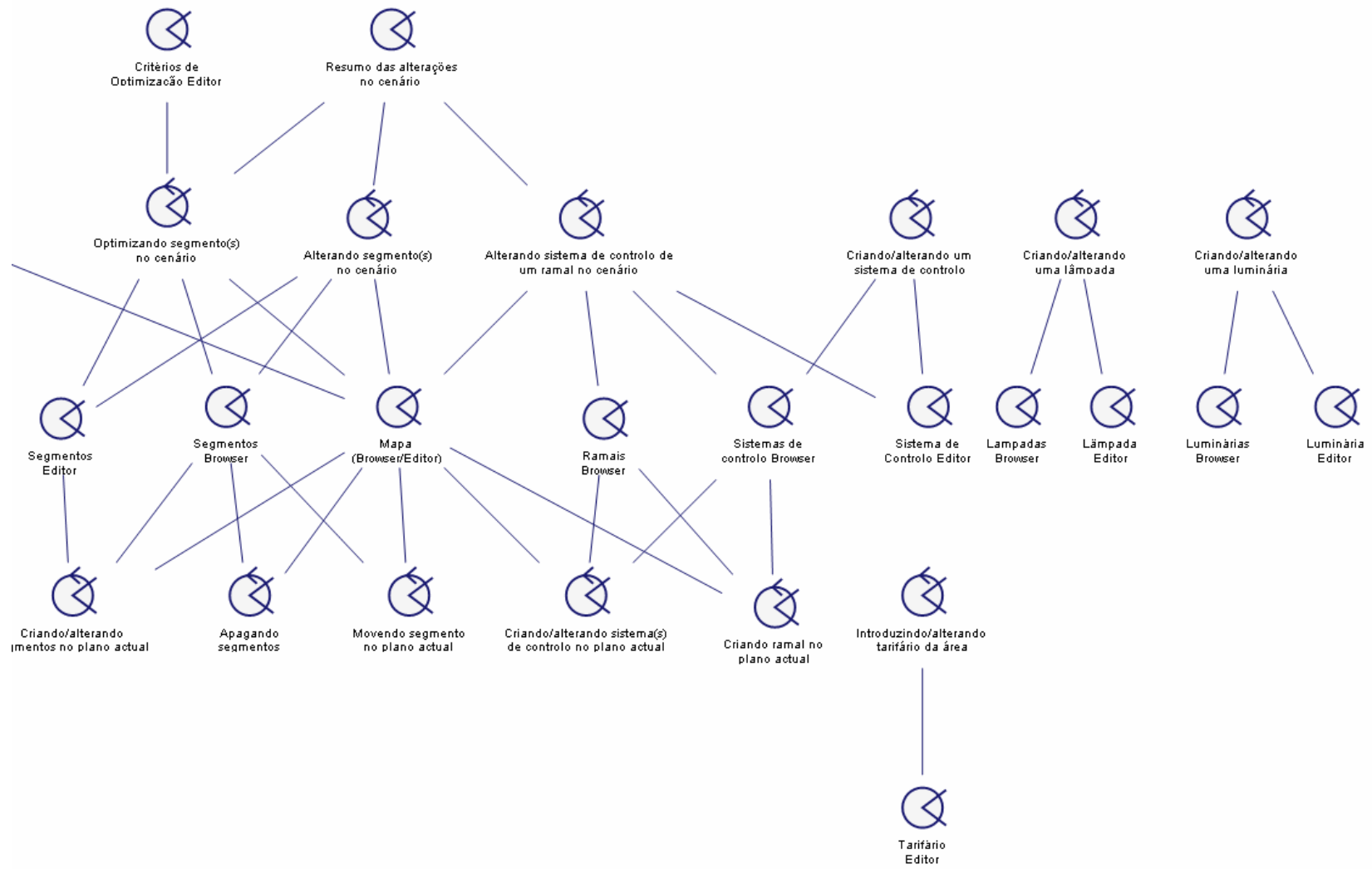
Modelo de análise subdividido (2)



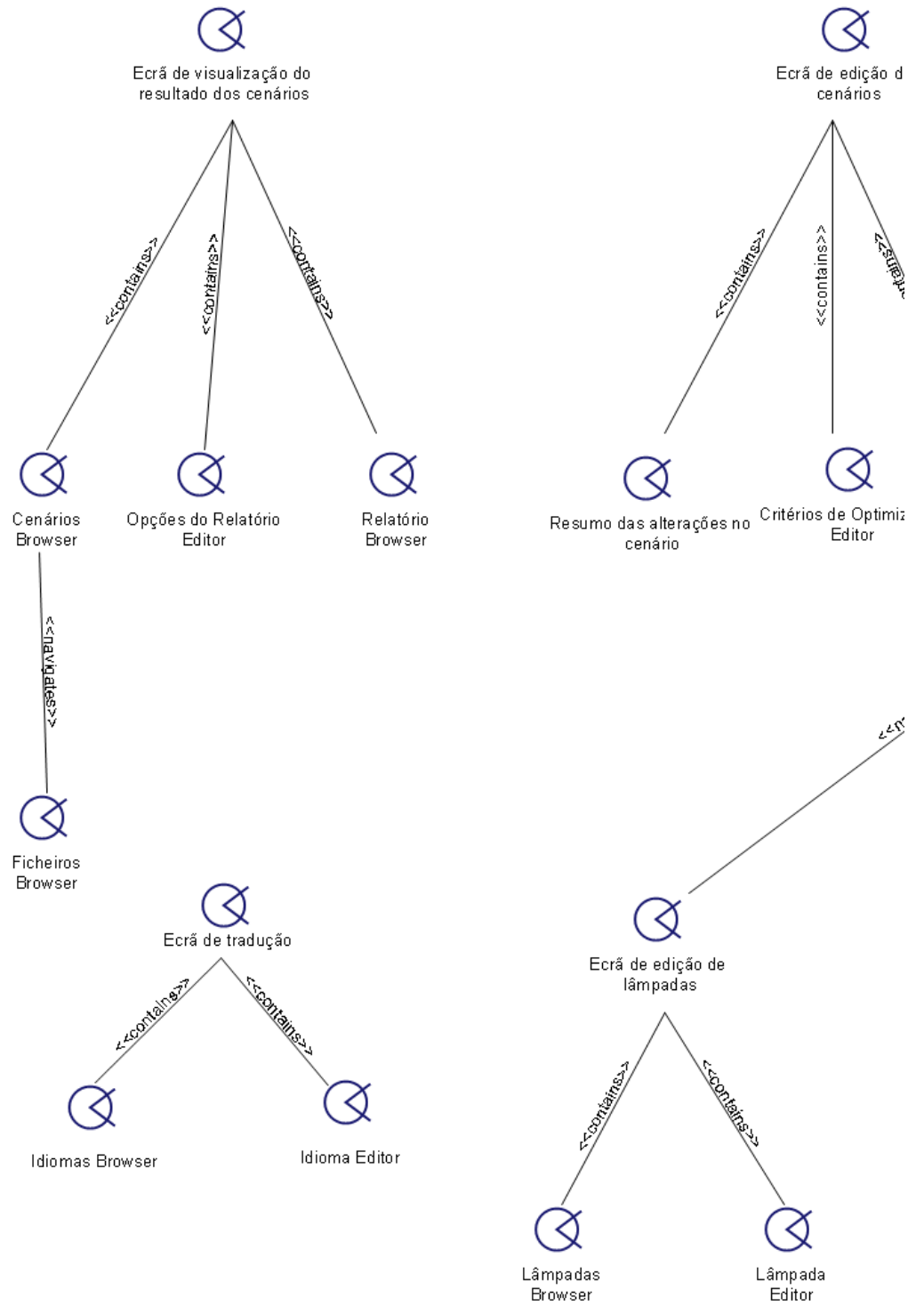
Modelo de análise subdividido (3)

