

Aplicação SIG: Gestão de Pontos de Interesse de Entidades

José Henrique Estêvão Gonçalves do Souto

Trabalho realizado sob a orientação de
José Luís Padrão Exposto

Mestrado de Sistemas de Informação

2011/2012

Aplicação SIG: Gestão de Pontos de Interesse de Entidades

Relatório da UC de Projeto
Mestrado de Sistemas de Informação
Escola Superior de Tecnologia e de Gestão

José Henrique Souto
2011/2012

A Escola Superior de Tecnologia e Gestão não se responsabiliza pelas opiniões expressas neste relatório.

Certifico que li este relatório e que na minha opinião, é adequado no seu conteúdo e forma como demonstrador do trabalho desenvolvido no âmbito da UC de Projeto.

José Luís Padrão Exposto Orientador

Certifico que li este relatório e que na minha opinião, é adequado no seu conteúdo e forma como demonstrador do trabalho desenvolvido no âmbito da UC de Projeto.

Arguente

Aceite para avaliação da UC de Projeto

Dedicatória

Aos meus pais e irmãos.

Agradecimentos

Aquando o início do meu percurso no mestrado, o mesmo tinha sido idealizado como retilíneo e uniforme, no entanto transformou-se num caminho sinuoso e inconstante. De tal modo que quero e devo fazer os seguintes agradecimentos:

Ao Prof. Doutor José Luís Exposto, por ter aceite ser meu orientador de projeto e durante esta longa caminhada ter partilhado conhecimentos tecnológicos, ter dado sugestões e ideias de implementação e principalmente pela compreensão que sempre manifestou durante o desenvolvimento desta solução pela minha dificuldade em conciliar as atividades académica e profissional e pelo constante apoio, motivação e por mostrar-me que a conclusão do mestrado estava mais próxima do que eu acreditava. Um muito obrigado Professor!

Ao colega de trabalho Nuno Alves pela partilha de conhecimentos na área SIG, ideias e sugestões.

Por fim quero agradecer às pessoas mais importantes da minha vida:

Aos meus pais, José e Maria de Fátima, por todo o carinho e amor que sempre demonstraram para comigo e pelo apoio constante e incondicional que sempre me deram.

Às minhas irmãs, Ana Cristina e Ângela, pelo carinho e amor que sempre me deram e pela união que sempre demonstrámos.

À minha avó Ana, pelo carinho e amor que sempre demonstrou e por mostrar que presentemente na sua juventude de 90 anos é uma irrequieta.

Ao meu saudoso avô Felisberto (*in memoriam*), por quem tenho muitas boas lembranças e orgulho.

Ao meu afilhado Martinho, por ter “entendido” a minha ausência nestes últimos meses, o que não me permitiu durante este tempo ensinar-lhe a dar toques na bola como um jogador profissional.

Para não correr o risco de me esquecer de alguém, um simples e sentido obrigado a todos!

Resumo

O uso de aplicações georreferenciadas e as ferramentas que permitem saber onde se encontram pontos de interesse estão cada vez mais disseminadas. No entanto, nem todas disponibilizam os serviços de categorização e gestão de pontos de interesse e respetiva georreferenciação com a possibilidade de consulta por parâmetros e geração de percursos sobre os mesmos pontos. Além disto, é mais raro encontrar aplicações deste tipo que também faça gestão de entidades (instituições, organizações ou empresas), associadas à respetiva gestão de pontos de interesse e política de privilégios de interação dos utilizadores sobre os pontos de interesse georreferenciados no mapa. Para tal, neste projecto implementou-se um sistema de informação geográfica (SIG), recorrendo à API do Google Maps, composto por um portal de administração (backoffice) onde se possa categorizar pontos de interesse e para as entidades personalizar essa gestão de pontos de interesse de acordo com permissões de grupos e utilizadores; na caracterização de pontos de interesse também é possível configurar quais os campos informativos a serem apresentados nos respetivos tooltips/speechbubbles sobre os mapas, além dos métodos de georreferenciação *geocoding* e *reverse geocoding*. Para o utilizador final criou-se um portal de mapas (frontoffice) que permite interagir e visualizar os respetivos pontos de interesse em função de uma determinada pesquisa e ainda ser possível gerar percursos sobre um determinado conjunto de pontos de interesse.

Palavras-chave: SIG, pontos de interesse, entidades, percursos, georreferenciação, tooltip, *geocoding*, *reverse geocoding*, backoffice, frontoffice.

Abstract

The use of geo-referenced applications and tools that enable the knowledge about the location of points of interest are increasingly widespread. However, not all provide services for the categorization and management of waypoints and its geo-referenced with the ability to query parameters and for the generation route on that points of interest. Moreover, it is rarer to find applications of this type which also makes the management of entities (institutions, organization or enterprises) associated with its points of interest and privileges interaction on the georeferenced points of interest on the map. To this end, this project implemented a geographic information system (GIS) consisting of an administration portal (back office) where it is possible to categorize points of interest and to customize certain entities that manage waypoints according to permissions of users and groups; characterization of the points of interest it is possible to configure which fields of information should be shown at their respective tooltips -/- speechbubbles on maps, in addition to methods of geocoding and reverse geocoding geotagging. The existence of a portal maps (frontoffice) allows the user to interact and view their points of interest according to a given search and still be able to generate routes on a given set of points of interest.

Keywords: GIS, points of interest, entities, pathways, geotagging, tooltip, geocoding, reverse geocoding, backoffice, frontoffice.

Conteúdo

1	Introdução	1
1.1	Motivação	2
1.2	Objetivos	3
1.3	Estrutura do documento	4
2	Enquadramento	6
2.1	Sistemas de informação geográfica	6
2.1.1	Estado da arte	7
2.1.2	Áreas de aplicação e apoio à tomada de decisão	9
2.1.3	Modelos de dados de mapas	11
2.1.3.1	Tipos de Mapas	11
2.1.3.2	Local de armazenamento	12
2.1.4	Serviços de mapas	13
2.2	Pontos de interesse	17
2.2.1	Tipos de aplicações	18
2.3	Algoritmos de planeamento de percursos	21
3	Análise e Modelação	23
3.1	Análise do problema	23
3.2	Diagrama de camadas	26
3.3	Diagrama de classes	28
3.4	Diagramas de casos de uso	30
3.5	Diagrama de sequência	39
3.6	Diagramas de estrutura de site	40
4	Tecnologias	41
4.1	Framework .NET 4.0	41
4.2	Arquitetura multicamada	44
4.3	SQL Server	47
4.4	Entity Framework	47
4.5	LINQ	49
4.6	C#	50
4.7	WCF	51
4.8	ASP.NET	53

4.9	XML.....	54
4.10	JavaScript.....	55
4.11	jQuery	56
4.12	Ajax	57
4.13	JSON.....	58
4.14	API Google Maps	59
4.15	Log4net.....	60
5	Implementação.....	62
5.1	Estrutura do ambiente de desenvolvimento	62
5.2	BackOffice	64
5.2.1	Autenticação e API Membership	64
5.2.2	Entidades	65
5.2.3	Grupos	67
5.2.4	Utilizadores	70
5.2.5	Categorização POI.....	73
5.2.6	Pontos de Interesse	75
5.3	FrontOffice.....	80
5.3.1	Pesquisas	81
5.3.2	Percursos	83
5.3.3	Categorias.....	84
6	Conclusões e trabalho futuro.....	85
6.1	Conclusões	86
6.2	Trabalho futuro	87

Lista de Tabelas

Tabela 2.1 - Comparação de características de modelos de dados de mapas.....	13
Tabela 2.2 - Comparação entre Google Maps, Yahoo! Maps e Bing Maps	16
Tabela 3.1 - Descrição das tabelas da solução SIG.....	29
Tabela 4.1 - Comparação entre Java e C#.....	50
Tabela 4.2 - Comparação de sintaxe entre JavaScrit e jQuery.....	57
Tabela 4.3 - Descrição dos níveis de registo do Log4net.....	60
Tabela 5.1 - Grupos e respetivas permissões	67

Lista de Figuras

Figura 2.1 - SIG da concessionária de autoestradas ASCENDI	9
Figura 2.2 - SIG do município de Lisboa - Lisboa Interactiva	10
Figura 2.3 - Interface do Google Maps	14
Figura 2.4 - Interface de Google Maps, GoogleEarth, Yahoo! Maps e Bing Maps.....	15
Figura 2.5 - Interface do Sapo Mapas	17
Figura 2.6 - Pesquisa de pontos de interesse em GIS Studio	19
Figura 2.7 - Alojamentos representados pela aplicação Sakura House.....	20
Figura 2.8 - População mundial representada pela Gapminder World	21
Figura 3.1 - Diagrama de camadas.....	27
Figura 3.2 - Diagrama de classes	28
Figura 3.3 - Diagrama de casos de uso do backoffice de funcionalidades de acesso	30
Figura 3.4 - Diagrama de caso de uso do backoffice - pontos de interesse (administrador)....	32
Figura 3.5 - Diagrama de caso de uso do backoffice - pontos de interesse (Administrador_<Entidade>)	34
Figura 3.6 - Diagrama de caso de uso do backoffice - pontos de interesse (Convidado)	36
Figura 3.7 - Diagrama de caso de uso do backoffice - pontos de interesse (Convidado_<Entidade>)	37
Figura 3.8 - Diagrama de caso de uso do frontoffice	38
Figura 3.9 - Diagrama de sequência da criação de ponto de interesse	39
Figura 3.10 - Diagrama de site do backoffice	40
Figura 3.11 - Diagrama de site do frontoffice	40
Figura 4.1 - Evolução da framework .NET	42
Figura 4.2 - Framework .NET 4.0	43
Figura 4.3 - Arquitetura Multicamadas [ArqCamada].....	45
Figura 4.4 - Enquadramento aplicacional de Entity Framework [ArqEF]	48
Figura 4.5 - Arquitectura do LINQ [ArqLINQ]	49
Figura 4.6 - Endpoint do WCF [EndWCF]	52
Figura 4.7 - Codebehind em ASP.NET [CodeASP]	54
Figura 4.8 - Modelo tradicional web e modelo Ajax [ModAJAX]	58
Figura 4.9 - Geração de mapa com API Google Maps	60
Figura 5.1 - Página Inicial do BackOffice.....	65
Figura 5.2 - Listagem de entidades	66
Figura 5.3 - Listagem de grupos.....	69
Figura 5.4 - Criação de novos grupos	70
Figura 5.5 - Listagem de utilizadores.....	71
Figura 5.6 - Criação de um novo utilizador	72
Figura 5.7 - Categorização de POIs	73
Figura 5.8 - Criação de nova categorização	74
Figura 5.9 - Listagem de pontos de interesse	75
Figura 5.10 - Criação de ponto de interesse (separador “Informação”).....	76
Figura 5.11 - Criação de ponto de interesse (separador “Localização”).....	77

Figura 5.12 - Criação de ponto de interesse (separador “Tooltip”)	78
Figura 5.13 - Consulta de ponto de interesse	79
Figura 5.14 - Pontos de Interesse (Permissões de grupo Convidado_<Entidade>)	80
Figura 5.15 - Pesquisa de Pontos de Interesse (FrontOffice)	81
Figura 5.16 - POI centrado no mapa através de evento da grelha resultados	82
Figura 5.17 - Percurso entre dois pontos de interesse	83
Figura 5.18 - Apresentação de categorização de pontos de interesse	84

Lista de Abreviações

AJAX	Asynchronous Javascript And XML
API	Application Programming Interface
ASP	Active Server Pages
BLL	Business Logic Layer
CAD	Computer Aided Design
CLS	Common Language Specification
CSS	Cascading Style Sheet
DAL	Data Access Layer
DOM	Document Object Model
DLL	Dynamic Link Library
GPS	Global Positioning System
HTML	HyperText Markup Language
JPEG	Joint Photographic Experts Group
JSON	JavaScript Object Notation
LINQ	Language Integrated Query
SGBD	Sistema de Gestão de Base de Dados
SIG	Sistemas de Informação Geográficos
SQL	Structured Query Language
TIF	Tagged Image File
UI	User Interface
WCF	Windows Communication Foundation
XML	eXtensible Markup Language

Capítulo 1

1 Introdução

Os Sistemas de Informação Geográficos (SIG) são sistemas que, recorrendo a meios computacionais cada vez mais sofisticados, capturam, guardam, analisam, gerem e apresentam dados georreferenciados. De tal modo que, estes sistemas estão cada vez mais presentes no nosso dia-a-dia e nas mais diversas áreas: trânsito e transportes, saúde, turismo e lazer, telecomunicações, agricultura, biologia, mercado imobiliário, geomarketing entre outras. O sucesso da aplicação Google Maps, da Google, com as melhorias que regularmente lhe são adicionadas, é um exemplo significativo da importância crescente da visualização de informação georreferenciada, quer para os utilizadores, quer para os produtores e difusores de informação. Isto pode também ser comprovado pelo crescente número de vendas de dispositivos recetores do Global Positioning System (GPS) [Matos 08].

Em particular, tem-se assistido a uma procura superior em soluções de georreferenciação de pontos de interesse que tenham resposta a diferentes pedidos de representação geográfica e que a informação apresentada corresponda às necessidades dos utilizadores, como por exemplo a aplicação nacional Gis Studio [GisStudio 12] da Sapo Mapas, a qual também disponibiliza um módulo de gestão de informação de entidades, e alguns sites municipais, como o do município de Lisboa que não sendo aplicações específicas de pontos de interesse apresentam a sua informação, em diferentes modos de visualização, gestão de escalas, tipologias entre outros componentes. Contudo, existe a necessidade de uma maior oferta de soluções de gestão de pontos de interesse que contemple diferentes módulos de informação, nomeadamente entidades.

No desenvolvimento de aplicações SIG é imperativo a utilização de serviços ou servidores de mapas, nomeadamente o Google Maps que com o recurso da respetiva API (Application

Programming Interface), oferece um vasto leque de ações e interação com mapas. Dos recursos que esta API disponibiliza sobre os mapas, podem-se destacar a colocação de marcadores (“pinos”) associado a “balões” informativos de determinados pontos geográficos, pesquisa desses mesmos pontos e respetiva centragem no mapa a uma determinada escala e cálculo de percursos entre localizações.

Ainda com base na API do Google Maps é possível georreferenciar pontos através de dois métodos, designados de *geocoding* e *reverse geocoding*. Utilizando o *geocoding*, é possível georreferenciar um ponto de interesse pelo modo “tradicional”, ou seja, através de um endereço originado pelo encadeamento de distrito, concelho, freguesia e localidade obtém-se a coordenada correspondente e georreferencia nessa coordenada no mapa o respectivo ponto; ao utilizar-se o *reverse geocoding* é efetuado o processo inverso, ou seja, ao clicar num determinado local do mapa será originada a coordenada correspondente e com base numa estrutura de dados resultante dessa ação é possível extrair os dados correspondentes dos diferentes níveis de endereço.

Com base nas potencialidades e facilidade de utilização da API Google Maps acima enunciadas, fez-se uso das mesmas no desenvolvimento deste projeto, integrando-as e ajustando-as às necessidades dos requisitos das funcionalidades que contém mapas, ou seja, com base na georreferenciação simples de pontos geográficos e geração de percursos entre duas localidades adaptaram-se as respetivas metodologias à georreferenciação de pontos de interesse de entidades e geração de percursos entre dois pontos de interesses da mesma ou diferentes entidades.

1.1 Motivação

As motivações que orientaram e tornaram este projeto tão entusiasmante e desafiador, devem-se essencialmente à procura de respostas e em algo que possa contribuir para um maior número de aplicações de qualidade e inovadoras na área de gestão e georreferenciação de pontos de interesse.

Concretamente, esta solução procura abstrair-se da implementação mais “comum” da gestão de pontos de interesse em que num portal de mapas é efetuada a gestão e disponibilização dos respetivos pontos. Para tal pretende-se concentrar uma gestão de pontos de interesse mais ampla num portal administrativo que contemple a associação de funcionalidades, tal como a gestão personalizada de pontos de interesse de entidades com base numa categorização de

níveis superiores dos mesmos pontos, em que este tipo de categorização é disponibilizado muito pouco nas soluções SIG.

Além disso são oferecidas possibilidades de configurar quais os campos informativos de pontos de interesse a serem apresentados nos respectivos tooltips/speechbubbles sobre os mapas, além dos métodos de georreferenciação *geocoding* e *reverse geocoding*.

Além de tudo o que foi referido, existe ainda uma tentativa de querer melhorar, atualizar e desenvolver novas técnicas que ofereçam ao utilizador formas de interagir e visualizar informação georreferenciada fiáveis, fáceis de interpretar e que destaquem a informação que o utilizador considere mais interessante.

1.2 Objetivos

Com base na motivação, o principal objetivo deste trabalho é apresentar uma solução SIG que ofereça uma maior amplitude de opções de gestão de pontos de interesse e respetiva georreferenciação do que algumas existentes no mercado.

As funcionalidades a serem desenvolvidas tem como objetivo melhorar e inovar, associadas a uma gestão de pontos de interesses com uma gama de opções de caracterização superior à maioria de aplicações existentes do género, disponibilizando para cada gestão de entidades uma gestão de pontos de entidades personalizada com recurso a uma categorização de categorias até três níveis de classificação de categorias, com a particularidade de as respetivas entidades efetuarem as respetivas gestões de pontos de interesse num portal administrativo com recurso a diferenciação de permissões e privilégios de utilizadores e grupos. Outra componente de cariz inovador e diferenciador é a configuração do “balão” informativo que um determinado ponto de interesse apresenta na sua localização no mapa, ou seja, por norma existe a atribuição de um conjunto de campos informativos para todos os pontos de interesses, mas neste projeto o objetivo é disponibilizar a opção de personalizar a atribuição somente dos campos adequados e com valores de modo a evitar valores de campos vazios e conseqüentemente a origem de “balões vazios” nos pontos de interesse. A disponibilização dos métodos *geocoding* e *reverse geocoding*, que possibilita a georreferenciação e aquisição de dados geográficos é outro dos objetivos a realizar neste projeto, visto que na maioria de soluções SIG é utilizado somente o *geocoding*.

Para se alcançar os objetivos propostos, esta solução será assente sobre uma arquitetura tecnológica extensível por camadas e orientada a serviços, de modo a que futuramente, se assim se entender, estender a solução e permitir a reutilização ou extensão de funcionalidades sem necessidade de código redundante e permitir o consumo de serviços para diferentes dispositivos, nomeadamente dispositivos móveis.

1.3 Estrutura do documento

Este documento está estruturado da seguinte forma:

- **Capítulo 1 – Introdução**
Neste capítulo são apresentadas as motivações e objetivos do projeto e a organização deste documento.
- **Capítulo 2 – Enquadramento**
Neste capítulo é apresentado o estado da arte de sistemas de informação geográfica e tipos aplicações que são utilizadas em diferentes áreas operacionais. É apresentado também o enquadramento dos pontos de interesse em SIG e algumas aplicações existentes no mercado. São ainda apresentados alguns algoritmos de pesquisa e de percurso utilizados em sistemas de informação geográfica.
- **Capítulo 3 – Análise e modelação**
Neste capítulo é apresentada a análise do problema e respetiva modelação que inclui diagrama de camada, diagrama de classes, diagramas de caso de uso, diagrama de sequência e diagramas de estrutura de sites.
- **Capítulo 4 – Tecnologias**
Neste capítulo são apresentadas a arquitetura definida para o projeto e respetivas tecnologias utilizadas.

- Capítulo 5 – Implementação

Neste capítulo é apresentada a implementação da solução tecnológica composta pelo backoffice e frontoffice.

- Capítulo 6 – Conclusões e trabalho futuro

Neste capítulo são apresentadas as conclusões deste projeto e ideias para trabalho futuro.

Capítulo 2

2 Enquadramento

Neste capítulo apresentam-se assuntos específicos que foram objeto de estudo e investigação preparatória para aquisição de conhecimentos na área do projeto.

Para Sistemas de Informação Geográfica através do estado de arte faz-se um enquadramento histórico e evolutivo de aplicações SIG, é efetuada a descrição de algumas áreas de aplicação e métodos de obtenção de coordenadas e mapas; de seguida são apresentados alguns serviços de mapas que se podem encontrar na internet.

Para Pontos de Interesse é apresentada uma breve definição e alguns tipos de aplicações com diferentes modos de representação gráfica.

Por fim são apresentados alguns algoritmos de georreferenciação de pontos de interesse enquadrados no planeamento de percursos.

2.1 Sistemas de informação geográfica

Atualmente os Sistemas de Informação Geográfica são utilizados nas mais diversas áreas de trabalho, o que torna difícil a sua definição. A seguir encontram-se algumas das definições mais utilizadas:

“Conjunto de poderosas ferramentas para recolha, armazenamento, organização e selecção, transformação e representação da informação de natureza espacial acerca do mundo real, para um determinado contexto (Burrough, 1986)”.

“Um SIG é uma tecnologia de informação que permite o armazenamento, análise e representação tanto de dados espaciais como de dados não espaciais (Parker, 1988)”.

“Um SIG é um sistema de apoio à decisão que envolve a integração de dados georreferenciados num ambiente orientado para a resolução de problemas (Cowan,1988)”.

“Os SIG são sistemas computacionais usados para armazenar e manipular informação geográfica. São sistemas concebidos para recolher, armazenar e analisar objectivos e fenómenos em relação aos quais a localização geográfica é uma característica importante (Aronoff, 1989)”.

“Um SIG integra cinco componentes principais: hardware, software, dados, pessoas e métodos (ESRI, 1997)”.

Um SIG é um sistema constituído por *hardware* (computador), *software* (programa executável) e um ambiente institucional, com o fim de armazenar, manipular, visualizar e analisar dados de natureza espacial.

2.1.1 Estado da arte

A solução mais antiga, e até hoje mais comum, de resolver problemas de análise de informações espaciais envolve a construção e a utilização de mapas. Embora a técnica de produção de mapas em papel esteja dominada, uma vez que a Cartografia é uma ciência muito antiga, o processo de produção e utilização de mapas é muito oneroso, principalmente considerando-se os aspetos de levantamento de dados em campo, armazenamento e atualização.

As primeiras tentativas de automatizar parte do processamento de dados com características espaciais aconteceram na Inglaterra e nos Estados Unidos, nos anos 50 do século XX, com o principal objetivo de reduzir os custos de produção e manutenção de mapas. Os primeiros Sistemas de Informações Geográficas surgiram na década de 1960, no Canadá, como parte de um esforço governamental para criar um inventário de recursos naturais. Estes sistemas, no entanto, eram muito difíceis de usar: não existiam monitores gráficos de alta resolução, os computadores necessários eram excessivamente onerosos, e a mão-de-obra tinha que ser

altamente especializada e, portanto também muito onerosa. Não existiam sistemas comerciais prontos para uso, e cada interessado precisava desenvolver seus próprios programas, o que acarretava muito tempo e, naturalmente, muitos recursos financeiros. Além disto, a capacidade de armazenamento e a velocidade de processamento eram muito baixas. Ao longo dos anos 70 do século XX, foram desenvolvidos novos e mais acessíveis recursos computacionais, tornando viável o desenvolvimento de sistemas comerciais. Foi então que a expressão Sistema de Informações Geográficas foi criada. Foi também nesta época que começaram a surgir os primeiros sistemas comerciais de CAD (Computer Aided Design, ou Projecto Assistido por Computador), que melhoraram em muito as condições para a produção de desenhos e plantas para engenharia, e serviram de base para os sistemas de cartografia automatizada. Também nos anos 70 foram desenvolvidos alguns fundamentos matemáticos voltados para a cartografia, sendo que o produto mais importante foi a topologia aplicada. Esta nova disciplina permitia realizar análises espaciais entre elementos cartográficos. No entanto, devido aos custos e ao facto destes sistemas ainda utilizarem exclusivamente computadores de grande porte, apenas grandes organizações tinham acesso à tecnologia. No decorrer dos anos 80 do século XX, com o aparecimento e evolução dos computadores pessoais e dos sistemas de gestão de bases de dados, ocorreu uma grande difusão do uso de SIG. A integração de muitas funções de análise espacial proporcionou também um alargamento do leque de aplicações de SIG.

No final da década de 80 e início da década de 90 do século XX, os Sistemas de Informações Geográficas eram orientados a pequenos projetos, considerando-se pequenas áreas geográficas com poucos detalhes, ainda eram precários os dispositivos de armazenamento, acesso e processamento de dados, além disso, somente em grandes empresas era possível encontrar redes de computadores. Desta forma, realizava-se o mapeamento de uma pequena área, inseria-se este mapeamento em computadores, realizavam-se algumas análises e elaboravam-se mapas e relatórios impressos com as informações geográficas desejadas. Em meados da década de 90, com a Internet já em voga, e conseqüentemente também as redes de computadores, os Sistemas de Informações Geográficas puderam ser orientados às empresas e/ou instituições, com a introdução do conceito da arquitetura cliente-servidor e a divulgação das bases de dados. Nesta época também, os programas computacionais de SIG integravam as funções de processamento de imagens digitais.

No final da década de 90 e início do século XXI, os Sistemas de Informações Geográficas começam a se tornarem corporativos e orientados à sociedade, com a utilização da Internet, de

bases de dados geográficos distribuídos e com os esforços realizados em relação a interoperabilidade dos sistemas.

2.1.2 Áreas de aplicação e apoio à tomada de decisão

Os domínios de aplicação de um SIG são tão numerosos como variados. Em função do empenho com que cada área profissional os utiliza assim se verifica o nível de desenvolvimento:

- **Autoestradas:** registo rodoviário, tráfego, sinistralidade, ocorrências, entre outros;
Exemplo: A figura 2.1 ilustra uma aplicação SIG da concessionária de autoestradas ASCENDI.

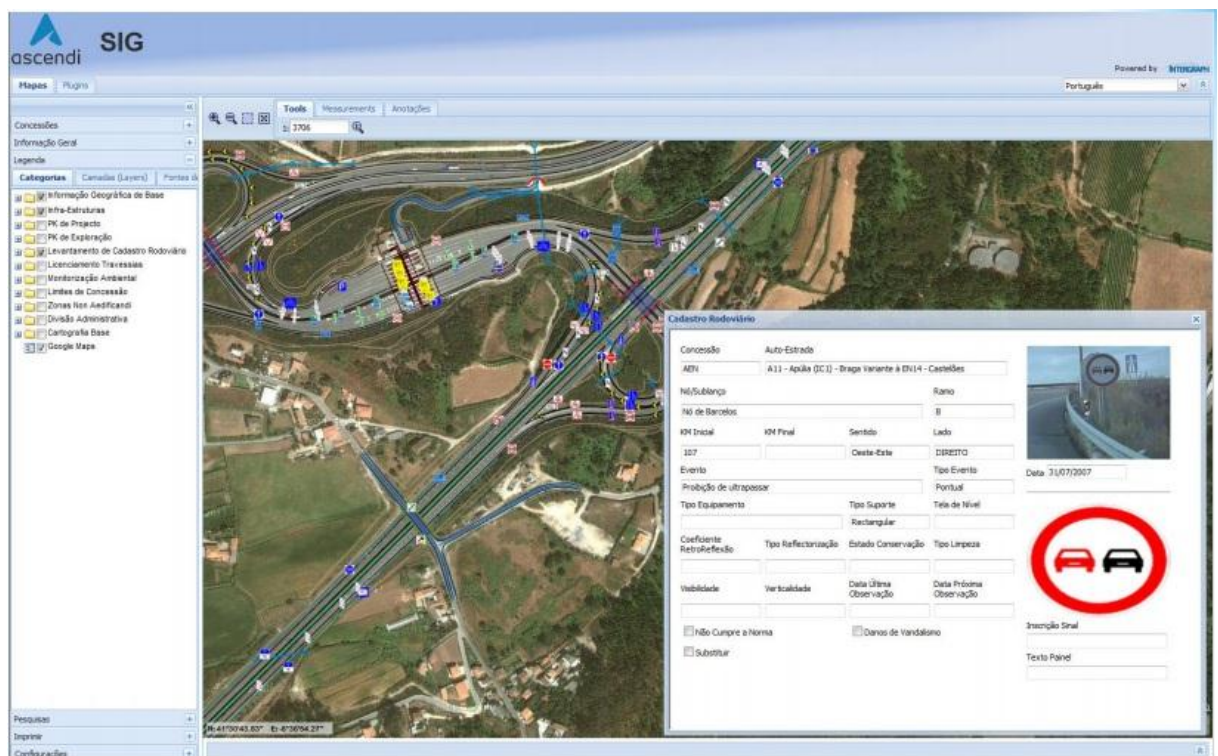


Figura 2.1 - SIG da concessionária de autoestradas ASCENDI

- **Turismo:** planeamento, gestão de infra-estruturas, itinerários turísticos;
- **Marketing:** localização dos clientes, análise da influência dos lugares;
- **Proteção civil:** gestão e prevenção de catástrofes;
- **Transportes:** planeamento dos transportes urbanos, otimização de itinerários;
- **Rede hidrológica:** gestão de aquíferos, prevenção de cheias;

- **Florestas:** localização de postos de vigia, gestão florestal;
- **Geologia:** prospeção mineira;
- **Biologia:** estudo da localização e das deslocações da população animal;
- **Planeamento urbano:** cadastro, ordenamento urbano, pesquisa de endereços;

Exemplo: A figura 2.2 ilustra uma aplicação SIG orientada ao município

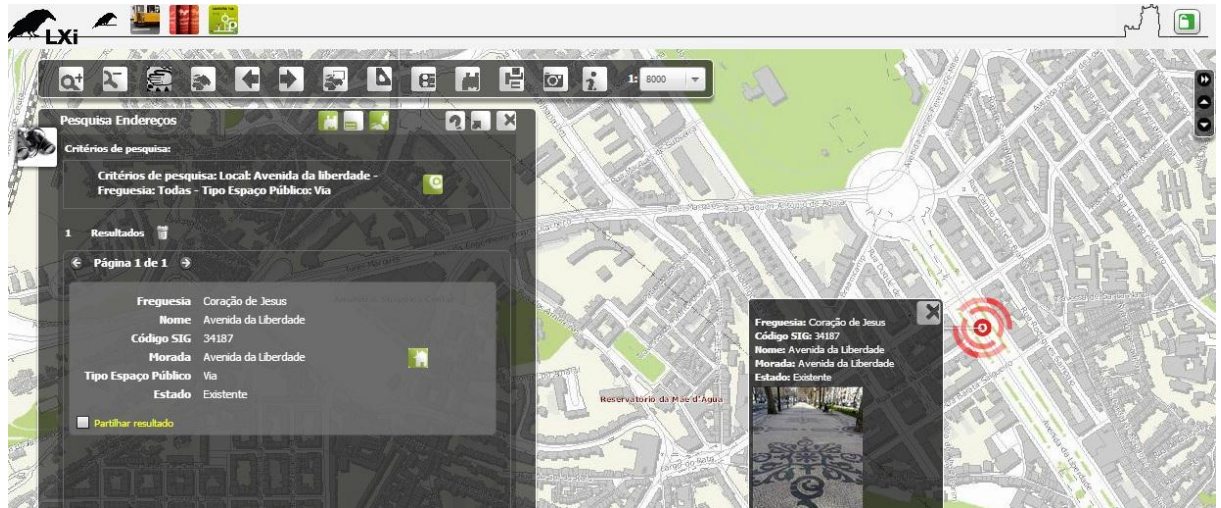


Figura 2.2 - SIG do município de Lisboa - Lisboa Interactiva

- **Telecomunicações:** instalação de antenas;
- **Mercado imobiliário:** gestão de imóveis;
- etc.

Pelo fato de os SIG estarem enquadrados num grande domínio aplicacional pode-se aferir que os mesmos constituem hoje instrumentos modernos indispensáveis e poderosos de gestão e de suporte à tomada de decisão. Em Portugal, os Sistemas de Informação Geográfica são cada vez mais considerados parte integrante de soluções e de projetos inovadores de gestão global de negócio. Um exemplo disso, são os projetos de modernização administrativa da administração pública local, com a disponibilização de informação aos munícipes e ao cidadão, utilizando a Internet, sendo esta representada graficamente em SIG.