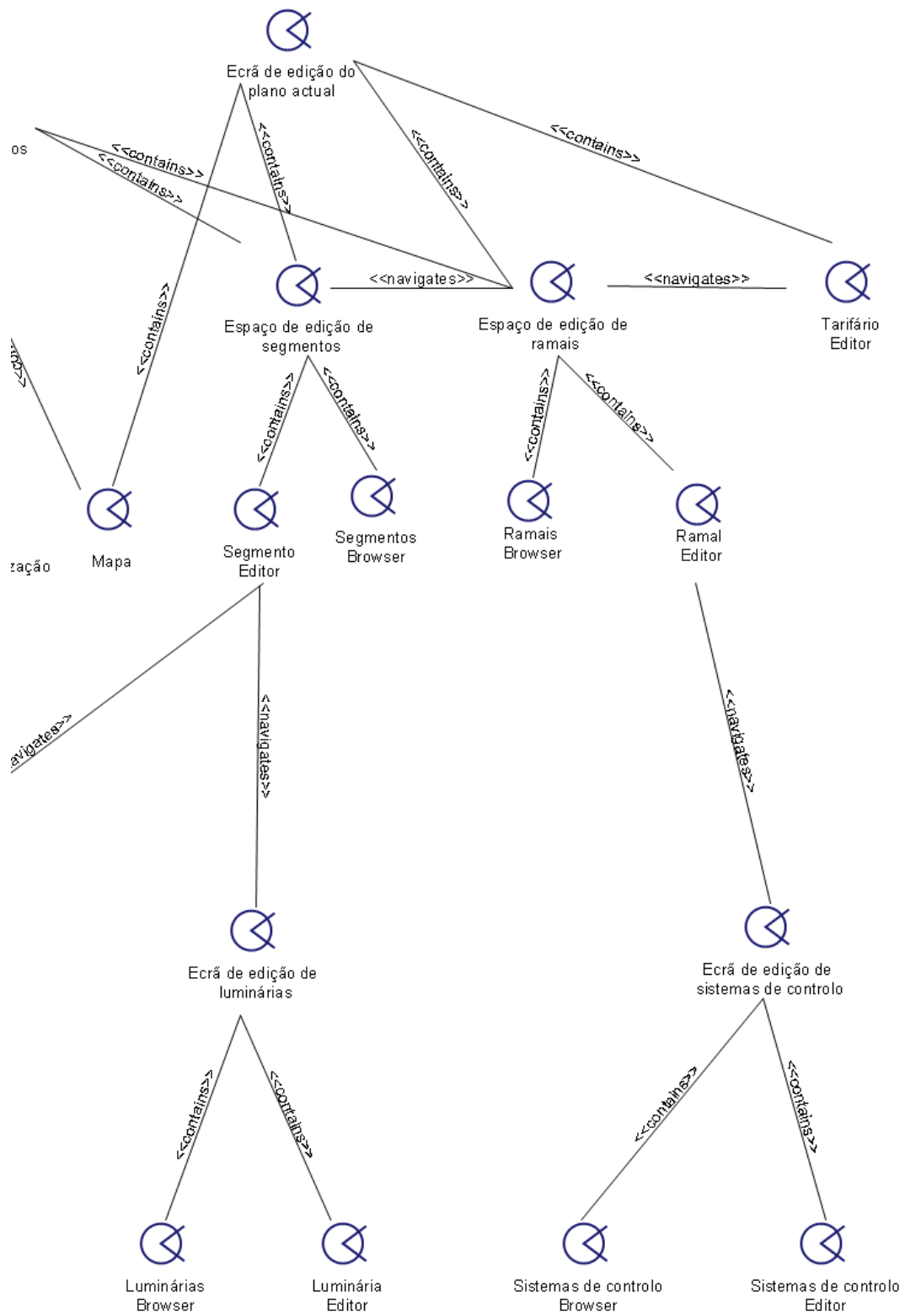


seq



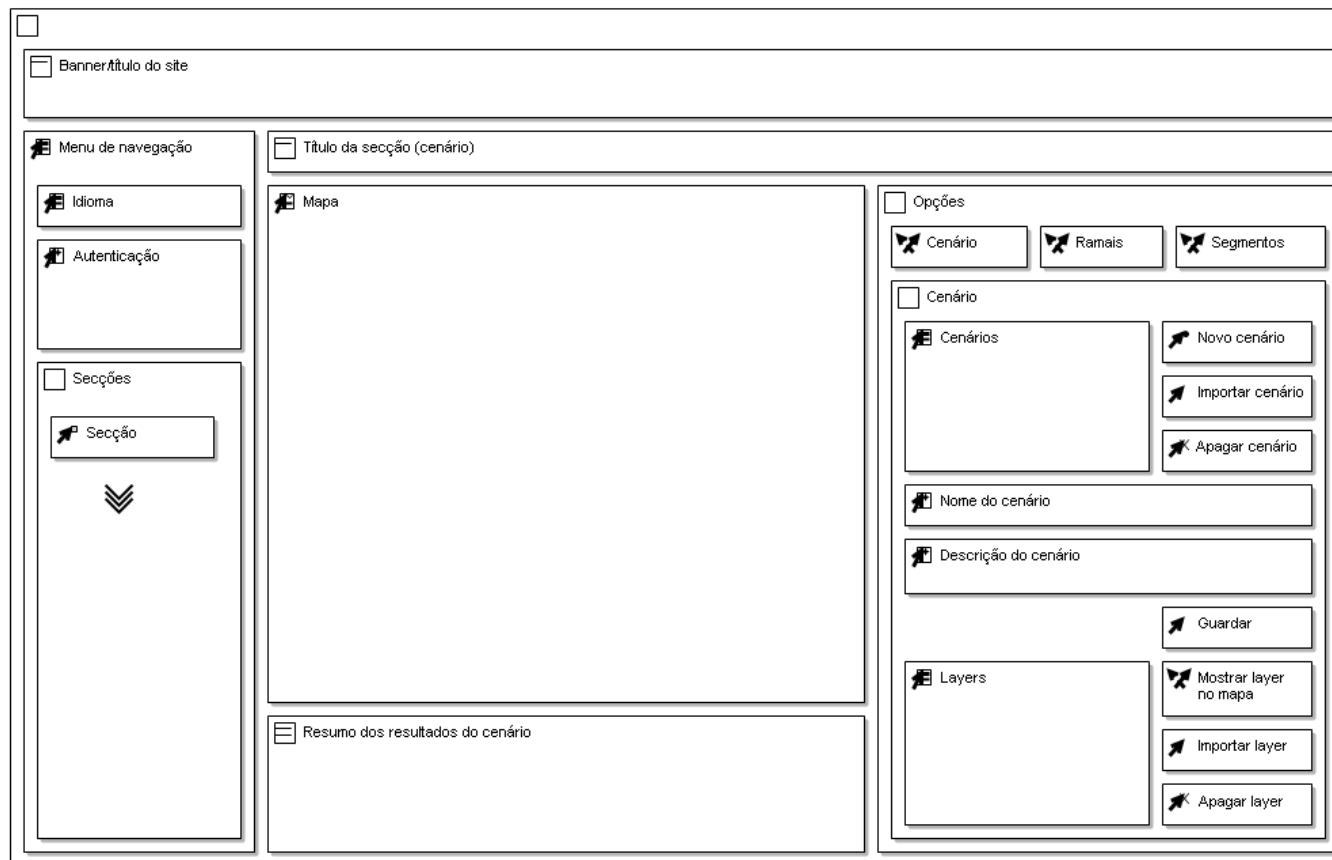
Modelo de navegação



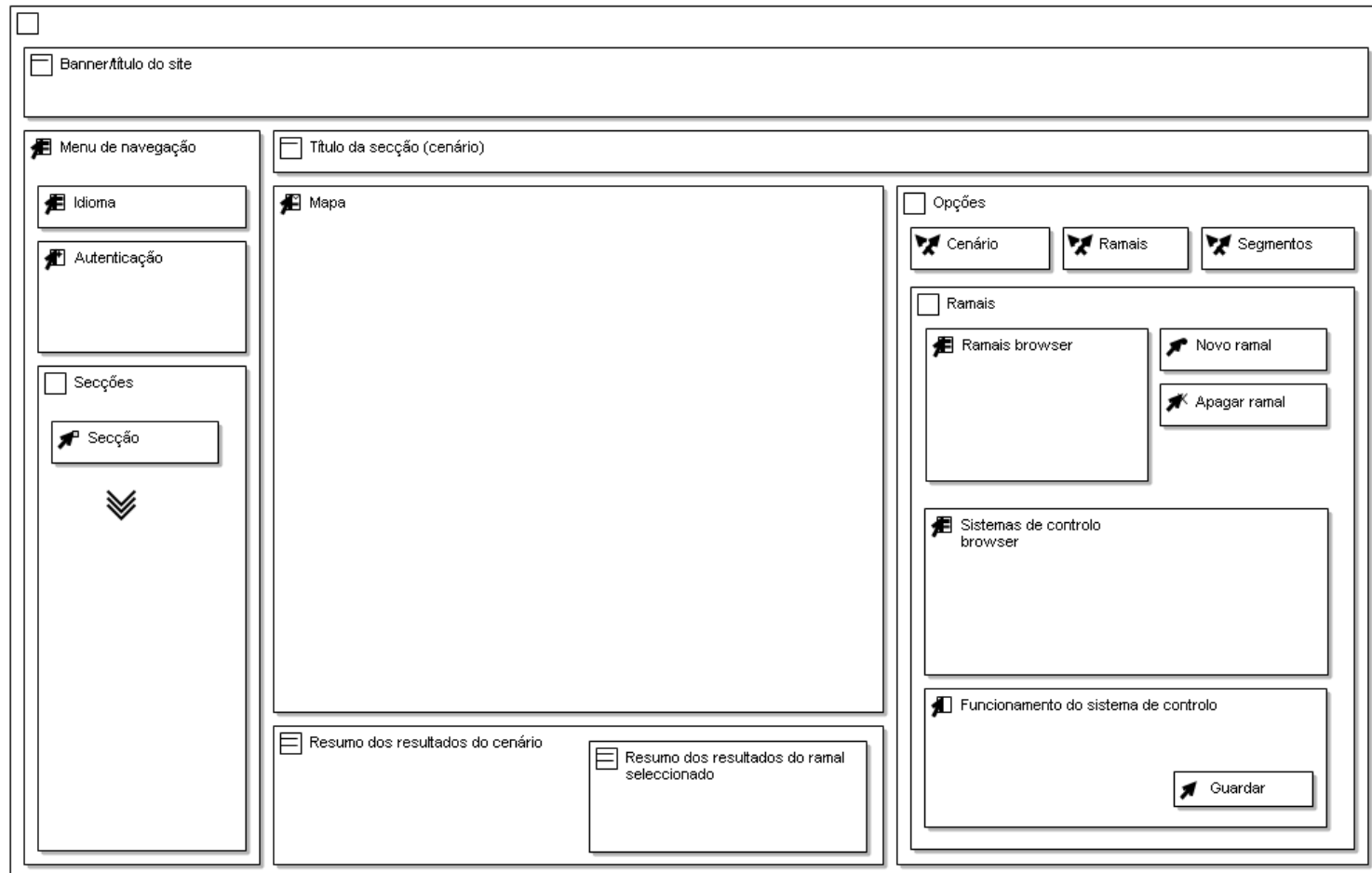


Protótipos abstractos canónicos

Ecrã cenários (tab opções gerais)



Ecrã cenários (tab ramais)



Ecrã cenários (tab segmentos)

Banner/título do site

Menu de navegação

- Idioma
- Autenticação
- Secções
 - Secção

Título da secção (cenário)

Mapa

Opções

- Cenário
- Ramais
- Segmentos

Segmentos

- Ramal seleccionado
- Segmentos browser
 - Novo segmento
 - Apagar segmento
 - Mostrar todos os segmentos / só os do ramal
- Nº de luminárias
- Luminária Escolher luminária
- Lâmpada Escolher lâmpada

Resumo dos resultados do cenário

Resumo dos resultados do segmento seleccionado

Guardar

Ecrã plano actual (tab opções gerais)

Banner/título do site

Menu de navegação

- Idioma
- Autenticação
- Secções
 - Secção

Título da secção (plano actual)

Mapa

Opções

- Plano actual
- Ramais
- Segmentos

Plano actual

- Definir localização actual como centro do mapa
- Layers
 - Mostrar layer no mapa
 - Importar layer
 - Apagar layer
- Condições do tarifário
 - Nova condição
 - Apagar condição
- Detalhes da condição
 - Guardar

Resumo dos resultados do plano actual

Ecrã plano actual (tab ramais)

Banner/título do site

Menu de navegação

- Idioma
- Autenticação
- Secções
 - Secção

Título da secção (plano actual)

Mapa

Opções

- Plano actual
- Ramais
- Segmentos

Ramais

- Ramais browser
 - Novo ramal
 - Apagar ramal
- Sistemas de controlo browser
 - Adicionar sistema de controlo
 - Apagar sistema de controlo
- Funcionamento do sistema de controlo
 - Guardar

Resumo dos resultados do plano actual

Resumo dos resultados do ramal seleccionado

Ecrã plano actual (tab segmentos)

Banner/título do site

Título da secção (plano actual)

Menu de navegação

- Idioma
- Autenticação

Secções

- Secção

≡

Mapa

Opções

- Plano actual
- Ramais
- Segmentos

Segmentos

- Ramal seleccionado
- Segmentos browser
 - Novo segmento
 - Apagar segmento
 - Mostrar todos os segmentos / só os do ramal
- Modo adicionar pontos ao segmento
- Nº de luminárias
- Luminária Escolher luminária
- Lâmpada Escolher lâmpada
- Guardar


Resumo dos resultados do plano actual


Resumo dos resultados do segmento seleccionado


Popup de escolha de uma lâmpada (a partir da secção "cenários")

Popup de escolha de uma lâmpada (a partir da secção "cenários")


Lâmpadas


 Lista de lâmpadas



 Filtrar lista de lâmpadas


 Detalhes da lâmpada seleccionada

Sugestões automáticas de lâmpada

Lâmpada com menor custo de instalação:  Ver lâmpada

Lâmpada com menor consumo:  Ver lâmpada


 


 Escolher lâmpada


Popup de escolha de uma lâmpada (a partir da secção "plano actual")


Popup de escolha de uma lâmpada (a partir da secção "plano actual")

Lâmpadas

 Lista de lâmpadas

 Filtrar lista de lâmpadas


 Detalhes da lâmpada seleccionada

 Escolher lâmpada


Popup de escolha de uma luminária


Popup de escolha de uma luminária

Luminárias

 Lista de luminárias

Filtrar lista de luminárias


 Detalhes da luminária seleccionada


 Escolher luminária


Popup de escolha de um sistema de controlo


Popup de escolha de um sistema de controlo

Sistemas de controlo

 Lista de sistemas de controlo

 Filtrar lista de sistemas de controlo

 Detalhes do sistema de controlo seleccionado

 Escolher sistema de controlo

Ecrã equipamentos – lâmpadas

Banner/título do site

Menu de navegação

- Idioma
- Autenticação
- Secções
 - Secção

Título da secção (equipamentos - lâmpadas)

Detalhes da lâmpada

- Marca
- Modelo
- Referência produto
- Referência int.
- Tecnologia
- Fixação
- Balastro
- Potência
- Fluxo luminoso
- Temperatura de cor
- Índice de rest. cor
- Comprimento
- Vida útil balastro
- Vida útil lâmpada
- Preço balastro
- Preço lâmpada
- Custo inst. balastro
- Custo inst. lâmpada

Imagem

- Alterar imagem

Lâmpadas

- Lista de lâmpadas
- Nova lâmpada
- Apagar lâmpada
- Filtrar lista de lâmpadas

Guardar

Ecrã equipamentos – luminárias

Banner/título do site

Menu de navegação

- Idioma
- Autenticação
- Secções
 - Secção

Título da secção (equipamentos - luminárias)

Detalhes da luminária

- Marca
- Modelo
- Referência produto
- Nº lâmpadas
- Potência máxima
- Fixação
- Comprimento max. lâmpada
- Referência int. lâmpadas compatíveis
- Reflector
- Difusor
- Eficiência reflector
- Eficiência difusor
- Ângulo abertura luminária
- Preço luminária
- Custo inst. luminária

Imagem

- Alterar imagem

Luminárias

- Lista de luminárias
 - Nova luminária
 - Apagar luminária
- Filtrar lista de luminárias

Guardar

Ecrã equipamentos – sistemas de controlo

Banner/título do site

Menu de navegação

- Idioma
- Autenticação
- Secções
 - Secção

Título da secção (equipamentos - sistemas de controlo)

Detalhes do sistema de controlo

- Marca
- Modelo
- Referência produto
- Tipo de sistema
- Potência máxima
- Referência int. lâmpadas compatíveis
- Sensor luminosidade
- Sensor movimento
- Programador horário
- Regulador fluxo
- Escalão mínimo fluxo luminoso
- Preço sistema
- Custo inst. sistema

Imagem

- Alterar imagem

Sistemas de controlo

- Lista de sistemas de controlo
- Novo sistema de controlo
- Apagar sistema de controlo
- Filtrar lista de sistemas de controlo

Guardar

Ecrã relatórios

Banner/título do site


Menu de navegação

Idioma

Autenticação

Secções

Secção



Título da secção (relatório)

Relatório

Opções

Tipo de visualização

Texto

Gráfico

Mapas a analisar

Plano actual

Cenários

Tipos de análise

Custo (inv. inicial)

Custo (exploração)

Custo (inv. inicial + exploração)

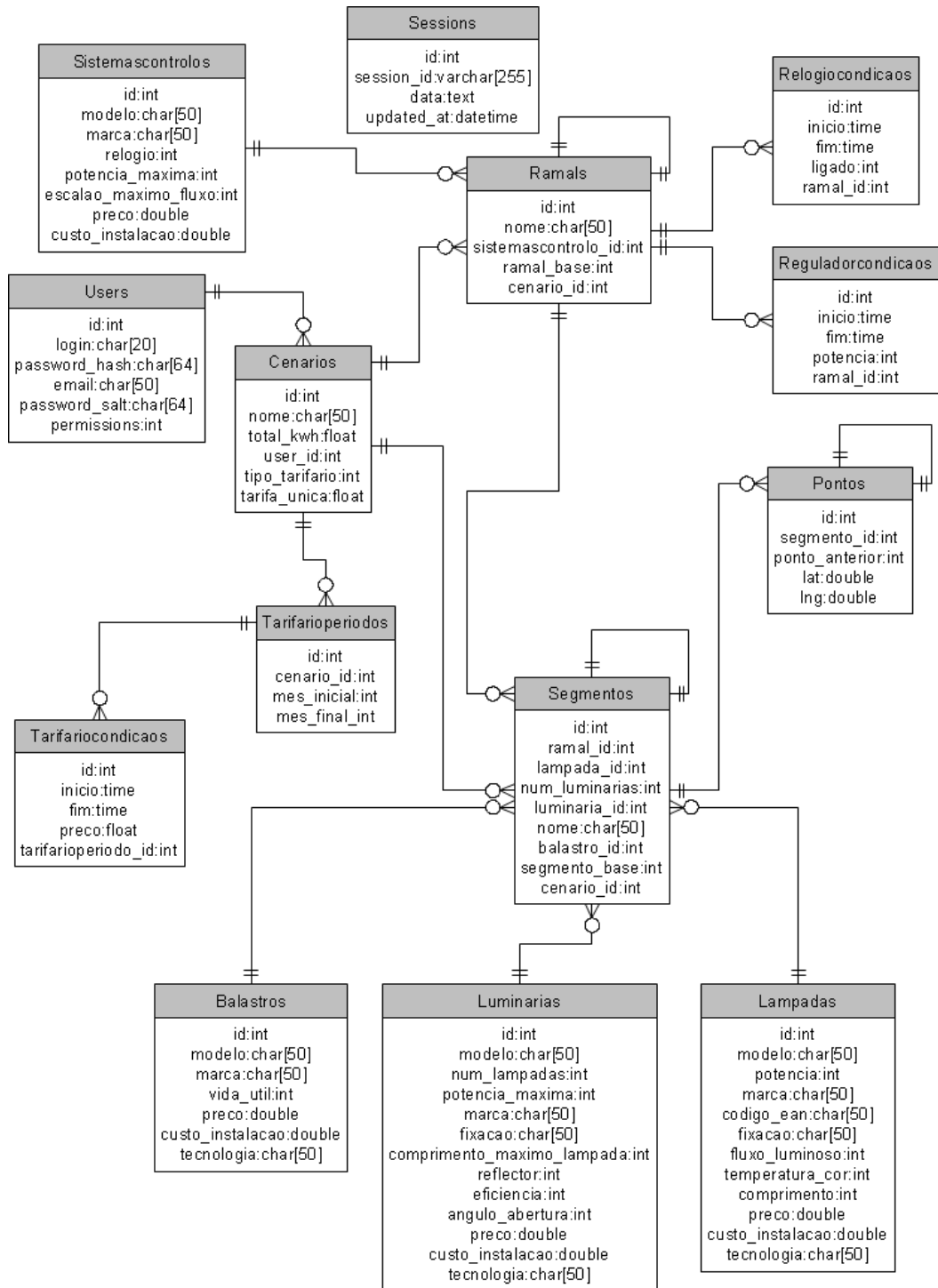
Consumo de energia

Impacte ambiental

Potência

Versão para impressão

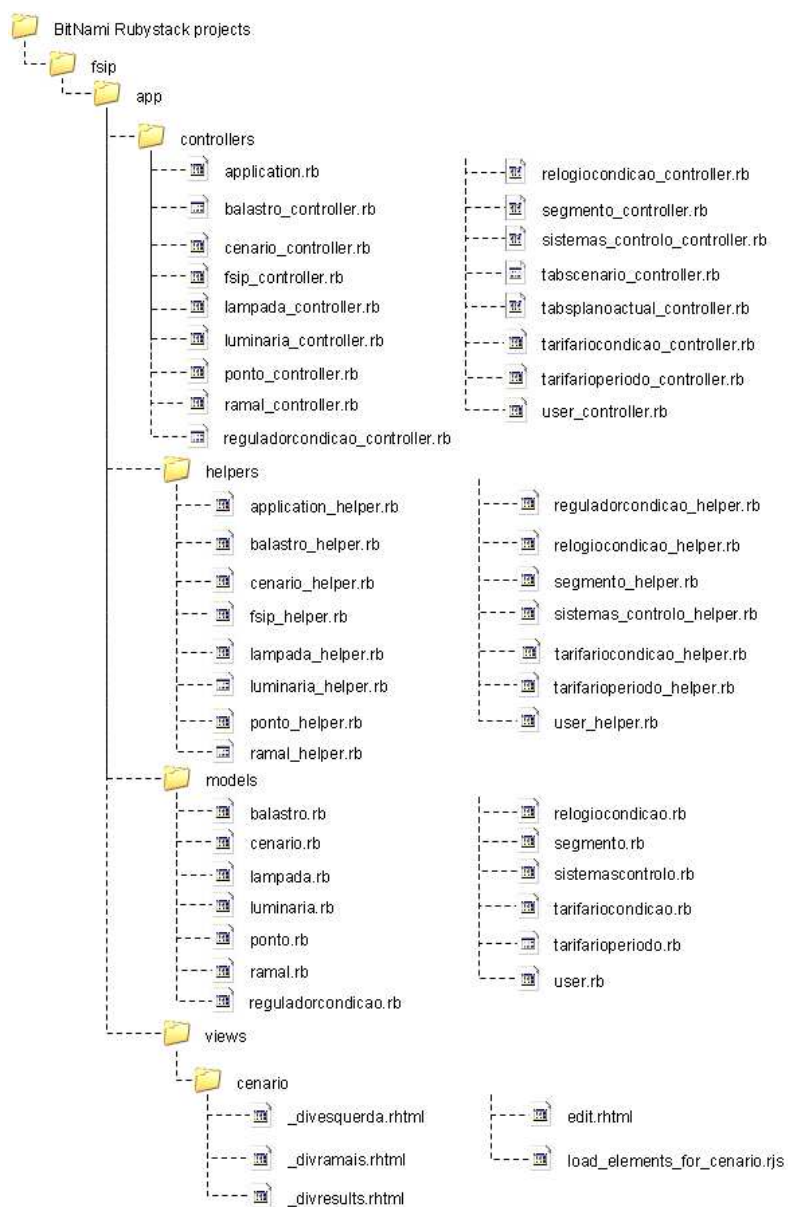
Diagrama de entidade-relacionamento da base de dados