Internamente, as autarquias utilizam os sistemas de informação geográfica como suporte das várias áreas de negócio, tais como a Gestão do Urbanismo, Espaços Verdes, Gestão de Redes Viárias, Gestão da Economia Local, entre tantas outras áreas de negócio cuja representação gráfica dos seus elementos representa uma mais-valia para as atividades diárias, como consulta, cruzamento de informação, tomada de decisão, etc. [Sinfic 06]

A utilização dos SIG na administração pública local permite:

- A criação de uma base de conhecimento estruturada e atualizada do concelho;
- Dispor de um instrumento de apoio à elaboração de estudos e projetos com interesse na gestão municipal;
- Dispor de um instrumento de apoio à decisão;
- Dotar os decisores políticos e técnicos de informação estratégica, destinada à definição de políticas gerais de gestão do território municipal e controlar a sua execução.

2.1.3 Modelos de dados de mapas

Uma das primeiras escolhas que é feita quando é desenvolvido um sistema de visualização de informação georreferenciada é o tipo de mapa a ser utilizado, bem como o local onde este vai ser armazenado, estando as duas escolhas interligadas.

2.1.3.1 Tipos de Mapas

Em relação aos tipos de mapas, existem dois à escolha: mapas *raster* e mapas vetoriais.

Um mapa do tipo *raster* é essencialmente uma imagem composta por um mapa de bits, em que a cada pixel da imagem é atribuída uma cor [Foley 97]. Neste tipo de mapas, a imagem final é normalmente composta, através da junção num mosaico, de um conjunto de imagens mais pequenas, que são então, identificadas e indexadas, de acordo com a sua localização, para um acesso mais fácil. As referidas imagens são normalmente armazenadas em formatos de imagem como o TIF ou o JPEG. Os mapas deste tipo têm como principal vantagem o fato de permitirem representar com exactidão imagens muito complexas (por exemplo, fotografias de satélite ou aéreas). As grandes desvantagens são o tamanho considerável de cada imagem, bem como a inexistência de informação sobre aquilo que estão a representar, tornando-se impossível saber, por exemplo, se aquilo que se está a tapar ao sobrepor um símbolo nessa imagem é, ou não, importante.

No caso dos mapas vetoriais, são utilizadas primitivas geométricas como pontos, linhas, curvas e polígonos, que representam os diferentes objetos numa imagem [Foley 97]. Neste caso, todas as características a representar são desenhadas a partir de formas geométricas. Assim, diferentes características são representadas por diferentes tipos de geometria: os pontos são utilizados para representar localizações simples como cidades pequenas; as linhas representam rios, estradas e outras características lineares; os polígonos são utilizados para

representar locais que cobrem uma área considerável, tais como lagos, florestas e grandes cidades.

Adicionalmente, cada uma destas características desenhadas num mapa pode conter informação relevante, tais como informações utilizadas no desenho do mapa (cor, espessura do traço, entre outras) ou informação sobre a característica em si (limites de velocidade de uma estrada, nome, entre outras). Deste modo, um mapa vetorial é constituído por um conjunto de entradas indexadas pela localização que contêm uma determinada característica do mapa.

A grande vantagem dos mapas vetoriais sobre os mapas *raster* consiste na possibilidade de ter informação sobre os objetos que se estão a desenhar. Por conseguinte, é possível, por exemplo, escolher o local adequado onde colocar um símbolo, de modo a não tapar nada de importante. O tamanho reduzido deste tipo de mapas, quando comparado com os mapas *raster* é também uma vantagem. A principal desvantagem dos mapas deste tipo é o fato de ou não serem tão realistas e atraentes para o utilizador, ou requererem uma significativa capacidade gráfica no dispositivo onde vão ser desenhados.

2.1.3.2 Local de armazenamento

Relativamente ao local onde os mapas são armazenados podem existir duas localizações distintas: a memória do dispositivo ou um servidor de mapas. A utilização da memória do dispositivo para guardar os mapas tem como principal vantagem o fato de as imagens estarem todas disponíveis no dispositivo, evitando descarregá-los constantemente de um servidor de mapas (o que requer não só uma ligação disponível, mas também tempo para o fazer). De salientar, contudo, que nem todos os dispositivos dispõem de espaço suficiente para guardar os mapas.

Poder-se-á concluir que o armazenamento em memória é mais indicado para mapas do tipo *raster*, que ocupam muito espaço, enquanto que os servidores de mapas são mais indicados para mapas vetoriais. No entanto, o aparecimento de servidores como o Google Maps [GoogleMaps], que disponibilizam gratuitamente (atualmente com limitação de pedidos ao mapa) os seus mapas (do tipo *raster*) tem provocado uma tendência contrária, tornando muito comum o uso de mapas *raster* descarregados através da Internet. A título ilustrativo, estão referenciados na (Tabela 2.1) alguns itens comparativos dos modelos de dados vetorial e *raster* [Pinto 09].

Tabela 2.1 - Comparação de características de modelos de dados de mapas

	Modelo Vectorial	Modelo Raster
Estrutura	Mais complexa	Mais simples
Topologia	Melhor definição	Fraca
Saídas gráficas	Com qualidade	Fraca qualidade
Manipulação de dados	Fácil	Impossível
Sobreposição de coberturas	Com erros	Fácil
Visualização	Rápida	Lenta

2.1.4 Serviços de mapas

Para o desenvolvimento de aplicações SIG recorre-se a serviços ou servidores de mapas de modo a georreferenciar a informação pretendida. Nas aplicações que envolvam mapas do tipo vetorial normalmente são utilizados servidores de mapas como o Geomedia da empresa Intergraph ou ArcGIS da Esri. No entanto, para mapas do tipo *raster*, no desenvolvimento de sistemas de informação geográfica normalmente recorre-se a serviços de mapas disponíveis na internet. Estes serviços partilham entre si muitas características e funcionalidades que disponibilizam ao utilizador comum e mesmo a desenvolvedores de aplicações SIG, dificultando de certo modo a escolha do serviço de mapas a utilizar. O mais popular é o Google Maps (antigo Google Local) [GMaps] da Google disponível gratuitamente (para uso não comercial e com limitações de pedidos) que alimenta muitos serviços baseados em mapas, incluindo o site do Google Maps, o Google Ride Finder, o Google Transit e mapas embutidos em sites externos através da API do Google Maps. Através de uma interface intuitiva ele oferece mapas de ruas, planeamento de rotas a pé, de carro, bicicleta (versão beta) a partir de pontos de início e de fim, e localização de pontos de interesse. Nas pesquisas de endereços é possível obter informação associada através de dois processos, *geocoding* (obtenção de coordenadas geográficas através de um endereço) e *reverse geocoding* (obtenção de um endereço através de coordenadas geográfica) sendo aplicado sobre essa coordenada um marcador visual.

O serviço também oferece, de forma gratuita para uso não comercial uma API (Application Programming Interface) acessível através da linguagem de programação JavaScript possibilitando que as funcionalidades do Google Maps sejam incorporadas na aplicação web.

O Google Maps utiliza uma variante próxima da projeção de Mercator, por isso não pode mostrar as áreas em torno dos pólos. Um produto relacionado é o Google Earth [GEarth], um programa independente que oferece mais visualização de recursos sobre o globo, incluindo as áreas polares [WikiGMaps12]. A interface do Google Maps é ilustrada na Figura 2.3. Este serviço disponibiliza uma funcionalidade que permite um utilizador criar no próprio mapa pontos de interesse ficando os mesmo registados na sua base de dados.

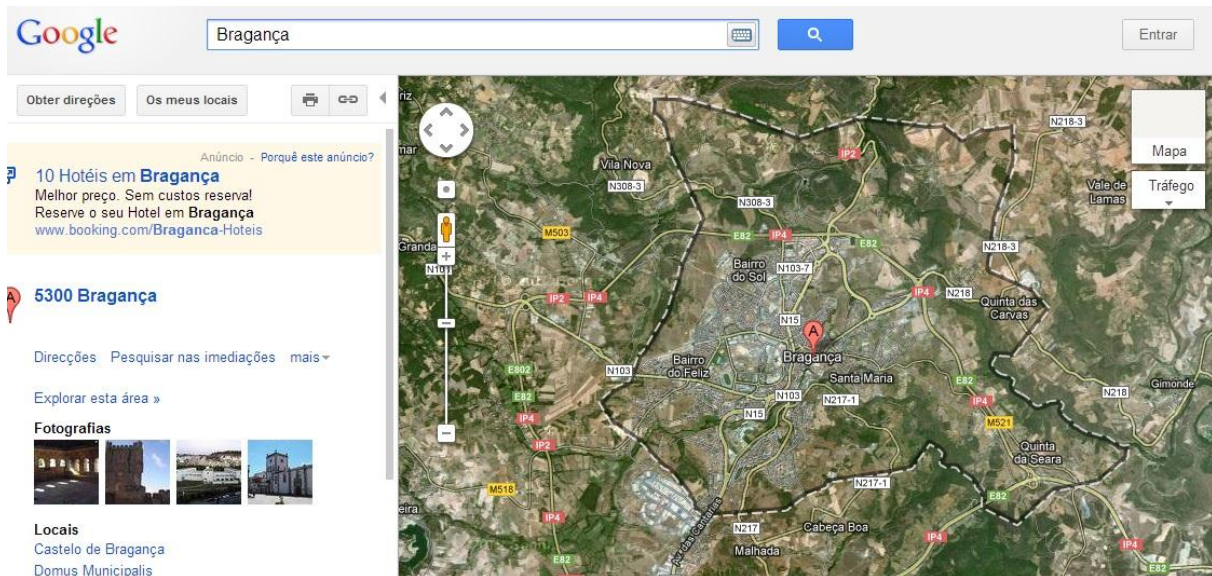


Figura 2.3 - Interface do Google Maps

Há outros serviços de mapas similares ao Google Maps que tem muitas características e funcionalidades em comum, nomeadamente o Bing Maps [BMaps 12] da Microsoft, anteriormente designado de Live Search Maps, Windows Live Maps, Windows Live Local, e MSN Virtual Earth e o Yahoo! Maps [YMaps] desenvolvido pela Yahoo.

O Bing Maps tem uma característica que não se encontra nos outros serviços de mapas; o *bird's-eye*, que oferece uma visão panorâmica de imagens a baixa altitude capturadas pelo seu sistema de dados. Ao contrário da visão de cima para baixo captada por satélite, a panorâmica de imagens é visível obliquamente com um ângulo de 45 graus, o que dá ao utilizador uma melhor percepção de profundidade para edifícios e geografia. Esta visão panorâmica está disponível em locais seleccionados em todo o mundo, incluindo as principais cidades dos Estados Unidos, Canadá, Japão e em mais de 80 locais da Europa. Panorâmica de imagens para um local pode ser visto de todos os quatro pontos cardeais. Estas imagens são

tipicamente muito mais detalhadas do que as vistas aéreas tiradas diretamente de cima [WikiBing 12].

O serviço Yahoo! Maps disponibiliza num painel lateral da interface mais opções e tipos de pesquisa por endereços de uma forma mais intuitiva que os outros serviços de mapas.

Em comum, também se pode afirmar que os serviços de mapas anteriormente descritos apresentam uma interface agradável e intuitiva de acordo com a ilustração da Figura 2.4.

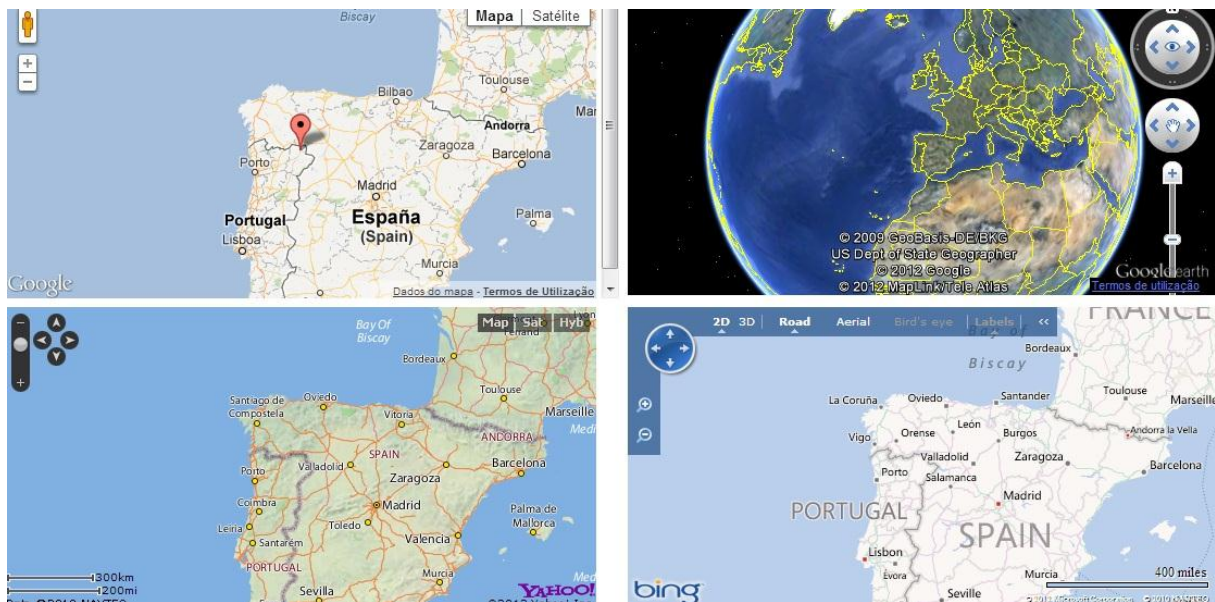


Figura 2.4 - Interface de Google Maps, GoogleEarth, Yahoo! Maps e Bing Maps

Apesar de estes serviços de mapas contarem muitas características e funcionalidades que seguem uma lógica comum; como é natural, existem algumas diferenças entre os mesmos, nomeadamente a quantidade de tipo de mapas disponibilizados e o nível de filtragem. O Bing Maps com dez tipos de mapas, é o que disponibiliza a maior variedade de tipo de mapas em relação ao Google Maps e Yahoo! Maps, que disponibilizam seis e três respetivamente. Em contrapartida o Bing Maps não apresenta qualquer nível de filtragem, enquanto o Google Maps apresenta um e o Yahoo! Maps vários.

A comparação entre estas características e outras dos serviços de mapas em estudo pode ser consultada na (Tabela 2.2).

Tabela 2.2 - Comparação entre Google Maps, Yahoo! Maps e Bing Maps

<i>Características</i>	<i>Google Maps</i>	<i>Yahoo! Maps</i>	<i>Bing Maps</i>
Navegadores Oficiais	IE6 +, Firefox 2 +, Safari 3 +, Opera 8,02 +, Google Chrome 1 +	Firefox 2 +, IE6 +, Opera 9 ou Safari 3	IE6 +, Firefox 2 +, Safari 3 +
Graus de movimento	Vertical, horizontal, profundidade de rotação, (beta), 360 panorâmico (street view), modo 3D	Vertical, profundidade, horizontal	Vertical, horizontal, profundidade, 360 panorâmico(Streetside), Modo 3D
Tipos de mapa	6: Mapa, satélite, híbrido, rua, tráfego, 3D	3: Mapa, satélite trânsito, híbrido, (EUA)	10: Road, aéreo, híbrido, bird's eye, tráfego, 3D, street, rua de Londres, mapa Ordnance Survey, mapas local
Tecnologias utilizadas	JavaScript, XMLHttpRequest (Ajax), Hidden Iframe, XSLTProcessor, JSON, XML, WebGL	JavaScript, XMLHttpRequest (Ajax), XML, JSON	JavaScript, Ajax,. NET
FrontEnd	DHTML	DHTML	DHTML
Backend	JSON	JSON, XML	. NET
Níveis de Filtragem	1	Muitos - SmartView	0
Integração de Transporte Público	Sim	Não	Sim - London Underground
Percurso a pé	Sim (beta)	Não	Sim
Percurso de bicicleta	Sim (beta)	Não	Não
Enviar para dispositivo	E-mail	E-mail, SMS	E-mail, SMS, GPS
API disponível	Sim	Sim - JavaScript e Flash	Sim - JavaScript e Silverlight

Além dos serviços de mapas anteriormente descritos, existe um português que foi criado para permitir exclusivamente a pesquisa e localização de localidades, ruas e entidades em Portugal Continental e ilhas – serviço de mapas do SAPO [SMapas 12].

Também este permite traçar itinerários entre duas localidades, com a possibilidade de escolha na forma de realização do percurso, como por exemplo a pé, mais rápido, de transportes, etc. Disponibiliza um grande número de serviços disponíveis para localização, como por exemplo: farmácias de serviço, o trânsito nos principais pontos do país e outras informações tais como restaurantes, saúde, segurança, transportes, cultura, etc. O SAPO Mapas disponibiliza uma funcionalidade inexistente nos outros serviços de mapas, a possibilidade de personalização do SAPO Mapas, onde existe uma panóplia de cores a seleccionar de acordo com a preferência de cada utilizador. O mapa pode também ser integrado nos sites dos utilizadores, através de uma API disponibilizada. A interface do Sapo Mapas é ilustrada de acordo com a Figura 2.5.

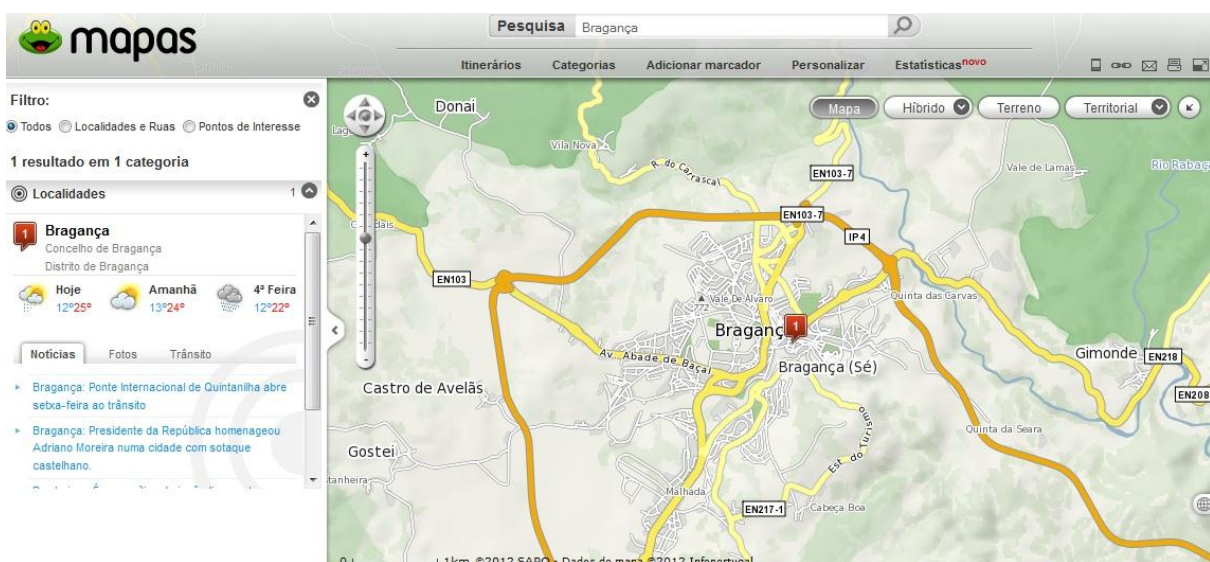


Figura 2.5 - Interface do Sapo Mapas

2.2 Pontos de interesse

Como já foi referido, as aplicações georreferenciadas são usadas em vários contextos, sendo por exemplo o turismo, lazer e serviços úteis três dos mais comuns. Quando se necessita obter informação georreferenciada de locais específicos que tem interesse para a sociedade de acordo com o seu contexto diz-se que estamos perante os pontos de interesse. Esses mesmos pontos são representados no mapa por ícones associados a diferentes categorias.

A definição e associação do grupo de categoria e respectiva simbologia é em função da área de interesse desse local que representa para a sociedade. Em algumas aplicações para uma determinada área de interesse a categoria associada tem mais do que um nível de classificação.

A representação de pontos de interesse em mapas deve ser tratada com rigor. É necessário perceber quais as características gráficas que fazem de um ícone uma boa representação para um ponto de interesse. Também tem que existir uma relação coerente entre a relevância de um ponto de interesse e a sua representação; isto é, de todos os ícones num mapa, aqueles que são mais importantes nunca devem passar despercebidos, nem o que é menos importante deve captar mais rapidamente a atenção do utilizador [Matos 08].

Apesar de existir aplicações SIG orientadas aos pontos de interesse em ambiente desktop, estas aplicações são mais populares em dispositivos móveis.

2.2.1 Tipos de aplicações

As aplicações SIG que efetuam a gestão de pontos de interesse ou simplesmente permitem a sua consulta, nem sempre conseguem disponibilizar uma excelente qualidade de informação e representação gráfica. No entanto há algumas aplicações que se esmeram por isso e de certo modo o conseguem, como é o caso do GIS Studio.

O GIS Studio é uma aplicação Web (SaaS - Software as a Service) que, mediante autenticação, permite inserir, gerir e editar a informação e conteúdos dos Pontos de Interesse (POIs) disponibilizados no Serviço de GIS e no SAPO Mapas, e cujas funcionalidades, permitem [GisStudio 12]:

- **Visualizar POIs** - Agrupados por categorias, como Compras, Desportos, Restaurantes, etc;
- **Edição de POIs** - Permite pesquisa, criação e edição de Pontos de Interesse e respetivos detalhes, como por exemplo: morada, telefone, e-mail, etc;
- **Supervisão de POIs** - Possibilidade de aceitar ou recusar as alterações efetuadas;
- **Gestão de listas de POIs** - Criar/apagar lista de POIs, adicionar/remover um POI a uma lista, duplicar listas, renomear uma lista;

- **Desenho de Polígonos** – Associar um conjunto de *shapes* (polígonos, linhas, pontos) a um POI representadas sobre o mapa;
- **Gestão de Entidade** - Gerir informação, permissões e utilizadores de Entidade.

Pode-se aferir que está-se perante uma ferramenta fácil de usar e que apresenta um design profissional e intuitivo, além do mais não exige conhecimentos especializados de Sistemas de informação geográfica (SIG) para as tarefas mais comuns.

Um exemplo de uma pesquisa de pontos de interesse na aplicação GIS Studio é ilustrada na Figura 2.6.

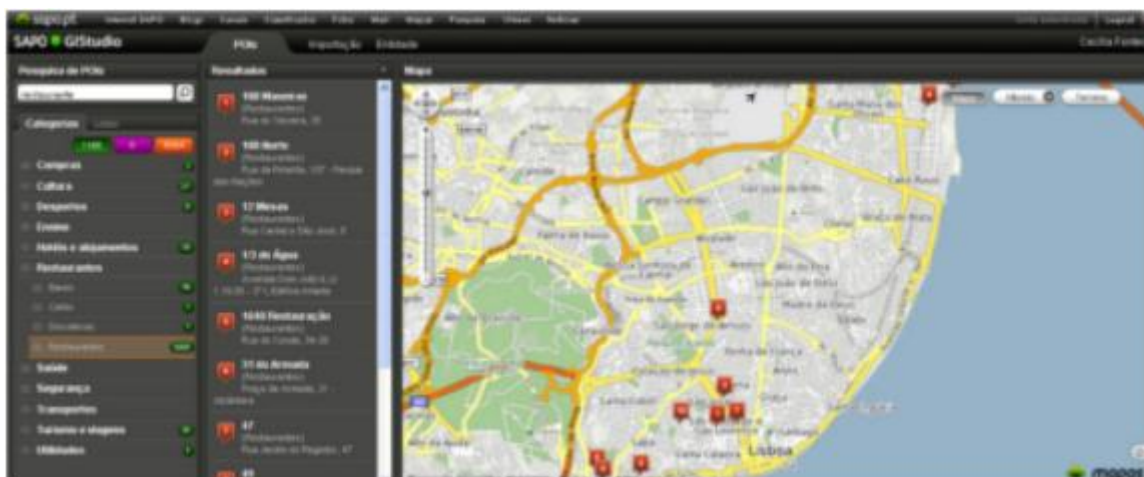


Figura 2.6 - Pesquisa de pontos de interesse em GIS Studio

Como já foi referido, há aplicações SIG de pontos de interesse que acedem a bases de dados muito completas, mas utilizam técnicas de representação gráfica rudimentares ou pouco sugestivas; por outro lado, há aplicações que, sendo mais limitadas na pesquisa de dados, dão maior ênfase à representação de informação. É o caso da aplicação Sakura House [Sakura]. É apenas possível pesquisar quartos em Tóquio, mas a informação é representada de uma forma clara e sugestiva. As flores cor-de-rosa, características do Japão, com um tamanho mais pequeno indicam apartamentos ou casas com alguns quartos. As flores maiores representam locais de maior dimensão, como grandes hotéis e escritórios desta organização. A aplicação também representa as linhas dos transportes com diferentes cores. Desta forma um utilizador sabe como se deslocar até um determinado alojamento (Figura 2.7).

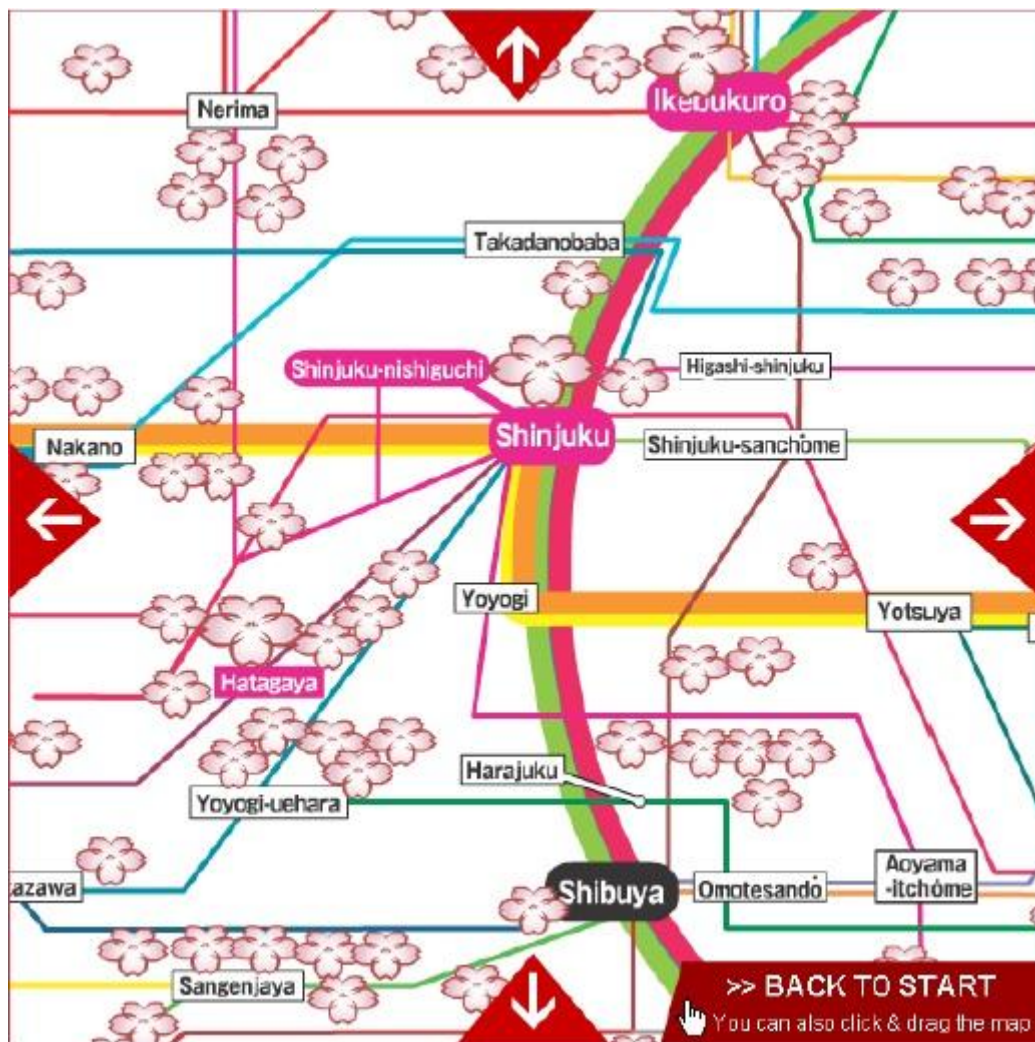


Figura 2.7 - Alojamentos representados pela aplicação Sakura House

Mas há ainda aplicações que complementam o acesso a bases de dados ricas e variadas com uma excelente representação visual da informação. É o caso da aplicação Gapminder World [Gapminder 12]. Esta aplicação está ligada a base de dados estatísticas muito completa e variada e que oferece grande flexibilidade de opções de pesquisa: pode, por exemplo, pesquisar-se a relação entre o salário médio e a esperança média de vida dos habitantes de todos os países do mundo nos últimos 200 anos. Igualmente cuidada é a representação da informação. Apenas usando círculos, consegue representar-se a informação de uma forma apelativa, perceptível e clara. Se escolhermos, por exemplo, observar a população atual em todos os países do mundo, obtemos o resultado representado na Figura 2.8.

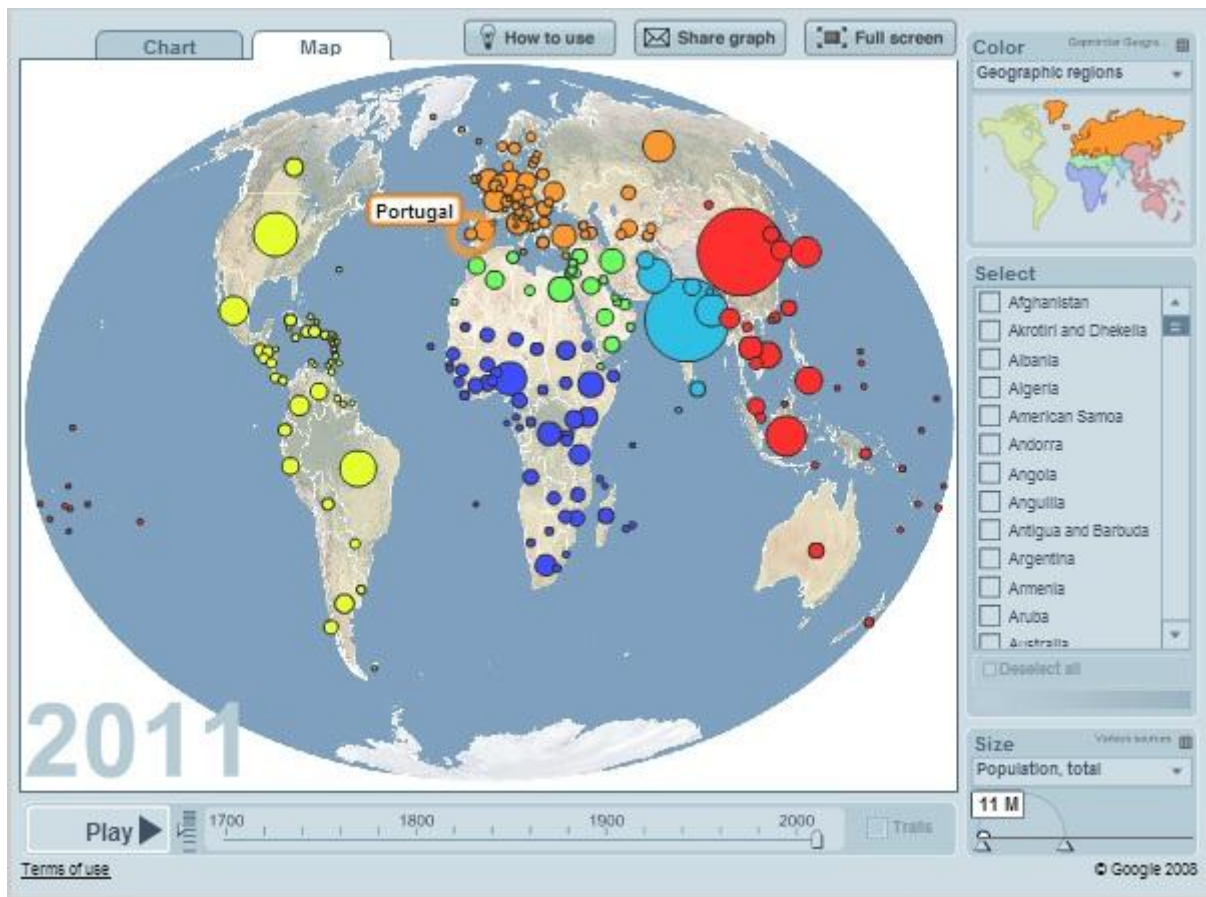


Figura 2.8 - População mundial representada pela Gapminder World

Cada cor representa os países de cada continente. Cada círculo está posicionado no país do qual se pretende representar a informação. Quanto mais pequeno é o círculo, menor é o valor da grandeza em questão e vice-versa. Pode apenas visualizar-se um país de cada vez, fazendo *click* no círculo correspondente ou no nome do país presente na lista lateral. Os outros países ficam mais transparentes, dando assim mais relevância ao que foi selecionado. A escala da grandeza também pode ser ajustada, alterando o tamanho dos círculos que representam o valor mínimo e máximo através de um *slider* disponível na interface da aplicação.

2.3 Algoritmos de planeamento de percursos

Nas aplicações georreferenciadas e em particular que envolvam pontos de interesse, uma das funcionalidades a que o utilizador dá muita importância é obtenção de caminhos/percursos de entre dois locais, de modo a retirar conclusões que o auxiliem numa tomada de decisão em relação a esse mesmo percurso. Mas o ideal é que as sugestões de percursos fornecidas pelas

aplicações SIG sejam ótimas, ou seja, em função de parâmetros que envolvam os pontos de origem e de destino, seja devolvido o melhor caminho entre esses dois pontos. Os serviços de mapas estudados, nomeadamente o Google Maps, utilizam algoritmos de grafos de modo a retornar os percursos ótimos. De seguida será apresentada uma breve introdução de alguns dos algoritmos mais populares nas soluções de percursos ótimos.

O algoritmo de Dijkstra foi desenvolvido por Edsger Dijkstra em 1959 para a resolução do problema de Shortest Path (Caminhos Mínimos). Este algoritmo tem por base um grafo no qual os nós são posições e os segmentos entre nós são os caminhos. Neste grafo, o peso dos caminhos pode ser dado pela distância, pelo tempo, pelos limites de velocidade, entre outros, dependendo da aplicação.

O Algoritmo de Dijkstra é um procedimento iterativo que partindo de um nó inicial do grafo, cresce em todos os sentidos, guardando o caminho para cada ponto bem como o peso do caminho total até cada um destes. Após inicialização, o mesmo procedimento é chamado no ponto com menor peso. Se um ponto for encontrado várias vezes, só a primeira é considerada (a de menor peso). O algoritmo termina quando é encontrado o ponto do destino [Dijkstra 12].

O algoritmo A* tem por base o algoritmo de Dijkstra, mas necessita da noção do espaço global ou seja, da noção da posição de cada ponto no mundo. A partir das posições consegue calcular a distância de cada ponto descoberto ao destino pretendido e usá-la para escolher que ponto explorar [A* 12].

Capítulo 3

3 Análise e Modelação

Neste capítulo é descrita a análise do problema, onde são identificadas algumas lacunas de funcionalidades relacionadas com gestão de pontos de interesse e requisitos em falta no mesmo tipo de aplicação e de seguida é apresentada uma solução para este caso.

De seguida é apresentada a modelação do projecto, em que consiste na apresentação de diagramas, nomeadamente um diagrama de camadas que descreve a disposição da arquitetura do projeto; um diagrama de classes que representa as entidades envolvidas na implementação do projeto; diagramas de caso de uso que ilustram o comportamento de ações das funcionalidades existentes no projeto, um diagrama de sequência que exemplifica a ordem de acontecimentos da invocação de um determinado evento; e diagramas de estruturas de site que indicam o modo como os menus e respetivas páginas web do backoffice e frontoffice estão organizados.

3.1 Análise do problema

Os sistemas de informação geográfica de gestão de pontos de interesse existentes a nível empresarial e pessoal atualmente oferecem muitas semelhanças de lógica e requisitos funcionais em que a maior parte deste tipo de sistemas cinge-se unicamente a um frontoffice com a disponibilização dos pontos de interesse num portal de mapas ou uma aplicação móvel e a gestão dessa informação feita diretamente na base de dados, havendo portanto poucas soluções que ofereçam um interface intuitivo de configuração e gestão de pontos de interesse. De um modo particular, os sistemas que permitem a gestão de pontos de interesse através de uma interface, a mesma por norma não tem disponível a possibilidade de se categorizar os referidos pontos, ou seja, atribuir por exemplo a categoria de “Alojamento” ou “Desporto” a

um hotel ou estádio de futebol respetivamente; em que as categorias são preenchidas previamente numa tabela da base de dados, sendo as mesmas disponibilizadas posteriormente na interface para leitura e associação aos pontos de interesse a serem criados. Este modo de categorização e nas soluções que permitem esta atribuição numa interface administrativa, por norma existe um único nível de grupos de categorias, ou seja, pode haver pontos de interesse que façam parte de um determinado grupo de categoria, apesar da possibilidade da existência de diferenças entre eles quanto à sua classificação, não havendo assim uma categorização pormenorizada e correta.

Relativamente aos dados presentes nos balões informativos/tooltips dos pontos de interesse quando se clica nestes no mapa, por norma, todos apresentam os mesmos campos de informação, mesmo quando não existem dados, podendo originar em certos casos do evento *click* o aparecimento de “balões” vazios, não havendo assim personalização da informação a mostrar nos tooltips.

Neste tipo de aplicação SIG é possível estender ou enquadrar outras temáticas de gestão de informação, especificamente a associação de entidades, ou seja, é possível associar-se a cada entidade (organizacional, municipal, turística,...) uma gestão personalizada de pontos de interesse e respetiva visualização nos mapas, no entanto há poucas soluções que apliquem este enquadramento.

Também não se verifica nestas aplicações a abundância a nível de diversidades de funcionalidades inerentes aos pontos de interesse, nomeadamente geração de percursos nos mapas entre os mesmos locais, de modo a fornecer aos utilizadores outro tipo de informação de pontos de interesse e apoio mais consistente na tomada de decisão sobre os mesmos.

Com base nas situações anteriormente identificadas na gestão de pontos de interesse, a implementação deste projeto procura responder a estas questões através de soluções eficientes e úteis. De modo a explicar e justificar as soluções apresentadas neste projeto, a seguir as mesmas serão detalhadas.

- a) Foi tomada a decisão de se criarem dois sistemas, um backoffice administrativo onde se pode efetuar a gestão de informação da solução e um frontoffice com recurso a um mapa onde se pode pesquisar e visualizar informação relacionada com os pontos de interesse.
- b) O backoffice apresentará a gestão de informação dividida por duas seções; uma de acesso exclusivo a administradores, composta pelas funcionalidades de *Entidades*,

Grupos e Utilizadores e outra composta pelas funcionalidades *Categorização de POIs* e *Pontos de Interesse* acessíveis a todos os utilizadores autenticados e de acordo com os seus privilégios. A seguir será apresentada uma breve descrição das funcionalidades que integram o sistema de backoffice.

- a. **Entidades** – Aqui será efetuada a gestão de entidades às quais posteriormente serão associadas os respetivos pontos de interesse. A informação das mesmas entidades só estará disponível no backoffice.
- b. **Grupos** – Gestão de grupos (Administrador, Administrador_<Entidade>, Convidado e Convidado_<Entidade>) dos utilizadores que tem acesso ao backoffice.
- c. **Utilizadores** – Gestão dos utilizadores autenticados e com acesso ao sistema de backoffice.
- d. **Categorização de POIs¹** – Esta funcionalidade permite gerir as categorias que posteriormente na criação de pontos de interesse serão atribuídas a estes. Como em geral as aplicações SIG de pontos de interesse disponibilizam somente a opção de categorizar a um único nível de categoria, neste projeto optou-se por três níveis de classificação de categoria; *Grupo*, *Categoria* e *Classe*. No entanto se a decisão fosse implementar ainda mais níveis, também seria possível, porque a estrutura de dados tal o permite, devido à utilização do *composite pattern*, que é um padrão de desenho indicado para estruturas irregulares e que permite tratar de forma igual objetos simples e objetos compostos. Sendo assim neste projeto é possível categorizar pontos de interesses do seguinte modo, sendo o último nível o identificador:
 - i. Grupo – Alojamento
 - ii. {Grupo, Categoria} - {Alojamento, Hotel} ou {Alojamento, Pousada}
 - iii. {Grupo, Categoria, Classe} - {Alojamento, Hotel, 5 Estrelas}
- e. **Pontos de Interesse** - Esta funcionalidade permite criar e gerir pontos de interesse associando os mesmos a categorias e entidades. Ao se criar um ponto de interesse é possível visualizar num mapa a sua localização e a configuração

¹ POI – Points Of Interest

do tooltip². Por exemplo o tooltip do ponto de interesse A pode conter os campos *Nome, Descrição, Telefone* e o do ponto de interesse B apresentar os campos *Nome, Email*.

- c) O frontoffice apresentará um mapa onde se poderá visualizar a georreferenciação de pontos de interesse e respetiva informação de acordo com os itens *Pesquisas, Percursos, Categorias*.
- a. **Pesquisas** – Através deste item é possível pesquisar e visualizar pontos de interesse no mapa de acordo com determinados critérios especificados no formulário.
 - b. **Percursos** – Nesta funcionalidade é possível através da seleção entre um ponto de interesse de origem e um ponto de interesse de destino gerar um percurso no mapa entre estes dois pontos de interesse e obter a respetiva informação do percurso.
 - c. **Categorias** – Esta funcionalidade disponibilizará as categorias de pontos de interesse existentes e selecionar as categorias dos pontos de interesse que pretende visualizar no mapa.

3.2 Diagrama de camadas

A arquitetura tecnológica definida para a aplicação SIG de gestão de pontos de interesse de entidades baseia-se no modelo de multicamadas, de modo a oferecer uma solução tecnológica sólida, modular, extensível, orientada a serviços e preparada para futuras atualizações.

Esta arquitetura é composta por três camadas tecnológicas interligadas entre si, cada uma com a sua função específica. As camadas presentes na solução são:

- **Data Access Layer** – Nesta camada existe a estrutura de dados e é onde se faz a ligação à base de dados através de consultas, além da existência de utilitários auxiliares no tratamento de dados.

² Balão informativo de um ponto georreferenciado

- **Business Service Layer** – Nesta camada estão presentes as entidades de negócio e componentes ou métodos que comunicam com a camada DAL. A interface de serviços permitirá comunicar com a camada superior.
- **Presentation Layer** – A camada da apresentação será composta pelas interfaces dos sistemas de backoffice e frontoffice, onde se irá consumir os serviços da camada inferior (BSL) na interação com o cliente. Nesta camada também será integrada a API³ da Google Maps e o acesso ao repositório de simbologia da categorização dos pontos de interesse.

O diagrama de camadas da Figura 3.1 representa as camadas da solução tecnológica.

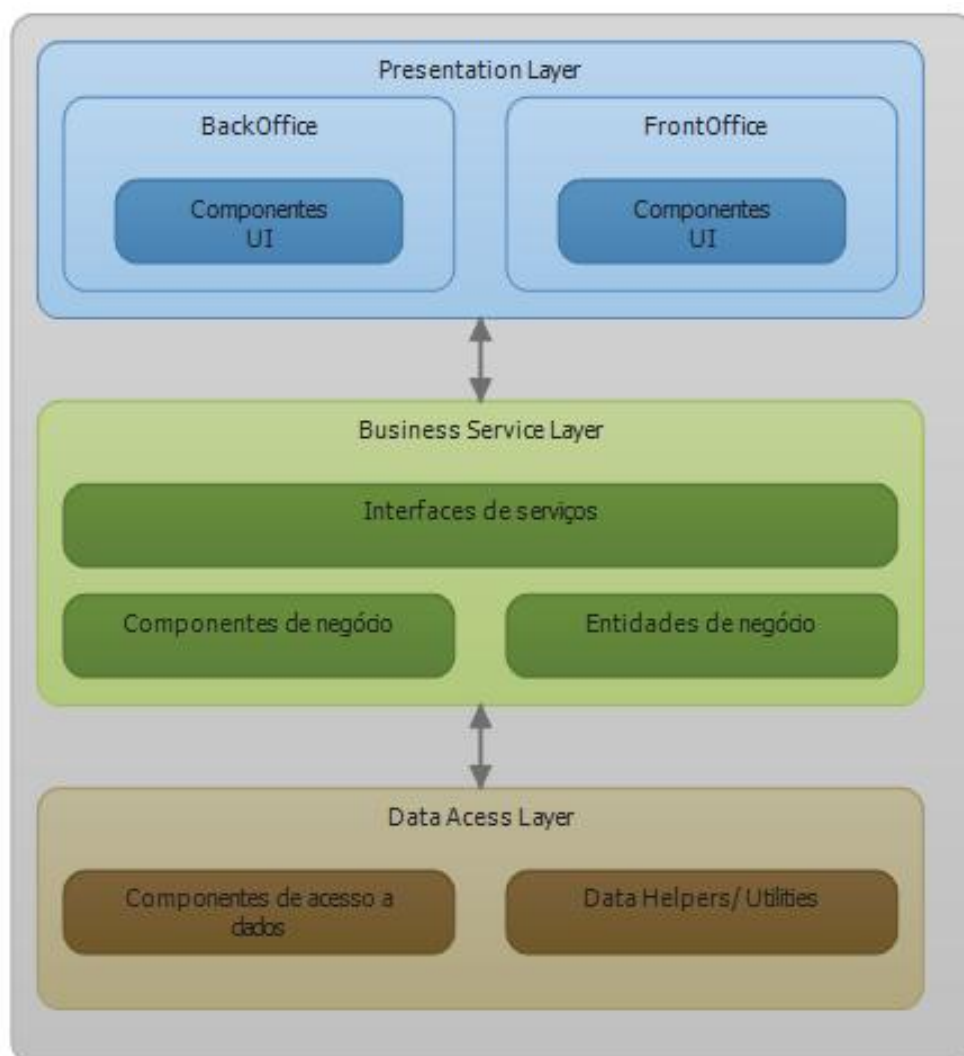


Figura 3.1 - Diagrama de camadas

³ API – Application Programming Interface

3.3 Diagrama de classes

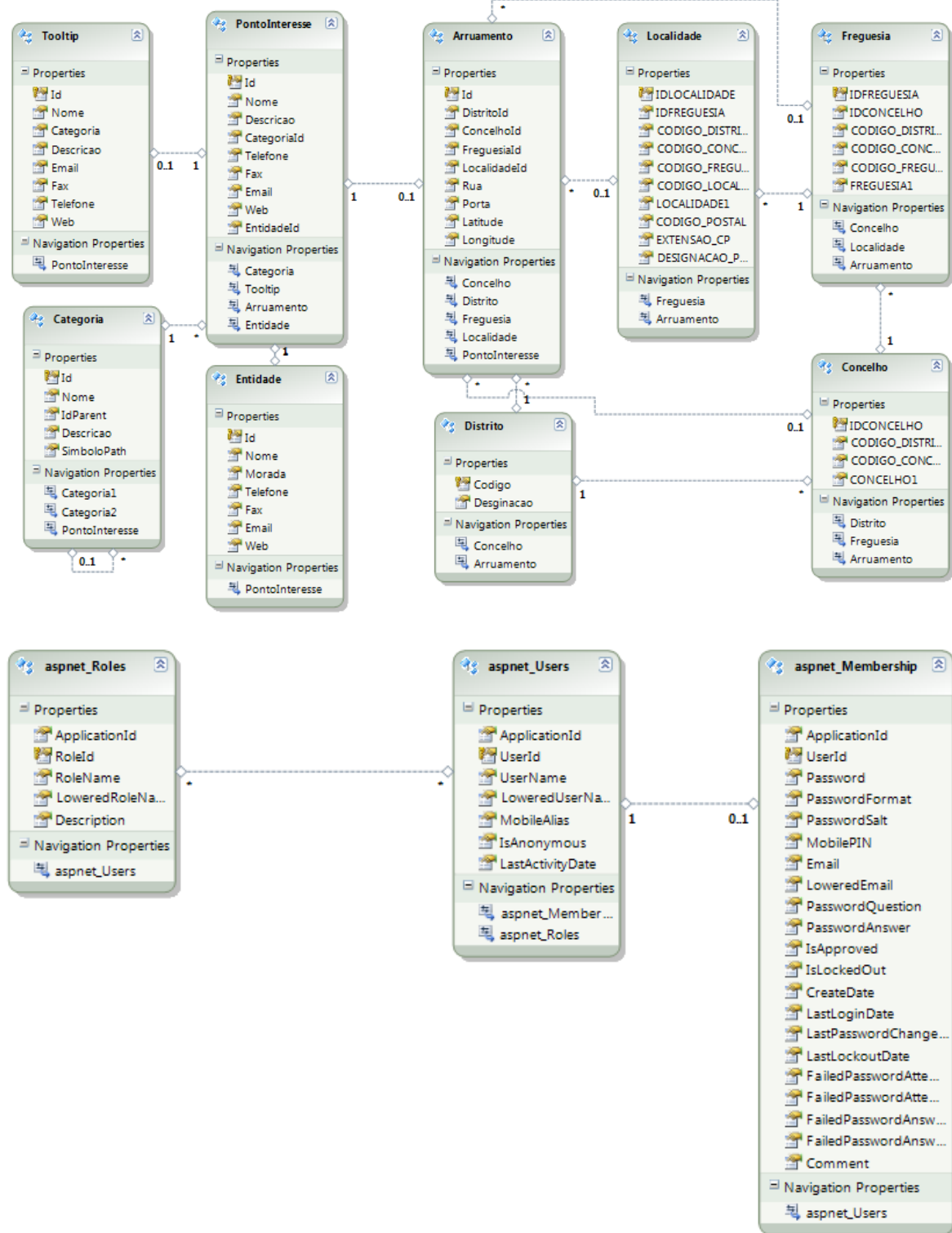


Figura 3.2 - Diagrama de classes

As classes do diagrama acima representado correspondem às tabelas homónimas presentes na base de dados. A (Tabela 3.1) apresenta uma breve descrição das referidas tabelas.

Tabela 3.1 - Descrição das tabelas da solução SIG

<i>Tabela</i>	<i>Descrição</i>
Entidade	Tabela onde é armazenada a informação das entidades da gestão de pontos de interesses.
Aspnet_Users	Tabela onde é armazenada a informação dos utilizadores.
Aspnet_Membership	Tabela onde é armazenada informação dos utilizadores referente a autenticação.
Aspnet_Roles	Tabela onde são armazenados os grupos aos quais os utilizadores podem pertencer.
Categoria	Tabela onde é armazenada a informação da categorização dos pontos de interesse. Esta tabela tem uma relação recursiva de modo a apresentar na mesma tabela os diferentes níveis de categorização. O campo IdParent corresponde ao identificador do nível superior da categoria.
PontoInteresse	Tabela onde é armazenada a caracterização dos pontos de interesses.
Tooltip	Tabela onde são armazenados os campos de informação de um determinado ponto de interesse que deve de aparecer no balão informativo.
Arruamento	Tabela onde é armazenada a localização dos pontos de interesses.
Distrito	Tabela onde é armazenada a identificação dos distritos de Portugal.
Concelho	Tabela onde é armazenada a identificação dos concelhos de Portugal.
Freguesia	Tabela onde é armazenada a identificação das freguesias de Portugal.
Localidade	Tabela onde é armazenada a identificação das localidades de Portugal.

3.4 Diagramas de casos de uso

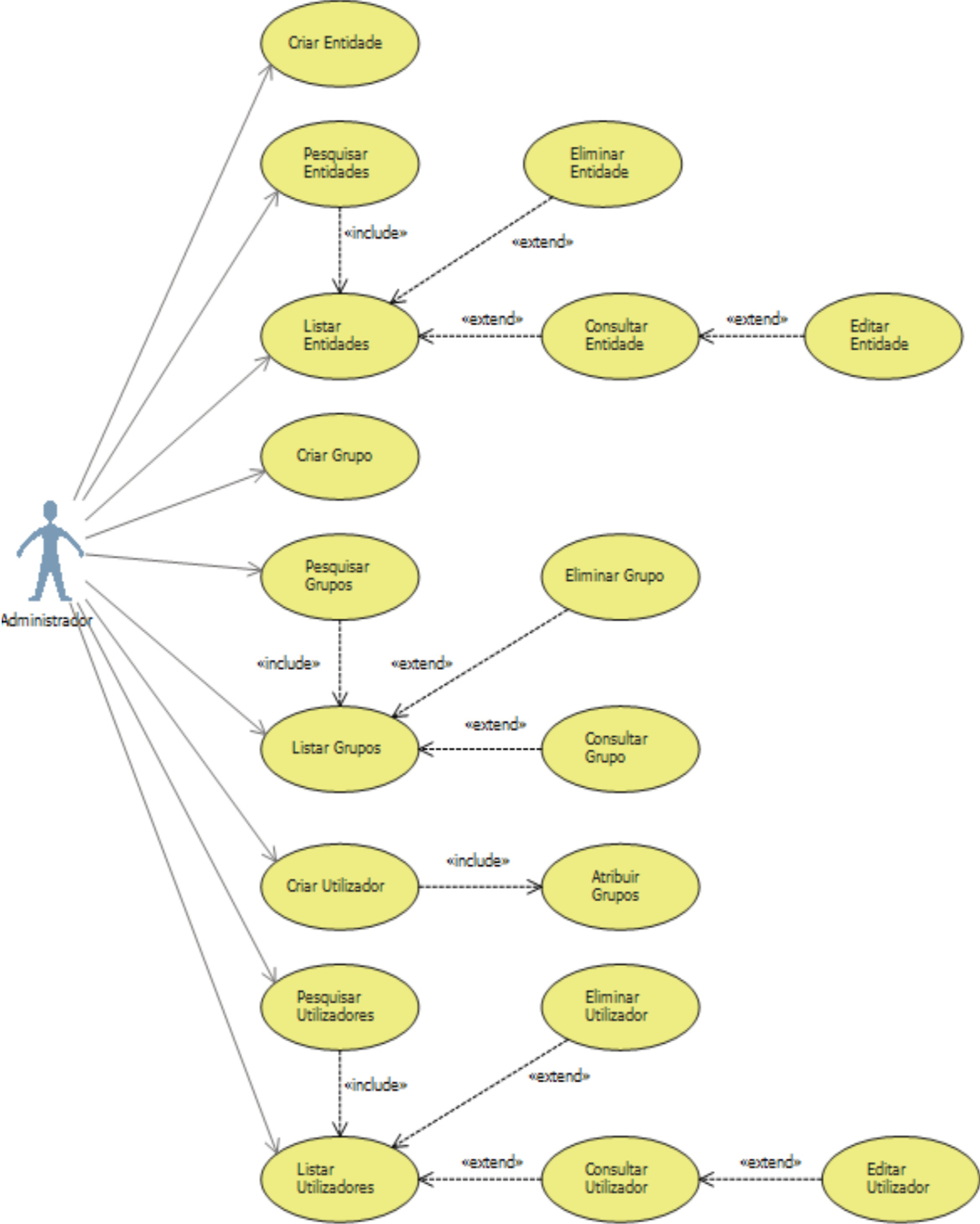


Figura 3.3 - Diagrama de casos de uso do backoffice de funcionalidades de acesso

De seguida é apresentada uma breve descrição dos casos de uso ilustrados na Figura 3.3. tendo como ator o Administrador para uma determinada iteração.

- Criar Entidade: Permite criar uma entidade.
- Pesquisar Entidades: Permite pesquisar entidades por nome e morada.
- Listar Entidades: Lista entidades de acordo com os parâmetros da pesquisa.
- Eliminar Entidade: Permite eliminar uma entidade da respetiva lista.
- Consultar Entidade: Permite consultar uma entidade da respetiva lista.
- Editar Entidade: Permite editar a entidade consultada.
- Criar Grupo: Permite criar um ou mais grupos.
- Pesquisar Grupos: Permite pesquisar grupos por nome.
- Listar Grupos: Lista grupos de acordo com o parâmetro da pesquisa.
- Eliminar Grupo: Permite eliminar um grupo da respetiva lista.
- Consultar Grupo: Permite consultar um grupo da respetiva lista.
- Criar Utilizador: Permite criar um utilizador.
- Atribuir Grupos: Permite associar um ou mais grupos ao utilizador na sua criação.
- Pesquisar Utilizadores: Permite pesquisar utilizadores por parâmetros.
- Listar Utilizadores: Lista utilizadores de acordo com parâmetros de pesquisa.
- Eliminar Utilizador: Permite eliminar um utilizador da respetiva lista.
- Consultar Utilizador: Permite consultar uma entidade da respetiva lista.
- Editar Utilizador: Permite editar o utilizador consultado.

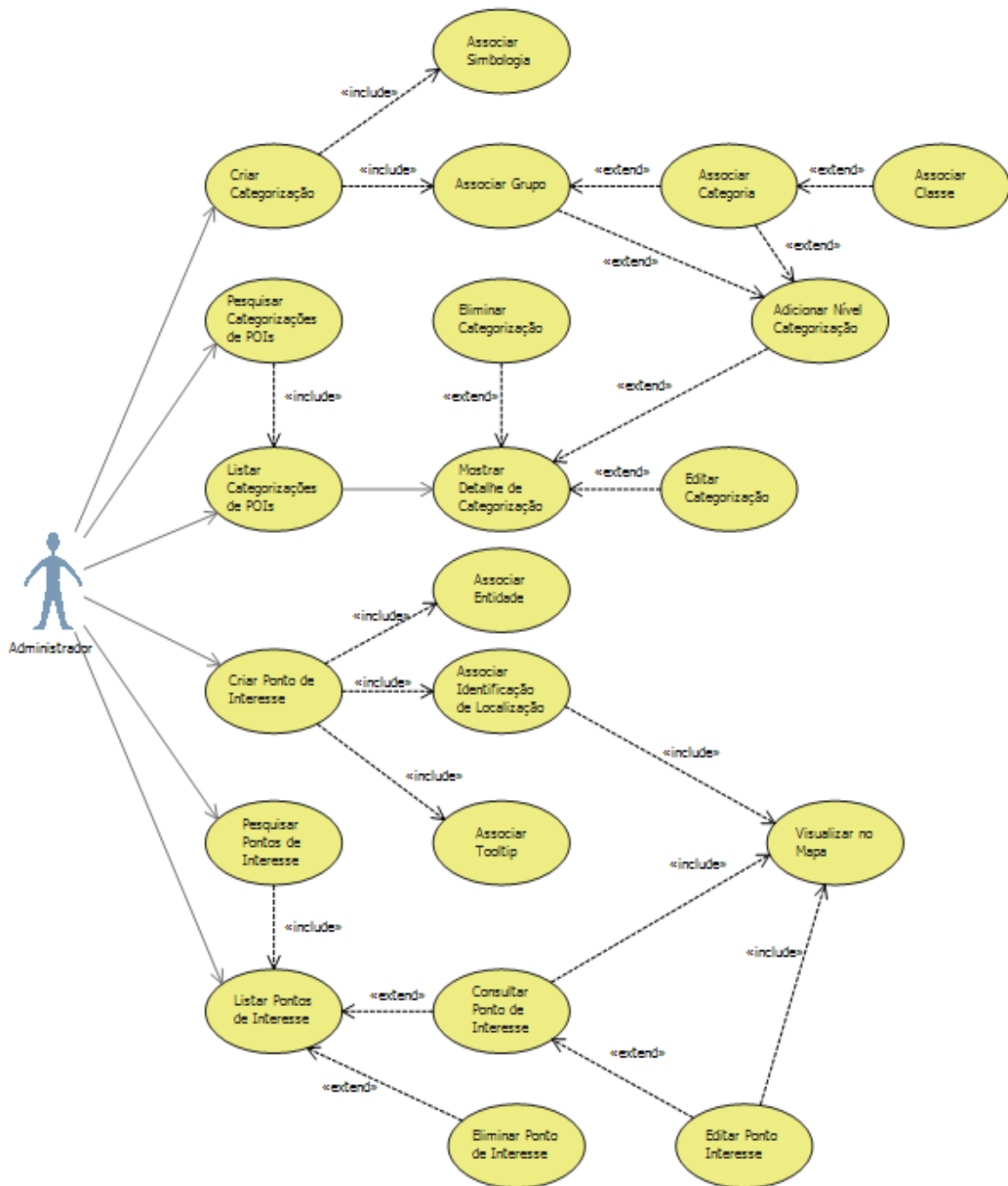


Figura 3.4 - Diagrama de caso de uso do backoffice - pontos de interesse (administrador)

De seguida é apresentada uma breve descrição dos casos de uso ilustrado na Figura 3.4, tendo como ator o Administrador para uma determinada iteração.

- Criar Categorização: Permite criar uma categorização de pontos de interesse.
- Associar Simbologia: Associa à categorização um ícone representativo.
- Associar Grupo: Permite efetuar categorização ao primeiro nível de classificação.

- Associar Categoria: Associa a categoria ao grupo categorizado e permite efetuar categorização ao segundo nível de classificação.
- Associar Classe: Associa a classe à categoria representada e permite efetuar categorização ao terceiro nível de classificação.
- Adicionar Nível de Categorização: Permite adicionar uma categoria ao grupo ou a uma classe a uma categoria, verificando-se o mesmo aquando a consulta dos detalhes da respetiva categorização.
- Pesquisar Categorizações de POIs: Permite pesquisar categorizações de pontos de interesse por um determinado nível de classificação.
- Lista Categorizações: Lista hierarquicamente as categorizações de acordo com o nível de classificação pesquisado.
- Mostrar Detalhe de Categorização: Ao ser listada as categorizações, a primeira categorização ou a selecionada da lista é apresentada a respectiva informação detalhada.
- Editar Categorização: Permite editar a categorização consultada (detalhada).
- Eliminar Categorização: Permite eliminar a categorização consultada (detalhada).
- Criar Ponto Interesse: Permite criar um ponto de interesse.
- Associar Entidade: Associa uma entidade ao ponto de interesse aquando a sua criação.
- Associar Identificação de Localização: Permite efetuar a aquisição dados geográficos para a localizar o ponto de interesse no mapa.
- Associar Tooltip: Permite configurar os campos informativos do ponto de interesse a serem apresentados no mapa.
- Pesquisar Pontos de Interesse: Permite pesquisar pontos de interesse de acordo com determinados parâmetros.
- Lista Pontos de Interesse: Lista os pontos de interesse de acordo com os parâmetros da pesquisa.
- Consultar Ponto de Interesse: Permite consultar um ponto de interesse da respetiva lista.
- Editar Ponto de Interesse: Permite editar o ponto de interesse consultado.

- Eliminar Ponto de Interesse: Permite eliminar um ponto de interesse da respetiva lista.
- Visualizar no mapa: Permite visualizar no mapa o ponto de interesse aquando a sua identificação de localização, consulta e edição.