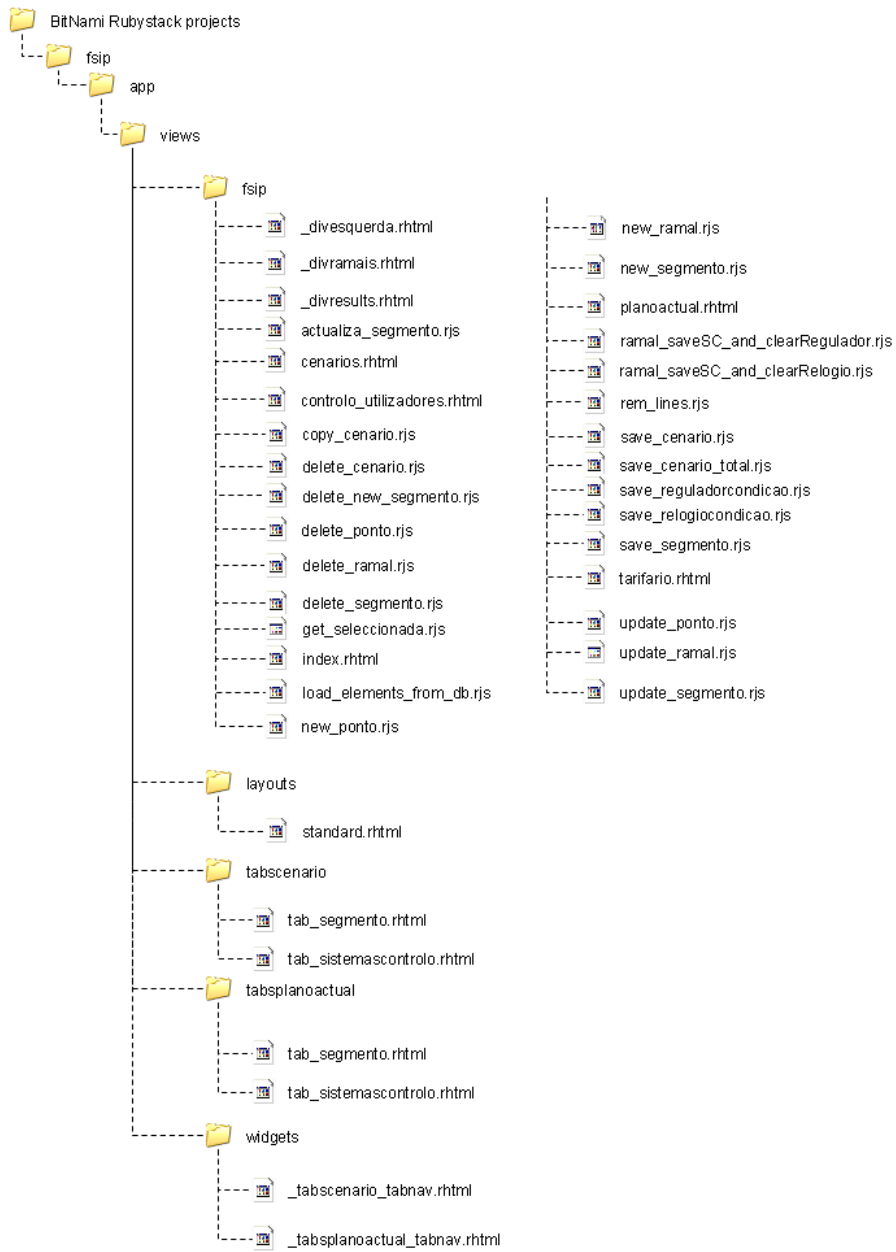


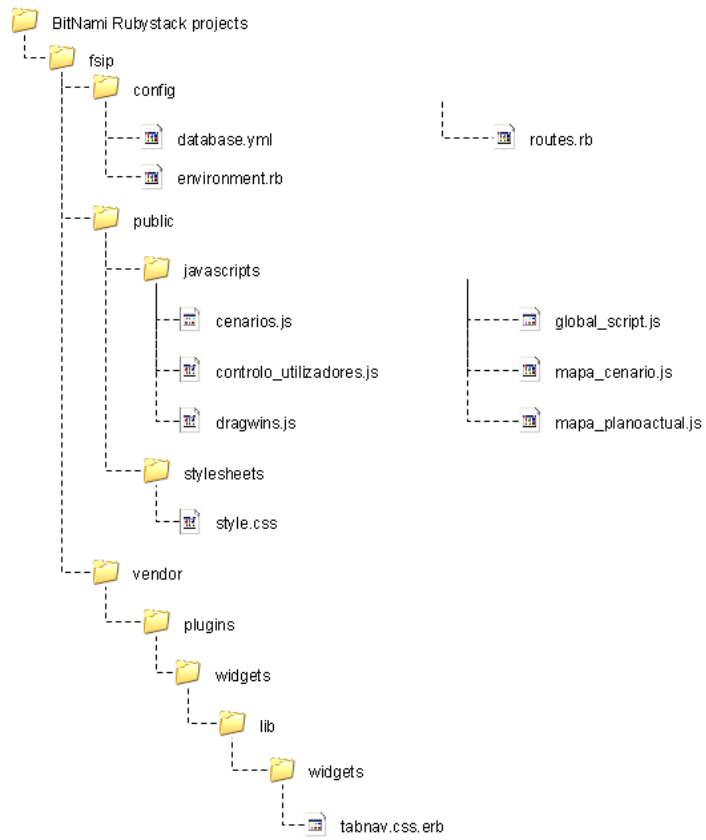
ANEXO II: FICHEIROS DA APLICAÇÃO

Diagramas de pastas e ficheiros

Estes diagramas incluem apenas os ficheiros e pastas que foram modificados ou gerados no decurso do desenvolvimento da aplicação. Na realidade, são necessários mais ficheiros para o correcto funcionamento da aplicação.







Modelos e controladores

Os ficheiros de modelos e controladores encontram-se, respectivamente, dentro de `/app/models` e `/app/controllers`. O funcionamento de cada modelo e controlador encontra-se explicado no relatório (sub-capítulo 4.3.), sendo que a nomenclatura dos ficheiros é fácil de associar aos respectivos modelos e controladores referidos.

Vistas

Os ficheiros de vistas são separados dentro de pastas correspondentes a cada controlador que tenha páginas web associadas a si, dentro de `/app/views`.

A pasta `/app/views/fsip` é a que contém mais ficheiros, naturalmente, por corresponder ao controlador `fsip_controller.rb`, que é o que define mais acções. Dentro desta pasta, existem diversos tipos de ficheiros:

- ficheiros RHTML que geram directamente uma página no browser. São neste caso **`index.rhtml`** (página inicial da aplicação), **`planoactual.rhtml`** (página da rede actual de iluminação sobre o mapa), **`tarifario.rhtml`** (página do tarifário), **`cenarios.rhtml`** (página de edição de cenários), e **`controlo_utilizadores.rhtml`** (página de controlo de utilizadores).

- ficheiros RHTML que geram *partials*, que são um fragmento de uma página html que pode ser reutilizado em várias páginas. Estes ficheiros são facilmente identificáveis pelo seu nome começar por *underscore* (`_`), e neste caso são: **`_divesquerda.rhtml`** (ficheiro que define o menu com as várias secções da aplicação que aparece na página do lado esquerdo), **`_divramais.rhtml`** (define o menu com as diversas opções sobre os ramos de electricidade do lado direito da página, nas páginas que incluem o mapa), e **`_divresults.rhtml`** (define a área com os “resultados rápidos” da simulação, indicando o consumo da electricidade da rede no fundo da página). Ao contrário dos outros tipos, este tipo de ficheiro não corresponde directamente a uma acção no respectivo controlador.

- ficheiros RJS, que geram código javascript do lado do servidor, como resposta a um pedido Ajax do cliente, referente a este controlador. Em certos casos, o ficheiro RJS é deixado vazio, com o objectivo de correr certas acções no controlador sem que a resposta corra qualquer código javascript; e a existência do ficheiro justifica-se para que o Ruby on Rails perceba que a acção respectiva no controlador corresponde a um

pedido Ajax, e não a uma página web (que é assumido por defeito). A pasta `/app/views/fsip` contém 24 ficheiros RJS.

A pasta `/app/views/cenario` também inclui vários ficheiros, utilizados na edição de um cenário pelo utilizador. `edit_rhtml` corresponde ao ficheiro principal da geração da página para tal efeito; `_divesquerda.rhtml`, `_divramais.rhtml` e `_divresults.rhtml` têm papel semelhante aos ficheiros com o mesmo nome utilizados em `/app/views/fsip`; e `load_elements_for_cenario.rjs` responde a um pedido Ajax que é efectuado no carregamento da página.

As *tabs* utilizadas na aplicação, tal como nos controladores, também necessitam de ficheiros específicos de vistas, ou seja, um ficheiro RHTML para cada *tab*. A pasta `/app/views/tabsplanoactual` inclui assim os ficheiros `tab_segmento.rhtml` e `tab_sistemascontrolo.rhtml`, que correspondem às *tabs* “Segmento” e “Sistemas de Controlo” na página da rede actual. A pasta `/app/views/tabscenario` inclui 2 ficheiros semelhantes, correspondentes às *tabs* com o mesmo nome na página de edição de um cenário. Para além destes ficheiros, uma pasta especial correspondente ao *plugin* Rails Widgets (`/app/views/widgets`), define todas as instâncias de *tabs* que são necessárias na aplicação – neste caso através dos ficheiros `_tabscenario_tabnav.rhtml` e `_tabsplanoactual_tabnav.rhtml`.

A pasta `/app/views/user` inclui `signup.rhtml`, que define uma página de registo de novos utilizadores na aplicação; e `_divesquerda.rhtml`, com comportamento semelhante aos ficheiros com o mesmo nome descritos anteriormente.

Finalmente, é utilizada uma pasta especial denominada de `/app/views/layouts`, que define características de visualização que se aplicam a todas as páginas, através do ficheiro `standard.rhtml`.

Outros ficheiros

Para além de modelos, vistas e controladores, a aplicação necessita de uma série de outros ficheiros para funcionar. Grande parte destes ficheiros é gerada automaticamente pelo Ruby on Rails no processo de criação de um projecto. Para além destes foi criado e configurado um grande número de ficheiros durante o desenvolvimento da aplicação.

Ficheiros javascript sobre a API do Google Maps

Apesar de o Ruby on Rails trazer a vantagem de grande parte do processo de escrita de código ser efectuado a um alto nível de abstracção, o que poupa muito tempo em diversas ocasiões, a API do Google Maps está acessível na sua maior parte apenas através de funções javascript - que nem mesmo a utilização do *plugin* YM4R consegue evitar. Sendo assim, uma grande parte do código da aplicação está concentrada em ficheiros javascript que manipulam directamente a API do Google Maps: **mapa_planoactual.js** e **mapa_cenario.js**. Estes dois ficheiros têm uma estrutura semelhante, e incluem várias funções idênticas, mas esta separação é necessária por definirem funções para dois casos diferentes, respectivamente: o uso do mapa no desenho da rede actual de electricidade, e o uso do mapa no desenho de um cenário de iluminação.

Estes dois ficheiros efectuam as seguintes acções:

- Carregamento de todos os dados relevantes à apresentação do mapa, dos detalhes de cada segmento/ramal, e dos dados necessários para o cálculo do espaço de "resultados rápidos". Isto é feito através de um pedido Ajax ao servidor imediatamente após o carregamento do ficheiro, o que obriga a um atraso adicional na apresentação da página. A razão de isto estar estruturado desta forma deriva da necessidade de carregar os dados para variáveis e arrays no âmbito do Javascript, o que é mais fácil através de um ficheiro RJS - no Ruby on Rails, os ficheiros RJS são executados a partir de pedidos Ajax. Neste caso, os ficheiros RJS específicos são **load_elements_for_cenario.rjs** e **load_elements_for_planoactual.rjs**.

- Actualizar a interface cada vez que o utilizador interage com um segmento ou ponto no mapa. Isto inclui mudar as cores dos segmentos, actualizar a lista de ramais, a tab activa com os dados relativos ao segmento ou ao sistema de controlo, actualizar a secção de resultados rápidos, e eventualmente abrir um menu no mapa (no caso de o

utilizador clicar com o botão direito do rato sobre um marker que represente um ponto).

- Actualizar o mapa cada vez que o utilizador interage com o espaço à direita do mapa; ou seja, cada vez que o utilizador altera ou selecciona diferentes ramais/segmentos, é necessário actualizar as cores dos segmentos no mapa, e mover o centro do mapa para o sítio onde o ramal/segmento está localizado.

- Calcular dinamicamente os “resultados rápidos” do consumo de electricidade, e, no caso de `mapa_cenario.js`, calcular as diferenças de consumo entre o cenário apresentado e a rede actual.

- Mandar pedidos Ajax para o servidor cada vez que é feita uma alteração a um ramal/segmento/ponto, ou é apagado um ramal/segmento/ponto, ou é criado um novo ramal/segmento/ponto. O servidor por sua vez tem de actualizar a base de dados com estas alterações e responder se a actualização teve ou não sucesso.

Foram também definidas um largo número de funções que são auxiliares a estas tarefas.

Outros ficheiros javascript

Para além dos dois ficheiros mencionados, são utilizados os seguintes ficheiros javascript:

- **global_script.js**, ficheiro que engloba funções que têm de ser chamadas por todas as páginas da aplicação, mesmo as páginas que não contêm um mapa. Na realidade, este ficheiro apenas define uma função de redimensionamento do mapa para a correcta apresentação no *browser* – mas esta função tem de ser chamada pelo corpo da página principal (<body>), que é definido no ficheiro `standard.rhtml`, e este *layout* aplica-se a todas as páginas da aplicação, mesmo as que não contêm mapa.

- **cenarios.js**, ficheiro que define funções para a página “Cenários”, em que é possível copiar e apagar cenários a partir da rede actual ou de outros cenários.

- **controlo_utilizadores.js**, ficheiro que define funções para a página “Controle de Utilizadores”, em que o administrador da aplicação pode definir as permissões para os utilizadores.

- **tarifario.js**, ficheiro que define funções para a página “Tarifários”, em que um utilizador poderá alterar o tarifário da área onde se insere a rede de electricidade a analisar.

- **dragwins.js**, ficheiro que define funções para a apresentação de pequenas “janelas” redimensionáveis dentro do browser, utilizadas para a pesquisa de lâmpadas, luminárias, balastos e sistemas de controlo nas páginas da aplicação que incluem o mapa. Este ficheiro é uma ligeira adaptação do código-fonte fornecido num artigo de Jeff Cogswell para o *website* DevSource.com [34].