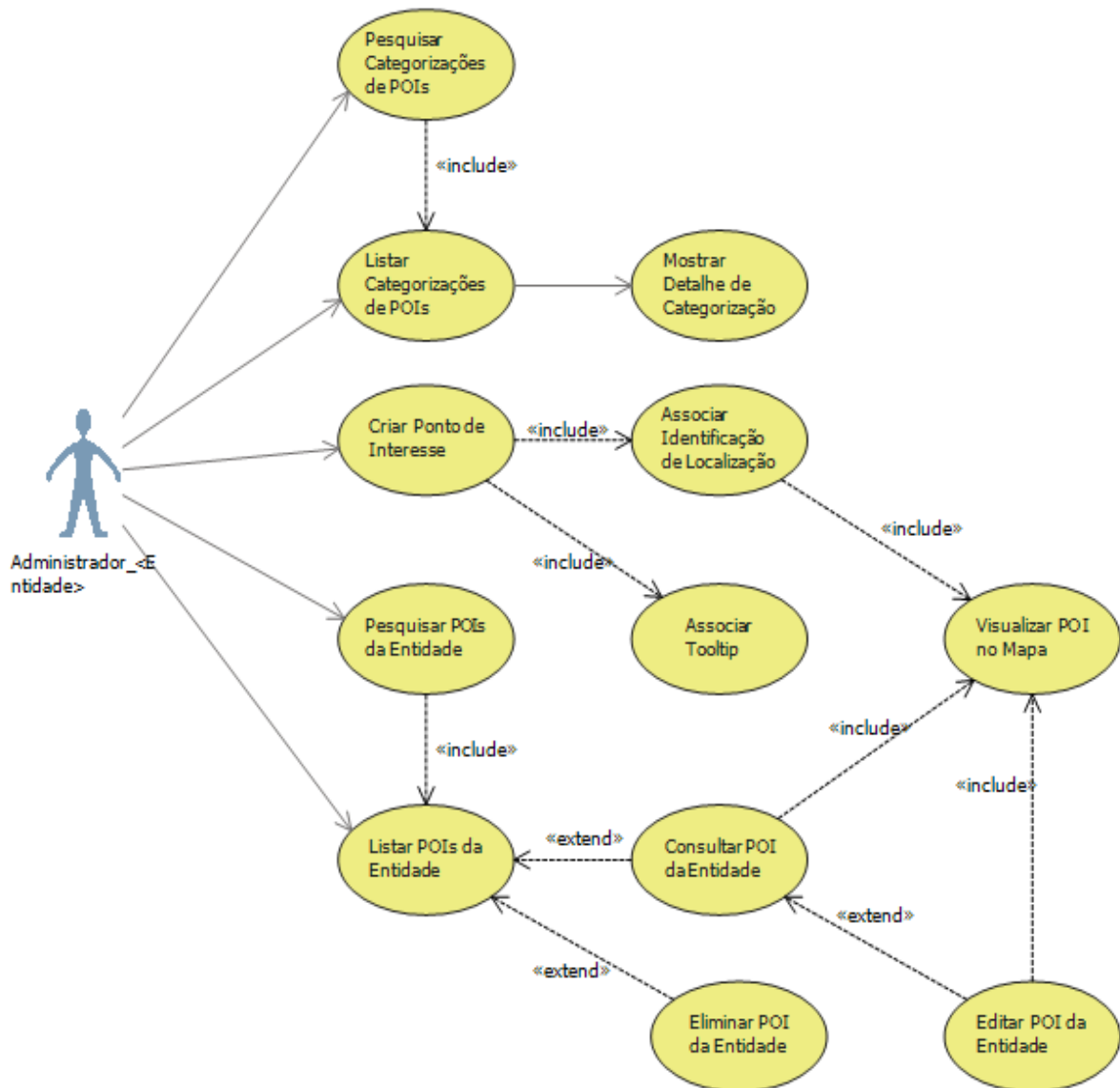


Figura 3.5 - Diagrama de caso de uso do backoffice - pontos de interesse (Administrador_<Entidade>)

De seguida é apresentada uma breve descrição dos casos de uso ilustrados na Figura 3.5. tendo como ator o Administrador_<Entidade> para uma determinada iteração.

- Pesquisar Categorizações de POIs: Permite pesquisar categorizações de pontos de interesse por um determinado nível de classificação.

- Lista Categorizações: Lista hierarquicamente as categorizações de acordo com o nível de classificação pesquisado.
- Mostrar Detalhe de Categorização: Ao selecionar uma categorização é apresentada a respectiva informação detalhada.
- Editar Categorização: Permite editar a categorização consultada (detalhada).
- Eliminar Categorização: Permite eliminar a categorização consultada (detalhada).
- Criar POI da Entidade: Permite criar um ponto de interesse da entidade <Entidade>.
- Associar Identificação de Localização: Permite efetuar a aquisição dados geográficos para a localizar o ponto de interesse no mapa.
- Associar Tooltip: Permite configurar os campos informativos do ponto de interesse a serem apresentados no mapa.
- Pesquisar POIs da Entidade: Permite pesquisar pontos de interesse da entidade <Entidade> de acordo com determinados parâmetros.
- Listar POIs da Entidade: Lista os pontos de interesse da entidade <Entidade> de acordo com os parâmetros da pesquisa.
- Consultar POI da Entidade: Permite consultar um ponto de interesse da entidade <Entidade> da respectiva lista.
- Editar POI da Entidade: Permite editar o ponto de interesse consultado da entidade <Entidade>.
- Eliminar POI da Entidade: Permite eliminar um ponto de interesse da respectiva lista da entidade <Entidade>.
- Visualizar no mapa: Permite visualizar no mapa o ponto de interesse da entidade <Entidade> aquando a sua identificação de localização ou consulta.

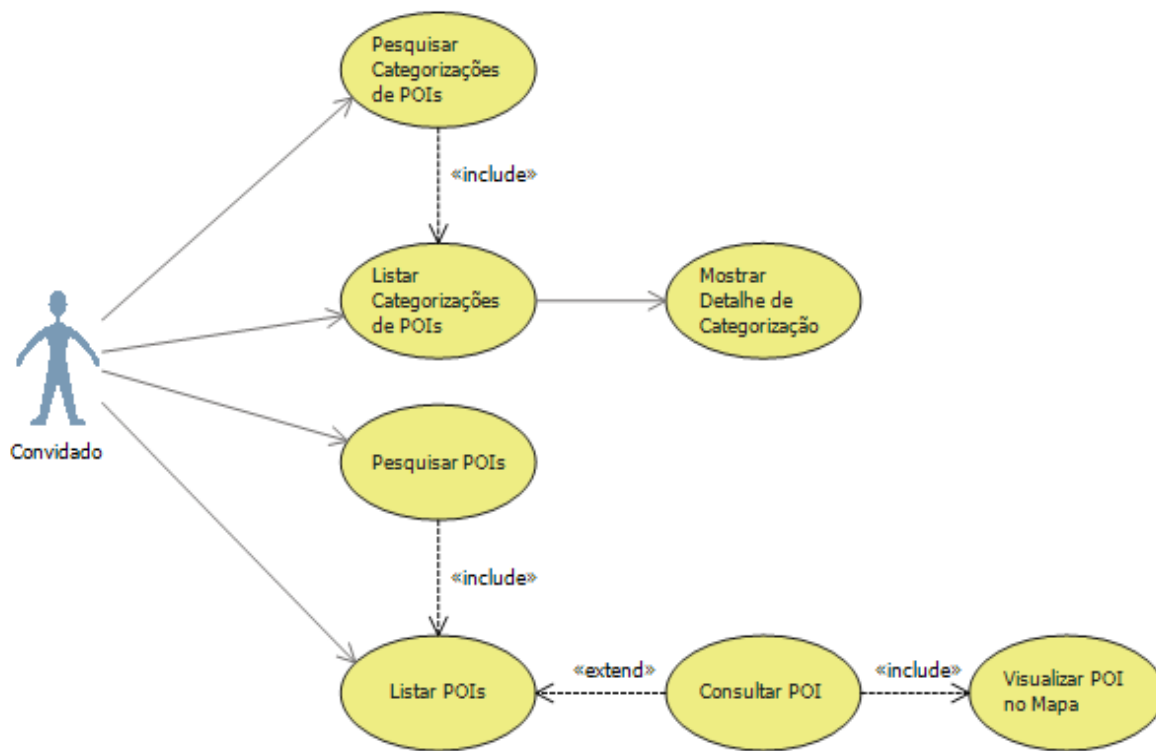


Figura 3.6 - Diagrama de caso de uso do backoffice - pontos de interesse (Convidado)

De seguida é apresentada uma breve descrição dos casos de uso ilustrados na Figura 3.6. tendo como ator o “Convidado” para uma determinada iteração.

- **Pesquisar Categorizações de POIs:** Permite pesquisar categorizações de pontos de interesse por um determinado nível de classificação.
- **Lista Categorizações:** Lista hierarquicamente as categorizações de acordo com o nível de classificação pesquisado.
- **Mostrar Detalhe de Categorização:** Ao ser listada as categorizações, a primeira categorização ou a selecionada da lista é apresentada a respectiva informação detalhada.
- **Pesquisar Pontos de Interesse:** Permite pesquisar pontos de interesse de acordo com determinados parâmetros.
- **Listar Pontos de Interesse:** Lista os pontos de interesse de acordo com os parâmetros da pesquisa.

- Consultar Ponto de Interesse: Permite consultar um ponto de interesse da respectiva lista.
- Visualizar no mapa: Permite visualizar no mapa o ponto de interesse aquando a sua consulta.

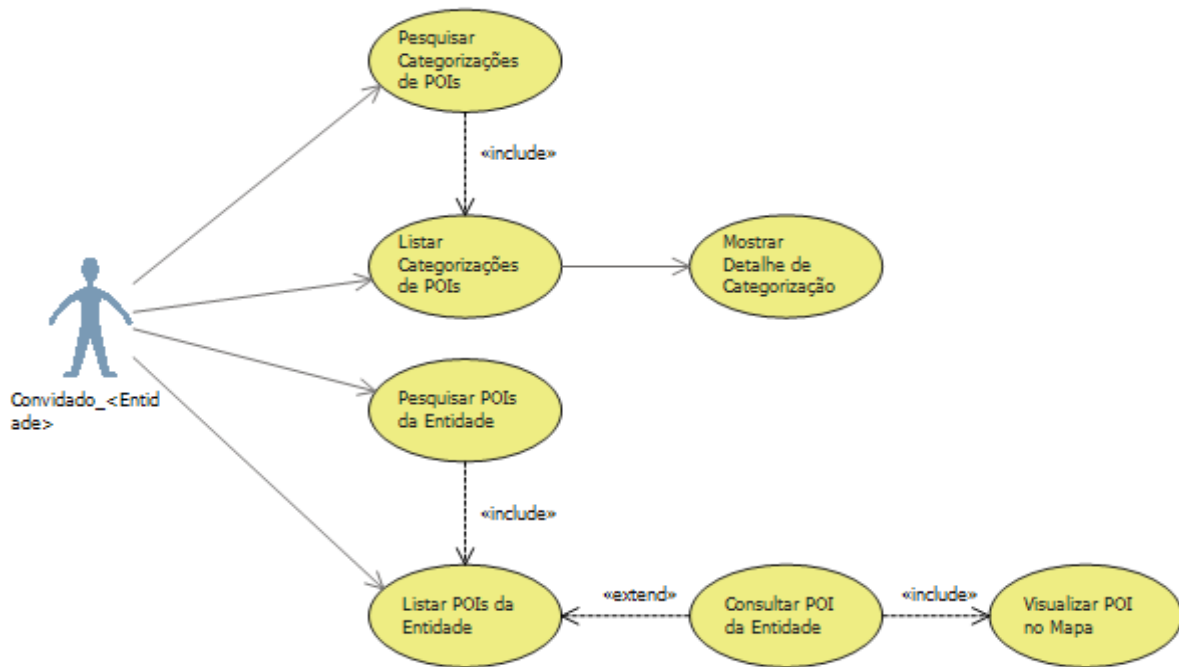


Figura 3.7 - Diagrama de caso de uso do backoffice - pontos de interesse (Convidado_<Entidade>)

De seguida é apresentada uma breve descrição dos casos de uso ilustrados na Figura 3.7. tendo como ator o “Convidado_<Entidade>” para uma determinada iteração.

- **Pesquisar Categorizações de POIs**: Permite pesquisar categorizações de pontos de interesse por um determinado nível de classificação.
- **Lista Categorizações**: Lista hierarquicamente as categorizações de acordo com o nível de classificação pesquisado.
- **Mostrar Detalhe de Categorização**: Ao ser listada as categorizações, a primeira categorização ou a selecionada da lista é apresentada a respectiva informação detalhada.
- **Pesquisar Pontos de Interesse**: Permite pesquisar somente pontos de interesse por <Entidade>.
- **Listar Pontos de Interesse**: Lista os pontos de interesse da entidade <Entidade> de acordo com os parâmetros da pesquisa

- Consultar Ponto de Interesse: Permite consultar um ponto de interesse da entidade <Entidade> da respetiva lista.
- Visualizar no mapa: Permite visualizar no mapa o ponto de interesse da entidade <Entidade> aquando a sua consulta.

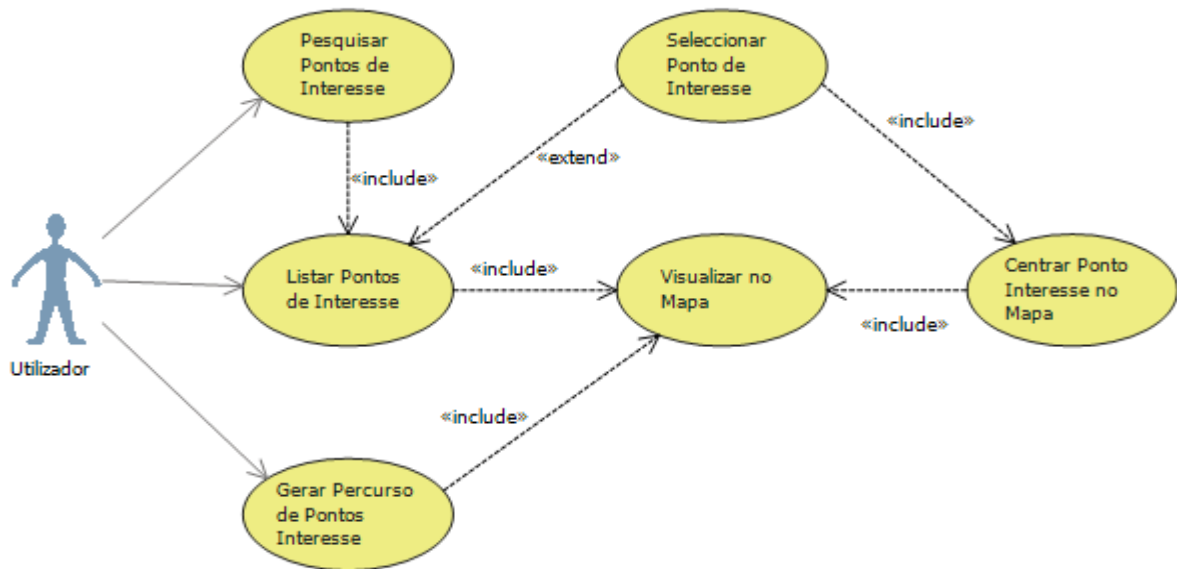


Figura 3.8 - Diagrama de caso de uso do frontoffice

De seguida é apresentada uma breve descrição dos casos de uso ilustrados na Figura 3.8. tendo como ator o “Utilizador” para uma determinada iteração.

- **Pesquisar Pontos de Interesse**: Permite pesquisar pontos de acordo com parâmetros.
- **Listar Pontos de Interesse**: Lista os pontos de interesse de acordo com os parâmetros da pesquisa.
- **Seleccionar Ponto de Interesse**: Seleciona um ponto de interesse da respetiva lista.
- **Centrar Ponto de Interesse no Mapa**: Permite obter dados geográficos para centrar o ponto de interesse no mapa.
- **Gerar Percurso de Pontos de Interesse**: Permite obter dados geográficos do percurso entre dois pontos de interesse a ser representado no mapa
- **Visualizar no mapa**: Permite visualizar no mapa a lista de ponto de interesse, tal como a centragem de um ponto de interesse e a geração do percurso entre dois pontos de interesse.

3.5 Diagrama de sequência

Não se fez um estudo exaustivo de diagramas de sequência, optando-se por apresenta um dos mais importantes da solução. O diagrama de sequência da Figura 3.9 ilustra a ordem sequencial de operações aquando a criação de um ponto de interesse.

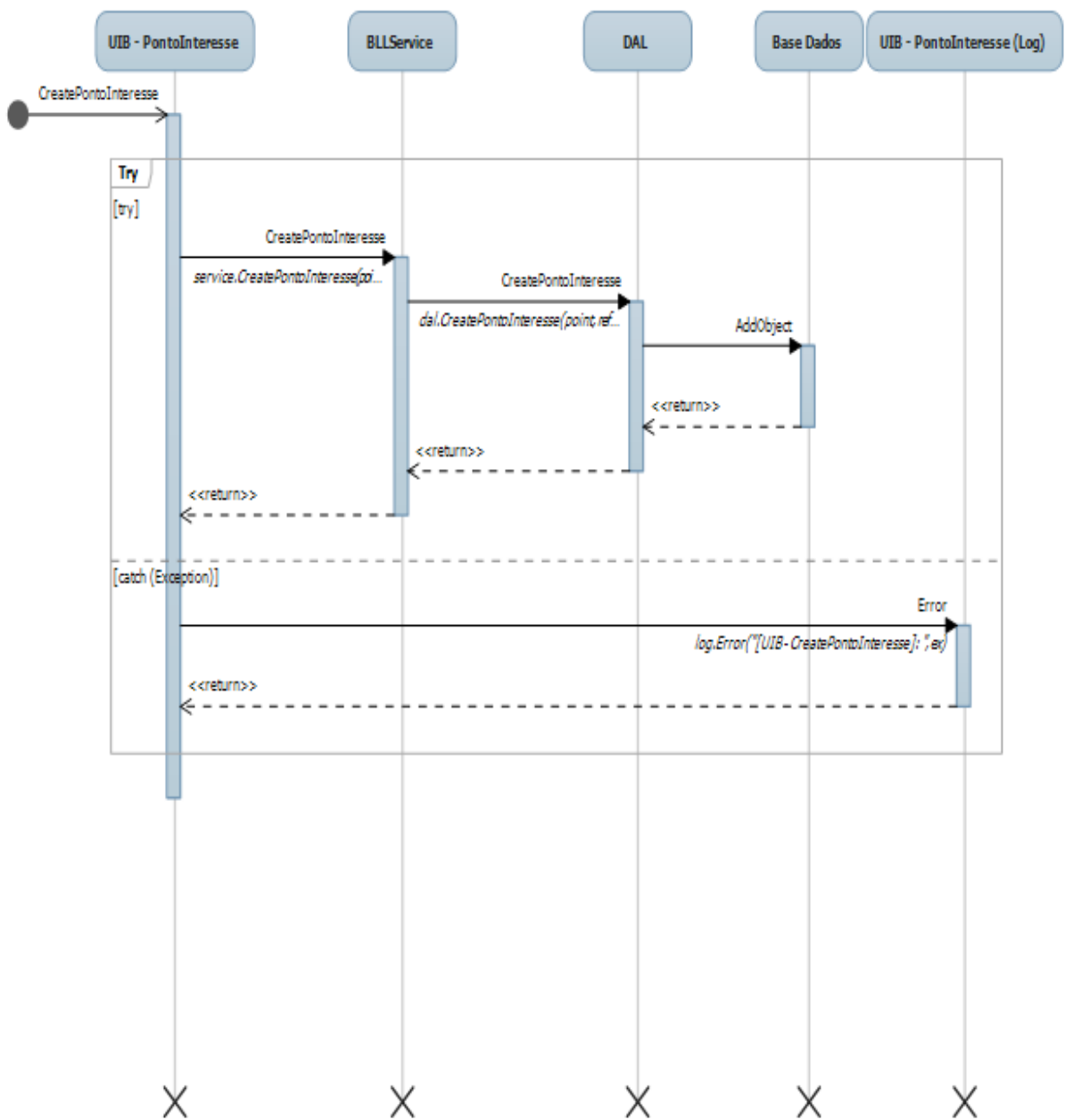


Figura 3.9 - Diagrama de sequência da criação de ponto de interesse

3.6 Diagramas de estrutura de site

O diagrama da Figura 3.10 ilustra os itens do menu da interface do backoffice e as respectivas páginas, com destaque que em “Pontos de Interesse” existe uma componente de geração de mapa.

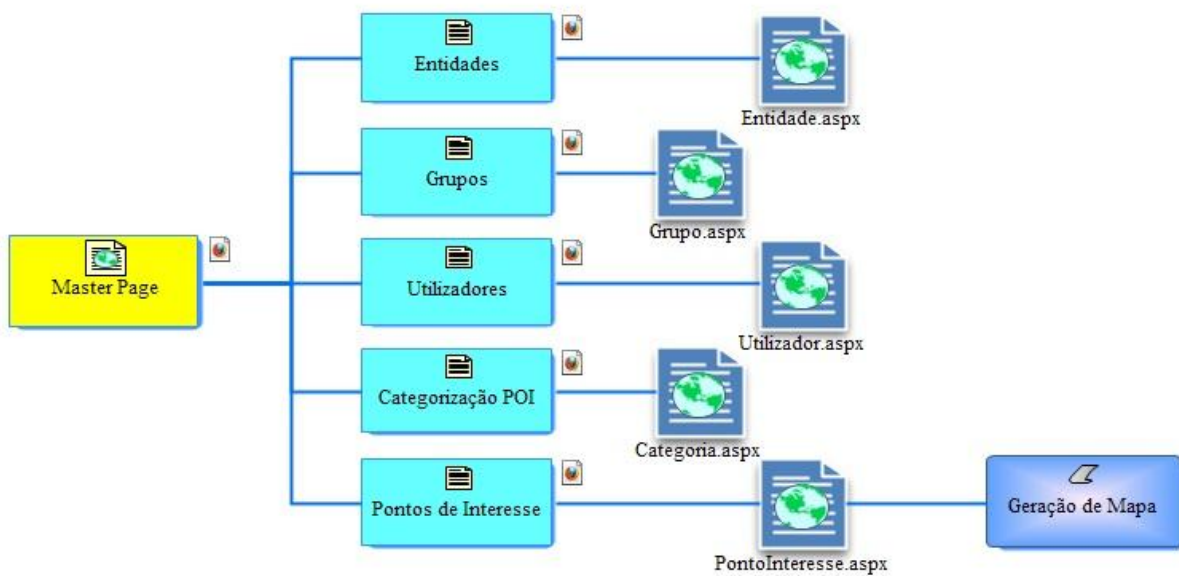


Figura 3.10 - Diagrama de site do backoffice

O diagrama da Figura 3.11 ilustra os itens do menu da interface do frontoffice e as respectivas páginas, as quais invocam a página de geração de mapa MapPOI.aspx.

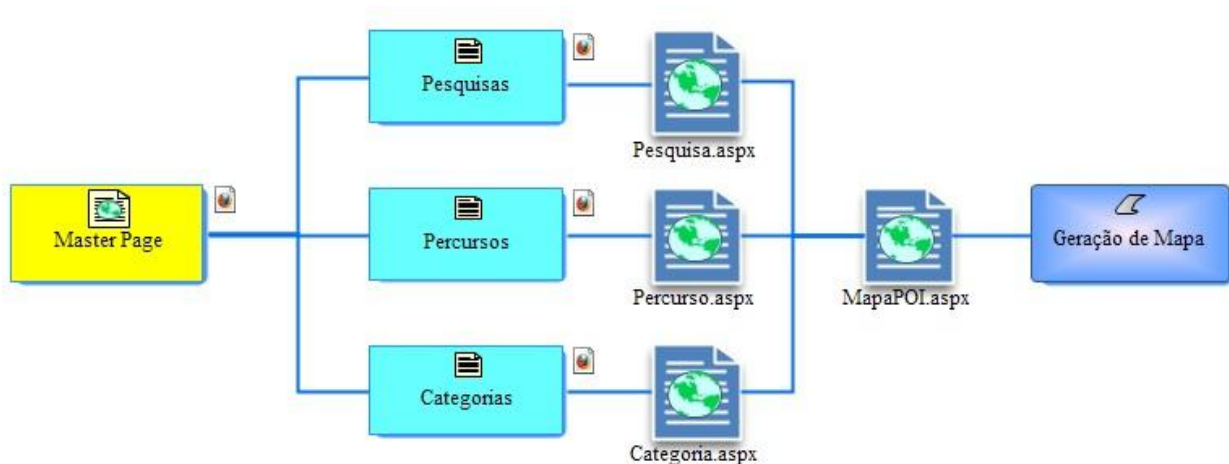


Figura 3.11 - Diagrama de site do frontoffice

Capítulo 4

4 Tecnologias

Neste capítulo são descritas as tecnologias utilizadas no projeto e respectivas justificações. Primeiro será descrita a framework .NET 4.0, que é a plataforma tecnológica onde assenta o ambiente de desenvolvimento desta solução e de seguida será descrita a arquitetura multicamada e seu enquadramento. Posteriormente, são apresentadas as tecnologias de desenvolvimento; para uma melhor compreensão as mesmas são descritas e justificadas de acordo com o seu enquadramento na arquitetura tecnológica do projeto; ou seja, primeiro são apresentadas as tecnologias relacionadas com a primeira camada de acesso a dados da arquitetura, que são o SQL Server, Entity Framework e LINQ; de seguida são apresentadas C# e WCF que fazem parte da camada lógica e serviços e por fim as tecnologias presentes na camada de apresentação são ASP.NET, XML, jQuery, Ajax, JSON e a API Google Maps. Para finalizar é apresentada a tecnologia Log.Net, que está presente em todas as camadas e permite registar os erros e exceções que possam ocorrer na solução.

4.1 Framework .NET 4.0

A framework .NET é uma plataforma de software criada pela Microsoft que permite aos programadores desenvolverem aplicações informáticas mais facilmente.

Esta plataforma possibilita que sistemas de aplicações, novos ou já existentes conetem os seus dados e transações independentemente do sistema operativo, tipo de computador ou dispositivo móvel que sejam utilizados, ou que a linguagem de programação tenha sido utilizada na sua criação. Os componentes principais da framework .NET são: o *Common*

Language Runtime (CLR), que fornece gestão de memória e outros serviços do sistema, e uma extensa biblioteca de classes denominada de *Base Class Library* (BCL), que expõe ao programador as características fundamentais do CLR, e como a BCL é baseada em classes, as mesmas podem inclusivamente serem estendidas pelo programador através de herança de classes [FNET 12].

A framework .NET tem tido uma evolução gradual de modo a tornar esta plataforma mais robusta e seja mais capaz de responder às necessidades aplicacionais. O seu lançamento teve início em 13 de Fevereiro de 2002 com a versão 1.0, passando pelas versões 1.1, 2.0, 3.0, 3.5, 4.0 e 4.5 (lançada em 15 de Agosto de 2012).

A Figura 4.1 ilustra a evolução da framework .NET até à versão 4.0; sendo esta a utilizada no desenvolvimento do projeto.

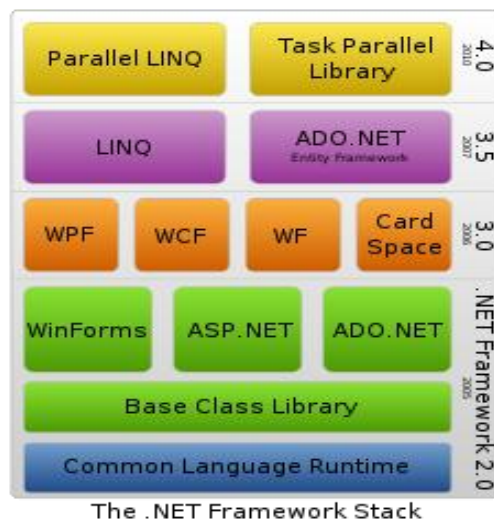


Figura 4.1 - Evolução da framework .NET

A framework .NET fornece um grande conjunto de serviços para o desenvolvimento de aplicações, sendo que alguns foram integrados/melhorados no decorrer da sua evolução. A saber alguns serviços [FNET4 12].

- **Gestão de memória** - Em muitas linguagens de programação, os programadores são responsáveis por alocar e libertar memória na manipulação do ciclo de vida dos objetos. Na framework .NET o CLR fornece esses serviços.
- **Um sistema de tipo comum** - Em linguagens de programação tradicionais, tipos básicos são definidos pelo compilador, o que complica a interoperabilidade entre linguagens. Na framework .NET, tipos básicos são definidos pelo tipo de sistema

- **Uma extensa biblioteca de classes** - Em vez de ter que escrever grandes quantidades de código para lidar com operações comuns de baixo nível de programação, os programadores podem usar uma biblioteca de classes na framework .NET.
- **Frameworks de desenvolvimento e tecnologias** - A framework .NET inclui bibliotecas para áreas específicas de desenvolvimento de aplicações, tais como ASP.NET para aplicações web, ADO.NET para acesso a dados e Windows Communication Foundation (WCF) para aplicações orientadas a serviços.
- **Interoperabilidade de linguagem** - Compiladores de linguagem que visam o .NET Framework emitem um código intermediário chamado Common Intermediate Language (CIL), que, por sua vez, é compilado em tempo de execução.
- **Compatibilidade da versão** - Com raras exceções, as aplicações que são desenvolvidas usando uma versão inferior da framework .NET podem ser executadas sem modificação numa versão superior.

A framework .NET 4.0 (ilustrada na Figura 4.2) fornece muitos recursos novos que permitem desenvolver aplicações de um modo mais produtivo e mais rápido. Tem também a capacidade de resolver conflitos de versão, ou seja, é possível a existência de várias versões do *Common Language Runtime* no mesmo computador.

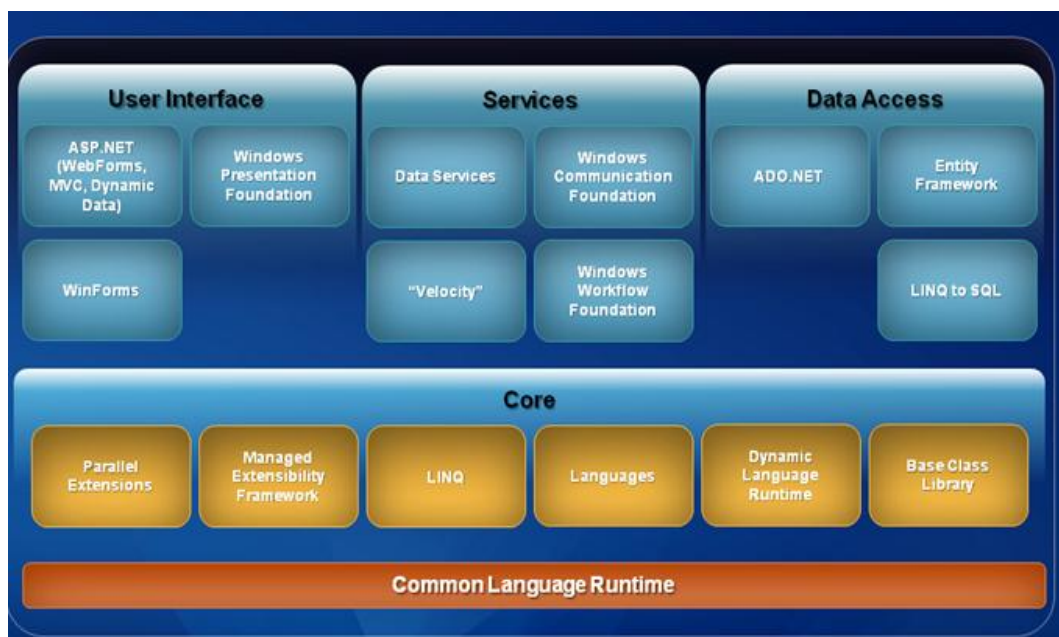


Figura 4.2 - Framework .NET 4.0

Outras inovações e melhorias na framework .NET 4.0 são, por exemplo:

- As instruções lambdas e parâmetros opcionais nas linguagens Visual Basic e C#;
- O aumento do nível de abstração e estabilidade do ADO.NET Entity Framework;
- Melhorias em ASP.NET com novos modelos de User Interface (UI) de JavaScript e capacidades de ligação de dados;
- Melhorias em Windows Presentation (WPF) e Windows Workflow Foundation (WWF) com nova linha de componentes de negócio;
- Melhorias em Windows Communication Foundation (WCF), permitindo programas de fluxo com actividades de mensagens, apoio de correlação, duração de comunicação de duas vias e aumento de recursos de hospedagem. Adição de novos recursos, como serviços de pesquisa, configuração simplificada e melhoria para filas, suporte a REST (REpresentational State Transfer), diagnósticos e desempenho.
- A adição de recursos de programação paralela.

4.2 Arquitetura multicamada

Arquitetura multicamada é uma arquitetura de software em que o sistema aplicacional é dividido em várias camadas, onde cada camada é responsável por um determinado aspeto ou funcionalidade como apresentação, lógica de negócio, persistência de dados [CAM 12].