Outros ficheiros javascript são necessários ao correcto funcionamento da aplicação, como por exemplo **prototype.js**, um ficheiro que define uma *framework* de ajuda ao desenvolvimento sobre Javascript [35]. Contudo, estes ficheiros não foram alterados no decurso do desenvolvimento da aplicação.

Todos os ficheiros javascript utilizados são alojados em /public/javascripts/.

Ficheiros de configuração diversos

Dentro da pasta /app/, para além dos modelos, vistas e controladores, é incluído também uma pasta com helpers – ficheiros que no Ruby on Rails existem para definir métodos adicionais que forneçam assistência aos ficheiros de vistas [36], e que correspondem aos controladores (ex: /app/helpers/fsip_helper.rb corresponde a /app/controllers/fsip_controller.rb). Nesta aplicação os helpers não foram utilizados para a definição de qualquer método - no entanto é necessário a sua inclusão.

Em /config/, é definido um ficheiro, cuja correcta definição é importante para o devido funcionamento onde quer que a aplicação seja instalada: **database.yml**, que define os detalhes da ligação à base de dados.

Em /public/stylesheets/, encontra-se outro ficheiro importante: **style.css**, que define os estilos CSS da aplicação, ou seja, todos os detalhes de apresentação como o tamanho de cada elemento da página, o tipo de letra, cor, etc. Em /public/images/, são alojadas todas as imagens da aplicação, como por exemplo pequenos ícones, ou o marcador de um ponto sobre o mapa.

Finalmente, em /vendor/plugins/ são alojados os 2 plugins para o Ruby on Rails utilizados para esta aplicação: Rails Widgets (/vendor/plugins/widgets/) e YM4R (/vendor/plugins/ym4r_gm/). Os ficheiros dentro destas pastas não foram alterados,

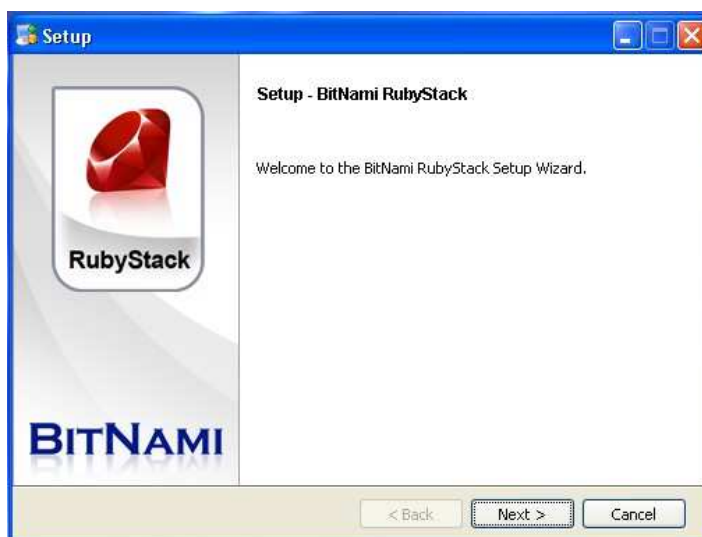
com a exceção de **tabnav.css.erb**, em que foram mudados alguns atributos de apresentação das *tabs* do Rails Widgets.

ANEXO III: MANUAL DE INSTALAÇÃO

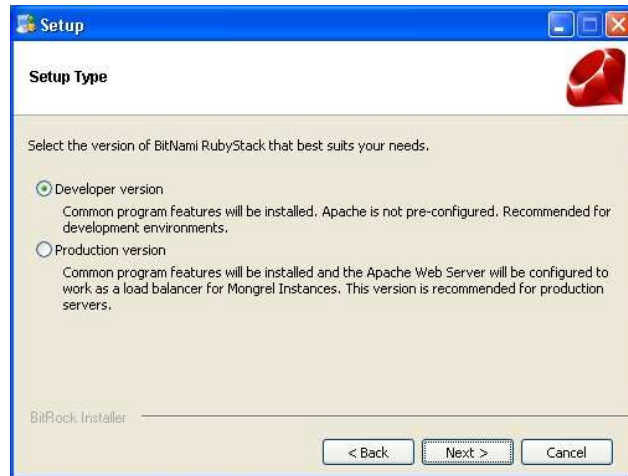
Este manual de instalação pretende descrever todos os passos necessários à correcta instalação da aplicação num computador servidor. Assume-se que para efectuar estes passos, a pessoa encarregue da instalação tem na sua posse o CD com a Ferramenta de simulação de iluminação pública. Estes procedimentos foram testados numa máquina com o sistema operativo Microsoft Windows XP Professional Service Pack 3, mas deverá ser possível instalá-lo de forma semelhante noutras versões do Windows; e também, provavelmente com alguns ajustes e configurações alternativas, em Linux e Mac OS X, dado que o BitNami RubyStack também se encontra disponível em versões particulares para estes sistemas operativos.

1. Instalação do BitNami RubyStack

Executar o instalador fornecido junto com o CD (ou alternativamente, fazer download da última versão disponível em <http://bitnami.org/stack/rubystack>):



Escolher “Developer version”:



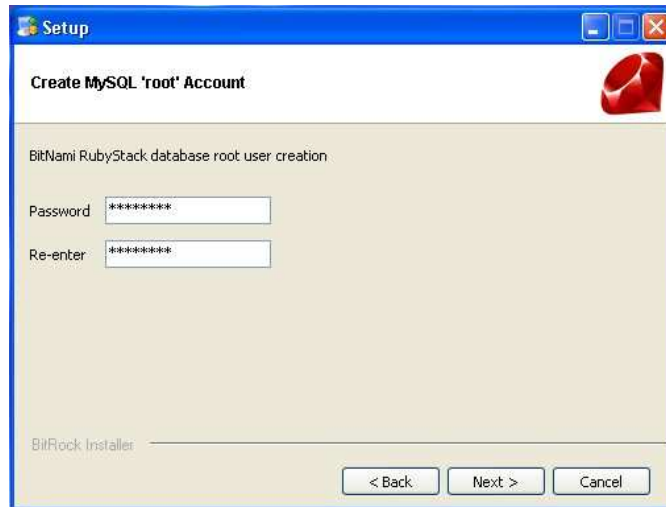
Marcar apenas “Ruby On Rails” (não é necessário instalar PHP Admin):



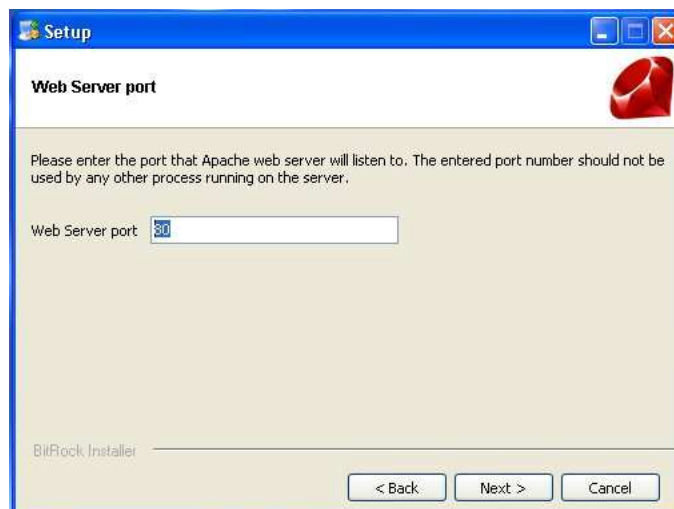
Escolher a pasta onde instalar, que por defeito é C:/Programas/BitNami RubyStack:



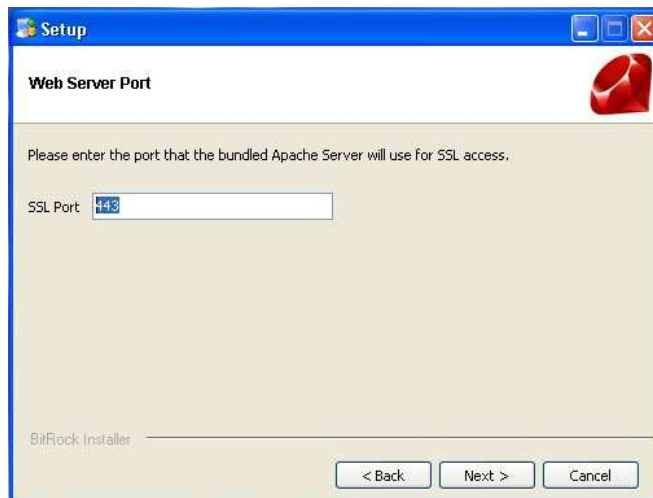
Introduzir a palavra-passe de administrador na base de dados. Nota: esta palavra-passe é importante pois será necessária em vários passos seguintes:



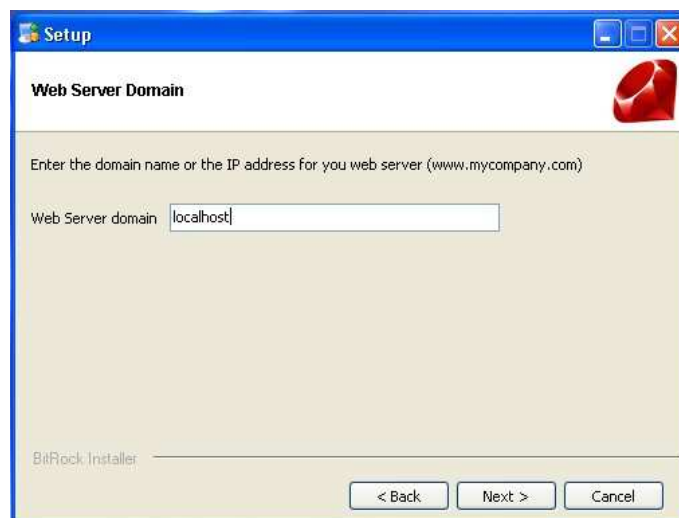
Escolher a porta de pedidos HTTP, que deverá sempre ser configurada como a porta 80:



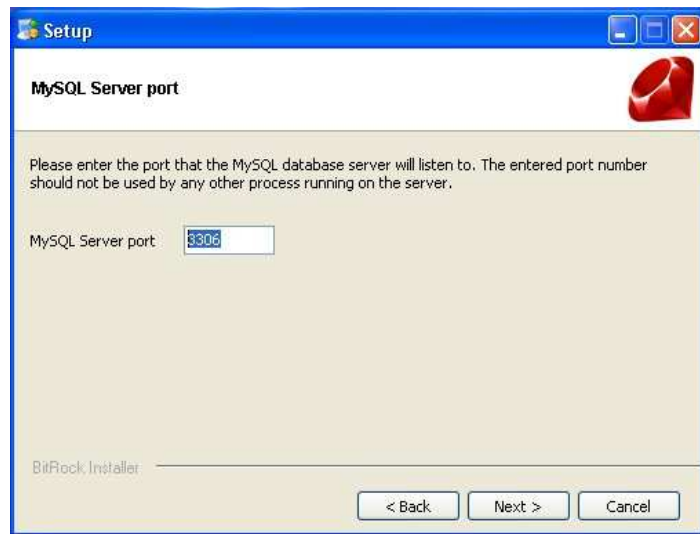
Escolher a porta de pedidos SSL, que deverá ser mantida como a porta **443**:



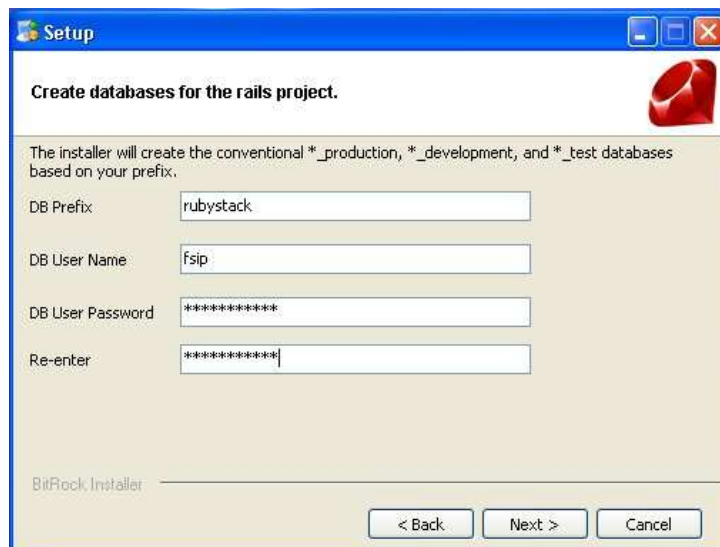
Introduzir o domínio web do servidor em questão (ex: `www.uma.pt`, `www.arem.pt`).
Caso o servidor esteja a ser configurado apenas para testes, poderá ser mantido como `127.0.0.1`, ou `localhost`:



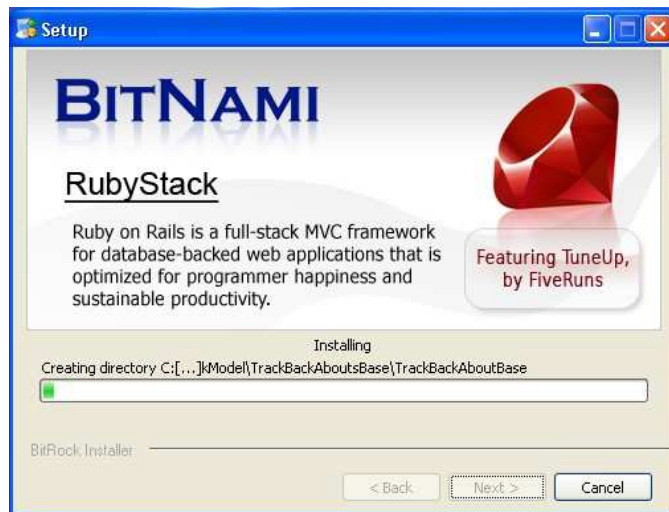
Escolher a porta MySQL, que deverá ser mantida como a porta **3306**:



Introduzir o nome de utilizador e palavra-passe pretendidos, para a base de dados de exemplo fornecida com o BitNami Rubystack. Esta conta de utilizador está apenas ligada a esta base de dados e não tem qualquer influência na aplicação a instalar:



Irá então proceder-se à instalação, que poderá demorar alguns minutos:



No final, não é necessário lançar imediatamente o BitNami RubyStack em execução.

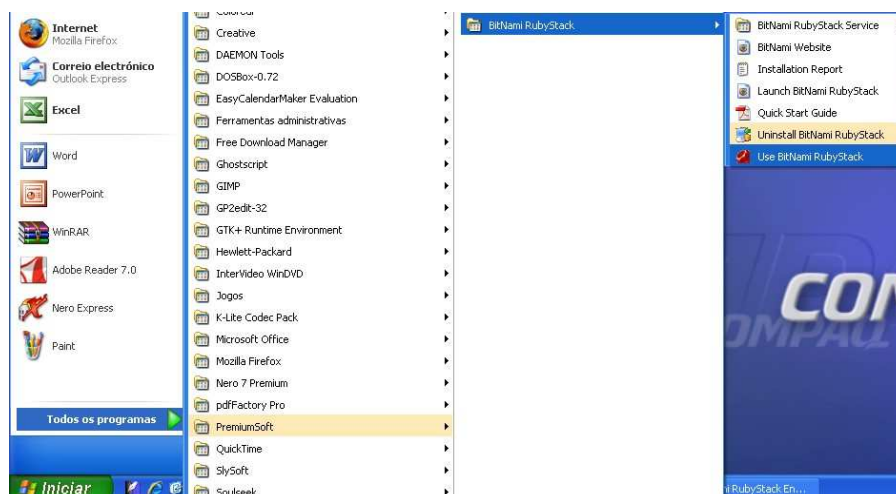
2. Instalar os ficheiros da aplicação

Copiar o conteúdo da pasta “fsip” (com todo o código-fonte e configurações da aplicação) a partir do CD para C:\Documents and Settings\Administrador\BitNami RubyStack projects\.

3. Instalar a versão do Rails adequada

A aplicação foi desenvolvida sobre Rails 1.2.5, e não é 100% compatível com versões mais recentes, que são disponibilizadas nas novas distribuições do BitNami RubyStack. Portanto, é necessário instalar a versão 1.2.5.

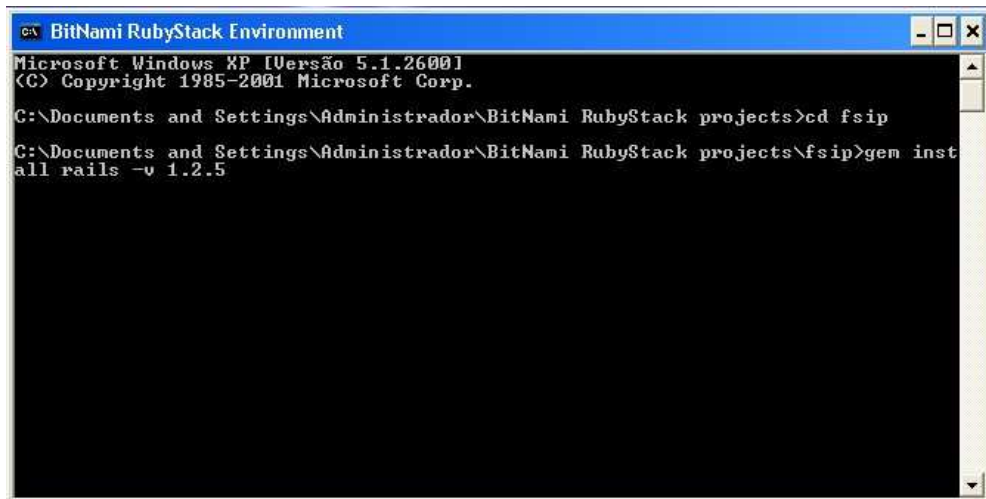
Lançar a consola no ambiente RubyStack (através de “Use BitNami RubyStack”):



E depois introduzir os seguintes comandos na consola:

```
cd fsip
```

```
gem install rails -v 1.2.5
```

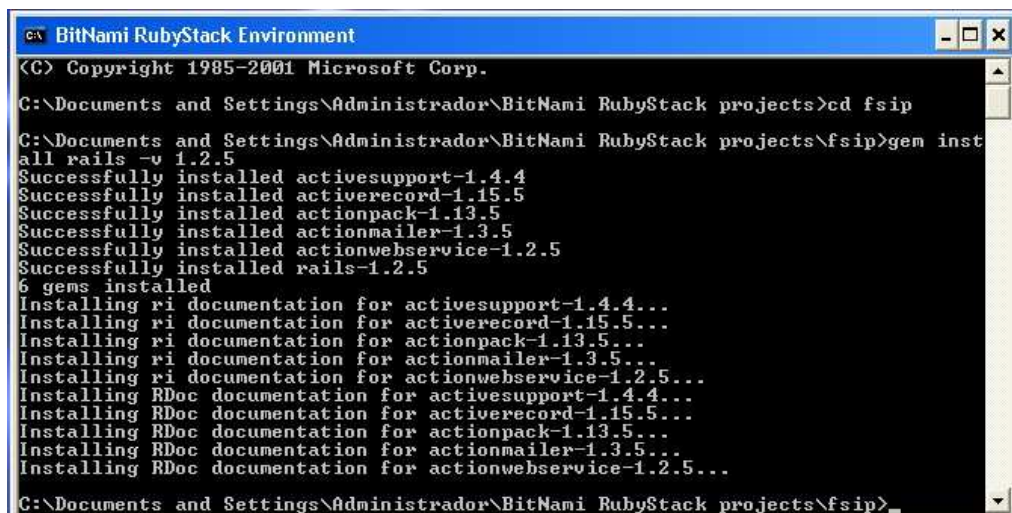


```
BitNami RubyStack Environment
Microsoft Windows XP [Versão 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrador\BitNami RubyStack projects>cd fsip

C:\Documents and Settings\Administrador\BitNami RubyStack projects\fsip>gem inst
all rails -v 1.2.5
```

Começará então o processo de download e instalação desta versão do Rails, o que poderá demorar vários minutos. É necessário que o servidor esteja já ligado à Internet neste ponto:



```
BitNami RubyStack Environment
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrador\BitNami RubyStack projects>cd fsip

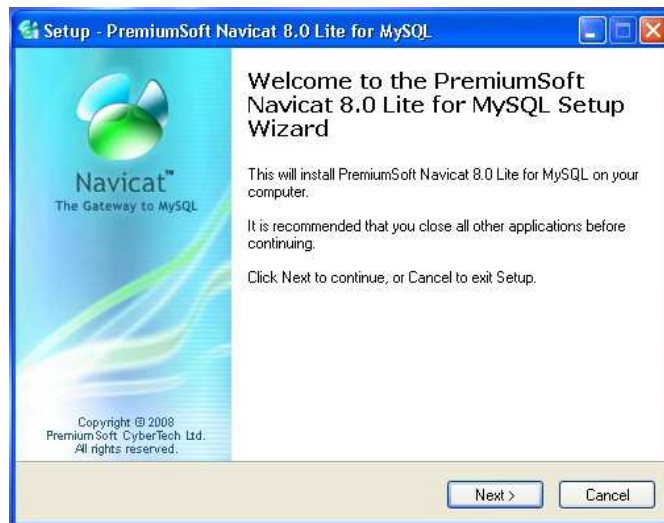
C:\Documents and Settings\Administrador\BitNami RubyStack projects\fsip>gem inst
all rails -v 1.2.5
Successfully installed activesupport-1.4.4
Successfully installed activerecord-1.15.5
Successfully installed actionpack-1.13.5
Successfully installed actionmailer-1.3.5
Successfully installed actionwebservice-1.2.5
Successfully installed rails-1.2.5
6 gems installed
Installing ri documentation for activesupport-1.4.4...
Installing ri documentation for activerecord-1.15.5...
Installing ri documentation for actionpack-1.13.5...
Installing ri documentation for actionmailer-1.3.5...
Installing ri documentation for actionwebservice-1.2.5...
Installing RDoc documentation for activesupport-1.4.4...
Installing RDoc documentation for activerecord-1.15.5...
Installing RDoc documentation for actionpack-1.13.5...
Installing RDoc documentation for actionmailer-1.3.5...
Installing RDoc documentation for actionwebservice-1.2.5...

C:\Documents and Settings\Administrador\BitNami RubyStack projects\fsip>
```