Projetar uma arquitectura multicamada não é uma tarefa fácil, no entanto, e em contrapartida produz um ambiente com maior flexível e escalável. As arquiteturas multicamadas mais comuns são compostas por três ou quatro camadas, sendo denominadas por camada de apresentação, camada lógica do negócio (BLL), camada de acesso de dados (DAL) e camada de dados. No entanto, as mesmas camadas podem apresentar variantes entre elas, dependendo do sistema que se queira implementar, ou seja, por exemplo pode haver um repositório de entidades que comunica com as camadas DAL e BLL, conforme ilustrado na Figura 4.3.

Num ambiente típico de uma arquitetura multicamada o cliente implementa a camada de apresentação, a BLL e DAL são implementadas em servidores de aplicações e a camada de dados em servidores de base de dados [Souto 07].

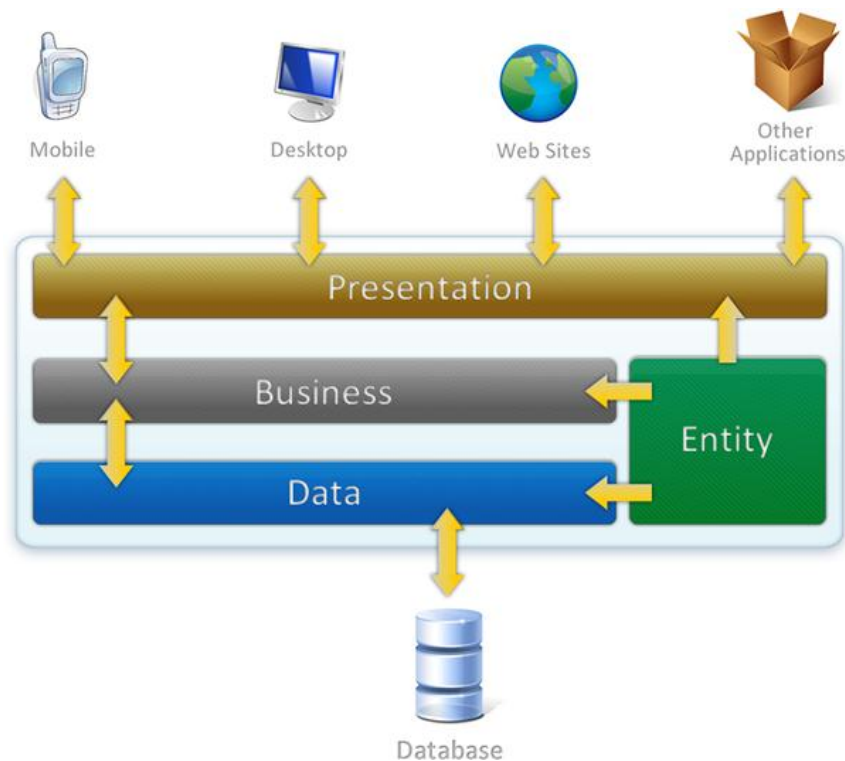


Figura 4.3 - Arquitetura Multicamadas [ArqCamada]

- **Camada de Apresentação:** corresponde ao interface do utilizador com o sistema, assumindo, geralmente, a forma de um *Browser*. Aqui são prestados os mais diversos serviços ao utilizador; nomeadamente a introdução e manipulação de dados sendo, ainda, permitido ao utilizador percorrer o sistema. Esta camada trabalha com os resultados da camada BLL e transforma esses mesmos resultados em algo editável ou de leitura pelo utilizador.
- **Camada Lógica do Negócio:** Permite que os utilizadores partilhem e controlem esta camada isolando-a das outras camadas da aplicação. A BLL funciona entre a camada de apresentação e a DAL, emitindo os pedidos dos dados do cliente à camada da base de dados com a camada de acesso dos dados.
- **Camada de Acesso dos Dados:** Fornece o acesso à base de dados executando um conjunto de instruções de consulta à base de dados. Esta camada é onde se colocará métodos genéricos relacionados com os dados.

- **Camada de Dados:** Esta camada é responsável pela gestão funcional dos dados assegurando os aspetos que se relacionam com a segurança (proteção contra acessos não autorizados) e integridade dos dados (controlo da concorrência e recuperação/tolerância a falhas), sendo armazenados num servidor de base de dados.

Relativamente ao modelo arquitetural da solução desta solução foi projetada uma variante na camada de lógica do negócio, ou seja, tendo em conta que nesta camada é utilizada a tecnologia WCF adaptou-se a camada como sendo também orientada a serviços designando-a de Business Service Layer (BSL); passando de certo modo a arquitetura tecnológica além de ser designada como multicamada também englobar o conceito de arquitetura orientada a serviços.

Vantagens

A arquitetura multicamada apresenta muitas vantagens, entre as quais se destacam as seguintes:

- A performance aumenta devido à carga ser balanceada entre o servidor de aplicações e bases de dados, o que permite atender a um maior número de pedidos.
- A gestão das aplicações é simultaneamente melhorada, pois a manutenção torna-se mais acessível, ou seja, quando é necessário efetuar alterações nas aplicações, o código torna-se mais modular o que reduz a possibilidade de *bugs* que comprometam a fiabilidade de toda a aplicação.
- A lógica aplicacional assume a dimensão menos centralizada pelo que não obriga a que haja uma ligação a um SGBD específico. Como consequência desta vantagem, é possível reutilizar-se o código, integrarem-se recursos heterogéneos e permitir que os clientes (browsers) funcionem de uma forma mais simples, ou seja, estes concentram-se essencialmente na apresentação da informação, podendo escolher o interface que desejarem.

Desvantagens

Apesar de tudo, a construção de uma arquitetura de multicamadas é um trabalho complexo, sendo um dos pontos críticos o fato de nem sempre ser fácil/óbvio separar a lógica das camadas.

4.3 SQL Server

Atualmente vivemos na era da informação, resultado das enormes quantidades de dados geradas a partir da recolha dos mesmos. Para armazenar os dados utilizamos aquilo que se denomina por base de dados, ou seja, coleção estruturada de dados relacionados acerca de um ou vários assuntos. Normalmente uma base de dados é manipulada através de um SGBD (Sistema de Gestão de Base de Dados), em que este pode ser bastante complexo na medida em que se encarrega da perseverança intacta dos dados bem como a organização e acesso aos mesmos [SQLServ 12].

Existem alguns SGBD's como o SQL Server, o Oracle entre outros. O SQL Server é um SGBD robusto usado por sistemas bastante exigentes do mercado, sendo possível a sua integração com a framework .NET, possibilitando assim a construção de rotinas que utilizam linguagens de programação como o C#. Este sistema tem como desvantagem o facto de funcionar apenas em algumas versões do Windows, ao contrário do seu mais direto concorrente Oracle.

No desenvolvimento do projeto optou-se pelo SQL Server 2008, visto que esta versão suporta o ADO.NET Entity Framework e permite modelar estruturas em árvore com recurso ao tipo de dado HIERARCHY ID [SQLServh 12], tendo sido fundamental na modelação de dados da solução implementada.

4.4 Entity Framework

O ADO.NET Entity Framework é a nova plataforma de acesso a dados desenvolvida pela Microsoft e incorporada inicialmente na framework .NET 3.5. Permite criar aplicações em que o acesso a dados é feito com base num modelo concetual e sem utilização de comandos diretos à base de dados. Isto permite que o programador se abstraia totalmente da base de dados (criação de ligações de acesso, comandos, parâmetros, etc.) e utilize apenas objetos durante o desenvolvimento [EF 10].

A versão do Entity Framework (EF) que está disponível na .NET Framework 4.0, tem um conjunto de novas funcionalidades e melhorias, como é o caso de suporte a POCO - *Plain Old*

CLR Objects (permite criar classes que não herdam, nem implementam nenhuma outra classes ou interface), abordagem Model-First (permite criar primeiro o modelo concetual e, com base nele, criar a base de dados), suporte para o uso de funções em LINQ-to-Entities, Complex Types (criação de tipos de dados complexos), *Deferred Loading* ou *Lazy Loading* (capacidade de carregar as propriedades de associação das entidades no momento em que são chamadas).

De acordo com a Figura 4.4 é possível ter a perceção do Entity Framework no enquadramento aplicacional.

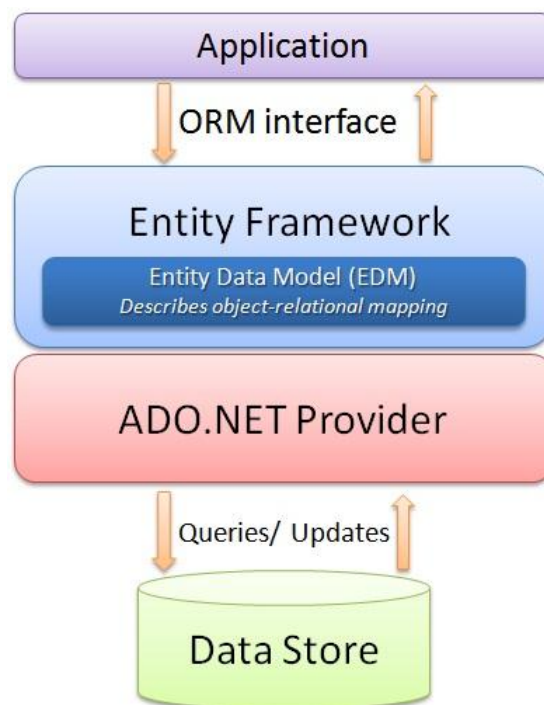


Figura 4.4 - Enquadramento aplicacional de Entity Framework [ArqEF]

O EDM (Entity Data Model) é um conceito que no EF foi implementado de forma a se obter uma maior produtividade. O modelo de entidades de dados e toda a informação de mapeamento de cada entidade para a sua tabela na base de dados é armazenada num ficheiro de representação XML num ficheiro de extensão *edmx*.

Pela descrição do EF constata-se que existem várias vantagens na utilização de um ORM (Object-relational mapping), que tornam a sua adoção quase inevitável. A par deste ORM existem outros com funções semelhantes como o NHibernate pertencente também à Microsoft e o Hibernate escrito em linguagem Java. No entanto, pelas vantagens enunciadas e por um entendimento mais fácil do enquadramento do EF na framework .NET 4.0, optou-se por este ORM no desenvolvimento da solução.

4.5 LINQ

LINQ (Language Integrated Query) é um componente introduzido na Framework .NET 3.0 com o propósito de colmatar a grande dificuldade em se executarem consultas a bases de dados e unificar o modelo de acesso a dados a diferentes fontes de informação como objetos, documentos XML e estruturas de dados. Utilizando LINQ pode-se programar o código de acesso a dados directamente em C# ou Visual Basic e verificar-se a sintaxe em tempo de compilação. Com LINQ não há necessidade de se utilizar diferentes tecnologias para acesso a dados como SQL, XPath, XQuery, entre outras, porque utiliza um modelo unificado de acesso a dados [LINQ 12]. A arquitetura da tecnologia LINQ é representada de acordo com a ilustração da Figura 4.5.

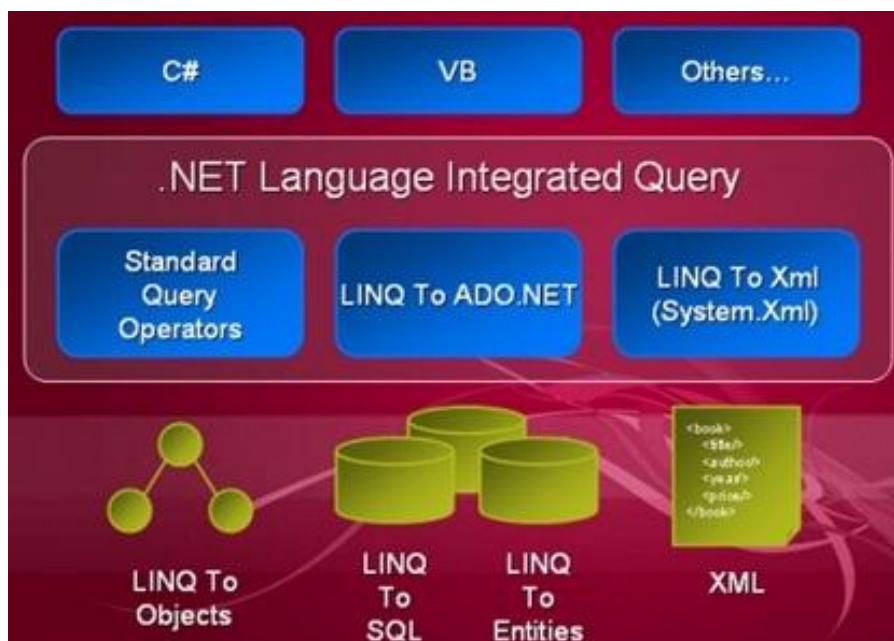


Figura 4.5 - Arquitectura do LINQ [ArqLINQ]

O LINQ é dividido em várias partes, como LINQ to Objects que permite fazer consultas a objectos em memória como arrays, LINQ to SQL que permite fazer consultas a base de dados, LINQ to Entities que permite fazer consultas a base de dados a partir da mesma linguagem usada para construir a lógica de negócio e LINQ to XML que permite criar, modificar e navegar por ficheiros XML.

4.6 C#

C# é uma linguagem de programação orientada a objetos fortemente tipada, desenvolvida pela Microsoft como parte da framework .NET [CSHARP 12]. A sua sintaxe orientada a objetos foi baseada na linguagem C++ mas inclui muitas influências de outras linguagens de programação como o Java; sendo que uma das vantagens da linguagem Java em relação ao C# é possuir compatibilidade com um maior número de plataformas, em que uma aplicação quando desenvolvida na linguagem Java é traduzida pelo seu compilador para os bytecodes, ou seja o código é transformado em código máquina de um processador virtual chamado de Java Virtual Machine (JVM), permitindo a ser executado em qualquer plataforma desde que esteja instalada uma JVM, enquanto que o C# além de suportado pela plataforma .NET também o é, por exemplo, no ambiente de Linux.

Apesar de estas linguagens apresentarem muitas semelhanças, nomeadamente no paradigma de programação orientada a objetos, a linguagem C# tem demonstrado uma evolução mais consistente e atualmente as vantagens são muitos superiores ao Java, como se pode constatar na (Tabela 4.1) que apresenta algumas dessas vantagens.

Tabela 4.1 - Comparação entre Java e C#

<i>Característica</i>	<i>Java</i>	<i>C#</i>
“Boxing” e “Unboxing” - conversão de tipos por valor para tipos por referência	Não. Exige conversão manual.	Sim.
Structs	Não.	Sim.
Passagem de parâmetros por referência	Não.	Sim, de duas maneiras: <i>ref</i> para parâmetros de entrada e saída e <i>out</i> para parâmetro apenas de saída.
Propriedades	Não. Podem ser simuladas com métodos Get/Set, com alguma dificuldade.	Sim, diretamente. A criação de “componentes” é bastante facilitada.
Inteiros sem sinal	Não.	Sim.

Sobrecarga de operadores	Não.	Sim.
Operadores de cast	Um, sintaxe semelhante ao C/C++.	Dois, um semelhante ao C/C++ e o outro “as”. Um retorna <i>null</i> e outro <i>exception</i> em caso de erro de conversão.
Eventos e delegates	Não.	Sim. Um “delegate” é um “ponteiro de função orientado a objetos”, permitindo a associação de um evento de uma classe ao código de uma forma simples e poderosa.
Especificadores de acesso	Quatro.	Cinco. O <i>internal</i> , adicional, especifica acesso apenas no mesmo “assembly”.
Cria objetos COM/COM+	Não.	Sim.

Pelas vantagens acima enunciadas e pelo fato de a linguagem C# também fazer parte da framework .NET optou-se por esta linguagem.

4.7 WCF

Windows Communication Foundation (WCF) é uma tecnologia de desenvolvimento de aplicações distribuídas e orientadas a serviços.

O WCF surgiu na framework .Net 3.0 com o propósito de unificar tecnologias como COM+, .Net Remoting, Web Services e MSMQ (Microsoft Message Queue), porque antes do WCF, era necessário que no desenvolvimento de aplicações o programador utilizasse tecnologias distintas para cada tipo de aplicação; um exemplo seria a criação de Web Services para disponibilizar na Internet algum serviço. Caso este serviço fosse disponibilizado na intranet, deveria ser criada uma aplicação que utilizasse .Net Remoting, porque esta tecnologia utiliza o protocolo TCP enviando ficheiros binários pela rede o que tornava a aplicação muito mais

rápida do que com Web Service (Http/XML). Com a criação do WCF isso deixa de existir, e torna a tarefa de desenvolvimento aplicacional por parte do programador mais simples e ainda proporciona um considerável ganho de performance em relação às tecnologias que o precedem [WCF 12] .

Para que se possa projetar, implantar e hospedar os serviços WCF, deve-se considerar alguns conceitos, nomeadamente os Endpoints que é o que o serviço expõe e seus componentes (Address, Binding e Contract), conforme ilustrado na Figura 4.6.



Figura 4.6 - Endpoint do WCF [EndWCF]

Os componentes de um Endpoint podem ser descritos do seguinte modo:

Address

É o local ou endereço onde reside o serviço. O Address possui dois pontos importantes que são a localização e o protocolo de transporte.

Um Address possui o seguinte formato: *[base address]/[optional URI]*

Já o base address tem o seguinte formato: *[transport]://[machine or domain][:optional port]*

Como transport, o WCF disponibiliza os seguintes canais de comunicação:

http, https, net.tcp, net.pipe, net.msmq e net.p2p

Para se ilustrar melhor o conceito de Address, de seguida são apresentados alguns exemplos:

`http://localhost:8080`

`http://localhost:8080/MeuWCFService`

`net.tcp://localhost:123/MeuWCFService`

`net.pipe://localhost/MeuWCFService`

`net.msmq://localhost/private/MeuWCFService`

`net.msmq://localhost/MeuWCFQueue`

Binding

É o responsável por definir como será a comunicação com o serviço (tcp, http, msmq, etc).

O WCF disponibiliza seis tipos de binding que são:

- BasicHttpBinding
- NetTcpBinding
- NetNamedPipeBinding
- WSHttpBinding
- WSDualHttpBinding
- NetMsmqBinding

Contract

No contrato é onde se definem quais as funcionalidades que um serviço irá expor para os clientes. Existem quatro tipos de contrato, que são:

Service Contracts: define quais operações estarão disponíveis no serviço para o cliente.

Data Contracts: define a estrutura de dados usada no serviço.

Fault Contracts: define os tipos de erros que serão despoletados pelo serviço.

Message Contracts: define a estrutura de mensagens usadas no serviço.

4.8 ASP.NET

O ASP.NET é uma tecnologia da Microsoft para desenvolvimento da camada UI (*User Interface*) de aplicações Web. Os programas em ASP.NET são aplicações centralizadas, residentes num ou mais servidores Web que respondem dinamicamente aos pedidos dos clientes. Estas respostas são dinâmicas, porque o ASP.NET intercepta pedidos para páginas e encaminha esses pedidos para ficheiros de código compilado *just-in-time* (JIT) que podem responder no momento. De acordo com o paradigma da programação orientada a objetos, deve-se sempre separar a parte de processamento da aplicação da parte de interface com o utilizador [ASP 12]. Sendo coisas diferentes, não têm necessariamente de estar misturadas. Uma das grandes inovações das ASP.NET é o *CodeBehind*. Com ele é possível separar o código do servidor (exemplo: Page1.cs) do código HTML (exemplo: Page1.aspx). As páginas

Web ASP.NET ou formulários Web são o elemento fulcral da camada UI no desenvolvimento de aplicações Web, estando os formulários integrados nos ficheiros de extensão “.aspx” .

Na Figura 4.7 pode-se verificar que a aplicação pode ser estruturada em vários componentes, componentes esses que estão divididos em classes só com código e em páginas ASP.NET. Cada vez que se desenvolve uma página ASP.NET é desenvolvida uma nova classe.

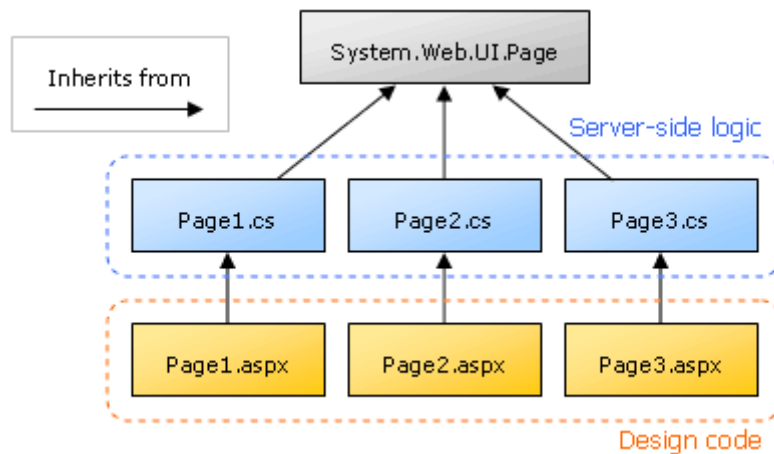


Figura 4.7 - Codebehind em ASP.NET [CodeASP]

O ASP.NET lida também com ficheiros de configuração (**Web.config** e **machine.config**) que contém iniciação e definições para uma aplicação específica, ou porção de uma aplicação. Outro ficheiro de configuração, chamado **machine.web**, contém iniciação e definições nível-máquina. O servidor ignora pedidos para ficheiros Web, porque servi-los poderia constituir uma quebra de segurança. Relativamente aos pedidos, a grande diferença entre um pedido estático e um pedido dinâmico é que um típico pedido Web referencia um ficheiro estático. O servidor lê o ficheiro e responde com o conteúdo do ficheiro requisitado. Com o ASP.NET, não existe tal limitação. Não é necessário responder a um pedido com um ficheiro, pode-se responder com o que se quiser, incluindo ficheiros HTML criados dinamicamente, XML, gráficos ou dados binários. De se referir que o ASP.NET usa todas as linguagens da plataforma .NET.

4.9 XML

Extensible Markup Language (XML) é linguagem de marcação de dados (meta-markup language) que especifica um formato para a criação de documentos com dados organizados de

forma hierárquica, como se pode constatar, frequentemente, em documentos de texto formatados, imagens vetoriais ou bases de dados. Pela sua portabilidade, já que é um formato que não depende das plataformas de hardware ou de software, uma base de dados pode, através de uma aplicação, escrever num ficheiro XML, e uma outra base de dados pode ler estes mesmos dados. O XML é considerado de grande importância na internet e em grandes intranets devido à capacidade de interoperabilidade dos computadores por ter um padrão flexível e aberto e independente de dispositivo [XML 12]. As aplicações podem ser construídas e actualizadas mais rapidamente e também permitem múltiplas formas de visualização dos dados estruturados. Uma característica importante é que uma vez tendo sido recebido o dado pelo cliente, tal dado pode ser manipulado, editado e visualizado sem a necessidade de reativar o servidor. Dessa forma, os servidores tem menor sobrecarga, reduzindo a necessidade de computação e reduzindo também a requisição do canal comunicativo entre cliente e servidor.

No desenvolvimento deste projeto foi indispensável a utilização do XML, porque permitiu através da leitura dos itens de categorização de pontos de interesse construir a hierarquia das respectivas categorias para disponibilizá-la em formato de “árvore” na interface da solução.

4.10 JavaScript

JavaScript (por vezes abreviado **JS**) é uma linguagem de script que é dinâmica, fracamente tipada e que possui funções de primeira classe.

É também uma linguagem de multi-paradigma, apresentando estilos de programação imperativa e funcional e com suporte a orientação de objetos.

O JavaScript é usado principalmente do lado do cliente de uma aplicação, implementado como parte de um navegador da Web e com o intuito de criar user interfaces (UI) avançadas e sites dinâmicos, ou seja, permitir a escrita de funções que são incluídas em páginas HTML e que interagem com o Document Object Model (DOM) da página. Alguns exemplos deste uso são:

- Abrir uma nova janela com controlo programático sobre o seu tamanho, posição e atributos;
- Validar valores de um formulário para garantir que são aceitáveis antes de serem enviados ao servidor;

- Mudar imagens à medida que o rato se movimenta sobre elas.

Pelo fato do código JavaScript funcionar localmente no navegador do utilizador, e não num servidor remoto, o navegador pode responder a tais ações rapidamente. Além disso, o código JavaScript pode detetar acções do utilizador que o HTML sozinho não pode, tais como pressionar teclas. Por ser o JavaScript a única linguagem que a maioria dos browsers populares suportam, tem-se tornado uma linguagem alvo para muitas frameworks em outras linguagens [JS 12].

4.11 jQuery

jQuery é uma biblioteca Open Source de JavaScript muito leve, rápida, intuitiva e que rapidamente (criada em 2006) tornou-se numa das mais populares bibliotecas para programar na Web. Com esta biblioteca melhora-se e simplifica-se de forma significativa a interação entre JavaScript e HTML, cuja função principal do jQuery é simplificar a sintaxe JavaScript com muito menos linhas de código do que com JavaScript. Outra das suas grandes vantagens é ser compatível com todos os browsers existentes [jQuery 12]. O slogan do jQuery é “Write less, do more”.

As principais vantagens do uso de jQuery sobre JavaScript são:

- Acesso direto a qualquer componente do DOM, ou seja, não há necessidade de várias linhas de código para aceder determinados pontos no DOM.
- O uso de regras de estilo não sofre qualquer tipo de limitação devido as inconsistências dos browsers. Mesmo os seletores CSS (Cascading Style Sheets) podem ser usados sem qualquer restrição.
- Manipulação de conteúdos, sem limitações, com poucas linhas de código.
- Suporte a todo o tipo de eventos de interação com o utilizador sem limitações impostas pelos navegadores web.
- Possibilidade de inserir uma grande variedade de efeitos de animação com uma simples linha de código.

- Uso simplificado e sem restrições com AJAX (**A**synchronous **J**avascript and **X**ML) e linguagens de programação, como o ASP.NET.
- Simplificação na criação de scripts.

A título de exemplo é apresentada na Tabela 4.2 dois itens comparativos entre as sintaxes de JavaScript e jQuery.

Tabela 4.2 - Comparação de sintaxe entre JavaScript e jQuery

<i>Sintaxe JavaScript</i>	<i>Sintaxe jQuery</i>
<code>document.getElementsByTagName("p")</code>	<code>\$("p")</code>
<code>document.getElementById("um").setAttribute("class", "cor")</code>	<code>\$("#um").attr("class", "cor")</code>

Face às vantagens acima enumeradas do jQuery sobre o JavaScript, no desenvolvimento deste projeto e no âmbito de programação do lado do cliente e interação com o utilizador poder-se-ia utilizar somente o jQuery, no entanto, devido a algumas situações específicas de desenvolvimento e pessoalmente ainda não ter o mesmo nível de domínio de programação do jQuery em relação ao JavaScript, optou-se pela utilização conjunta destas tecnologias.

4.12 Ajax

Ajax não é uma linguagem de programação mas sim uma técnica para criar melhores aplicações Web, mais rápidas e interativas. Embora o nome sugira uma linguagem de programação, Ajax não passa de uma sigla **A**synchronous **J**avascript and **X**ML, ou seja, esta técnica permite que JavaScript ou uma plataforma com base nesta tecnologia (por exemplo jQuery) comunique diretamente com o servidor usando o objeto XMLHttpRequest do browser. Assim é possível trocar dados entre o servidor e o cliente sem precisar de recarregar a página, poupando tempo e tráfego. Ajax é baseado em JavaScript, XML, HTML e CSS, estes Web Standards são suportados pela maioria dos browsers permitindo assim que as aplicações Ajax sejam independentes da plataforma e do browser. Em aplicações sem Ajax cada vez que o cliente dá entrada de um novo input, seja um *form*, um *click* de um botão ou até mesmo um *link*, é feito um pedido ao servidor por GET ou POST, o servidor gera a resposta e envia para o cliente que por sua vez gera uma nova página com os dados enviados

pele servidor [Ajax 12]. A comparação entre os modelos tradicional da web e o modelo Ajax encontra-se ilustrado na Figura 4.8.

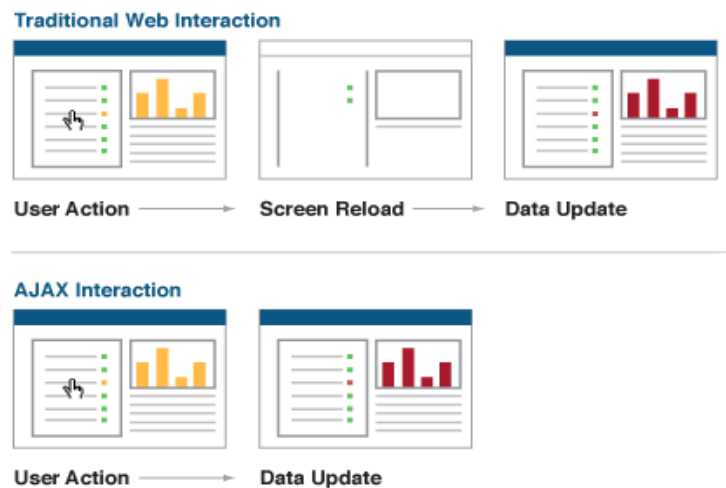


Figura 4.8 - Modelo tradicional web e modelo Ajax [ModAJAX]

Com o Ajax é possível pedir ao servidor apenas os dados referentes à modificação que se deseja fazer, deixando o restante conteúdo da página no estado inicial. Para isso usa-se o objeto XMLHttpRequest que através do JavaScript (ou por exemplo jQuery) e da manipulação dinâmica da DOM (*Document Object Model*), permite a criação de aplicações mais interativas e com uma maior capacidade de resposta .

4.13 JSON

JSON (JavaScript Object Notation) é uma formatação leve de troca de dados, também conhecido como um sucessor do XML. JSON é em formato texto muito simples e ao mesmo tempo elegante, além de ser um formato para troca de informações entre o browser e o servidor. O JSON também é mais rápido e mais leve que o XML. Enquanto que no XML tratam-se todos os dados como string, os objetos JSON são tipados (string, number, array, boolean). Um objeto JSON pode ter a forma de duas estruturas:

- Uma coleção de pares nome/valor
- Uma lista de valores

A simplicidade de JSON tem resultado como uma alternativa a XML em AJAX.

Uma das vantagens reivindicadas de **JSON** sobre XML como um formato para intercâmbio de dados neste contexto, é o fato de ser muito mais fácil escrever um analisador JSON. Em JavaScript, JSON pode ser analisado usando a função `eval()`. Isto foi importante para a aceitação de JSON dentro da comunidade AJAX devido à presença deste recurso de JavaScript em todos os browsers atuais [JSON 09].

Na prática, os argumentos a respeito da facilidade de desenvolvimento e desempenho do analisador são raramente relevados devido aos interesses de segurança no uso de `eval()` e a crescente integração de processamento XML nos browsers modernos. Por esta razão JSON é tipicamente usado em ambientes onde o tamanho do fluxo de dados entre o cliente e o servidor é de grande. Enquanto JSON é frequentemente posicionado "em confronto" com XML, não é incomum ver tanto JSON como XML sendo usados na mesma aplicação, apesar da performance de JSON ser superior ao XML.

Para o intercâmbio de dados entre cliente e servidor na solução deste projeto, nomeadamente manipulação de dados de geração de mapa optou-se por estruturar os dados segundo a filosofia do JSON em conjunto com as tecnologias AJAX e jQuery.

4.14 API Google Maps

Uma Application Programming Interface (API) é um conjunto de rotinas, padrões, estruturas de dados, classes e objetos estabelecidos por uma biblioteca de software ou por serviços de sistemas operativos com o objetivo de dar suporte à criação de aplicações para uma determinada tecnologia.

A Google criou uma API para facilitar a vida de programadores que gostariam de criar aplicações que tivessem o Google Maps integrado sem que para isso a aplicação estivesse no servidor da Google. Através desta API, é possível colocar o Google Maps completo em qualquer outro site, com interfaces e funcionalidades customizadas através da linguagem JavaScript. Essa API é formada por um conjunto de classes JavaScript que fornecem uma interface para o utilizador construir aplicações para exibir mapas, fazer pesquisas de endereço, ampliar, acrescentar pontos de interesse e descrições no mapa, adicionar legendas, entre outras funcionalidades. A Figura 4.9 ilustra a geração de um mapa centrado sobre um endereço e identificado por um marcador. O código com recurso à API do Google Maps que originou o mapa referido é o apresentado a seguir à figura.