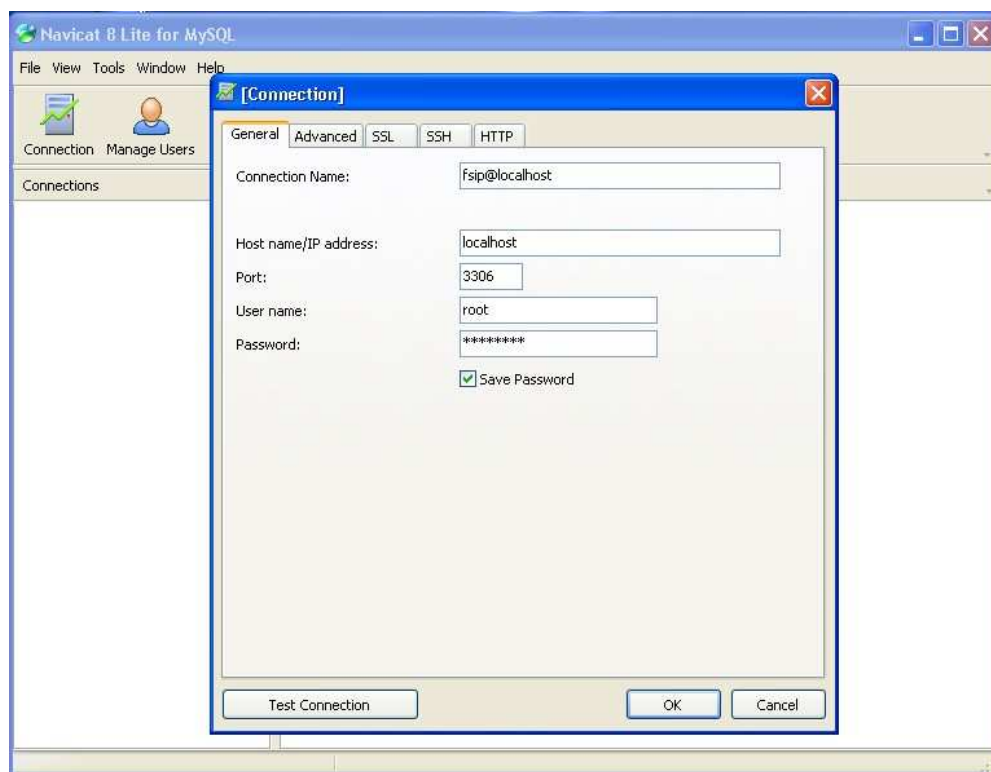
4. Instalar a base de dados

Para tal, recomenda-se a utilização do Navicat for MySQL, uma ferramenta de administração de base de dados que oferece uma interface gráfica. Tal como o BitNami RubyStack, esta aplicação é fornecida no CD da Ferramenta de simulação da iluminação pública, na sua versão livre para fins não-comerciais - mas também é

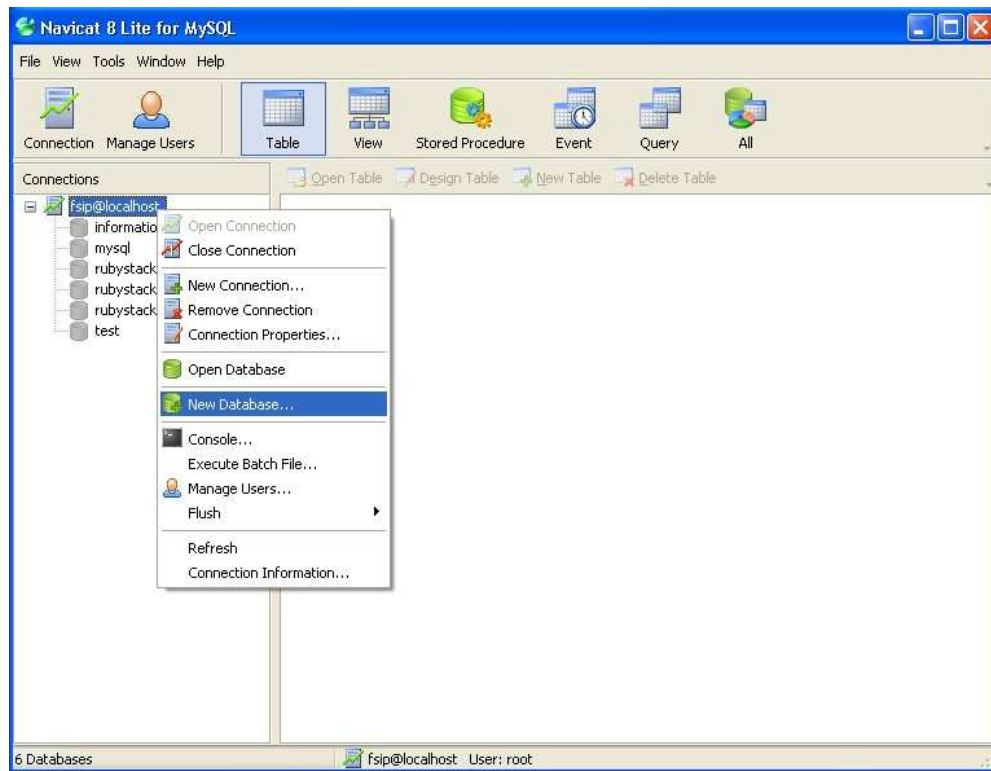
possível fazer o download da última versão em <http://mysql.navicat.com/download.html>.



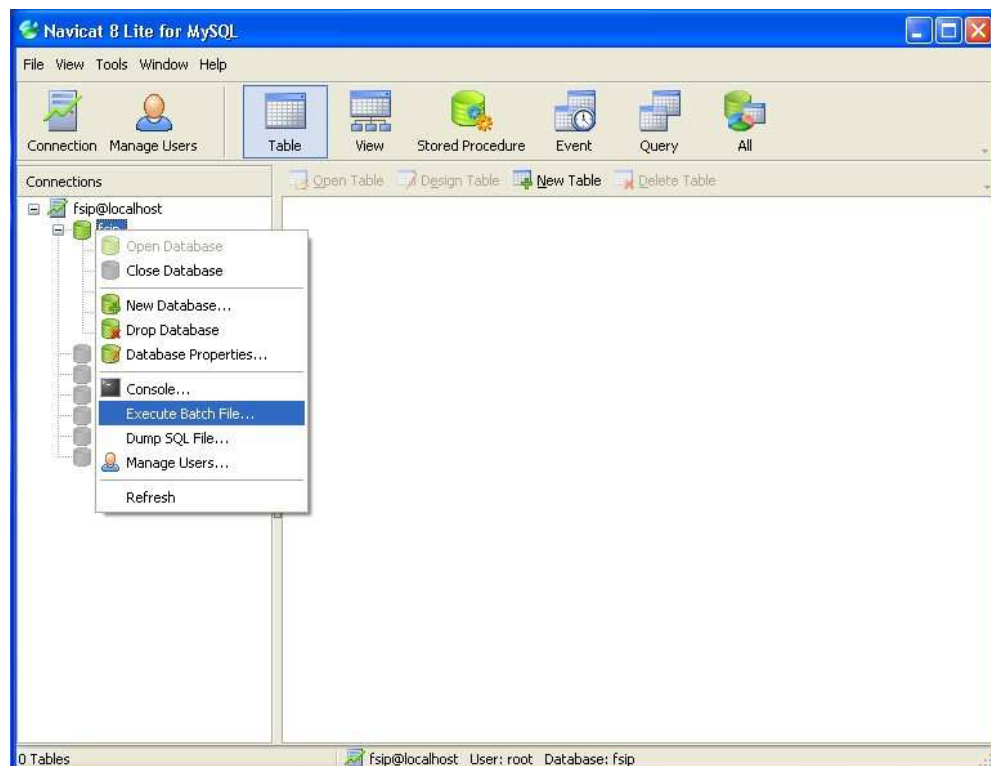
Após instalado o Navicat, é necessário criar uma ligação à base de dados através de File -> New connection. O nome desta ligação é irrelevante, mas os outros dados têm de estar correctos - host name: **localhost**, port: **3306**, username: **root**, e a password é a password introduzida anteriormente nas configurações do BitNami RubyStack:



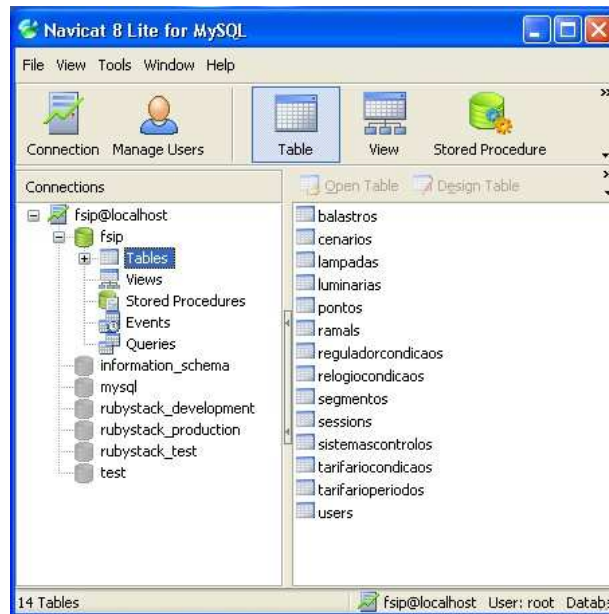
Tendo a ligação a funcionar, é necessário criar a base de dados. Para tal, utilizamos a opção “New database” e o nome da base de dados será **fsip**:



Neste ponto podemos então popular a base de dados com as tabelas e os campos mínimos necessários ao funcionamento de aplicação. Para tal, é necessário o ficheiro construir-bd.sql, fornecido junto com o CD. Através da opção “Execute batch file” sobre a base de dados criada, escolher o ficheiro construir-bd.sql:



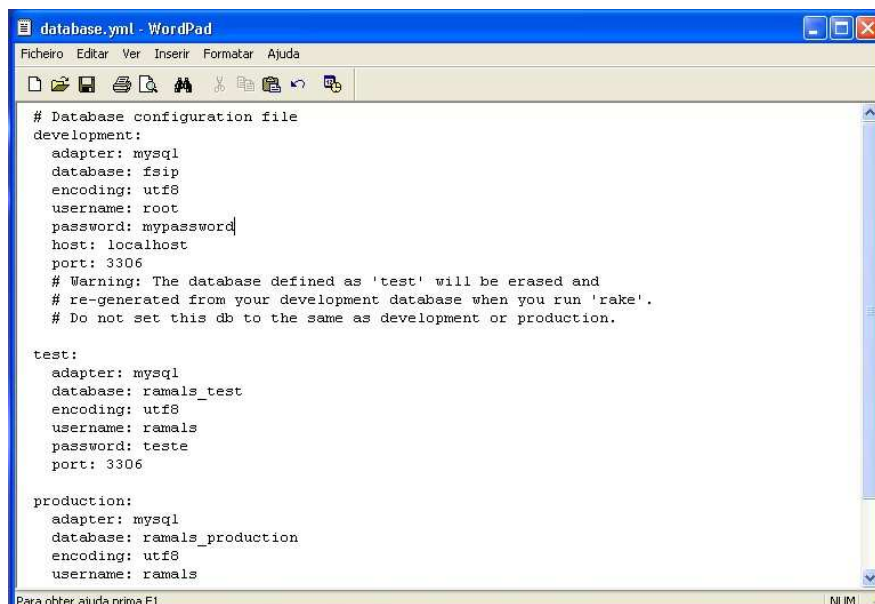
A base de dados ficará pronta para ser utilizada:



5. Editar as configurações de acesso à base de dados

É necessário editar as configurações da aplicação para permitir o acesso à conta de utilizador da base de dados MySQL, definida aquando da instalação do BitNami Rubystack.

Para tal, abrir o ficheiro C:\Documents and Settings\Administrador\BitNami RubyStack projects\fsip\config\database.yml e editar a password nas configurações da base de dados “development”, que mais uma vez deverá corresponder à password utilizada anteriormente:



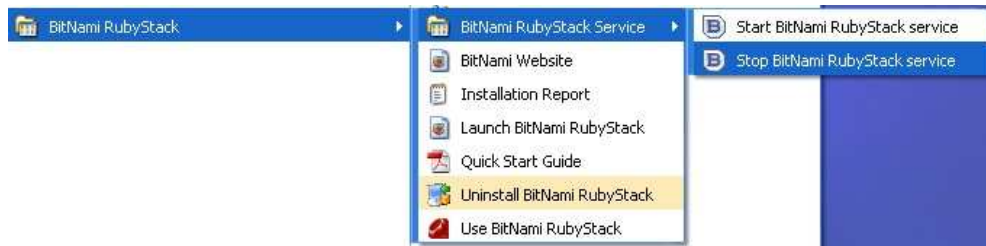
```
# Database configuration file
development:
  adapter: mysql
  database: fsip
  encoding: utf8
  username: root
  password: mypassword
  host: localhost
  port: 3306
  # Warning: The database defined as 'test' will be erased and
  # re-generated from your development database when you run 'rake'.
  # Do not set this db to the same as development or production.

test:
  adapter: mysql
  database: ramals_test
  encoding: utf8
  username: ramals
  password: teste
  port: 3306

production:
  adapter: mysql
  database: ramals_production
  encoding: utf8
  username: ramals
```

6. Reiniciar o serviço BitNami RubyStack

Este passo poderá prevenir alguns erros. Pode ser feito através de “Stop BitNami RubyStack service” seguido de “Start BitNami RubyStack service”.

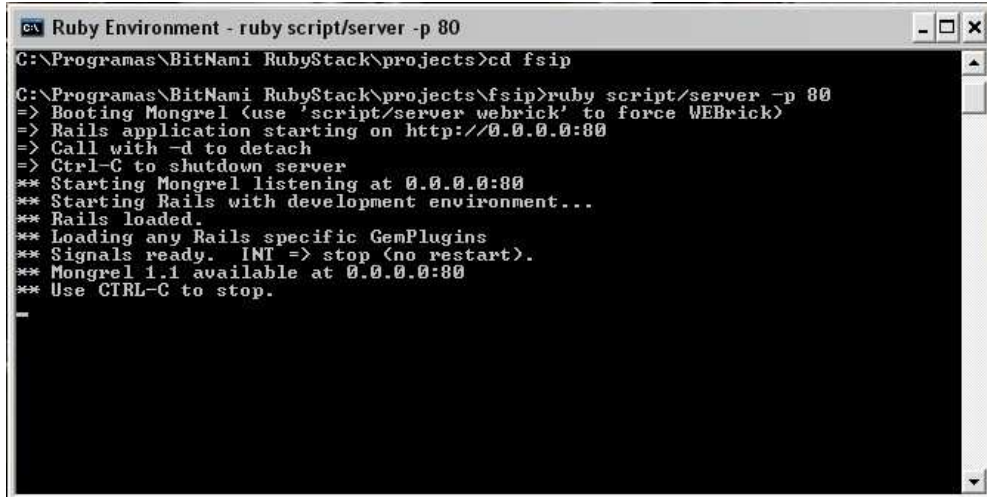


7. Lançar a aplicação

A aplicação deverá estar pronta para ser lançada. Primeiro há que lançar a consola, como explicado anteriormente no passo 3), e introduzir os seguintes comandos, que lançam o servidor:

```
cd fsip
```

```
ruby script/server -p 80
```

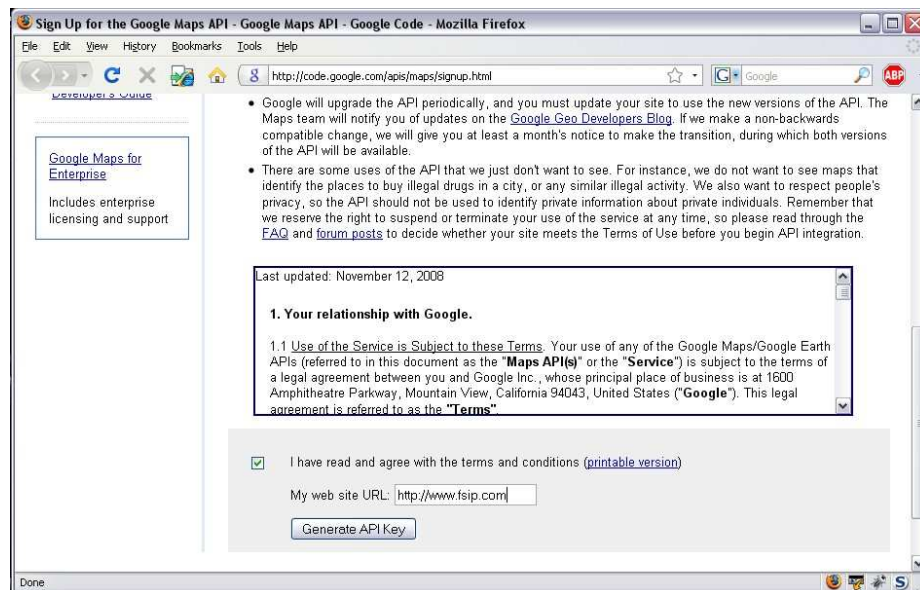
A screenshot of a terminal window titled 'Ruby Environment - ruby script/server -p 80'. The terminal shows the following output:

```
C:\Programas\BitNami RubyStack\projects>cd fsip
C:\Programas\BitNami RubyStack\projects\fsip>ruby script/server -p 80
=> Booting Mongrel (use 'script/server webrick' to force WEBrick)
=> Rails application starting on http://0.0.0.0:80
=> Call with -d to detach
=> Ctrl-C to shutdown server
** Starting Mongrel listening at 0.0.0.0:80
** Starting Rails with development environment...
** Rails loaded.
** Loading any Rails specific GemPlugins
** Signals ready. INT => stop (no restart).
** Mongrel 1.1 available at 0.0.0.0:80
** Use CTRL-C to stop.
```

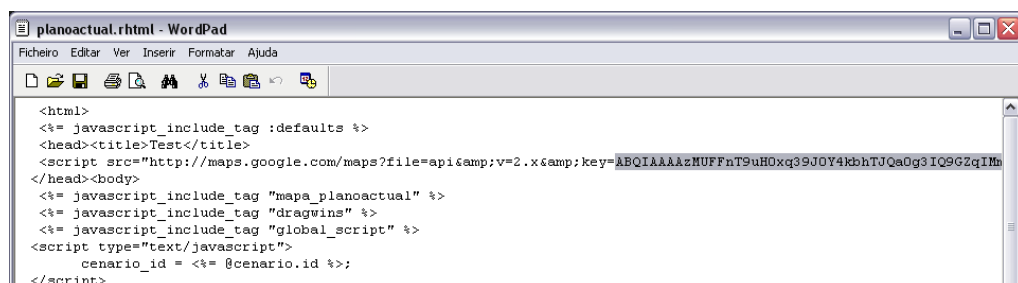
Neste momento a aplicação está acessível no browser através do endereço <http://localhost>. Não existem quaisquer utilizadores na base de dados da aplicação (nota: este é o sistema de autenticação interno da FSIP, e completamente independente dos nomes de utilizadores mencionados até agora neste manual), portanto o administrador do sistema deverá registar-se pela interface da aplicação como se fosse um utilizador normal. O nível de permissões deverá depois ser alterado manualmente na tabela “users” da base de dados – os administradores deverão ter o valor “4” na coluna “permissions”, que corresponde ao nível de permissões mais alto.

8. Geração de uma chave na API do Google Maps

O último passo necessário para pôr a aplicação em funcionamento é a geração de uma chave na API do Google Maps para o domínio em que se pretende que a aplicação seja acedida, ex: www.uma.pt ou www.areas.pt (de momento esta limitação faz com que seja apenas possível aceder à aplicação a partir de um único domínio). A chave na API do Google Maps pode ser gerada em <http://code.google.com/apis/maps/signup.html>:



A chave obtida deverá agora substituir a chave antiga existente nos ficheiros <C:\DocumentsandSettings\Administrador\BitNamiRubyStackprojects\fsip\app\views\cenario\edit.rhtml> e <C:\DocumentsandSettings\Administrador\BitNamiRubyStackprojects\fsip\app\views\fsip\planoactual.rhtml> - a linha <http://maps.google.com/maps?file=api&v=2.x&key=> deverá ser seguida da nova chave:



Após gravadas estas alterações, a aplicação deverá estar 100% funcional.

ANEXO IV: BIBLIOGRAFIA

- [1] <http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=43487> - QuickStudy: Application Programming Interface (API)
- [2] <http://www.codeproject.com/KB/architecture/WhatIsAFramework.aspx> - CodeProject: What Is A Framework?
- [3] http://www.jfwk.com/what_is.html - JFramework (JFW) - What is a Framework?
- [4] www.gis.com/whatisgis - What Is GIS?
- [5] <http://www.opensource.org/docs/definition.php> - The Open Source Definition (Annotated)
- [6] http://www.webopedia.com/TERM/p/plug_in.html - What is plug-in? - A Word Definition From The Webopedia Computer Dictionary
- [7] <http://codeidol.com/ajax/ajaxdp/Page-Architecture/Popup/> - CodeIdol - Thinking about Ajax Design Patterns - Page Architecture - Popup
- [8] United Nations, "United Nations Framework Convention On Climate Change", 1992
- [9] Instituto do Ambiente, "Portuguese National Inventory Report On Greenhouse Gases, 1990-2004", 2006
- [10] <http://code.google.com/apis/maps/> - Google Maps API - Google Code
- [11] <http://www.google.com/enterprise/maps/> - Google Maps API Premier
- [12] <http://earth.google.com/comapi/> - Google Earth COM API Documentation
- [13] <http://mapas.sapo.pt/> - SAPO Mapas
- [14] <http://maps.yahoo.com/> - Yahoo! Maps

- [15] <http://maps.live.com/> - Live Search Maps
- [16] <http://www.mapquest.com/> - MapQuest Maps
- [17] <http://grass.itc.it/> - GRASS GIS
- [18] <http://www.qgis.org/> - QGIS Community
- [19] <http://www.gvsig.gva.es/index.php?id=gvsig&L=2> - gvSIG
- [20] <http://www.autodesk.com> - Autodesk
- [21] <http://bitnami.org/stack/rubystack> - BitNami RubyStack
- [22] <http://ym4r.rubyforge.org/> - YM4R
- [23] <http://wiki.rubyonrails.org/rails/pages/UnderstandingRailsMVC> - Understanding Rails MVC in Ruby on Rails
- [24] Nunes N., "Modelação por Objectos para o Desenvolvimento Centrado nos Utilizadores e o Desenho de Interfaces com o Utilizador", 2001
- [25] Constantine L., "Canonical Abstract Prototypes for Abstract Visual and Interaction Design", 2003
- [26] <http://www.pr.gov.br/batebyte/edicoes/2001/bb115/requisitos.htm> - Zanolenci E., Burnett R., "Requisitos funcionais e não-funcionais, as duas faces da moeda aplicáveis à engenharia de software"
- [27] Constantine L., "Users, Roles, and Personas", 2006
- [28] Constantine L., "Activity Modeling: Toward a Pragmatic Integration of Activity Theory with Usage-Centered Design", 2006
- [29] Constantine L., "Structure and Style in Use Cases for User Interface Design", 2000
- [30] <http://instantrails.rubyforge.org/tutorial/index.html> - Rolling with Ruby on InstantRails
- [31] <http://www.seesaw.it/en/toolbox/widgets/> - Rails Widgets
- [32] <http://www.welie.com/patterns/showPattern.php?patternID=main-navigation> - Main Navigation - Interaction Design Pattern Library - Welie.com

[33] <http://www.welie.com/patterns/showPattern.php?patternID=crumbs> -
Breadcrumbs - Interaction Design Pattern Library - Welie.com

[34] <http://www.devsource.com/c/a/Techniques/Draggable-Collapsible-Dynamic-DIV-windows-with-JavaScript/> - Draggable, Collapsible, Dynamic DIV windows with
Javascript

[35] <http://www.prototypejs.org/> - Prototype JavaScript framework: Easy Ajax and
DOM manipulation for dynamic web applications

[36] The Pragmatic Programmers, *"Agile Web Development With Rails"*, 2nd Edition,
2006