Figura 4.9 - Geração de mapa com API Google Maps

```
function geoMorada() {  
    var morada = "Bragança";  
    geocoder.geocode( { 'address': morada}, function(results, status) {  
        if (status == google.maps.GeocoderStatus.OK) {  
            var posicao = results[0].geometry.location;  
            map.setCenter(posicao);  
            var marker = new google.maps.Marker({  
                map: map,  
                position: posicao  
            });  
        } else {  
            alert("Sem sucesso: " + status);  
        }  
    }  
}
```

4.15 Log4net

Log4net é uma biblioteca open source que permite que aplicações da framework .NET registem os outputs para uma variedade de fontes (por exemplo o browser ou ficheiros). Log4net é uma variante da biblioteca Log4j que é usado em Java. Log4net proporciona um mecanismo simples de log (ou registo) de informação [LOG 12]. Estas informações são registadas através de um ou mais níveis de registos. Níveis esses que são os descritos na (Tabela 4.3).

Tabela 4.3 - Descrição dos níveis de registo do Log4net

<i>Nível</i>	<i>Descrição</i>
OFF	A classificação mais alta possível, e destina-se a desativar o registo.
FATAL	Erros graves que causam interrupção prematura.

ERRO	Outros erros de execução ou condições inesperadas.
WARN	Situações de tempo de execução inesperadas, mas que não são necessariamente "erros".
INFO	Informação de eventos de tempo de execução (início / fecho).
DEBUG	Informações pormenorizadas sobre o processo de execução do sistema.
TRACE	Informações mais detalhadas sobre o processo de execução do sistema.

No Aplicação do projeto de modo a não se visualizar no browser erros ou exceções originadas eventualmente pelo sistema optou-se por guardar num ficheiro de texto os registos de log do sistema de acordo com a tecnologia Log4net.

Capítulo 5

5 Implementação

Neste capítulo é descrita a implementação de todas as funcionalidades da solução aplicacional tal como as respetivas justificações e opções inerentes ao desenvolvimento tecnológico.

Primeiro será descrita a organização e estrutura tecnológica escolhida no ambiente de desenvolvimento para a implementação da aplicação. De seguida é apresentada o modo de autenticação na aplicação e respetivo enquadramento no modelo de dados. Posteriormente, para o BackOffice e FrontOffice são descritas e ilustradas as respetivas funcionalidades; para as mesmas são também identificadas as operações funcionais e apresentada a respetiva lógica e enquadramento técnico.

5.1 Estrutura do ambiente de desenvolvimento

De modo a se implementar a aplicação em questão e seguir os pressupostos descritos anteriormente na análise, modelação e tecnologias, principalmente o modelo de arquitetura por camadas, optou-se a nível do ambiente de desenvolvimento (Visual Studio 2010) por se criar uma *solution* que integrasse três *projects* (cada um associado à respetiva camada). De seguida será apresentada a respetiva organização das camadas e descrição dos principais componentes.

DAL (Tipo de *Project* – Class Library)

- **SIG.edmx** – Ficheiro associado à tecnologia Entity Framework (EF) em que “transcreve” o modelo de dados presente no sistema de gestão de base de dados.

- **Entidades** – Pasta com classes associadas às respetivas entidades homónimas presentes em SIG.edmx e onde se efectuam *queries* no tratamento de informação de dados.
- **ModelSIG.tt** – Ficheiro onde é gerado através do EF o modelo de classes e onde se pode caracterizá-las de modo (por exemplo) as camadas superiores terem acesso a operações inerentes a este modelo, nomeadamente a tecnologia WCF.

BLLServiceLayer (Tipo de *Project* – WCF Service Application)

- **References** – Componente onde se inclui a referência do *project* DAL, de modo as camadas DAL e BLLServiceLayer poderem comunicar entre si.
- **Serviços** – Criação dos serviços, compostos por interfaces e classes funcionais. Exemplo: IEntidade.cs, Entidade.svc e Entidade.svc.cs.
- **Web.config** – Ficheiro de configuração do ABC (Address, Binding, Contract) dos serviços.

UIBackOffice e UIFrontOffice (Tipo de *Project* – ASP.NET Web Application)

- **References** – Componente onde se inclui a referência do *project* BLLServiceLayer, de modo as camadas BLLServiceLayer e UIBackOffice/UIFrontOffice poderem comunicar entre si.
- **Service References** – Componente onde se inclui as referências aos serviços, de modo a estes poderem ser invocados nas camadas de *User Interface*.
- **Páginas Web** – Criação dos layouts das funcionalidades e respetiva invocação de lógica.
- **Scripts** – Ficheiros associados às tecnologias JavaScript e jQuery onde contém métodos de interação com o cliente e métodos de manipulação e interação com o mapa respetivamente.

- **Símbolos** – Pasta onde contém os ícones associados às respectivas categorias de pontos de interesse a serem visualizados sobre o mapa.
- **Web.config** – Configurações adequadas ao funcionamento correto da aplicação Web.

5.2 BackOffice

Na implementação das funcionalidades e toda a envolvimento do BackOffice tentou-se criar um sistema de lógica e interface padrão e coerente entre as diferentes funcionalidades de modo futuramente ser mais fácil e intuitivo a adição de funcionalidades ao sistema e/ou atualização das mesmas funcionalidades e a criação de uma interação “amigável” com o cliente. Como este é um sistema de gestão de informação, o mesmo, de um modo genérico para cada funcionalidade apresenta as operações CRUD (*Create, Read, Update, Delete*), associadas da seguinte forma:

- Create - Criação de novo item
- Read – Pesquisa, consulta e listagem de informação
- Update - Edição
- Delete - Eliminação

5.2.1 Autenticação e API Membership

Como já foi referido, a solução deste sistema de informação geográfica de gestão de pontos de interesse apresenta funcionalidades de gestão associados aos referidos pontos. No entanto, neste subcapítulo deve-se destacar as funcionalidades de *Grupos, Utilizadores* e o método de autenticação, pois decidiu-se optar por integrar no modelo de dados da solução as tabelas associadas à API Membership de ASP.NET. Esta API providência um vasto leque de opções para configurar os utilizadores de uma aplicação em que se destacam as seguintes características:

- Criar e apagar utilizadores, quer programaticamente quer através de um utilitário da própria API.
- Fazer *reset a passwords* e enviar automaticamente emails para os utilizadores.

- Pesquisar utilizadores por parâmetros, como por exemplo por email, e apresentá-los como uma lista e respetivos detalhes.
- Com base numa série de controlos incluídos na framework .NET, estes permitem fazer o login e registar novos utilizadores.
- Configurar grupos de acesso dentro da aplicação, conhecidos por *Roles*.

Pelas características enumeradas da API Membership pode-se constatar que esta é uma abordagem prática e eficiente na autenticação e gestão e configuração de utilizadores.

Deste modo optou-se por se implementar o layout de login e registo de utilizadores (ver Anexo A.1) e das funcionalidades de gestão de *Utilizadores* e *Grupos* com base nas premissas da API Membership.

Se o utilizador não estiver autenticado no sistema o mesmo visualizará a página de entrada ilustrada em Figura 5.1.

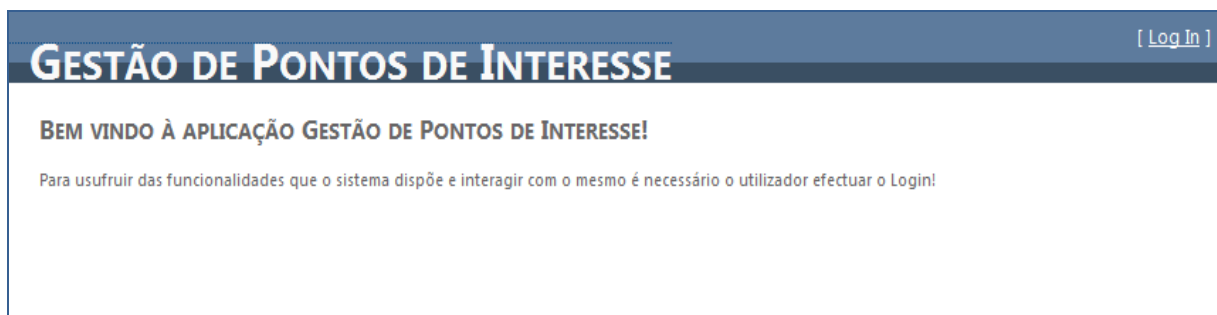


Figura 5.1 - Página Inicial do BackOffice

Se o utilizador estiver autenticado irá visualizar uma página com a indicação das funcionalidades organizadas num menu. Para se validar a autenticação do utilizador e dar a respetiva permissão de visualização de conteúdos pode-se recorrer ao seguinte código:

```
protected void Page_Load(object sender, EventArgs e)
{
    mainMenu.Visible = (Page.User.Identity.IsAuthenticated) ? true : false;
}
```

5.2.2 Entidades

Nesta funcionalidade pode-se efetuar a gestão de entidades (municipais, organizacionais,...) às quais posteriormente se podem associar na funcionalidade *Pontos de Interesse* as respetivas

gestões. O layout principal de *Entidades* apresenta uma secção de operações funcionais e outra onde são listadas as próprias entidades como ilustra a Figura 5.2.

ID	Nome	Morada	Eliminar
1	CMLisboa	Paços do Concelho - Praça do Município, Lisboa	Eliminar
2	CMPBarca	Rua Cons ^o Rocha Peixoto, Ponte da Barca	Eliminar
3	CMBragança	Forte S. João de Deus 5300-263, Bragança	Eliminar
4	ProTurismoLda	Av. Liberdade, Lisboa	Eliminar
6	EventosParty	Av. Aliados, Porto	Eliminar
12	RadicalFree	Rua do Souto, Braga	Eliminar

Figura 5.2 - Listagem de entidades

As operações funcionais representadas pelos botões no layout permitem o seguinte:

- **Pesquisar** – Pesquisa entidades de acordo com os parâmetros de “Nome” e “Morada” e em modo de texto livre e lista numa grelha as respetivas entidades.
- **Limpar** – Limpa os valores introduzidos nas caixas de texto de pesquisa e lista na grelha de resultados todas as entidades.
- **Novo** – Permite criar uma nova entidade. Ao se clicar neste botão é apresentado um formulário com campos que identificam a identidade (ver Anexo A.2). Nesse formulário após se clicar no botão “Gravar” é redirecionado de volta ao painel principal da funcionalidade com a grelha de resultados a apresentar também a entidade anteriormente criada.

Outras operações associadas às entidades estão presentes na grelha de resultados em que se ao clicar no valor da coluna “ID” é redirecionado para um painel de detalhes da respetiva entidade e onde se pode editar a mesma. Na coluna “*Eliminar*” da mesma linha da grelha de entidades é possível eliminar essa mesma entidade.

5.2.3 Grupos

Nesta funcionalidade como já foi referido, é onde se gere e se diferenciam os grupos que posteriormente na funcionalidade *Utilizadores* são atribuídos aos utilizadores de modo a que entre estes se diferencie os acessos e permissões que cada um usufruirá.

De acordo com a solução apresentada para este sistema SIG optou-se por se indicar e sugerir um critério de nomes padrão de grupos de modo a tornar a sua gestão coerente e intuitiva; ou seja, além dos grupos “Administrador” e “Convidado” em que os utilizadores de acordo com as permissões associadas a estes grupos terão um controlo a nível geral do BackOffice e acesso a informação de todas as entidades; já a adição de grupos com designação “Administrador_<Entidade>” e “Convidado_<Entidade>” em que <Entidade> é o nome de uma determinada entidade, permitem que os utilizadores de acordo com as permissões associadas a estes grupos terão somente o controlo de informação que englobe exclusivamente as respetivas entidades. O propósito de haver os dois tipos de administrador, em que um tem controlo total sobre a informação de todas as entidades e outro associado a uma determinada entidade pode gerir exclusivamente essa mesma entidade, não havendo a necessidade desse mesmo administrador depender do superior, podendo assim gerir pontos de interesse do domínio da sua entidade. De seguida na (Tabela 5.1) são apresentados os diferentes grupos e respetivas permissões para as diferentes funcionalidades.

Tabela 5.1 - Grupos e respetivas permissões

<i>Grupo</i>	<i>Permissões</i>
Administrador	Acesso total
Administrador_<Entidade>	<ul style="list-style-type: none">• “Entidades”, “Grupos” e “Utilizadores” – Sem acesso e permissões.• “Categorização de POIs” – Permissões de consulta e pesquisa.• “Pontos de Interesse” – Permissões de pesquisa, criação, consulta, edição, eliminação e listagem de

Administrador_<Entidade>	pontos de interesse exclusivamente da entidade <Entidade>.
Convidado	<ul style="list-style-type: none"> • “Entidades”, “Grupos” e “Utilizadores” – Sem acesso e permissões. • “<i>Categorização de POIs</i>” – Permissões de consulta e pesquisa. • “<i>Pontos de Interesse</i>” – Permissões de pesquisa, consulta e listagem de pontos de interesse associados a todas as entidades.
Convidado_<Entidade>	<ul style="list-style-type: none"> • “<i>Entidades</i>”, “<i>Grupos</i>” e “<i>Utilizadores</i>” – Sem acesso e permissões. • “<i>Categorização de POIs</i>” – Permissões de consulta e pesquisa. • “<i>Pontos de Interesse</i>” – Permissões de pesquisa, consulta e listagem de pontos de interesse exclusivamente da entidade <Entidade>.

O layout principal de *Grupos* apresenta uma secção de operações funcionais e outra onde são listados os próprios grupos como ilustra a Figura 5.3.

Bem vindo **hsouto** ! [[Log Out](#)]

GESTÃO DE PONTOS DE INTERESSE

Acesso

Entidades

Grupos

Utilizadores

Backoffice

Categorização POI

Pontos de Interesse

Grupo

Pesquisar
Limpar
Novo

Lista de Grupos

Grupo	Número de Utilizadores	Eliminar
Administrador	2	Eliminar
Administrador_Bragança	1	Eliminar
Administrador_CMLisboa	2	Eliminar
Administrador_CMPBarca	2	Eliminar
Administrador_EventosParty	1	Eliminar
Administrador_ProTurismoLda	1	Eliminar
Administrador_RadicalFree	2	Eliminar
Convidado	1	Eliminar
Convidado_Bragança	2	Eliminar
Convidado_CMLisboa	3	Eliminar

1 2

Figura 5.3 - Listagem de grupos

As operações funcionais representadas pelos botões no layout permitem o seguinte:

- **Pesquisar** – Pesquisa grupos de acordo com o nome introduzido e em modo de texto livre (nome contido em grupos) e lista numa grelha os respectivos grupos.
- **Limpar** – Limpa os valores introduzidos nas caixas de texto de pesquisa e lista na grelha de resultados todos os grupos.
- **Novo** – Permite criar um ou um conjunto de novos grupos.

Outras operações associadas aos grupos estão presentes na grelha de resultados em que se ao clicar no valor da coluna “ID” é redirecionado para um painel de detalhes do respetivo grupo e onde se pode editar o mesmo. Na coluna “Eliminar” da mesma linha da grelha de grupos é possível eliminar esse mesmo grupo. De se realçar que na lista de grupos, cada um apresenta a indicação do número de utilizadores pertencentes a esse mesmo grupo. Para tal utilizou-se o seguinte código com base na API Membership.

```
foreach (string role in Roles.GetAllRoles())
{
    RoleDataForMP rd = new RoleDataForMP();
    rd.RoleName = role;
    rd.NumberOfUsersInRole = Roles.GetUsersInRole(role).Length;
    roleList.Add(rd);
}
```

Ao se clicar no botão “Novo” é apresentado o layout de acordo com a ilustração da Figura 5.4. Pelo fato da funcionalidade de “Grupos” conter um único atributo informativo, optou-se ao se clicar na operação “Novo” a interface respetiva oferecer a possibilidade criar vários grupos para esta ação. Para tal implementou-se uma estrutura de dados que armazenasse temporariamente os grupos e os apresentasse numa grelha à medida que os mesmos eram adicionados e só na operação “Gravar” é que os criava definitivamente.

GESTÃO DE PONTOS DE INTERESSE Bem vindo **hsouto** ! [[Log Out](#)]

Grupos

Perfil: Administrador Convidado

Entidade: RadicalFree

Grupo	Eliminar
Convidado	Eliminar
Administrador_CMPBarca	Eliminar
Convidado_CMLisboa	Eliminar
Convidado_CMBragança	Eliminar
Convidado_RadicalFree	Eliminar
Administrador_ProTurismoLda	Eliminar
Convidado_EventosParty	Eliminar
Convidado_CMPBarca	Eliminar
Administrador_CMLisboa	Eliminar
Administrador_CMBragança	Eliminar

Figura 5.4 - Criação de novos grupos

As operações funcionais representadas pelos botões no layout permitem o seguinte:

- **Criar Novo Grupo** – Adiciona um grupo, que é a concatenação do tipo de perfil, *underscore* e entidade, à grelha de novos grupos. Se não se escolher uma entidade o novo grupo adicionado será o homónimo do tipo de perfil selecionado.
- **Gravar** – Guarda no sistema de gestão de base de dados os grupos criados anteriormente e apresenta-os na grelha do painel principal da funcionalidade.
- **Cancelar** – Cancela a criação de novos grupos.

5.2.4 Utilizadores

Esta funcionalidade permite, como o nome indica, a gestão de utilizadores que consiste em gerir a informação que identifica o utilizador e associar-lhe um ou mais grupos. Apesar de o

utilizador ao iniciar a aplicação ter a possibilidade de registar-se no sistema, o mesmo só terá acesso às funcionalidades se o administrador posteriormente ao editar o seu registo atribuir-lhe um ou mais grupos. Se o processo completo de registo de utilizadores for efetuado pelo administrador, este inicialmente pode associar-lhes os respetivos grupos.

O layout principal de *Utilizadores* apresenta uma secção de operações funcionais e outra onde são listados os próprios utilizadores e respetivos grupos associados, como ilustra a Figura 5.5.

The screenshot shows the 'GESTÃO DE PONTOS DE INTERESSE' application. On the left is a navigation menu with items: Acesso, Entidades, Grupos, Utilizadores (highlighted), Backoffice, Categorização POI, and Pontos de Interesse. The main content area has a header with 'Bem vindo hsouto ! [Log Out]'. Below the header is a search form with two input fields labeled 'Utilizador' and 'Email', and three buttons: 'Pesquisar', 'Limpar', and 'Novo'. Below the search form is a section titled 'Lista de Utilizadores' containing a table with the following data:

Utilizador	Email	Grupos	Última Entrada	Acesso	Eliminar
administrador	administrador@gmail.com	Administrador	22-11-2012 04:35:40	<input checked="" type="checkbox"/>	Eliminar
anaalves	anaalves@gmail.com	Convidado	19-11-2012 04:18:19	<input checked="" type="checkbox"/>	Eliminar
anacristina	anacristina@gmail.com	Convidado_CMPBarca	17-11-2012 04:24:45	<input checked="" type="checkbox"/>	Eliminar
angela	angela@gmail.com	Convidado_CMPBarca	17-11-2012 04:24:06	<input checked="" type="checkbox"/>	Eliminar
henriquesouto	henriquesouto@gmail.com	Convidado_Bragança; Convidado_CMLisboa	19-11-2012 04:31:01	<input checked="" type="checkbox"/>	Eliminar
hsouto	hsouto@gmail.com	Administrador	22-11-2012 04:48:47	<input checked="" type="checkbox"/>	Eliminar
josesouto	josesouto@gmail.com	Administrador_CMLisboa; Administrador_CMPBarca	17-11-2012 04:20:30	<input checked="" type="checkbox"/>	Eliminar
mariafatima	mariafatima@gmail.com	Administrador_CMLisboa; Administrador_CMPBarca	17-11-2012 04:21:52	<input checked="" type="checkbox"/>	Eliminar

At the bottom of the table, there is a pagination bar showing '1 2'.

Figura 5.5 - Listagem de utilizadores

As operações funcionais representadas pelos botões no layout permitem o seguinte:

- **Pesquisar** – Pesquisa utilizadores de acordo com os parâmetros de “Utilizador” e “Email” em modo de texto livre e lista numa grelha os respetivos utilizadores.
- **Limpar** – Limpa os valores introduzidos nas caixas de texto de pesquisa e lista na grelha de resultados todos os utilizadores.
- **Novo** – Permite criar um novo utilizador.

Outras operações associadas aos utilizadores estão presentes na grelha de resultados em que se ao clicar no valor da coluna “ID” é redirecionado para um painel de detalhes do respetivo utilizador e onde se pode editar o mesmo. Na coluna “Eliminar” da mesma linha da grelha de

utilizadores é possível eliminar esse mesmo utilizador, tal como as referências aos grupos a que pertence.

Ao se clicar no botão “Novo” é apresentado o layout de acordo com a ilustração da Figura 5.6. Nesta operação teve-se o cuidado de minimizar a redundância de atribuição de grupos aos utilizadores optando-se por apresentar no formulário a lista de grupos disponíveis oferecendo assim numa única invocação desta operação a possibilidade de associação de um ou mais grupos ao novo utilizador.

The screenshot shows a web application interface for user management. The main header is 'GESTÃO DE PONTOS DE INTERESSE' with a user greeting 'Bem vindo hsouto ! [Log Out]'. A sidebar on the left contains navigation links: 'Acesso', 'Entidades', 'Grupos', 'Utilizadores', 'Backoffice', 'Categorização POI', and 'Pontos de Interesse'. The main content area is titled 'Novo Utilizador' and contains a form with the following fields and controls:

- Utilizador:** Text input containing 'IPBraganca'.
- Email:** Text input containing 'geral@ipb.pt'.
- Password (min. de 6 caracteres):** Password input field with masked characters.
- Acesso:** Checkboxes for 'Permitido?' (checked).
- Grupo(s):** A list box containing 'Administrador', 'Administrador_Bragar' (checked), 'Administrador_CMList', and 'Administrador_CMPBa'.
- Buttons:** 'Gravar' and 'Cancelar' buttons at the bottom.

Figura 5.6 - Criação de um novo utilizador

Para se criar um novo utilizador recorreu-se novamente à API Membership conforme demonstra o seguinte código.

```
internal void CreateUtilizador()
{
    .
    .
    .
    ArrayList alGroups = new ArrayList();
    try
    {
        foreach (ListItem item in chkListUserGrupos.Items)
        {
            if (item.Selected)
                alGroups.Add(item.Text);
        }
        roles = (string[])alGroups.ToArray(typeof(string));
        UtilizadorMembership.Insert(user, isApproved, email, password, roles);
    }
    catch (Exception ex)
    {
        log.Error("[UIB (Utilizador) - CreateUtilizador]: ", ex);
    }
}
```

5.2.5 Categorização POI

A “Categorização POI” permite criar e gerir classificações de categorias as quais posteriormente serão associadas a pontos de interesse. Para a implementação deste layout face ao tipo de informação a apresentar e respetiva disposição entendeu-se que a interface ilustrada na Figura 5.7 era a mais adequada. Esta interface é composta por uma secção de pesquisa em que cada controlo contém os valores de classificação de categoria correspondente ao respetivo nível de hierarquia. Outra secção corresponde à representação hierárquica das classificações de categorias. O layout de “Categorização POI” por omissão apresenta a classificação da primeira categoria do primeiro nível selecionado com a respetiva informação associada na secção de detalhe.

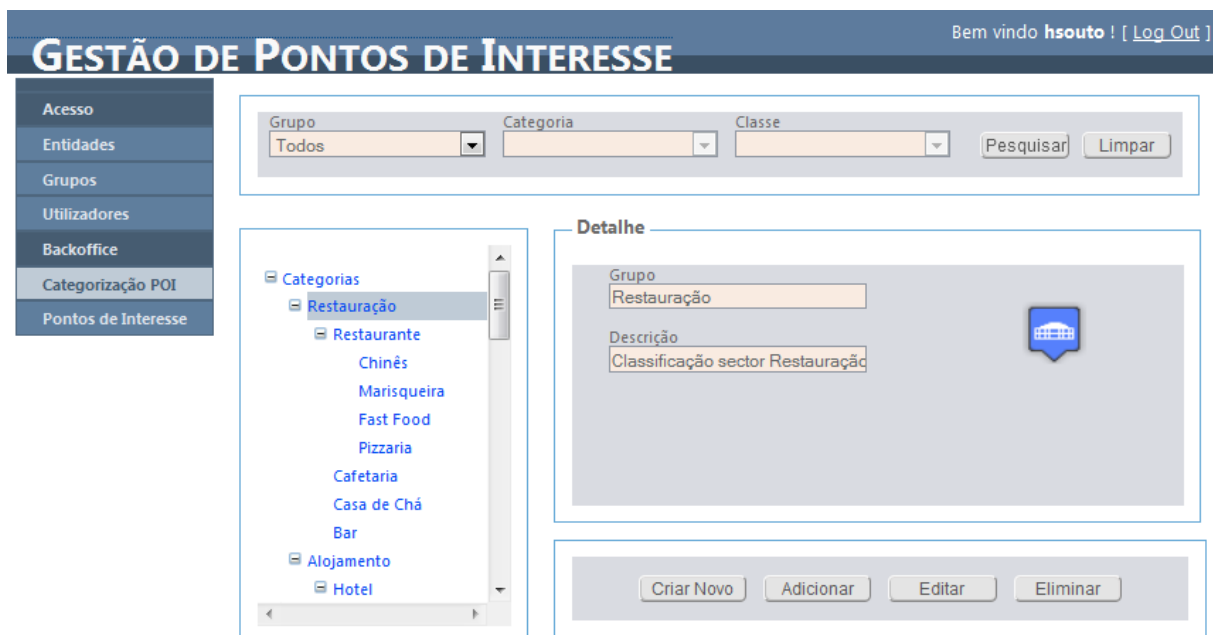


Figura 5.7 - Categorização de POIs

As operações funcionais representadas pelos botões no layout permitem o seguinte:

- **Pesquisar** – Pesquisa classificações de categorias de acordo com os valores dos seus níveis. Se por exemplo se pesquisar por “Grupo” e “Categoria” serão apresentadas as respetivas classificações (“Grupo”, “Categoria” e “Classes”) hierarquicamente no formato em “árvore” com o detalhe de informação do grupo pesquisado (Ver Anexo A.5).
- **Limpar** – Limpa os valores selecionados das classificações de categoria e retorna todas as classificações.

O layout apresenta também uma secção de operações de gestão da categorização de pontos de interesse composta pelos seguintes botões:

- **Criar Novo** – É apresentado o layout ilustrado na Figura 5.8 e oferece a categorização para todos os níveis de classificação de categorias numa única operação, ou seja, é possível criar uma nova categorização de acordo com as variantes {Grupo}, {Grupo, Categoria} ou {Grupo, Categoria, Classe}.

The screenshot displays the 'Gestão de Pontos de Interesse' web application. At the top, there is a navigation menu with options: 'Acesso', 'Entidades', 'Grupos', 'Utilizadores', 'Backoffice', 'Categorização POI', and 'Pontos de Interesse'. The main content area is divided into several sections. On the left, there is a tree view of categories under 'Saúde', 'Educação', 'Religião', 'Serviços', and 'Transporte'. The 'Saúde' category is expanded, showing sub-items like 'Hospital', 'Farmácia', 'Igreja', and 'Catedral'. The 'Religião' category is also expanded, showing 'Igreja' and 'Catedral'. The 'Serviços' category is expanded, showing 'Banco', 'Correios', 'Administrativo', and 'Polícia'. The 'Transporte' category is expanded, showing 'Polícia'. The main content area is divided into several sections. At the top, there is a search bar with three dropdown menus: 'Grupo' (set to 'Todos'), 'Categoria', and 'Classe'. Below the search bar, there is a 'Criar Novo' form. The form has three input fields: 'Grupo' (containing 'Comércio'), 'Símbolo' (with an 'Upload' button and a 'Procurar...' button), and 'Descrição' (containing 'Inclui shoppings, mercado,...'). Below the input fields, there is a checkbox labeled 'Criar Categoria?' which is checked. At the bottom of the form, there are two buttons: 'Guardar' and 'Cancelar'.

Figura 5.8 - Criação de nova categorização

Para tal optou-se por apresentar no layout os campos de informação a serem preenchidos do primeiro nível de classificação (grupo); se o administrador pretender acrescentar uma categoria (segundo nível) a esse grupo ao clicar no botão de “Criar Categoria?” é estendido um painel onde se pode configurar essa mesma categoria (campos de informação iguais ao de grupo) e se o administrador pretender também associar a esta categoria uma classe (terceiro nível) deverá clicar no botão de “Criar Classe?” sendo estendido um novo painel onde se pode configurar a respetiva classe. Ao se guardar esta nova categorização a mesma será adicionada à estrutura de “árvore” de categorização.

- **Adicionar** – Esta operação demonstra uma grande utilidade na gestão da funcionalidade de “Categorização POI” porque permite para uma classificação de um determinado nível adicionar uma classificação de nível inferior, ou seja, se estiver

seleccionada uma categoria na “árvore” de categorização e clicar-se no botão “Adicionar” é possível acrescentar uma nova classe a essa categoria. Deste modo, esta operação oferece um maior dinamismo na classificação de categorias.

- **Editar** – Para uma determinada classificação de categoria seleccionada na “árvore” é possível editar a mesma.
- **Eliminar** – Para uma determinada classificação de categoria seleccionada na “árvore” é possível eliminar a mesma. Se por exemplo eliminar-se um grupo todas as suas categorias e classes também serão removidas.

5.2.6 Pontos de Interesse

Como já foi referido, esta funcionalidade permite criar e gerir pontos de interesse associando-os a categorias e entidades, ou seja, esta funcionalidade apesar de apresentar uma lista de todos os pontos de interesse de todas as entidades (se o utilizador tiver permissões para tal) a lógica de suporte torna independente essa gestão de pontos de interesse de uma determinada entidade. O layout principal de *Pontos de Interesse* apresenta uma secção de operações funcionais e outra onde são listados os pontos de interesse como ilustra a Figura 5.9.

GESTÃO DE PONTOS DE INTERESSE
Bem vindo **hsouto** ! [[Log Out](#)]

- Acesso
- Entidades
- Grupos
- Utilizadores
- Backoffice
- Categorização POI
- Pontos de Interesse

Lista de Pontos de Interesse

ID	Nome	Entidade	Categoria	Distrito	Concelho	Freguesia	Localidade	Eliminar
1	MacDonalds Roma	CMLisboa	Fast Food	Lisboa	Lisboa	São João de Deus	Lisboa	Eliminar
2	Estádio da Luz	CMLisboa	Estádio	Lisboa	Lisboa	São Domingos de Benfica	Lisboa	Eliminar
3	Castelo S. Jorge	CMLisboa	Castelo	Lisboa	Lisboa	Castelo	Lisboa	Eliminar
4	Estádio de Alvalade	CMLisboa	Estádio	Lisboa	Lisboa	Alvalade	Lisboa	Eliminar
5	Papo Cheio	CMPBarca	Restaurante	Viana do Castelo	Ponte da Barca	Ponte da Barca	Cruzeiro	Eliminar
6	Escola EB2 Diogo Bernardes	CMPBarca	Educação	Viana do Castelo	Ponte da Barca	Paço Vedro de Magalhães	Paço Vedro	Eliminar
7	Sé Catedral de Bragança	CMBragança	Catedral	Bragança	Bragança	Sé	Bragança	Eliminar

1 2 3

Figura 5.9 - Listagem de pontos de interesse

As operações funcionais representadas pelos botões no layout permitem o seguinte:

- **Pesquisar** – Pesquisa pontos de interesse de acordo com os valores dos parâmetros representados na secção correspondente e retorna os respetivos pontos de interesse

numa grelha. De se referir que para manter a disposição de controlos presentes no layout e coerência do espaço da secção de pesquisa na funcionalidade de “Pontos de Interesse” optou-se por disponibilizar as categorias hierarquicamente no controlo de lista (dropdownlist).

- **Limpar** – Limpa os valores de todos os controlos e lista na grelha de resultados todos os pontos de interesse.
- **Novo** – Permite criar um novo ponto de interesse. Como este item tem de certo modo uma caracterização ampla, optou-se por agrupar essas características por tipos e apresentá-las numa interface amigável e intuitiva através dos separadores “Informação”, “Localização” e “Tooltip”. A Figura 5.10 ilustra o layout das características do ponto de interesse presente no separador “Informação”.

The screenshot displays the 'GESTÃO DE PONTOS DE INTERESSE' web application. The top navigation bar includes the title and a user greeting 'Bem vindo hsouto ! [Log Out]'. A left sidebar contains menu items: 'Acesso', 'Entidades', 'Grupos', 'Utilizadores', 'Backoffice', 'Categorização POI', and 'Pontos de Interesse'. The main content area is divided into three tabs: 'Informação', 'Localização', and 'Tooltip'. The 'Informação' tab is active, showing a form with the following fields: 'Nome' (Papo Cheio), 'Entidade' (CMPBarca), 'Descrição' (Cozinha Regional), 'Telefone' (258741032), 'Fax' (258741078), 'Web' (http://www.papocheio.pt), and 'Email' (papocheio@hotmail.com). A 'Categoria' dropdown menu is open, showing a hierarchical list: 'Restauração' (expanded) and 'Restaurante' (selected), with sub-options: 'Chinês', 'Marisqueira', 'Fast Food', and 'Pizzaria'. 'Gravar' and 'Cancelar' buttons are at the bottom of the form. Below the form is a map of Portugal with a 'Mapa' dropdown menu.

Figura 5.10 - Criação de ponto de interesse (separador “Informação”)

Este separador contém a informação que caracteriza o ponto de interesse que se está a criar; de se destacar que é aqui se atribui a entidade à qual este ponto de interesse irá pertencer e a categoria do mesmo. Optou-se na operação “Novo” incluir um mapa com recurso à API

Google Maps de modo a auxiliar o processo de criação do ponto de interesse, nomeadamente a sua localização e visualização dos respetivos campos de informação na tooltip/speechbubble (balão informativo).

A Figura 5.11 ilustra o processo de localização do ponto de interesse presente no separador “Localização”. Este processo permite localizar um ponto de interesse de acordo com a hierarquia de localização (distrito, concelho, freguesia, localidade), como um local no mapa é representado por uma coordenada (latitude, longitude) é possível localizar esse ponto de interesse num diferente nível de hierarquia visto que o valor de cada um tem uma coordenada associada. Nesta solução à medida que se aumenta o nível de profundidade de localização o *zoom* no mapa é aumentado proporcionalmente e o local georreferenciado, tal como a atualização automática dos valores das caixas de texto de latitude e longitude.

De modo a tornar esta solução mais prática e intuitiva optou-se por disponibilizar ao administrador a possibilidade de georreferenciar o ponto através das metodologias de *geocoding* e *reverse geocoding*.

GESTÃO DE PONTOS DE INTERESSE Bem vindo **hsouto** ! [[Log Out](#)]

Localização

Distrito	Viana do Castelo	Rua	Rua Dr. Joaquim Moreira de Barros
Concelho	Ponte da Barca	Porta	54
Freguesia	Ponte da Barca	Latitude	41.8060963
Localidade	Cruzeiro	Longitude	-8.416387999999984

Gravar Cancelar

Satélite

Figura 5.11 - Criação de ponto de interesse (separador “Localização”)

Utilizando o *geocoding* é possível georreferenciar um ponto de interesse pelo modo “tradicional”, ou seja, através de um endereço originado pelo encadeamento de distrito,

concelho, freguesia e localidade obtém-se a coordenada correspondente e georreferencia nessa coordenada no mapa o respetivo ponto; ao utilizar-se o *reverse geocoding* é efetuado o processo inverso, ou seja, ao clicar num determinado local do mapa será originada a coordenada correspondente e com base numa estrutura de dados resultante dessa acção é possível extrair os dados correspondentes dos diferentes níveis de endereço.

Já a Figura 5.12 apresenta o separador “Tooltip”, que ilustra quais os campos de informação podem ser visíveis na tooltip desse ponto de interesse.

Em geral as aplicações SIG de pontos de interesse quando mostram informação na tooltip/speechbubble por regra são apresentados os mesmos campos para todos os pontos de interesse, o que nem sempre é o mais adequado e por vezes pode ocorrer a situação desses campos não terem valores associados, originando assim “balões vazios”. O propósito da criação desta componente funcional é em resolver as lacunas anteriormente identificadas na mostra de campos na tooltip, ou seja, validando se um determinado campo informativo do ponto de interesse tem associado um valor e se achar-se adequado seleccionar esse mesmo campo, em que será visível no balão informativo ao se clicar no ponto de interesse no mapa.



Figura 5.12 - Criação de ponto de interesse (separador “Tooltip”)

Após se criar o ponto de interesse, o mesmo vai ser adicionado à grelha do painel principal. Na grelha de resultados ao se clicar no ponto de interesse é possível efetuar a sua consulta e

visualizar a georreferenciação no mapa de acordo com os critérios escolhidos aquando a sua criação, conforme ilustra a Figura 5.13.

GESTÃO DE PONTOS DE INTERESSE
Bem vindo **hsouto** ! [[Log Out](#)]

Acesso
 Entidades
 Grupos
 Utilizadores
 Backoffice
 Categorização POI
Pontos de Interesse

Informação	Localização	Tooltip
Nome	<input type="text" value="Papo Cheio"/>	Entidade: <input type="text" value="CMPBarca"/>
Descrição	<input type="text" value="Cozinha Regional"/>	Categoria: <input type="text" value="Restaurante"/>
Telefone	<input type="text" value="258741032"/> Fax: <input type="text" value="258741078"/>	
Web	<input type="text" value="http://www.papocheio.pt"/>	
Email	<input type="text" value="papocheio@hotmail.com"/>	

Figura 5.13 - Consulta de ponto de interesse

O mapa gerado na Figura 5.13 foi com base no seguinte código parcial que se recorreu à API do Google Maps.

```
function mapaPOI(latitude, longitude, zoom, icon, infoWindowHtml) {
    var latlng = new google.maps.LatLng(latitude, longitude);
    var options = {
        zoom: zoom,
        center: latlng,
        mapTypeControl: true,
        mapTypeControlOptions:
        {
            style: google.maps.MapTypeControlStyle.DROPDOWN_MENU,
            mapTypeIds: [google.maps.MapTypeId.ROADMAP,
                google.maps.MapTypeId.TERRAIN,
                google.maps.MapTypeId.HYBRID,
                google.maps.MapTypeId.SATELLITE]
        },
        navigationControl: true,
        draggableCursor: 'hand'
    };
    map = new google.maps.Map($("#mapa"), options);
    placeMarkerPOI(latlng, icon, infoWindowHtml);
}
}
```

A título de exemplo, com referência a permissões de utilizadores, se o utilizador não pertencer ao grupo de “Administrador”, mas sim (por exemplo) ao grupo Convidado_<Entidade> para a funcionalidade “Pontos de Interesse” é apresentado o layout conforme ilustrado na Figura 5.14.

ID	Nome	Entidade	Categoria	Distrito	Concelho	Freguesia	Localidade
5	Papo Cheio	CMPBarca	Restaurante	Viana do Castelo	Ponte da Barca	Ponte da Barca	Cruzeiro
6	Escola EB2 Diogo Bernardes	CMPBarca	Educação	Viana do Castelo	Ponte da Barca	Paço Vedro de Magalhães	Paço Vedro
16	Banco Português Imobiliário	CMPBarca	Banco	Viana do Castelo	Ponte da Barca	Ponte da Barca	Bairro Municipal

Figura 5.14 - Pontos de Interesse (Permissões de grupo Convidado_<Entidade>)

De acordo com as permissões atribuídas aos utilizadores do tipo de grupo acima indicado, pode-se confirmar a veracidade das mesmas para a funcionalidade em questão, ou seja, verifica-se que o utilizador não tem acesso às funcionalidades administrativas (“Entidades”, “Grupos” e “Utilizadores”); para o painel principal verifica-se a inexistência do botão “Novo”, não tendo permissões de criar pontos de interesse; constata-se também a inexistência da coluna “Eliminar” da grelha de pontos de interesse, não tendo também permissão de eliminação de pontos de interesse. Também se confirma que os pontos de interesse listados pertencem exclusivamente à entidade da qual o utilizador tem permissões. Se clicar-se num determinado ponto de interesse da grelha o layout não apresentará o botão “Editar”, deste modo é validado que o utilizador tem somente permissões de leitura.

5.3 FrontOffice

Na implementação do FrontOffice seguiu-se a mesma linha de coerência que no BackOffice, ou seja, optou-se também por disponibilizar uma interface intuitiva e neste caso ainda mais apelativa. O layout contém um controlo do tipo *accordion* onde estão contidas as

funcionalidades do FrontOffice e um mapa na área principal da página onde serão efetuadas operações que envolvam os pontos de interesse, concretamente a sua georreferenciação e percurso entre dois pontos de interesse.

As funcionalidades presentes no FrontOffice estão organizadas em:

- Pesquisas
- Percursos
- Categorias

5.3.1 Pesquisas

Esta funcionalidade permite os utilizadores pesquisarem pontos de interesse com base num conjunto de critérios do painel de pesquisas conforme ilustra a Figura 5.15.

Um dos propósitos da implementação do FrontOffice é a divulgação dos pontos de interesse das diferentes entidades e permitir o utilizador interagir com aplicação de modo a auxiliá-lo na obtenção de informação de locais que lhe desperte o interesse e eventualmente tomar uma decisão, seja em relação à sua localização ou ao caminho entre dois pontos de interesse. Deste modo, nesta funcionalidade ao pesquisar pontos de interesse em função de um ou mais critérios esses pontos de interesse são listados numa grelha e georreferenciados no mapa com a simbologia de categorização adequada.

PORTAL MAPAS - PONTOS DE INTERESSE [Login]

PESQUISAS

Entidade
CMLisboa

Ponto de Interesse


Distrito
Selecione

Concelho
Selecione

Freguesia
Selecione

Localidade
Selecione

Pesquisar



ID	Nome	Entidade	Categoria	Distrito	Concelho	Freguesia	Localidade
1	MacDonalds Roma	CMLisboa	Fast Food	Lisboa	Lisboa	São João de Deus	Lisboa
2	Estádio da Luz	CMLisboa	Estádio	Lisboa	Lisboa	São Domingos de Benfica	Lisboa
3	Castelo S. Jorge	CMLisboa	Castelo	Lisboa	Lisboa	Castelo	Lisboa
4	Estádio de Alvalade	CMLisboa	Estádio	Lisboa	Lisboa	Alvalade	Lisboa
8	Estação de Metro das Olaias	CMLisboa	Metro	Lisboa	Lisboa	Alto do Pina	Lisboa
9	Estação de Metro de Alvalade	CMLisboa	Metro	Lisboa	Lisboa	Alvalade	Lisboa
10	Estação dos Comboios	CMLisboa	Comboio	Lisboa	Lisboa	Santa Maria dos Olivais	Lisboa

Figura 5.15 - Pesquisa de Pontos de Interesse (FrontOffice)

Dependendo do número de pontos de interesse e sua localização, com recurso à API Google Maps é possível enquadrá-los todos no mapa à escala que melhor se ajusta a esse enquadramento. O código seguinte mostra como esse processo é efetuado.

```
function putPOIsMap(data) {
    var bounds = new google.maps.LatLngBounds();
    var marker;
    var myOptions = {
        mapTypeId: google.maps.MapTypeId.ROADMAP,
        mapTypeControl: false
    };
    map = new google.maps.Map($("#mapaFO"), myOptions);

    $.each(data.d, function (i, item) {
        var pos = new google.maps.LatLng(item.Latitude, item.Longitude);
        var icon = "Simbolos/" + item.Simbologia;
        bounds.extend(pos);
        marker = new google.maps.Marker({
            position: pos,
            map: map,
            icon: icon
        });
    });
    map.fitBounds(bounds);
}
```

De modo esta funcionalidade ter um maior dinamismo decidiu-se que ao clicar num determinado ponto de interesse da grelha o mesmo seja centrado no mapa à escala adequada com a respetiva informação presente no tooltip/speechbubble, conforme ilustra a Figura 5.16.

ID	Nome	Entidade	Categoria	Distrito	Concelho	Freguesia	Localidade
1	MacDonalds Roma	CMLisboa	Fast Food	Lisboa	Lisboa	São João de Deus	Lisboa
2	Estádio da Luz	CMLisboa	Estádio	Lisboa	Lisboa	São Domingos de Benfca	Lisboa
3	Castelo S. Jorge	CMLisboa	Castelo	Lisboa	Lisboa	Castelo	Lisboa
4	Estádio de Alvalade	CMLisboa	Estádio	Lisboa	Lisboa	Alvalade	Lisboa
8	Estação de Metro das Olaias	CMLisboa	Metro	Lisboa	Lisboa	Alto do Pina	Lisboa
9	Estação de Metro de Alvalade	CMLisboa	Metro	Lisboa	Lisboa	Alvalade	Lisboa
10	Estação dos Combóios	CMLisboa	Combóio	Lisboa	Lisboa	Santa Maria dos Olivais	Lisboa

Figura 5.16 - POI centrado no mapa através de evento da grelha resultados

5.3.2 Percursos

Como já foi referido, a funcionalidade “*Percursos*” permite gerar um percurso entre dois pontos de interesse. Por norma as aplicações SIG que permitem gerar percursos fazem-o entre dois locais. Nesta solução tal também se verifica, mas como existem entidades, é possível gerar percursos entre dois pontos de interesse no domínio de uma única entidade ou entre entidades diferentes. Ao se clicar no botão “Gerar Percurso” do painel “Percursos” é gerado no mapa o percurso correspondente a ponto de interesse inicial de uma determinada entidade até um ponto de interesse final (que pode pertencer à mesma entidade ou não) e no painel aparece a sequência informativa de locais intermédios até ao ponto de interesse final; a Figura 5.17 ilustra esse percurso entre dois pontos de interesse de uma determinada entidade.

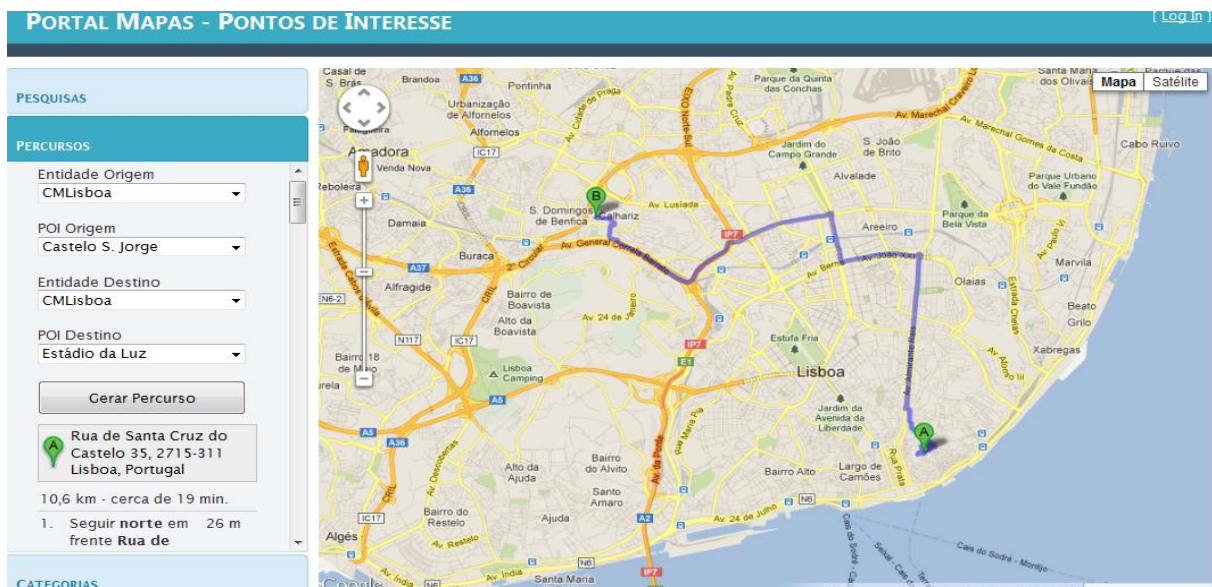


Figura 5.17 - Percurso entre dois pontos de interesse

Um exemplo de percurso entre pontos de interesse de diferentes entidades pode ser consultado no Anexo B.2. O código parcial seguinte mostra o processo de geração de percurso.

```
function geraPercursoMapaFO() {
    directionsDisplay = new google.maps.DirectionsRenderer();
    var mapOptions = {
        zoom: 14,
        mapTypeId: google.maps.MapTypeId.ROADMAP
    }
    map = new google.maps.Map($("#mapaFO"), mapOptions);
    directionsDisplay.setMap(map);
    directionsDisplay.setPanel($("#directionsPanel"));
    calcRoute();
}
```

5.3.3 Categorias

A funcionalidade “Categorias” foi implementada com o propósito de mostrar a categorização de pontos de interesse e a mesma servir como critério de pesquisa na funcionalidade homónima, ou seja, quando se pesquisam pontos de interesse os mesmos serão georreferenciados no mapa se nesta funcionalidade houver categorias selecionadas, conforme ilustra a Fig. 5.18.

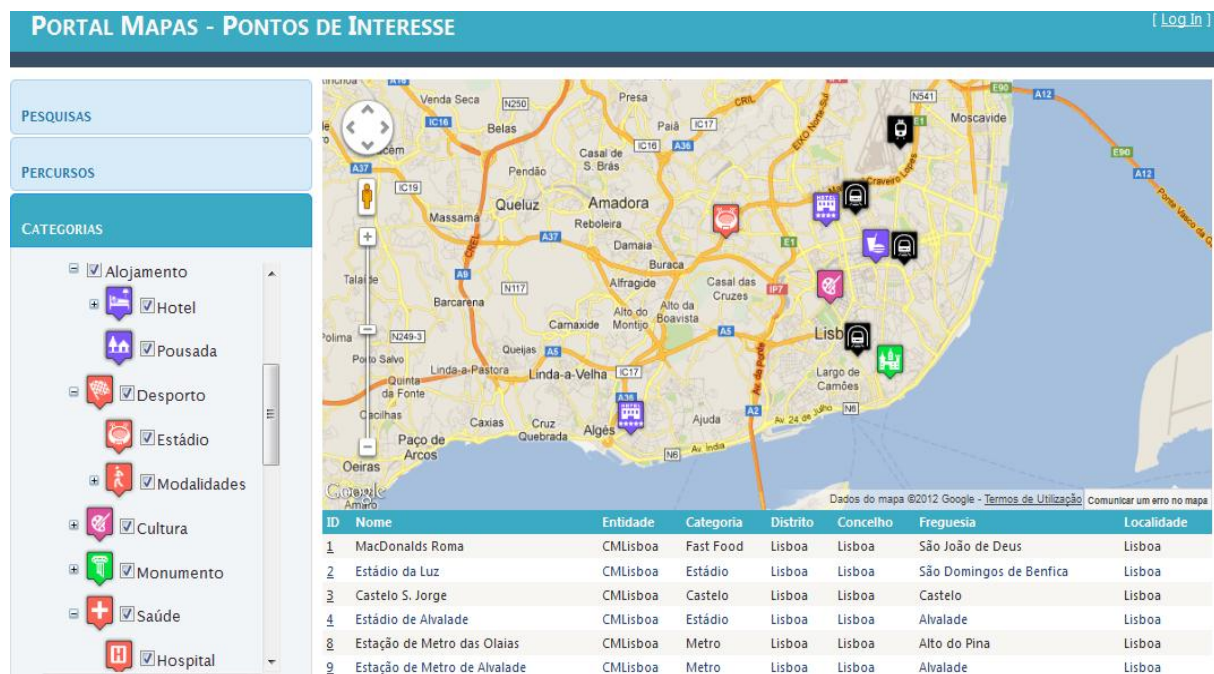


Figura 5.18 - Apresentação de categorização de pontos de interesse

Capítulo 6

6 Conclusões e trabalho futuro

Os Sistemas de Informação Geográficos (SIG) são sistemas que, tem recorrido a meios computacionais cada vez mais sofisticados, os quais, capturam, guardam, analisam, gerem e apresentam dados georreferenciados. De tal modo que, estes sistemas tem apresentado uma evolução assinalável estando cada vez mais presentes no nosso dia-a-dia e nas mais diversas áreas. Uma das áreas que tem revelado uma maior notoriedade e utilidade para a sociedade são os pontos de interesse, visto que são orientados ao auxílio de localização de determinados pontos geográficos e respetiva tomada de decisão por parte do cidadão.

Uma das aplicações de sistemas de informação geográfica que tem reforçado a sua importância e popularidade junto dos consumidores, produtores e difusores de informação é o Google Maps que através de constantes atualizações tem oferecido melhorias e novos recursos nas mais variadas áreas de negócio da vertente SIG, Também este serviço tem-se dedicado de um modo geral à otimização de representação de informação de pontos de interesse nos mapas e de um modo particular a funcionalidades extensíveis à localização dos respetivos pontos de interesse, nomeadamente percursos de diferentes modos entre dois pontos geográficos. A relevância e importância desta área SIG têm contribuído para uma procura superior em soluções de georreferenciação de pontos de interesse em comparação com outras áreas de sistemas de informação geográfica. Soluções estas que tem procurado sempre inovar e melhorar, nomeadamente através da adição de outros módulos de informação que visam representar informação nos mapas de acordo com determinados critérios, por exemplo, representação de pontos de interesse de acordo com a sua categoria

ou classificação e em função de uma determinada entidade (organizacional, municipal,...), como a solução deste projeto.

A abordagem arquitetural e tecnológica seguida neste projeto teve como pressupostos a modularidade, funcionalidade e reutilização, garantindo assim uma solução orientada a serviços através de camadas independentes e projetando soluções futuras ou extensões desta em que seja possível a reutilização de serviços e funcionalidades sem redundância de código. Nesta solução foram implementadas funcionalidades com o propósito de atender requisitos administrativos na área de gestão de pontos de interesse e divulgação dos mesmos.

6.1 Conclusões

Este projeto cumpriu os objetivos estipulados aquando a definição e arquitetura do mesmo, devido a solução responder aos requisitos delineados, apresentando um portal administrativo regido pela gestão de pontos de interesse com base em entidades e diferentes categorias, em que na criação dos pontos de interesse é possível georreferenciá-los por distrito, concelho, freguesia e localidade de acordo com a utilização dos métodos de georreferenciação de *geocoding* e *reverse geocoding*, os quais em conjunto, são utilizados com pouco frequência neste tipo de soluções. Também é apresentado um portal de mapas onde com sucesso é possível efetuar pesquisas de pontos de interesse de acordo com determinados filtros e geração de percursos sobre mapas, tal como a apresentação da categorização de pontos de interesse.

Pelo fato de esta solução ter sido implementada sobre uma arquitetura de camadas orientada a serviços, centralizando os mesmos, oferecendo um dinamismo e reutilização superior a outros tipos de solução aplicacional, pode-se concluir que a abordagem tecnológica e arquitetural neste projeto foi uma ótima solução.

A título pessoal, a implementação deste projeto serviu para ampliar e aprofundar os meus conhecimentos tecnológicos e aplicação dos mesmos na área SIG, reforçando também a motivação em continuar na procura de respostas e aperfeiçoamento de soluções nesta área de negócio.

6.2 Trabalho futuro

Após a definição de objetivos realistas na implementação deste projeto, com a ambição e entusiasmo inerente ao desenvolvimento de soluções de gestão de pontos de interesse e toda a panóplia de funcionalidades envolvente, tomou-se como objetivo elevar as expectativas e exigências, talvez irrealista, da adição de uma grande variedade de funcionalidades na implementação da solução que na realidade e por limitação de tempo não foi possível concretizá-las. A saber, de seguida são apresentadas algumas dessas funcionalidades e respetiva projeção.

- Estatísticas - Esta funcionalidade (BackOffice) deverá permitir através da interação do utilizador com o portal de mapas do FrontOffice a aquisição de informação e tratamento da mesma de modo a gerar resultados estatísticos com o propósito de auxiliar as entidades de gestão de pontos de interesse na tomada de decisão, como por exemplo, apresentação de pontos de interesse mais visualizados no portal por diferentes parâmetros (entidade, utilizador, categoria,...).
- Históricos – Esta funcionalidade deverá permitir gerar históricos de informação de pontos de interesse pelas funcionalidades de acesso ao sistema, ou seja, por entidades, grupos e utilizadores.
- Pesquisas avançadas de pontos de interesse – Esta funcionalidade no FrontOffice deverá permitir pesquisas de pontos de interesse sobre os mapas com base num domínio mais amplo de critérios e opções, como por exemplo efetuar pesquisas de acordo com os pontos de interesses mais visitados, ou de acordo com um determinado percurso.
- Geração de percursos através de POIs intermédios e armazenamento de percursos na base de dados – Apesar de esta funcionalidade, ter sido projetada e implementada parcialmente num protótipo, deverá permitir que o percurso entre dois pontos de interesse seja possível adicionar pontos intermédios e recalculá-lo e otimizar o percurso em questão e armazenar esses mesmos percursos na base de dados em que posteriormente sejam passíveis de consulta no FrontOffice.

- Aplicação Móvel – Esta aplicação deverá permitir em dispositivos móveis a pesquisa de pontos de interesse e adição de novos, geração de percurso no mapa e a possibilidade importação de dados geográficos de acordo com os diferentes tipos de ficheiro de importação/exportação de dados existentes na área SIG.

Referências bibliográficas

- [A* 12] Algoritmo A*, http://en.wikipedia.org/wiki/A-star_algorithm, 2012
- [Ajax 12] Ajax, [http://en.wikipedia.org/wiki/Ajax_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming)), 2012
- [ASP 12] Descrição de ASP.NET, http://www.tutorialspoint.com/asp.net/asp.net_introduction.htm, 2012
- [CAM 12] Arquit. camadas, <http://www.devmedia.com.br/arquitetura-em-camadas-com-c/12037>, 2012
- [CSHARP 12] Comparação C# e Java, http://www.mas.com.br/Artigos/CSharp_Java.htm, 2012
- [BMaps 12] Bing Maps, <http://www.bing.com/maps/>, 2012
- [Dijkstra 12] Algoritmo de Dijkstra, http://en.wikipedia.org/wiki/Dijkstra%27s_algorithm, 2012
- [EF 10] Entity Framework, <http://pontonetpt.org/blogs/jpaulino/archive/2010/12/19/introdu-231-227-o-ao-entity-framework.aspx>, 2010
- [FNET 12] Evolução Framework .NET, http://en.wikipedia.org/wiki/.NET_Framework_version_history#.NET_Framework_4.5, 2012
- [FNET4 12] Framework .NET 4.0, [http://msdn.microsoft.com/en-us/library/ms171868\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/ms171868(v=vs.100).aspx), 2012
- [Foley 97] Foley, J. D., van Dam, A., van Dam, S., Hughes, J., Phillips, R. *Introduction to Computer Graphics*. Addison-Wesley 2nd Ed., 1997.
- [GEarth 12] Google Earth, <http://www.google.com/intl/pt-PT/earth/index.html>, 2012
- [Gil 10] Gil, Filipe João Cerdeira. *Exploração de Técnicas para a Representação de Pontos de Interesse em Mapas*, Dissertação de Mestrado. Departamento de Informática, Faculdade de Ciências da Universidade de Lisboa, 2010.
- [GisStudio 12] GIS Studio, http://ajuda.sapo.pt/servicos.html?faq_id=34366&servico_id=3436, 2012
- [GMaps 12] Google Maps, <http://maps.google.com>, 2012
- [JS 12] Descrição de JavaScript, <http://www.howtcreate.co.uk/tutorials/javascript/introduction>, 2012
- [JSON 09] Descrição de JSON, <http://www.loiane.com/2009/06/breve-introducao-ao-json/>, 2009
- [jQuery 12] jQuery, <http://www.linhadecodigo.com.br/artigo/2068/introducao-a-jquery.aspx>, 2012
- [LINQ 12] LINQ, <http://www.linhadecodigo.com.br/artigo/1729/introducao-ao-linq.aspx>, 2012

- [LOG 12] Descrição do Log4net, <http://www.codeproject.com/Articles/8245/A-Brief-Introduction-to-the-log4net-logging-librar>, 2012
- [Matos 08] Matos, Paulo Miguel C.P. P. *Visualização de Informação Georreferenciada em Dispositivos Móveis*, Dissertação de Mestrado. Departamento de Informática, Faculdade de Ciências da Universidade de Lisboa, 2008.
- [Pinto 09] Pinto, Inês. *Introdução aos Sistemas de Informação Geográfica*, Instituto de Investigação Científica Tropical, 2009.
http://www.idcplp.net/archive/doc/georrefIntroducaoSIG_InesPinto.pdf
- [Sinfic 06] Sinfic, <http://www.sinfic.pt/SinficNewsletter/sinfic/Newsletter56/Dossier2.html>, 2006
- [SMapas 12] Sapo Mapas, <http://mapas.sapo.pt/>, 2012
- [SQLServ 12] SQL Server, http://en.wikipedia.org/wiki/Microsoft_SQL_Server, 2012
- [SQLServh 12] Hierarquia ID em SQL Server 2008,
<http://pratchev.blogspot.pt/2008/05/hierarchies-in-sql-server-2008.html>, 2012
- [Souto 07] Souto, José Henrique E. G. *D-WebEngine: Motor de construção de sítios Web*, Relatório de Licenciatura de Engenharia Informática, Escola Superior de Tecnologia e Gestão, Instituto Politécnico de Bragança, 2007.
- [WCF 12] Descrição de WCF, <http://www.lucianolima.com.br/category/WCF.aspx>, 2012
- [WikiBing 12] Definição de Bing Maps, http://en.wikipedia.org/wiki/Bing_Maps, 2012
- [WikiGMaps 12] Definição de Google Maps, http://en.wikipedia.org/wiki/Google_Maps, 2012
- [XML 12] Descrição de XML, http://www.gta.ufrj.br/grad/00_1/miguel/link1.htm, 2012
- [YMaps 12] Yahoo Maps, <http://maps.yahoo.com/>, 2012

Referências de Figuras

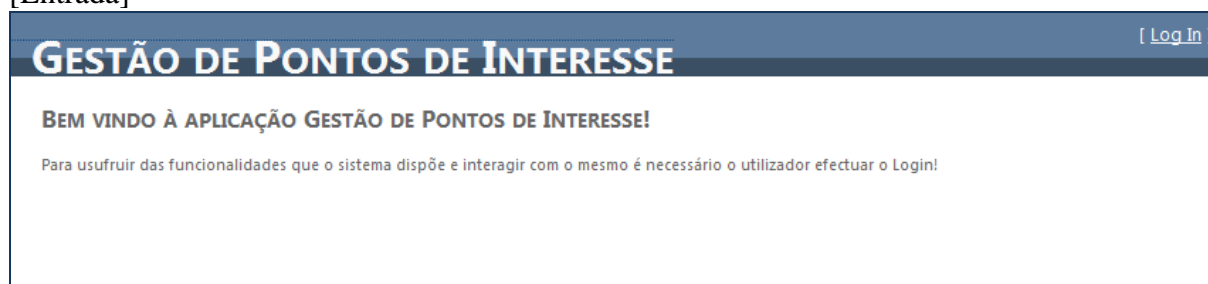
- [Ascendi] http://www.oern.pt/documentos/Eliana_Carreira_-_Ascendi.pdf
- [ArqCamada] <http://devbrasil.net/profiles/blogs/aplicacoes-em-n-camadas-com-1>
- [ArqEF] <http://www.codeproject.com/Articles/363040/An-Introduction-to-Entity-Framework-for-Absolute-B>
- [ArqLINQ] <http://msdn.microsoft.com/msdnmag/issues/07/06/CSharp30/pt/fig01.gif>
- [CodeASP] <http://www.sitepoint.com/c-asp-net-developers/>
- [EndWCF] <http://blogs.msdn.com/b/paolos/archive/2009/11/17/customizing-and-extending-the-biztalk-wcf-adapters.aspx>
- [Lisboa] <http://lxi.cm-lisboa.pt/lxi/>
- [Mapas] <http://geshout.com/mapscompare/all.php>
- [ModAJAX] <http://www.lukew.com/ff/entry.asp?180>

Anexo A

A – BackOffice (Layouts)

A.1. Entrada, login e registo de utilizador

[Entrada]



GESTÃO DE PONTOS DE INTERESSE [Log In]

BEM VINDO À APLICAÇÃO GESTÃO DE PONTOS DE INTERESSE!

Para usufruir das funcionalidades que o sistema dispõe e interagir com o mesmo é necessário o utilizador efectuar o Login!

[Login]

Registe-se se não tem uma conta.' Below this is a form titled 'Informação da conta' with fields for 'Nome:' (containing 'hsouto'), 'Password:' (masked with dots), and a checkbox for 'Manter-me ligado.' which is checked. A 'Log In' button is located below the form." data-bbox="114 657 874 900"/>

GESTÃO DE PONTOS DE INTERESSE [Log In]

LOG IN

Por favor introduza o seu nome e password. [Registe-se](#) se não tem uma conta.

Informação da conta

Nome:

Password:

Manter-me ligado.

[Registo]

GESTÃO DE PONTOS DE INTERESSE [\[Log In \]](#)

CRIAR CONTA

Use o formulário abaixo para criar uma nova conta.
As passwords requerem um mínimo de 6 caracteres.

Informação da Conta

Utilizador:

E-mail:

Password:

Confirme a Password:

A.2. Entidades

[Criação de entidade]

GESTÃO DE PONTOS DE INTERESSE Bem vindo **hsouto** ! [\[Log Out \]](#)

Entidades

Nome: Morada:

Telefone: Fax: Email:

Web:

[Lista de entidades]

GESTÃO DE PONTOS DE INTERESSE Bem vindo **hsouto** ! [\[Log Out \]](#)

Lista de Entidades

Nome: Morada:

ID	Nome	Morada	Eliminar
1	CMLisboa	Paços do Concelho - Praça do Município, Lisboa	Eliminar
2	CMPBarca	Rua Cons ^o Rocha Peixoto, Ponte da Barca	Eliminar
3	CMBragança	Forte S. João de Deus 5300-263, Bragança	Eliminar
4	ProTurismoLda	Av. Liberdade, Lisboa	Eliminar
6	EventosParty	Av. Aliados, Porto	Eliminar

A.3. Grupos

[Criação de grupos]

Bem vindo **hsouto** ! [[Log Out](#)]

GESTÃO DE PONTOS DE INTERESSE

- Acesso
- Entidades
- Grupos**
- Utilizadores
- Backoffice
- Categorização POI
- Pontos de Interesse

Grupos

Perfil: Administrador Convidado

Entidade:

Grupo	Eliminar
Convidado	Eliminar
Administrador_CMPBarca	Eliminar
Convidado_CMLisboa	Eliminar
Convidado_CMBragança	Eliminar
Convidado_RadicalFree	Eliminar
Administrador_ProTurismoLda	Eliminar
Convidado_EventosParty	Eliminar
Convidado_CMPBarca	Eliminar
Administrador_CMLisboa	Eliminar
Administrador_CMBragança	Eliminar

[Lista de grupos]

Bem vindo **hsouto** ! [[Log Out](#)]

GESTÃO DE PONTOS DE INTERESSE

- Acesso
- Entidades
- Grupos**
- Utilizadores
- Backoffice
- Categorização POI
- Pontos de Interesse

Grupo:

Lista de Grupos

Grupo	Número de Utilizadores	Eliminar
Administrador	2	Eliminar
Administrador_Bragança	1	Eliminar
Administrador_CMLisboa	2	Eliminar
Administrador_CMPBarca	2	Eliminar
Administrador_EventosParty	1	Eliminar
Administrador_ProTurismoLda	1	Eliminar
Administrador_RadicalFree	2	Eliminar
Convidado	1	Eliminar
Convidado_Bragança	2	Eliminar
Convidado_CMLisboa	3	Eliminar

1 2

[Pesquisa de grupos]

GESTÃO DE PONTOS DE INTERESSE Bem vindo **hsouto** ! [[Log Out](#)]

Acesso
Entidades
Grupos
Utilizadores
Backoffice
Categorização POI
Pontos de Interesse

Grupo
Administrador
Pesquisar Limpar Novo

Lista de Grupos

Grupo	Número de Utilizadores	Eliminar
Administrador	2	Eliminar
Administrador_Bragança	1	Eliminar
Administrador_CMLisboa	2	Eliminar
Administrador_CMPBarca	2	Eliminar
Administrador_EventosParty	1	Eliminar
Administrador_ProTurismoLda	1	Eliminar
Administrador_RadicalFree	2	Eliminar

A.4. Utilizadores

[Novo utilizador]

GESTÃO DE PONTOS DE INTERESSE Bem vindo **hsouto** ! [[Log Out](#)]

Acesso
Entidades
Grupos
Utilizadores
Backoffice
Categorização POI
Pontos de Interesse

Novo Utilizador

Utilizador: IPBraganca Email: geral@ipb.pt
Password (min. de 6 caracteres): Acesso: Permitido?
Grupo(s): Administrador Administrador_Bragar Administrador_CMLisb Administrador_CMPBa

Gravar Cancelar

[Detalhe de utilizador]

GESTÃO DE PONTOS DE INTERESSE Bem vindo **hsouto** ! [[Log Out](#)]

Acesso
Entidades
Grupos
Utilizadores
Backoffice
Categorização POI
Pontos de Interesse

Detalhe Utilizador

Utilizador: henriquesouto Email: henriquesouto@gmail.com
Acesso: Permitido?
Grupo(s): Convidado Convidado_Bragança Convidado_CMLisboa Convidado_CMPBarca

Editar Voltar

[Edição de utilizador]

Bem vindo **hsouto** ! [[Log Out](#)]

GESTÃO DE PONTOS DE INTERESSE

Acesso

Entidades

Grupos

Utilizadores

Backoffice

Categorização POI

Pontos de Interesse

Edição Utilizador

Utilizador

Email

Grupo(s)

Convidado
 Convidado_Bragança
 Convidado_CMLisboa
 Convidado_CMPBarca

Password (min. de 6 caracteres)

Acesso

 Permitido?

[Lista de utilizadores]

Bem vindo **hsouto** ! [[Log Out](#)]

GESTÃO DE PONTOS DE INTERESSE

Acesso

Entidades

Grupos

Utilizadores

Backoffice

Categorização POI

Pontos de Interesse

Utilizador

Email

Lista de Utilizadores

Utilizador	Email	Grupos	Última Entrada	Acesso	Eliminar
administrador	administrador@gmail.com	Administrador	22-11-2012 04:35:40	<input checked="" type="checkbox"/>	Eliminar
anaalves	anaalves@gmail.com	Convidado	19-11-2012 04:18:19	<input checked="" type="checkbox"/>	Eliminar
anacristina	anacristina@gmail.com	Convidado_CMPBarca	17-11-2012 04:24:45	<input checked="" type="checkbox"/>	Eliminar
anqela	angela@gmail.com	Convidado_CMPBarca	17-11-2012 04:24:06	<input checked="" type="checkbox"/>	Eliminar
henriquesouto	henriquesouto@gmail.com	Convidado_Bragança; Convidado_CMLisboa	19-11-2012 04:31:01	<input checked="" type="checkbox"/>	Eliminar
hsouto	hsouto@gmail.com	Administrador	22-11-2012 04:48:47	<input checked="" type="checkbox"/>	Eliminar
josesouto	josesouto@gmail.com	Administrador_CMLisboa; Administrador_CMPBarca	17-11-2012 04:20:30	<input checked="" type="checkbox"/>	Eliminar
mariafatima	mariafatima@gmail.com	Administrador_CMLisboa; Administrador_CMPBarca	17-11-2012 04:21:52	<input checked="" type="checkbox"/>	Eliminar

1 2

A.5. Categorização POI

[Nova categorização]

Bem vindo **hsouto** ! [[Log Out](#)]

GESTÃO DE PONTOS DE INTERESSE

- Acesso
- Entidades
- Grupos
- Utilizadores
- Backoffice
- Categorização POI**
- Pontos de Interesse

Grupo: Todos | Categoria: | Classe: |

Criar Novo

Grupo: | Símbolo:

Descrição: |

Criar Categoria?

[Lista de categorias e detalhe da seleccionada]


Bem vindo **hsouto** ! [[Log Out](#)]

GESTÃO DE PONTOS DE INTERESSE

- Acesso
- Entidades
- Grupos
- Utilizadores
- Backoffice
- Categorização POI**
- Pontos de Interesse

Grupo: Todos | Categoria: | Classe: |

Detalhe

Grupo: | 

Descrição:

[Pesquisa de categorização]

Bem vindo **hsouto** ! [Log Out]

GESTÃO DE PONTOS DE INTERESSE


- Acesso
- Entidades
- Grupos
- Utilizadores
- Backoffice
- Categorização POI
- Pontos de Interesse

Grupo: Restauração | Categoria: Todos | Classe: | Pesquisar | Limpar

Detalhe

Grupo: Restauração

Descrição: Restauração



Criar Novo | Adicionar | Editar | Eliminar

A.5. Pontos de Interesse

[Lista de pontos de interesse]

Bem vindo **hsouto** ! [Log Out]

GESTÃO DE PONTOS DE INTERESSE

- Acesso
- Entidades
- Grupos
- Utilizadores
- Backoffice
- Categorização POI
- Pontos de Interesse

Nome: | Entidade: Seleccione | Categoria: Seleccione

Distrito: Seleccione | Concelho: | Freguesia: | Localidade: |

Pesquisar | Limpar | Novo

Lista de Pontos de Interesse

ID	Nome	Entidade	Categoria	Distrito	Concelho	Freguesia	Localidade	Eliminar
1	MacDonalds Roma	CMLisboa	Fast Food	Lisboa	Lisboa	São João de Deus	Lisboa	Eliminar
2	Estádio da Luz	CMLisboa	Estádio	Lisboa	Lisboa	São Domingos de Benfica	Lisboa	Eliminar
3	Castelo S. Jorge	CMLisboa	Castelo	Lisboa	Lisboa	Castelo	Lisboa	Eliminar
4	Estádio de Alvalade	CMLisboa	Estádio	Lisboa	Lisboa	Alvalade	Lisboa	Eliminar
5	Papo Cheio	CMPBarca	Restaurante	Viana do Castelo	Ponte da Barca	Ponte da Barca	Cruzeiro	Eliminar
6	Escola EB2 Diogo Bernardes	CMPBarca	Educação	Viana do Castelo	Ponte da Barca	Paço Vedro de Magalhães	Paço Vedro	Eliminar
7	Sé Catedral de Bragança	CMBragança	Catedral	Bragança	Bragança	Sé	Bragança	Eliminar

1 2 3

[Criação de ponto de interesse – “Informação”]

Bem vindo **hsouto** ! [[Log Out](#)]

GESTÃO DE PONTOS DE INTERESSE

Acesso

Entidades

Grupos

Utilizadores

Backoffice

Categorização POI

Pontos de Interesse

Informação	Localização	Tooltip
Nome: <input type="text" value="Papo Cheio"/>	Entidade: <input type="text" value="CMPBarca"/>	
Descrição: <input type="text" value="Cozinha Regional"/>		
Telefone: <input type="text" value="258741032"/> Fax: <input type="text" value="258741078"/>		
Web: <input type="text" value="http://www.papocheio.pt"/>		
Email: <input type="text" value="papocheio@hotmail.com"/>		
<input type="button" value="Gravar"/> <input type="button" value="Cancelar"/>		<div style="border: 1px solid #ccc; padding: 5px;"> <p>Categoria</p> <ul style="list-style-type: none"> Restauração <ul style="list-style-type: none"> <li style="background-color: #e0e0e0;">Restaurante Chinês Marisqueira Fast Food Pizzaria </div>



Mapa

[Criação de ponto de interesse – “Localização”]

Bem vindo **hsouto** ! [[Log Out](#)]

GESTÃO DE PONTOS DE INTERESSE

Acesso

Entidades

Grupos

Utilizadores

Backoffice

Categorização POI

Pontos de Interesse

Informação	Localização	Tooltip
Distrito: <input type="text" value="Viana do Castelo"/>	Rua: <input type="text" value="Rua Dr. Joaquim Moreira de Barros"/>	
Concelho: <input type="text" value="Ponte da Barca"/>	Porta: <input type="text" value="54"/>	
Freguesia: <input type="text" value="Ponte da Barca"/>	Latitude: <input type="text" value="41.8060963"/>	
Localidade: <input type="text" value="Cruzeiro"/>	Longitude: <input type="text" value="-8.416387999999984"/>	
<input type="button" value="Gravar"/> <input type="button" value="Cancelar"/>		

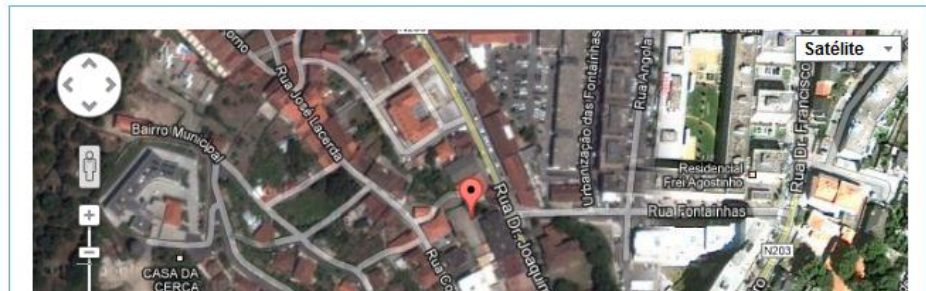


Satélite

[Criação de ponto de interesse – “Tooltip”]

- Acesso
- Entidades
- Grupos
- Utilizadores
- Backoffice
- Categorização POI
- Pontos de Interesse**

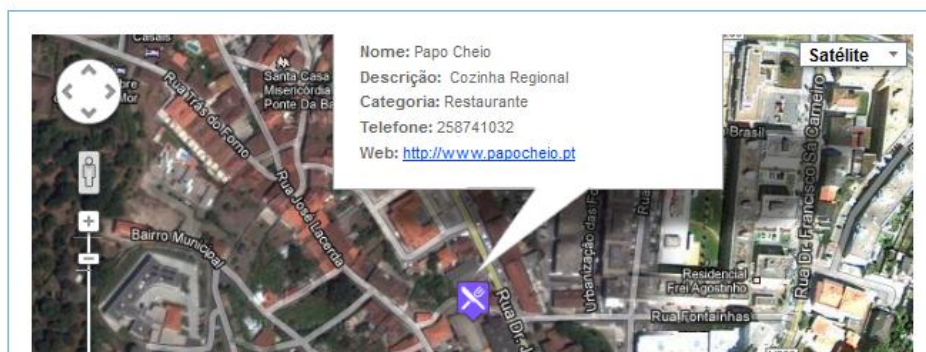
Informação	Localização	Tooltip
Campos de Ponto de Interesse a mostrar na tooltip:		
<input checked="" type="checkbox"/> Nome	<input checked="" type="checkbox"/> Descrição	<input checked="" type="checkbox"/> Categoria
<input checked="" type="checkbox"/> Web	<input type="checkbox"/> Email	<input checked="" type="checkbox"/> Telefone <input type="checkbox"/> Fax
<input type="button" value="Gravar"/> <input type="button" value="Cancelar"/>		



[Detalhe de ponto de interesse]

- Acesso
- Entidades
- Grupos
- Utilizadores
- Backoffice
- Categorização POI
- Pontos de Interesse**

Informação	Localização	Tooltip
Nome	<input type="text" value="Papo Cheio"/>	Entidade
Descrição	<input type="text" value="Cozinha Regional"/>	Categoria
Telefone	<input type="text" value="258741032"/>	Fax
	<input type="text" value="258741078"/>	
Web	<input type="text" value="http://www.papocheio.pt"/>	
Email	<input type="text" value="papocheio@hotmail.com"/>	
<input type="button" value="Editar"/> <input type="button" value="Voltar"/>		



[Pesquisa de pontos de interesse]

Bem vindo **hsouto** ! [[Log Out](#)]

GESTÃO DE PONTOS DE INTERESSE

- Acesso
- Entidades
- Grupos
- Utilizadores
- Backoffice
- Categorização POI
- Pontos de Interesse**

Nome <input type="text"/>	Entidade CMLisboa ▼	Categoria -----Metro ▼	
Distrito Seleccione ▼	Concelho <input type="text"/>	Freguesia <input type="text"/>	Localidade <input type="text"/>
<input type="button" value="Pesquisar"/> <input type="button" value="Limpar"/> <input type="button" value="Novo"/>			

Lista de Pontos de Interesse

ID	Nome	Entidade	Categoria	Distrito	Concelho	Freguesia	Localidade	Eliminar
8	Estação de Metro das Olaias	CMLisboa	Metro	Lisboa	Lisboa	Alto do Pina	Lisboa	Eliminar
9	Estação de Metro de Alvalade	CMLisboa	Metro	Lisboa	Lisboa	Alvalade	Lisboa	Eliminar
17	Estação de Metro da Avenida	CMLisboa	Metro	Lisboa	Lisboa	São José	Lisboa	Eliminar

Anexo B

B – FrontOffice (Layouts)

B.1. Pesquisas

[Pesquisa de pontos de interesse]

PORTAL MAPAS - PONTOS DE INTERESSE [Log In]

PESQUISAS

Entidade
CMLisboa

Ponto de Interesse

Distrito
Selecione

Concelho
Selecione

Freguesia
Selecione

Localidade
Selecione



Dados do mapa ©2012 Google - [Termos de Utilização](#) [Comunicar um erro no mapa](#)

ID	Nome	Entidade	Categoria	Distrito	Concelho	Freguesia	Localidade
1	MacDonalds Roma	CMLisboa	Fast Food	Lisboa	Lisboa	São João de Deus	Lisboa
2	Estádio da Luz	CMLisboa	Estádio	Lisboa	Lisboa	São Domingos de Benfca	Lisboa
3	Castelo S. Jorge	CMLisboa	Castelo	Lisboa	Lisboa	Castelo	Lisboa
4	Estádio de Alvalade	CMLisboa	Estádio	Lisboa	Lisboa	Alvalade	Lisboa
8	Estação de Metro das Orlaias	CMLisboa	Metro	Lisboa	Lisboa	Alto do Pina	Lisboa
9	Estação de Metro de Alvalade	CMLisboa	Metro	Lisboa	Lisboa	Alvalade	Lisboa
10	Estação dos Comboios	CMLisboa	Comboio	Lisboa	Lisboa	Santa Maria dos Olivais	Lisboa

PERCURSOS

CATEGORIAS

[Centragem no mapa do pontos de interesse selecionado]

PORTAL MAPAS - PONTOS DE INTERESSE [Log In]

PESQUISAS

Entidade
CMLisboa

Ponto de Interesse

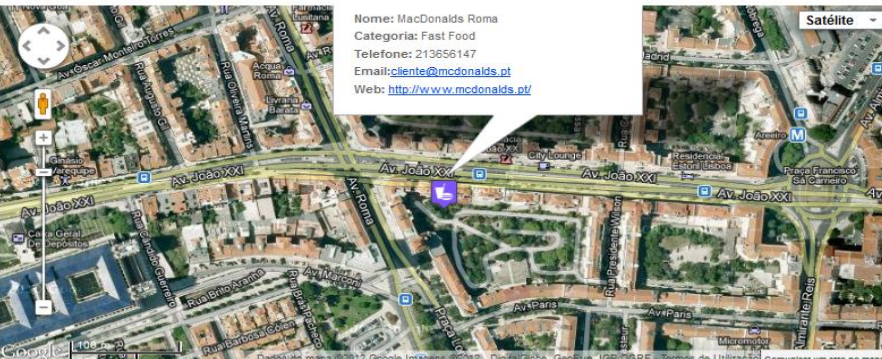
Distrito
Selecione

Concelho
Selecione

Freguesia
Selecione

Localidade
Selecione

Pesquisar



ID	Nome	Entidade	Categoria	Distrito	Concelho	Freguesia	Localidade
1	MacDonalds Roma	CMLisboa	Fast Food	Lisboa	Lisboa	São João de Deus	Lisboa
2	Estádio da Luz	CMLisboa	Estádio	Lisboa	Lisboa	São Domingos de Benfca	Lisboa
3	Castelo S. Jorge	CMLisboa	Castelo	Lisboa	Lisboa	Castelo	Lisboa
4	Estádio de Alvalade	CMLisboa	Estádio	Lisboa	Lisboa	Alvalade	Lisboa
8	Estação de Metro das Olaias	CMLisboa	Metro	Lisboa	Lisboa	Alto do Pina	Lisboa
9	Estação de Metro de Alvalade	CMLisboa	Metro	Lisboa	Lisboa	Alvalade	Lisboa
10	Estação dos Comboios	CMLisboa	Comboio	Lisboa	Lisboa	Santa Maria dos Olivais	Lisboa

B.2. Percursos

[Percurso entre POIs da mesma entidade]

PORTAL MAPAS - PONTOS DE INTERESSE [Log In]

PESQUISAS

PERCURSOS

Entidade Origem
CMLisboa

POI Origem
Castelo S. Jorge

Entidade Destino
CMLisboa

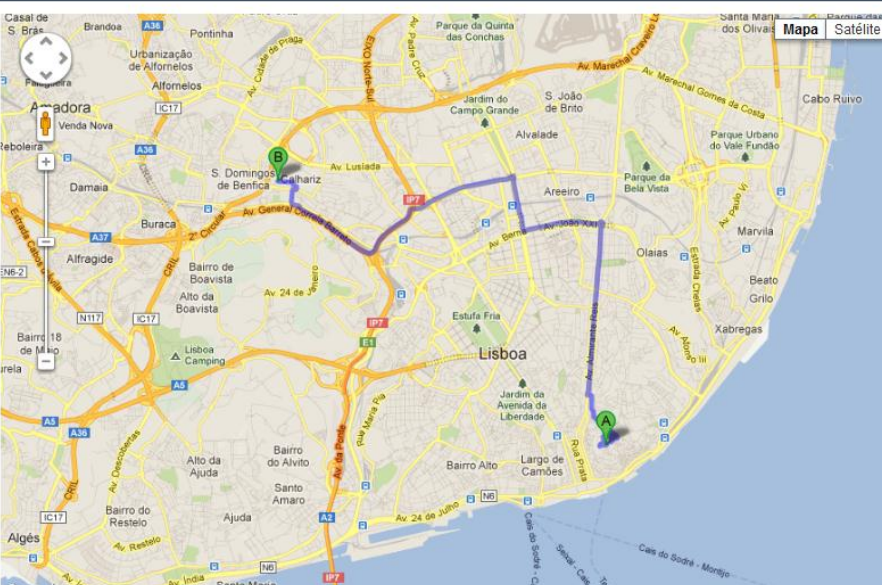
POI Destino
Estádio da Luz

Gerar Percurso

Rua de Santa Cruz do Castelo 35, 2715-311 Lisboa, Portugal

10,6 km - cerca de 19 min.

1. Seguir norte em 26 m frente Rua de



[Percurso entre POIs de diferentes entidades]

PORTAL MAPAS - PONTOS DE INTERESSE [\[Log In \]](#)

PESQUISAS

PERCURSOS

Entidade Origem
CMPBarca

POI Origem
Papo Cheio

Entidade Destino
CMBragança

POI Destino
Sé Catedral de Bragança

Gerar Percurso

Rua Doutor Joaquim Moreira de Barros 24, 4980-634 Ponte da Barca, Portugal

257 km - cerca de 3 horas 15 min.

1. Seguir **sudeste** 0.2 km

CATEGORIAS

B.2. Categorias

[Categorias de pontos de interesse]

PORTAL MAPAS - PONTOS DE INTERESSE [\[Log In \]](#)

PESQUISAS

PERCURSOS

CATEGORIAS

- Alojamento
- Hotel
- Pousada
- Desporto
- Estádio
- Modalidades
- Cultura
- Monumento
- Saúde
- Hospital

ID	Nome	Entidade	Categoria	Distrito	Concelho	Freguesia	Localidade
1	MacDonalds Roma	CMLisboa	Fast Food	Lisboa	Lisboa	São João de Deus	Lisboa
2	Estádio da Luz	CMLisboa	Estádio	Lisboa	Lisboa	São Domingos de Benfica	Lisboa
3	Castelo S. Jorge	CMLisboa	Castelo	Lisboa	Lisboa	Castelo	Lisboa
4	Estádio de Alvalade	CMLisboa	Estádio	Lisboa	Lisboa	Alvalade	Lisboa
8	Estação de Metro das Olaias	CMLisboa	Metro	Lisboa	Lisboa	Alto do Pina	Lisboa
9	Estação de Metro de Alvalade	CMLisboa	Metro	Lisboa	Lisboa	Alvalade	Lisboa

Anexo C

C – Implementação por camadas (Entidade)

C.1. Camada DAL

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using log4net;
using System.Data.Linq.SqlClient;
using PredicateBuilderUtils;
using System.Data.Objects;

namespace DAL.Entidades
{
    public class EntidadeDAL
    {
        private static readonly ILog log =
log4net.LogManager.GetLogger(typeof(EntidadeDAL));

        #region CRUD Entidades
        public void CreateEntidade(Entidade ent)
        {
            try
            {
                using (SIGDataContext dc = new SIGDataContext())
                {
                    dc.Entidade.AddObject(ent);
                    dc.SaveChanges();
                }
            }
            catch (Exception ex)
            {
                log.Error("[DAL - CreateEntidade]: ", ex);
            }
        }
    }
}
```

```

public List<Entidade> ReadEntidade(int Id)
{
    try
    {
        using (SIGDataContext dc = new SIGDataContext())
        {
            var items = from e in dc.Entidade
                        where e.Id.Equals(Id)
                        orderby e.Nome
                        select e;
            return items.Of<Entidade>().ToList();
        }
    }
    catch (Exception ex)
    {
        log.Error("[DAL - ReadEntidade]: ", ex);
        return null;
    }
}

public void UpdateEntidade(Entidade ent)
{
    try
    {
        using (SIGDataContext dc = new SIGDataContext())
        {
            Entidade entidade = dc.Entidade.First(e => e.Id ==
ent.Id);

            entidade.Nome = ent.Nome;
            entidade.Morada = ent.Morada;
            entidade.Telefone = ent.Telefone;
            entidade.Fax = ent.Fax;
            entidade.Email = ent.Email;
            entidade.Web = ent.Web;
            dc.SaveChanges();
        }
    }
    catch (Exception ex)
    {
        log.Error("[DAL - UpdateEntidade]: ", ex);
    }
}

public void DeleteEntidade(int idEntidade)
{
    try
    {
        using (SIGDataContext dc = new SIGDataContext())
        {
            Entidade ent = dc.Entidade.First(e => e.Id ==
idEntidade);

            dc.Entidade.DeleteObject(ent);
            dc.SaveChanges();
        }
    }
    catch (Exception ex)
    {
        log.Error("[DAL - DeleteEntidade]: ", ex);
    }
}

```

```

public List<Entidade> ListEntidades()
{
    try
    {
        using (SIGDataContext dc = new SIGDataContext())
        {
            var entidade = from e in dc.Entidade
                            orderby e.Id
                            select e;
            return entidade.OfType<Entidade>().ToList();
        }
    }
    catch (Exception ex)
    {
        log.Error("[DAL - ListEntidades]: ", ex);
        return null;
    }
}

public List<Entidade> ListEntidadesByFiltro(string nomeEntidade,
string morada)
{
    #region Variables
    IQueryable<Entidade> filterQuery = null;
    #endregion

    try
    {
        using (SIGDataContext dc = new SIGDataContext())
        {
            filterQuery = GetFilterQueryEntidade(dc, nomeEntidade,
morada);

            var entidade = from e in filterQuery
                            orderby e.Nome
                            select e;
            return entidade.OfType<Entidade>().ToList();
        }
    }
    catch (Exception ex)
    {
        log.Error("[DAL - ListEntidadesByFiltro]: ", ex);
        return null;
    }
}

internal static IQueryable<Entidade>
GetFilterQueryEntidade(SIGDataContext dc, string nomeEntidade, string
morada)
{
    #region Variables
    IQueryable<Entidade> filterQuery = dc.Entidade;
    #endregion

    try
    {
        #region Where Conditions
        if (!string.IsNullOrEmpty(nomeEntidade))
            filterQuery = filterQuery.Where(e =>
e.Nome.Contains(nomeEntidade));
        #endregion
    }
}

```

```

        if (!string.IsNullOrEmpty(morada))
            filterQuery = filterQuery.Where(e =>
e.Morada.Contains(morada));
        #endregion
    }
    catch (Exception ex)
    {
        log.Error("Exception in GetFilterQueryEntidade", ex);
    }
    return filterQuery;
}
#endregion
}
}

```

C.2. Camada BLL Service Layer

C.2.1 Interface

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.Text;

namespace BLLServiceLayer.Entidade
{
    [ServiceContract]
    public interface IEntidade
    {
        [OperationContract]
        void CreateEntidade(DAL.Entidade ent);

        [OperationContract]
        List<DAL.Entidade> ReadEntidade(int Id);

        [OperationContract]
        void UpdateEntidade(DAL.Entidade ent);

        [OperationContract]
        void DeleteEntidade(int idEntidade);

        [OperationContract]
        List<DAL.Entidade> ListEntidades();

        [OperationContract]
        List<DAL.Entidade> ListEntidadesByFiltro(string nomeEntidade,
string morada);
    }
}

```

C.2.2 Serviço

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.Text;
using log4net;
using DAL.Entidades;

namespace BLLServiceLayer.Entidade
{
    public class Entidade : IEntidade
    {
        private static readonly ILog log =
log4net.LogManager.GetLogger(typeof(Entidade));

        #region CRUD
        public void CreateEntidade(DAL.Entidade ent)
        {
            EntidadeDAL entidade = new EntidadeDAL();
            try
            {
                entidade.CreateEntidade(ent);
            }
            catch (Exception ex)
            {
                log.Error("[BLL - CreateEntidade]: ", ex);
            }
        }

        public List<DAL.Entidade> ReadEntidade(int Id)
        {
            EntidadeDAL entidade = new EntidadeDAL();
            try
            {
                return entidade.ReadEntidade(Id);
            }
            catch (Exception ex)
            {
                log.Error("[BLL - ReadEntidade]: ", ex);
                return null;
            }
        }

        public void UpdateEntidade(DAL.Entidade ent)
        {
            EntidadeDAL entidade = new EntidadeDAL();
            try
            {
                entidade.UpdateEntidade(ent);
            }
            catch (Exception ex)
            {
                log.Error("[BLL - UpdateEntidade]: ", ex);
            }
        }
    }
}
```

```

    }
}

public void DeleteEntidade(int idEntidade)
{
    EntidadeDAL entidade = new EntidadeDAL();
    try
    {
        entidade.DeleteEntidade(idEntidade);
    }
    catch (Exception ex)
    {
        log.Error("[BLL - DeleteEntidade]: ", ex);
    }
}

public List<DAL.Entidade> ListEntidades()
{
    EntidadeDAL entidade = new EntidadeDAL();
    try
    {
        return entidade.ListEntidades();
    }
    catch (Exception ex)
    {
        log.Error("[BLL - ListEntidades]: ", ex);
        return null;
    }
}

public List<DAL.Entidade> ListEntidadesByFiltro(string
nomeEntidade, string morada)
{
    EntidadeDAL entidade = new EntidadeDAL();
    try
    {
        return entidade.ListEntidadesByFiltro(nomeEntidade,morada);
    }
    catch (Exception ex)
    {
        log.Error("[BLL - ListEntidadesByFiltro]: ", ex);
        return null;
    }
}

#endregion
}
}

```

C.2. User Interface BackOffice

```
internal static ServiceEntidade.EntidadeClient serviceEntidade = new
ServiceEntidade.EntidadeClient();

.
.
.
internal void PreencheGridPesquisa()
{
    try
    {
        gvEntidade.DataSource = serviceEntidade.ListEntidades();
        gvEntidade.DataBind();
    }
    catch (Exception ex)
    {
        log.Error("[UIB (Entidade) - PreencheGridPesquisa]: ", ex);
    }
}
```